electronics

Article

Power Allocation Based on Multi-Agent Deep Deterministic Policy Gradient for Underwater Acoustic Communication Networks

Xuan Geng * and Xinyu Hui

College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China;
202130310017@shmtu.edu.cn
* Correspondence: xuangeng@shmtu.edu.cn

Abstract: This paper proposes a reinforcement learning-based power allocation for underwater acoustic communication networks (UACNs). The objective function is formulated as maximizing channel capacity under constraints of maximum power and minimum channel capacity. To solve this problem, a multi-agent deep deterministic policy gradient (MADDPG) approach is introduced, where each transmitter node is considered as an agent. Given the definition of a Markov decision process (MDP) model for this problem, the agents learn to collaboratively maximize the channel capacity by deep deterministic policy gradient (DDPG) learning. Specifically, the power allocation of each agent is obtained by a centralized training and distributed execution (CTDE) method. Simulation results show the sum rate achieved by the proposed algorithm approximates that of the fractional programming (FP) algorithm and improves by at least 5% compared with the DQN (deep Q-learning network) -based power allocation algorithm.

Keywords: power allocation; MADDPG; channel capacity

1. Introduction

Underwater acoustic communication networks (UACNs) have many applications for underwater environments, such as underwater environment monitoring, target tracking, and ocean data collection, which have attracted a lot of research [1]. The authors studied the channel state information (CSI) prediction in UACNs based on machine learning [2,3]. Q. Ren et al. investigated the energy-efficient data collection method for an underwater magnetic induction (MI)-assisted system [4]. In this paper, we focus on the power allocation study because it plays an important role in UACNs optimization. First, the total channel capacity can be improved through power allocation for transmitters, which reduces the negative impact of the limited bandwidth of underwater acoustic channels. Second, power allocation among nodes balances energy consumption and reduces total energy consumption, which is suitable for an energy-limited system. Finally, power allocation can reduce the interference between nodes and improve the service quality of the network. Therefore, considering the particular environment of UACNs, power allocation can overcome problems such as restricted bandwidth, limited energy, and interference, which have a substantial impact on underwater acoustic communication.

According to the characteristics of the underwater acoustic communication environment, many studies have proposed power allocation algorithms to optimize channel capacity [5–9]. K. Shen et al. analyzed the multiple-ratio concave–convex fractional programming (FP) problem and its application in solving power control problems [5]. Jin et al. proposed a joint optimization of slot scheduling and power allocation of sensor nodes to maximize the channel capacity for clustered networks [6]. Authors in [7] investigated a joint power allocation and transmission scheduling algorithm for UACNs, where the transmission start-up time and transmission power are co-optimized to maximize the total transmission capacity. Zhao et al. proposed power allocation based on genetic algorithms and adaptive greedy algorithms [8], which can maximize the channel capacity and system

check for updates

Citation: Geng, X.; Hui, X. Power Allocation Based on Multi-Agent Deep Deterministic Policy Gradient for Underwater Acoustic Communication Networks. *Electronics* **2024**, *13*, 295. https://doi.org/10.3390/electronics13020295

Academic Editor: Ahmed Abu-Siada

Received: 27 September 2023
Revised: 6 December 2023
Accepted: 7 December 2023
Published: 9 January 2024

robustness. To adapt to the dynamic underwater acoustic channel, Qarabaqi et al. proposed an adaptive power allocation method that models the channel as an autoregressive process and allows the transmitter to adaptively adjust the power allocation based on channel state information to maximize the signal interference noise ratio (SINR) at the receiver [9]. However, these algorithms require full channel state information (CSI).

Due to the dynamic channel and long propagation delay underwater, it is not efficient to obtain full CSI and execute model-based optimization. Therefore, mode-free-based reinforcement learning (RL) has been introduced to optimize the power control problem, whose model is data driven. The Q-learning and deep Q-networks (DQN) algorithms have been applied to solve power allocation problems in UACNs [10–12]. However, the Q-learning-based algorithms result in large action spaces that severely impact computational complexity. In contrast, the deterministic policy gradient (DPG) approach applies to the continuous action space. In response, the authors in [13] proposed to combine DQN and DPG into a deep deterministic policy gradient (DDPG) algorithm based on the actor–critic (AC) framework, which can solve high-dimensional continuous action space problems. Based on this, S. Han et al. proposed a DDPG strategy to optimize the continuous power allocation [14]. However, it takes the nodes as individual agents and does not consider the collaborative learning of the agents.

The multi-agent deep deterministic policy gradient (MADDPG) [15], as one of the AC algorithms, has been applied to much research such as unmanned aerial vehicle (UAV) [16], vehicle networks [17], and other resource allocation because of its high efficiency and collaboration. It also has been applied to power allocation in wireless mobile networks [18]. Inspired by these works, we proposed a power allocation algorithm based on MADDPG for UACNs in this paper, because the multiple underwater nodes generate high-dimensional action and state space, and the collaboration of nodes has the advantage in learning. We take the transmitter nodes as agents and multiple agents can cooperate and share information for network training. Accordingly, we propose to maximize the channel capacity as the objective function, with the constraints of maximum power and minimum channel capacity. We model the power allocation problem as a Markov decision process (MDP) and apply the MADDPG approach to optimize power allocation. The actor and critic network of DDPG is trained using a central trainer, and its parameters are broadcast to multiple agents. Each agent updates its own actor network and inputs state to obtain actions for execution. This centralized training and distributed execution (CTDE) method iteratively trains the neural network until convergence to obtain a power allocation strategy. The main contributions of this study are as follows.

We propose a MADDPG-based power allocation scheme for UACNs. A MDP model is formulated and then the MADDPG is used to solve it. To the best of our knowledge, we first study using the MADDPG approach to solve the power allocation problem in UACNs. Although the MADDPG structure comes from [15], we define the action, state, observation space, and reward function according to the objective function, and thus the MADDPG can be applied to the underwater network power allocation problem.

The consideration of the history information of CSI in the MDP model makes the proposed algorithm applicable to the underwater network involving mobility. Through the CTDE process, the multiple agents are trained collaboratively and make power allocation decisions adaptively to adapt to the changing underwater environment. Our approach is therefore better suited to underwater channels that vary due to fading and node movement.

