Distributed Adaptive Networks: A Graphical Evolutionary Game-Theoretic View

Chunxiao Jiang, *Member, IEEE*, Yan Chen, *Member, IEEE*, and K. J. Ray Liu, *Fellow, IEEE*

Abstract—Distributed adaptive filtering has been considered as an effective approach for data processing and estimation over distributed networks. Most existing distributed adaptive filtering algorithms focus on designing different information diffusion rules, regardless of the nature evolutionary characteristic of a distributed network. In this paper, we study the adaptive network from the game theoretic perspective and formulate the distributed adaptive filtering problem as a graphical evolutionary game. With the proposed formulation, the nodes in the network are regarded as players and the local combiner of estimation information from different neighbors is regarded as different strategies selection. We show that this graphical evolutionary game framework is very general and can unify the existing adaptive network algorithms. Based on this framework, as examples, we further propose two error-aware adaptive filtering algorithms. Moreover, we use graphical evolutionary game theory to analyze the information diffusion process over the adaptive networks and evolutionarily stable strategy of the system. Finally, simulation results are shown to verify the effectiveness of our analysis and proposed methods.

Index Terms—Adaptive filtering, graphical evolutionary game, distributed estimation, adaptive networks, data diffusion.

I. INTRODUCTION

RECENTLY, the concept of adaptive filter network derived from the traditional adaptive filtering was emerging, where a group of nodes cooperatively estimate some parameters of interest from noisy measurements [1]. Such a distributed estimation architecture can be applied to many scenarios, such as wireless sensor networks for environment monitoring, wireless Ad-hoc networks for military event localization, distributed cooperative sensing in cognitive radio networks and so on [2], [3]. Compared to the classical centralized architecture, the distributed one is not only more robust when the center node may be dysfunctional, but also more flexible when the nodes are with mobility. Therefore, distributed adaptive filter network has been considered as

an effective approach for the implementation of data fusion, diffusion and processing over distributed networks [4].

In a distributed adaptive filter network, at every time instant t, node i receives a set of data $\{d_i(t), \boldsymbol{u}_{i,t}\}$ that satisfies a linear regression model as follow

$$d_i(t) = \boldsymbol{u}_{i,t}\boldsymbol{w}^0 + v_i(t), \qquad (1)$$

where $\boldsymbol{w}^0$ is a deterministic but unknown $M \times 1$ vector, $d_i(t)$ is a scalar measurement of some random process $\boldsymbol{d}_i$, $\boldsymbol{u}_{i,t}$ is the $1 \times M$ regression vector at time t with zero mean and covariance matrix $\boldsymbol{R}_{u_i} = \mathbb{E}(u_{i,t}^* u_{i,t}) > 0$, and $v_i(t)$ is the random noise signal at time t with zero mean and variance σ_i^2. Note that the regression data $\boldsymbol{u}_{i,t}$ and measurement process $\boldsymbol{d}_i$ are temporally white and spatially independent, respectively and mutually. The objective for each node is to use the data set $\{d_i(t), \boldsymbol{u}_{i,t}\}$ to estimate parameter $\boldsymbol{w}^0$.

In the literatures, many distributed adaptive filtering algorithms have been proposed for the estimation of parameter $\boldsymbol{w}^0$. The incremental algorithms, in which node i updates $\boldsymbol{w}$, i.e., the estimation of $\boldsymbol{w}^0$, through combining the observed data sets of itself and node $i - 1$, were proposed, e.g., the incremental LMS algorithm [5]. Unlike the incremental algorithms, the diffusion algorithms allow node i to combine the data sets from all neighbors, e.g., diffusion LMS [6], [7] and diffusion RLS [8]. Besides, the projection-based adaptive filtering algorithms were summarized in [9], e.g., the projected subgradient algorithm [10] and the combine-project-adapt algorithm [11]. In [12], the authors considered the node's mobility and analyzed the mobile adaptive networks.

While achieving promising performance, these traditional distributed adaptive filtering algorithms mainly focused on designing different information combination rules or diffusion rules among the neighborhood by utilizing the network topology information and/or nodes' statistical information. For example, the relative degree rule considers the degree information of each node [8], and the relative degree-variance rule further incorporates the variance information of each node [6]. However, most of the existing algorithms are somehow intuitively designed to achieve some specific objective, sort of like bottom-up approaches to the distributed adaptive networks. There is no existing work that offers a design philosophy to explain why combination and/or diffusion rules are developed and how they are related in a unified view. Is there a general framework that can reveal the relationship among the existing rules and provide fundamental guidance for better design of distributed adaptive filtering algorithms? In our quest to answer the question, we found that in essence the parameter updating process in distributed adaptive networks follows similarly the

evolution process in natural ecological systems. Therefore, based on the graphical evolutionary game, in this paper, we propose a general framework that can offer a unified view of existing distributed adaptive algorithms, and provide possible clues for new future designs. Unlike the traditional bottom-up approaches that focus on some specific rules, our framework provide a top-down design philosophy to understand the fundamental relationship of distributed adaptive networks.

The main contributions of this paper are summarized as follows.

1) We propose a graphical evolutionary game theoretic framework for the distributed adaptive networks, where nodes in the network are regarded as players and the local combination of estimation information from different neighbors is regarded as different strategies selection. We show that the proposed graphical evolutionary theoretic framework can unify existing adaptive filtering algorithms as special cases.

2) Based on the proposed framework, as examples, we further design two simple error-aware distributed adaptive filtering algorithms. When the noise variance is unknown, our proposed algorithm can achieve similar performance