The MDP model proposed in this paper can provide more QoS requirements in design. In the study, we guarantee QoS by requiring a minimum channel capacity. However, other QoS metrics, such as throughput, delay, or success transmission rate, can also be combined into the objective function. As a result, the MDP model can be adjusted to meet these QoS requirements and the MADDPG structure is still valid in these cases.

Simulation results show the total channel capacity of the proposed MADDPG power allocation performs better than that of DQN-based [19] and DDPG-based [13] algorithms with independent agent training. Also, the proposed method has a much lower running time compared with the FP algorithm, particularly with large networks.

2. System Model and Problem Formulation

2.1. System Model

In this paper, we consider a UACN consisting of M transmitter nodes and N receiver nodes, where the transmitter nodes are deployed at the water bottom and each node is configured with an underwater acoustic transducer, as shown in Figure 1. We use $\mathcal{M} \triangleq \{1, 2, \cdots M\}$ and $\mathcal{N} \triangleq \{1, 2, \cdots N\}$ to denote the transmitter nodes and the receiver nodes, respectively. Thus, there are $M \times N$ links in the system. When a node transmits the signal to the target receiver, the signals from other transmitter nodes are considered as interference. We assume that the transmission of all nodes in the network starts and ends at the same time slot for a duration of T_s.

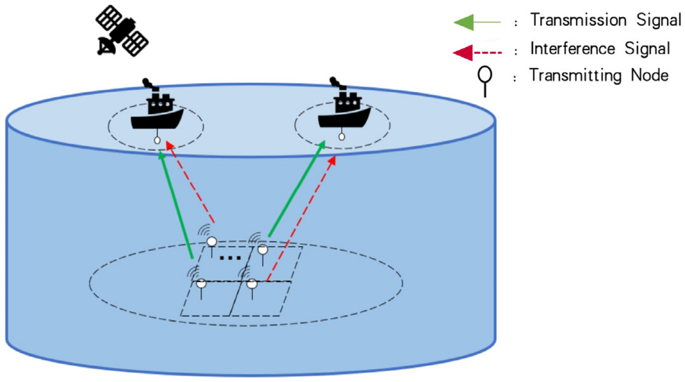

Figure 1. UACNs system model.

We assume the channel is slowly time varying and quasi-stationary with flat fading during a time slot. At the time slot t, the channel gain $g_{j,i}^{(t)}$ between receiver node $\mathcal{N}_j$ $(\mathcal{N}_j \in \mathcal{N})$ and transmitter node $\mathcal{M}_i$ $(\mathcal{M}_i \in \mathcal{M})$ is denoted by [20]:

$$g_{j,i}^{(t)} = \kappa \cdot 10 \log d_{j,i}^{(t)} + d_{j,i}^{(t)} \cdot 10 \log a(f), \tag{1}$$

where $i \in \mathcal{M}, j \in \mathcal{N}$. f represents the signal transmission frequency, and $d_{j,i}^{(t)}$ represents the distance between $\mathcal{N}_j$ and $\mathcal{M}_i$ in the time slot t. The expansion factor κ is typically 1.5. The $a(f)$ represents the absorption coefficient. According to the *Francois* $\propto$ *Garrison* model [20], the coefficient $a(f)$ is expressed by

$$a(f) = \frac{A_1 P_1 f_1 f^2}{f^2 + f_1^2} + \frac{A_2 P_2 f_2 f^2}{f^2 + f_2^2} + A_3 P_3 f^2, \tag{2}$$

where $A_1,\ A_2, A_3$ denote the impacts from boric acid, magnesium sulfate salt, and pure water components, respectively. They are functions of seawater temperature, the potential of hydrogen (pH), sound speed, and salinity. The symbols P_1, P_2, P_3 denote the water depth pressure of boric acid, magnesium sulfate salt, and pure water components. The f_1, f_2 denote the relaxation frequency of the boric acid and magnesium sulfate salts, which also depend on seawater temperature and salinity [21].

Considering the real network scenario of acoustic communication, the signal is interfered with underwater noise. The power spectral density of the noise $N(f)$ is denoted by

$$N(f) = N_t(f) + N_s(f) + N_w(f) + N_{th}(f), \tag{3}$$

where $N_t(f)$, $N_s(f)$, $N_w(f)$, $N_{th}(f)$ denote turbulence noise, shipping noise, wave noise, and thermal noise, respectively. These noises are mainly affected by signal frequency, shipping activity coefficient, and wind speed, as discussed in [21].

2.2. Problem Formulation

At time slot t, the SINR of the communication link (j,i) formed from the transmitter node $\mathcal{M}_i$ to receiver node $\mathcal{N}_j$ is expressed as

$$SINR_j^{(t)} = \frac{\left|g_{j,i}^{(t)}\right|^2 p_i^{(t)}}{\sum_{k\in\mathcal{M},\, k\neq i}\left|g_{j,k}^{(t)}\right|^2 p_k^{(t)} + \sigma_n^2} \quad i,k \in \mathcal{M}, j \in \mathcal{N}, \tag{4}$$

where $g_{j,i}^{(t)}$ denotes the channel gain of the link (j,i). The symbols $p_i^{(t)}$ and $p_k^{(t)}$ denote the transmit power of $\mathcal{M}_i$ and $\mathcal{M}_k$ at time slot t, respectively. The σ_n^2 is noise power. Accordingly, the channel capacity of $\mathcal{N}_j$ in time slot t is denoted by

$$C_j^{(t)} = \log_2\left(1 + SINR_j^{(t)}\right). \tag{5}$$

Our objective is to maximize total channel capacity by optimizing the power allocation, which is subject to maximum transmitting power and quality of service (QoS) requirements. This optimization problem is then formulated as

$$\begin{aligned} \mathcal{P}_1 : &\underset{p_i^{(t)}}{\text{maximize}} \sum_{j=1}^{N} C_j^{(t)} \\ s.t.\ &0 \leq p_i^{(t)} \leq P_{max}, \forall i \in M \\ &C_j^{(t)} \geq q_{th}, \forall j \in N, \end{aligned} \tag{6}$$

where P_{max} is the maximum transmit power and q_{th} is a threshold. The q_{th} ensures the minimum channel capacity of a single link, which is regarded as the QoS requirement. To solve $\mathcal{P}_1$, we formulate this problem as a Markov decision process (MDP) model and then apply the MADDPG to solve the problem, which can obtain the power allocation policy.

3. Reinforcement Learning

3.1. Introduction to Actor–Critic

In RL, the agent interacts with the environment and learns the optimal policy to maximize the expected total reward over a time horizon. At time slot t, the agent takes action $a^{(t)} \in \mathcal{A}$ in state $s^{(t)} \in \mathcal{S}$, where $\mathcal{A}$ and $\mathcal{S}$ represent action space and state space, respectively. After that, the environmental feedbacks reward $r^{(t)}$ to the agent, and then the agent moves to the next state $s'^{(t)}$. It then forms a sample of experience $(a^{(t)}, s^{(t)}, r^{(t)}, s'^{(t)})$ and stores it into replay memory $\mathcal{D}$. The agent trains the neural network to maximize the discounted future reward when it obtains enough experience samples and then obtains optimal decision strategy. The discounted future reward $R^{(t)}$ is defined as [22]:

$$R^{(t)} = \sum_{\eta=0}^{\infty} \gamma^{\eta} r^{(t+\eta+1)}, \tag{7}$$

where γ is a discount factor.

The policy updates include the value function-based method and the policy gradient-based method. The actor–critic framework combines these two methods. As shown in Figure 2, the AC network consists of an actor neural network and a critic neural network, with network parameters θ and μ, respectively. Considering continuous action and state space, we exploit DDPG to solve our objective function; therefore, the actor network updates θ using the deterministic policy gradient π_θ, while the critic updates μ using the gradient of the loss function.

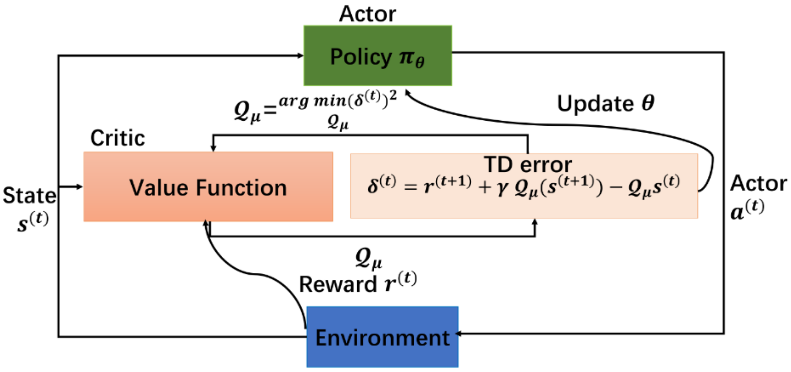

Figure 2. AC framework.

The actor and critic are defined as follows:

Actor: The actor network updates policy π_θ, which maps state space $\mathcal{S}$ into action space $\mathcal{A}$, which is denoted by

$$\pi_\theta(\mathcal{S}\ ) : \mathcal{S} \mapsto \mathcal{A}. \tag{8}$$

According to policy $\pi_\theta(\mathcal{S})$, the actor selects the action by the following rules:

$$a^{(t)} = \pi_\theta(\mathcal{S}) + \mathcal{U}^{(t)} \ \left(a^{(t)} \in \mathcal{A}\right), \tag{9}$$

where $\mathcal{U}^{(t)}$ is a random process.

Critic: The critic network estimates the action value $Q_\mu\left(s^{(t+1)}\right)$. It evaluates the new state by the temporal difference (TD) error, which is

$$\delta^{(t)} = r^{(t+1)} + \gamma Q_\mu\left(s^{(t+1)}\right) - Q_\mu\left(s^{(t)}\right). \tag{10}$$

The action selection weight will be enhanced if the TD error is positive. Otherwise, it is decreased with a negative TD error. The critic network and actor network parameters are updated as follows:

(1) Updates: μ AC uses replay buffer $\mathcal{D}$ to store empirical samples $(a^{(t)}, s^{(t)}, r^{(t)}, s'^{(t)})$. The critic network randomly selects G mini-batch samples $\left\{\left(a^{(g)}, s^{(g)}, r^{(g)}, s'^{(g)}\right)\right\}_{g=1}^{G}$ for network training, and updates the parameters by minimizing the mean-squared loss function between the target Q-value and the estimated Q-value. The loss function is formulated by [13]:

$$L(\mu) = \frac{1}{G}\sum_{g=1}^{G}\left[\left(y^{(g)} - Q_\mu\left(s^{(g)}, a^{(g)}\right)\right)\right]^2, \tag{11}$$

$$\mu \leftarrow \mu + \alpha_\mu \nabla L(\mu), \tag{12}$$

where $y^{(g)} = r^{(g)} + \gamma Q_{\mu'}\left(s'^{(g)}, \pi'_\theta\left(s'^{(g)}\right)\right)$ denotes the Q-value calculated by the target network and $\alpha_\mu \in (0,1)$ is the step size of the iterative update. The target network with parameter μ' is used to maintain the stability of the Q-value, where μ' is updated periodically by μ as

$$\mu' \leftarrow \tau\mu + (1-\tau)\mu', \tau \ll 1, \tag{13}$$

where τ is used to slowly update the target network.

(2) Update *θ:* The actor network is performed by a deterministic strategy, whose parameters are also trained from randomly selected samples. The goal of the actor is to find strategies that maximize the average long-term reward. The network parameters θ are updated by [13]:

$$\nabla_\theta J(\theta) \approx \mathbb{E}_{s^{(g)}\sim\mathcal{D}}\left[\nabla_{\pi_\theta} Q_\mu\left(s^{(g)}, a^{(g)}\right)\Big|_{a^{(g)}=\pi_\theta(s^{(g)})} * \nabla_\theta \pi_\theta\left(s^{(g)}\right)\right], \tag{14}$$

$$\theta \leftarrow \theta + \alpha_\theta \nabla J(\theta), \tag{15}$$

where $\alpha_\theta \in (0,1)$ is the step size of an iterative update to ensure that the critic is updated faster than the actor. The operation ∇ represents gradient descent for functions. Similar to the critic network, the t update for the actor target network parameter θ' is

$$\theta' \leftarrow \tau'\theta + \left(1-\tau'\right)\theta', \tau' \ll 1, \tag{16}$$

where τ' is used to update the target network.

3.2. MADDPG

The UACNs environment contains multiple nodes. It is more efficient to use multi-agent reinforcement learning like MADDPG than DDPG with independent training by a single agent. However, training multiple agents leads to instability and invalid experience replay. To address these challenges, MADDPG utilizes a centralized training and decentralized execution (CTDE) framework, where a central trainer handles the learning process using the DDPG and broadcasts the training parameters to each agent. The central trainer includes the actor network, target actor network, critic network, and target critic network. The single agent only contains an independent actor network, whose parameters come from the central trainer. The single agent inputs the state to its actor network and obtains the action. This separation of training and execution allows more stable and efficient multi-agent learning. Each agent benefits from the shared learning while acting independently during execution.

There are M agents in the UACNs, and for the agent $\mathcal{M}_i$, the parameters of the actor network and its local policy are denoted by θ_i and π_{θ_i} respectively. Therefore, the network parameters related to M agents are described by $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M)$ and $\boldsymbol{\pi} = \left(\pi_{\theta_1}, \ldots, \pi_{\theta_M}\right)$. The learning processes of multiple agents can be represented by a MDP model, which is defined by the state $\mathcal{S}$, action $\mathcal{A}_1, \ldots, \mathcal{A}_M$, observation $\mathcal{O}_1, \ldots, \mathcal{O}_M$, and state transfer function Γ. Agent $\mathcal{M}_i$ uses the deterministic policy $\pi_{\theta_i}(\mathcal{O}) : \mathcal{O}_i \mapsto \mathcal{A}_i$ for action selection and moves to the next state according to the state transition function $\mathit{\Gamma} : \mathcal{S} \times \mathcal{A}_i \times \cdots \times \mathcal{A}_M \mapsto \mathcal{S}$. It then receives the reward $r_i : \mathcal{S} \times \mathcal{A}_i \mapsto \mathcal{R}_i$ and also obtains the observation $o_i : \mathcal{S} \mapsto \mathcal{O}_i$.

The central trainer updates the parameters of the critic network by minimizing the loss function [15]:

$$L(\theta_i) = \mathbb{E}_{s,a,r,s'\sim\mathcal{D}}\left[\left(y^{(g)} - Q_i^\pi\left(s^{(g)}, a_1^{(g)}, \cdots, a_M^{(g)}\right)\right)^2\right], \tag{17}$$

where $y^{(g)} = r_i^{(g)} + \gamma Q_i^{\pi'}\left(s'^{(g)}, a_1'^{(g)}, \cdots, a_M'^{(g)}\right) |_{a_i^{(g)}=\pi_{\theta_i}(o_i)}$ is the Q-value of the target network, and $\boldsymbol{\pi}' = \left(\pi'_{\theta_1}, \cdots, \pi'_{\theta_M}\right)$ is the set of target policies, which is updated by Equation (16).

The actor network of agent $\mathcal{M}_i$ performs parameter updates by the gradient descent algorithm with the deterministic policy π_{θ_i}. The loss function is [15]:

$$\nabla_{\theta_i} J(\theta_i) \approx \mathbb{E}_{s^{(g)},a^{(g)}\sim\mathcal{D}}\left[\nabla_{a_i} Q_{\pi_{\theta_i}}\left(s^{(g)}, a_1^{(g)}, \ldots, a_M^{(g)}\right)\Big|_{a_i^{(g)}=\pi_{\theta_i}(o_i)} * \nabla_{\theta_i}\pi_{\theta_i}(o_i)\right], \tag{18}$$

where the replay buffer $\mathcal{D}$ stores samples from M agents at each time slot, which are $\left(s^{(t)}, a_1^{(t)}, \dots, a_M^{(t)}, r_1^{(t)}, \dots, r_M^{(t)}, s'^{(t)}\right)$. After the training in the central trainer, the parameters of the ith actor network are broadcasted to each agent. Each agent then uses the received parameters to independently update the actor network.

4. Power Allocation Based on MADDPG

4.1. MDP Model

In this paper, we regard each transmitter node as an agent. Therefore, there are multiple agents in the system. The multi-agents must consider both their observations and other agents' actions, and their actions also affect surrounding agents' policies. To obtain a collaborative advantage, the CTDE framework is used in this paper to train the network by the centralized trainer, as is shown in Figure 3. Each agent interacts with the environment and other agents with the information exchange demonstrated by ①/②/③/④. The central trainer includes the actor and critic training networks of all agents and their respective target networks. Training samples come from experiences $\left(s^{(t)}, a_1^{(t)}, \cdots, a_M^{(t)}, r_1^{(t)}, \cdots, r_M^{(t)}, s^{('t)}\right)$ sent by each agent and stored in replay memory $\mathcal{D}$. During the centralized training, the central trainer selects the samples randomly from $\mathcal{D}$ and updates AC network parameters via DDPG. After central training, the trainer broadcasts new actor parameters to each agent. For the distributed execution, the individual agent executes its action output by its local actor and then receives rewards and moves to the new state. A new sample experience is then obtained by each agent, which is sent back to the central trainer.

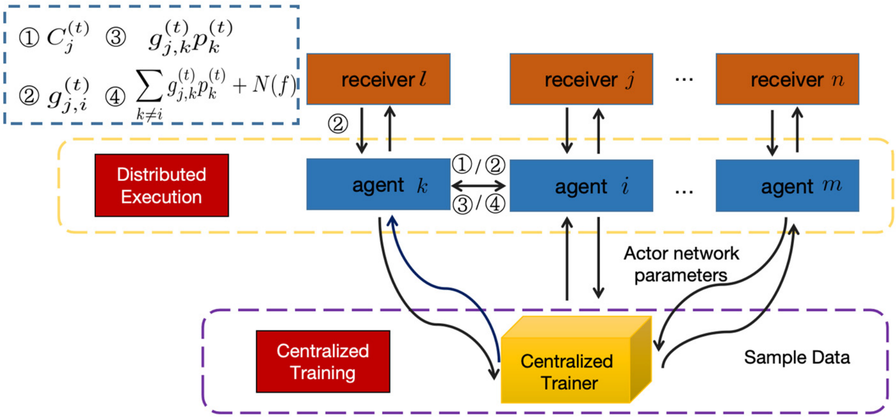

Figure 3. Diagram of CTDE.

Based on the objective function, we define the action, state, and reward functions to formulate a MDP model.

(1) Action Space $\mathcal{A}$: We assume all nodes have the same maximum power constraint. During the time slot, the action $a_i^{(t)}$ of agent $\mathcal{M}_i$ is defined as the transmission power allocated by an agent. This results in the action space being defined as

$$\begin{aligned} A &= \left\{a_i^{(t)} = p_i \middle| 0 \le p_i \le P_{max}\right\}, \\ p_i &= P_{min} + \frac{x}{X}(P_{max} - P_{min}),\ x = 0, \cdots, X, \end{aligned} \tag{19}$$

where X is the number of discretized power levels and $P_{min} = 0$ is assumed.

(2) State Space $\mathcal{S}$: The state of the $\mathcal{M}_i$ at time slot t consists of two parts, i.e., $s_i^t = \left\{o_i^{(t)}, a_i^{(t)}\right\}$, in which $o_i^{(t)} = \left\{\phi_i^{(t)}, \rho_i^{(t)}\right\}$ is the current observation. The symbol $\phi_i^{(t)} = \| g_{(j,i)}^{(t-1)}, g_{(j,i)}^{(t)} \|$ represents the channel state information, containing the channel gain from the transmitter $\mathcal{M}_i$ to receiver $\mathcal{N}_j$ at time slot $t-1$ and t. The $\rho_i^{(t)}$ denotes the system

state information. It includes the interference and noise received by $\mathcal{N}_j$ at the previous two time slots, and channel capacity $C_{j,i}^{(t-1)}$ and $C_{l,k}^{(t-1)}$ for the adjacent link, which is denoted by

$$\rho_i^{(t)} = \left[\sum\nolimits_{k\in M,k\neq i} g_{j,k}^{(t-2)} p_k^{(t-2)} + N(f), \sum\nolimits_{k\in M,k\neq i} g_{j,k}^{(t-1)} p_k^{(t-1)} + N(f), C_{j,i}^{(t-1)}, C_{l,k}^{(t-1)}\right], \tag{20}$$

where $N(f)$ denotes noise power. The historical information of CSI considered in state space makes our proposed algorithm applicable to varying channel environments.

(3) Reward: In the objective function, we require each user to maintain a minimum channel capacity. Therefore, we define the reward function such that channel capacity yields a positive reward, while interference results in a negative reward. We also consider the minimum channel capacity constraint. At time slot t, the reward function $r_i^{(t)}$ is defined as

$$r_i^{(t)} = C_j^{(t)} - \zeta_1 \sum_{l\in N,l\neq j} I_{l,i}^{(t)} - \zeta_2 \left| C_j^{(t)} - q_{th} \right|, \tag{21}$$

where $C_j^{(t)}$ denotes the channel capacity. The elements $\sum\limits_{l\in N,l\neq j} I_{l,i}^{(t)}$ represents the interference caused by signals from $\mathcal{M}_i$ to receivers $\mathcal{N}_l$ $(l \neq j)$, calculated as $I_{l,i}^{(t)} = C_{l\setminus i}^{(t)} - C_l^{(t)}$, with the $C_{l\setminus i}^{(t)}$ being the channel capacity of $\mathcal{N}_l$ without interference from $\mathcal{M}_i$. The coefficient ζ_1 adjusts the penalty ratio for interference. The $\left| C_j^{(t)} - q_{th} \right|$ represents the deviation from the minimum capacity threshold q_{th}, weighted by ζ_2.

4.2. Power Allocation Algorithm

Based on the MDP model defined in Section 4.1, we present a power allocation algorithm using MADDPG. At time slot t, agent $\mathcal{M}_i$ inputs the state $s_i^{(t)}$ into its local actor network, and outputs action $a_i^{(t)} = p_i^{(t)}$ according to the policy π_{θ_i}. Agent $\mathcal{M}_i$ then transmits signals to the receiver $\mathcal{N}_j$ with power $p_i^{(t)}$. If the signals are received successfully, the receiver $\mathcal{N}_j$ feedbacks the channel gain. Meanwhile, agent $\mathcal{M}_j$ sends communication requests to the neighboring agent $\mathcal{M}_k$. The $\mathcal{M}_k$ responds to $\mathcal{M}_i$ with information including of $g_{l,k}^{(t)}$, $g_{j,k}^{(t)}$, $p_k^{(t)}$, and $N(f)$. With the receiving information and the stored history information in $t-1$ and $t-2$, the $\mathcal{M}_i$ calculates the current observation $o_i^{(t)}$ and obtains reward $r_i^{(t)}$ to form the state $s_i^{(t)} = \left\{ o_i^{(t)}, a_i^{(t)} \right\}$. Note that this process is carried out in parallel for all agents. Although agents can interact with all neighbors for information exchange, we assume that each agent interacts only with its two nearest neighbors in the simulation due to long propagation delays in underwater channels.

After the communication process completes at the end of time slot t, each agent sends experience data $\left(s^{(t)}, a_1^{(t)}, \cdots, a_M^{(t)}, r_1^{(t)}, \cdots, r_M^{(t)}, s'^{(t)} \right)$ to the central trainer and the experience data is stored in $\mathcal{D}$. Once sufficient sample data is collected, the central trainer randomly selects a batch of G samples to update the actor–critic network parameters through gradient descent. The updating process has been described in Equations (17) and (18). The trainer broadcasts the updated actor parameters after completing centralized training. The agents then update their actor networks for action selection in the next time slot. If the environment changes slowly, the actor network parameters by centralized training can broadcast to the agent at intervals of several time slots. For an individual agent, the centralized training reduces computational requirements and saves energy.

We conclude the proposed method in Algorithm 1.

Algorithm 1: MADDPG power allocation

```
Initialization: Randomly initialize θ and μ; Initialize $\mathcal{D}$, $G$, target network parameter update period $T_u$
for each episode do
    Initialize the environment and state space $\mathcal{S}$.
    for $t = 1$ to $T_m$ do
        for $i = 1$ to $M$ do
            Input state $s_i^{(t)}$ to $Actor_i$, and agent $\mathcal{M}_i$ outputs action: $a_i^{(t)} = p_i^{(t)}$
            $\mathcal{M}_i$ interacts with $\mathcal{M}_j$ to obtain $g_{j,i}^{(t)}$
            $\mathcal{M}_i$ interacts with $\mathcal{M}_k$ to obtain $g_{l,k}^{(t)}$, $g_{j,k}^{(t)}$, $p_k^{(t)}$ and $N(f)$
            Calculates observations : $o_i^{(t)}$
            Receives reward : $r_i^{(t)}$
            Forms state : $s_i^{(t)} = \left\{o_i^{(t)},\ a_i^{(t)},\right\}$
            Updates next state : $s_i'^{(t)} = s_i^{(t)}$
            Forms sample data and transmits to $\mathcal{D}$ : $(s_i^{(t)}, a_i^{(t)}, r_i^{(t)}, s_i'^{(t)})$
        end for
        Selects $G$ batches samples from $\mathcal{D}$ : $\left(s^{(g)}, a^{(g)}, r^{(t)}, s'^{(t)}\right)$
        Calculate $y^{(g)} = r_i^{(g)} + \gamma Q_i^{\pi'}\left(s'^{(g)}, a_1'^{(g)}, \cdots, a_M'^{(g)}\right)|_{a_i^{(g)}=\pi_{\theta_i}(o_i)}$
        Update Critic network by Equation (17)
        Update Actor network by Equation (18)
        Broadcast $Actor_i$ network parameter to agent
    end for
    if $t == T_u$ then
        Update the target network parameters for critic and actor by Equations (13) and (16)
    end if
end for
```

5. Simulation Results

In this section, we evaluate the performance of the proposed MADDPG power allocation through simulations. We assume that $M = 10$ and $N = 10$, and all source nodes are deployed underwater, as shown in Figure 4. The source nodes are located 20 m underwater with 100 m between adjacent nodes. The receivers are on the water surface, with communication distances of $20 \sim 500$ m from the source nodes. The simulation parameters of the underwater acoustic environment are shown in Table 1, where the wind speed, salinity, pH, temperature, and sound speed data are measured from the Yellow Sea of China in 2015 [23]. We also assume the underwater acoustic channel is slow time varying and quasi-static flat fading, which means $g_{j,i}^{(t)}$ is constant within a time slot.

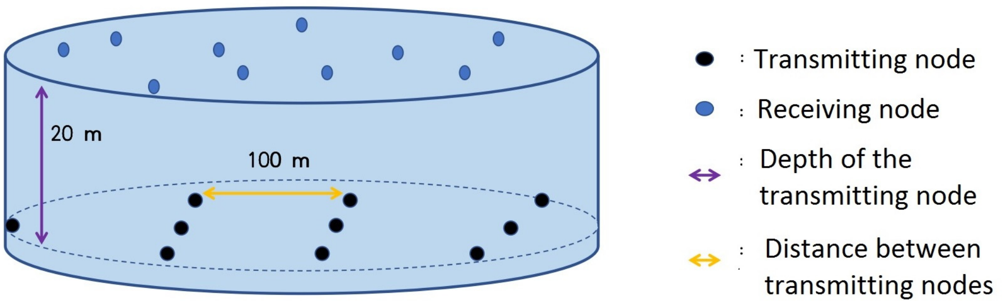
20 m
100 m
● : Transmitting node
● : Receiving node
↔ : Depth of the transmitting node
↔ : Distance between transmitting nodes

Figure 4. Node deployment.

Table 1. Simulation parameters.

Parameters	Value
Frequency (f)	20,000 Hz
Maximum Doppler frequency (f_d)	12 Hz
Shipping activity coefficient (μ)	0.5
Maximum transmission power (P_{max})	1 W
Time slot length (T_s)	2.0 s
Wind speed (w)	0.1 m/s
Salinity (H)	31.8%
pH	8.17
Temperature (T)	9.6 °C
Speed of sound (c)	1480.9 m/s

According to [24], the underwater sensor nodes are anchored and restricted by a cable, which can float in water. The nodes move at a speed of 0.83–1.67 m/s within the limit of the cable length [24]. We adopt a moving speed of 0.9 m/s in the simulation. Therefore, the maximum Doppler frequency is 12 Hz. To avoid the space–time uncertainty caused by the long propagation delay of underwater acoustic transmission, we assume that the time slot length is long enough to complete information exchange and power allocation in the same time slot. Based on transmission distance and sound velocity, the time slot is assumed to be 2 s.

We compared the proposed MADDPG algorithm with fractional programming (FP) power allocation algorithm [5], DQN training-based power allocation algorithm [19], DDPG algorithm without collaboration [13], random power allocation, and maximum transmitting power (full power).

Figure 5 shows that the proposed MADDPG algorithm obtains a better sum rate compared with other power allocation strategies. The sum rate here refers to the sum channel capacity of all links. The sum rate of the proposed MADDPG power allocation remains above 1.7 bps/Hz, in which the bps means bits per second. The FP algorithm is a model-driven method and has full CSI, whereas the deep learning methods such as DQN, DDPG, and MADDPG are only data driven without full CSI, resulting in lower performance than the FP algorithm. In the DQN and DDPG algorithms, each single agent is trained independently without interacting with the surrounding environment. However, the agents in MADDPG interact with each other and can use global data for centralized training, which obtains a better performance than DQN and DDPG. Random power and full power do not optimize power allocation, thus resulting in the worst performance.

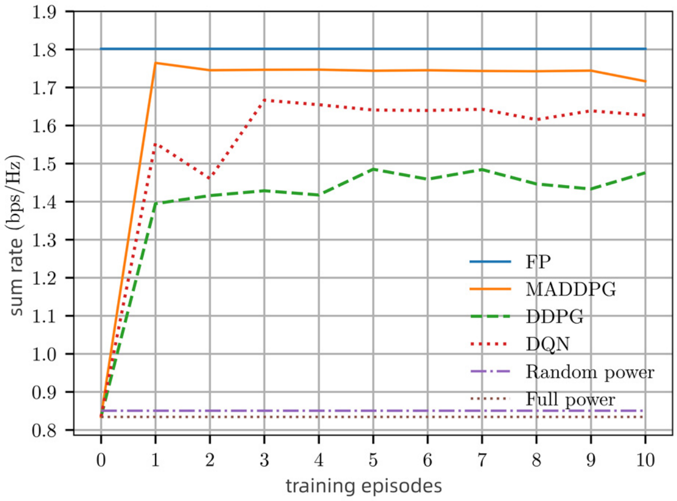

Figure 5. Comparison of sum rate with different algorithms.

Figure 6 compares the spectral efficiency (SE) performance of different algorithms in single-episode training. It can be seen that the SE of MADDPG is close to FP and outperforms other algorithms. Moreover, the MADDPG power allocation obtains convergence within 5000 training time steps, which has the same convergence rate as DQN.

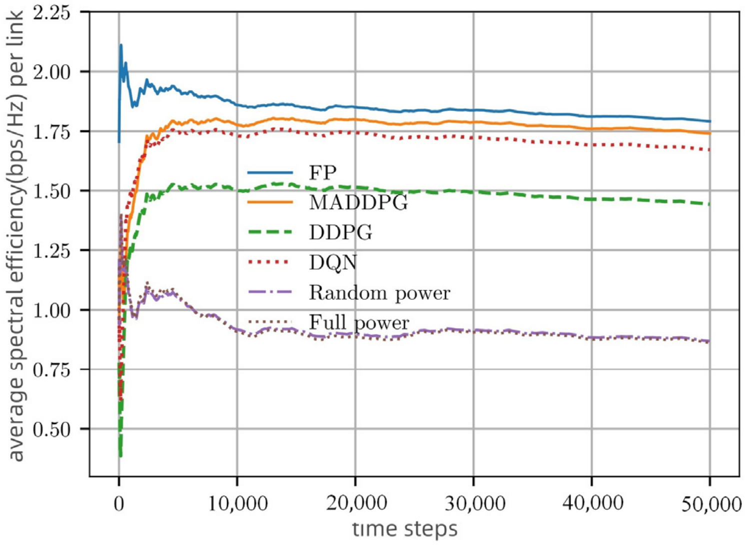

Figure 6. SE performance of different algorithms for single-episode training.

In the objective function $\mathcal{P}_1$, the threshold q_{th} is required to ensure the minimum channel capacity of each link. Figure 7 compares the sum rate of MADDPG power allocation with ($q_{th} = 1$ bps/Hz) and without ($q_{th} = 0$ bps/Hz) minimum channel capacity constraint. The algorithm considering $q_{th} = 1$ bps/Hz maintains a channel capacity of approximately 1.75 bps/Hz which performs 75% higher than that of without considering minimum channel capacity. Therefore, considering the minimum channel capacity constraint and penalty for interference in the reward function of our algorithm, each link can ensure a minimum channel capacity, which improves the system sum rate.

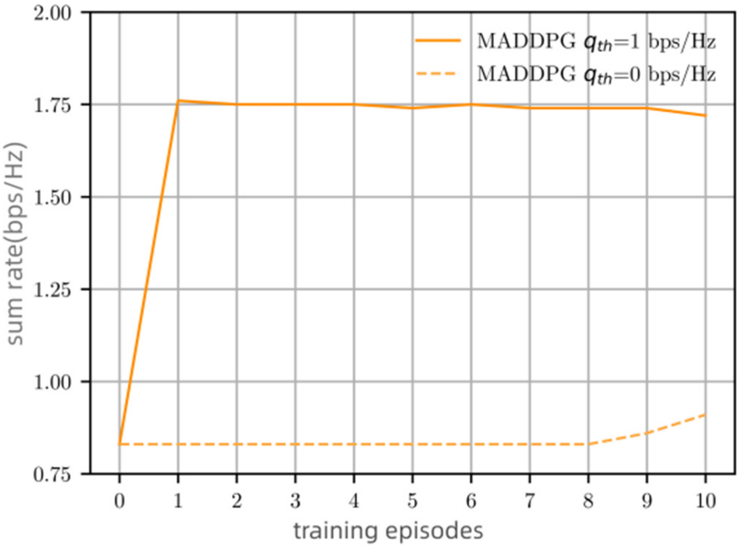

Figure 7. The influence on sum rate.

Figure 8 compares the computational complexity of different algorithms as the number of network nodes increases. We use the program running time to complete one-time power allocation as a metric. From Figure 8, we can see that the number of nodes affects the complexity of the algorithm. The algorithms of MADDPG, DQN, and DDPG have approximately the same complexity, while FP is much higher. The per-iteration complexity of FP is $O(M^2)$, while the others are $O(M)$ or less. Note that random and maximum power allocation algorithms are excluded from Figure 8 since they do not need additional calculations.

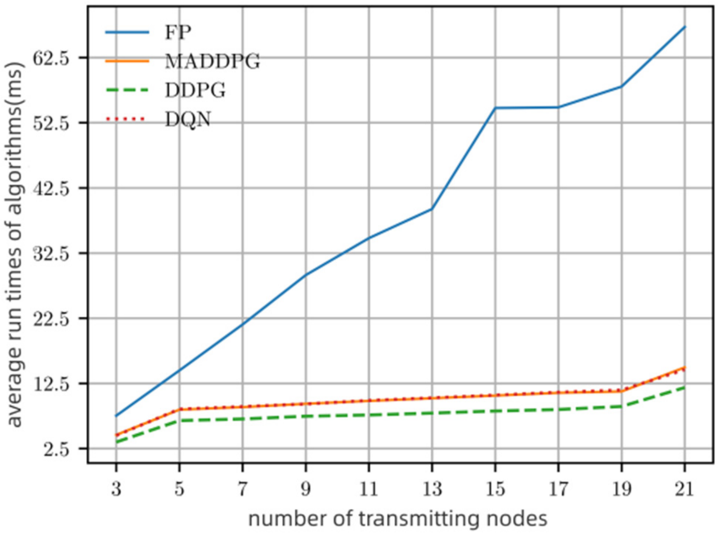

Figure 8. Average running time versus the number of network nodes.

6. Conclusions

This paper proposes a power allocation algorithm based on reinforcement learning, which optimizes the channel capacity of UACNs. The action, state, observation, and reward functions of a MDP model are designed to solve the objective function. To take advantage of collaborative training, the MADDPG structure is applied to this problem, which is implemented by centralized training and distributed execution. An actor–critic network of all agents is trained by a centralized trainer, while the independent actor network of each agent is used to execute actions. The minimum channel capacity constraint ensures the QoS requirement of each link. Simulation results demonstrate that the proposed algorithm outperforms the DQN- and DDPG-based power allocation algorithms of both sum rate and spectral efficiency. Furthermore, as the number of network nodes increases, the proposed method has a much lower running time compared with the FP algorithm.

Author Contributions: Formal analysis, X.G.; data curation, X.H.; conceptualization, X.G. and X.H.; methodology, X.G. and X.H.; writing—original draft, X.G. and X.H.; writing—review and editing, X.G. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is partially funded by the Innovation Program of Shanghai Municipal Education Commission of China under Grant, No. 2101070010E00121, the Shanghai Sailing Program, No. 20YF1416700, and the National Natural Science Foundation of China, No. 62271303.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Qiu, T.; Zhao, Z.; Zhang, T.; Chen, C.; Chen, C.L.P. Underwater internet of things in smart ocean: System architecture and open issues. *IEEE Trans. Ind. Inform.* **2020**, *16*, 4297–4307. [CrossRef]
2. Liu, L.; Cai, L.; Ma, L.; Qiao, G. Channel state information prediction for adaptive underwater acoustic downlink OFDMA system: Deep neural networks based approach. *IEEE Trans. Veh. Technol.* **2021**, *70*, 9063–9076. [CrossRef]
3. Hu, X.; Huo, Y.; Dong, X.; Wu, F.-Y.; Huang, A. Channel prediction using adaptive bidirectional GRU for underwater MIMO communications. *IEEE Internet Things J.* **2023**. [CrossRef]
4. Ren, Q.; Sun, Y.; Li, S.; Wang, B.; Yu, Z. Energy-efficient data collection over underwater MI-assisted acoustic cooperative MIMO WSNs. *China Commun.* **2023**, *20*, 96–110. [CrossRef]

5. Shen, K.; Yu, W. Fractional programming for communication systems—Part I: Power control and beamforming. *IEEE Trans. Signal Process.* **2018**, *66*, 2616–2630. [CrossRef]
6. Jin, X.; Liu, Z.; Ma, K. Joint slot scheduling and power allocation for throughput maximization of clustered UACNs. *IEEE Internet Things J.* **2023**, *10*, 17085–17095. [CrossRef]
7. Wang, C.; Zhao, W.; Bi, Z.; Wan, Y. A joint power allocation and scheduling algorithm based on quasi-interference alignment in underwater acoustic networks. In Proceedings of the OCEANS 2022, Chennai, India, 21–24 February 2022; pp. 1–6.
8. Zhao, Y.; Wan, L.; Chen, Y.; Cheng, E.; Xu, F.; Liang, L. Power allocation for non-coherent multi-carrier FSK underwater acoustic communication systems with uneven transmission source level. In Proceedings of the 2022 14th International Conference on Signal Processing Systems (ICSPS), Zhenjiang, China, 18–20 November 2022; pp. 616–622.
9. Qarabaqi, P.; Stojanovic, M. Adaptive power control for underwater acoustic communications. In Proceedings of the 2011 IEEE-Spain OCEANS, Santander, Spain, 6–9 June 2011; pp. 1–7.
10. Yang, L.; Wang, H.; Fan, Y.; Luo, F.; Feng, W. Reinforcement learning for distributed energy efficiency optimization in underwater acoustic communication networks. *Wirel. Commun. Mobile Comput.* **2022**, *2022*, 5042833. [CrossRef]
11. Wang, H.; Li, Y.; Qian, J. Self-adaptive resource allocation in underwater acoustic interference channel: A reinforcement. learning approach. *IEEE Internet Things J.* **2020**, *7*, 2816–2827. [CrossRef]
12. Su, Y.; Liwang, M.; Gao, Z.; Huang, L.; Du, X.; Guizani, M. Optimal cooperative relaying and power control for IoUT networks with reinforcement learning. *IEEE Internet Things J.* **2020**, *8*, 791–801. [CrossRef]
13. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.M.O.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
14. Han, S.; Li, L.; Li, X.; Liu, Z.; Yan, L.; Zhang, T. Joint relay selection and power allocation for time-varying energy harvesting-driven UACNs: A stratified reinforcement learning approach. *IEEE Sens. J.* **2022**, *22*, 20063–20072. [CrossRef]
15. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-agent actor-critic for mixed cooperative competitive environments. *arXiv* **2017**, arXiv:1706.02275.
16. Ding, R.; Xu, Y.; Gao, F.; Shen, X.S. Trajectory design and access control for air–ground coordinated communications system with multiagent deep reinforcement learning. *IEEE Internet Things J.* **2021**, *9*, 5785–5798. [CrossRef]
17. Huang, X.; He, L.; Zhang, W. Vehicle speed aware computing task offloading and resource allocation based on multi-agent reinforcement learning in a vehicular edge computing network. In Proceedings of the 2020 IEEE International Conference on Edge Computing (EDGE), Beijing, China, 18–24 October 2020; pp. 1–8.
18. Nasir, Y.S.; Guo, D. Deep actor-critic learning for distributed power control in wireless mobile networks. In Proceedings of the 2020 54th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 1–4 November 2020; pp. 398–402.
19. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
20. Francois, R.E.; Garrison, G.R. Sound absorption based on ocean measurements. Part II: Boric acid contribution and equation for total absorption. *J. Acoust. Soc. Am.* **1982**, *72*, 1879–1890. [CrossRef]
21. Domingo, M.C. Overview of channel models for underwater wireless communication networks. *Phys. Commun.* **2008**, *1*, 163–182. [CrossRef]
22. Sutton, R.S.; Barto, A.G. *Reinforcement Learning—An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
23. Huang, J.; Gao, G.; Cheng, T.; Hu, D.; Sun, D. Hydrological features and air-sea CO2 fluxes of the Southern Yellow Sea in the winter of 2015. *J. Shanghai Ocean Univ.* **2017**, *26*, 757–765.
24. Hong, F.; Zhang, Y.; Yang, B.; Guo, Y.; Guo, Z. Review on time synchronization techniques in underwater acoustic sensor networks. *Acta Electonica Sin.* **2013**, *41*, 960–965.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.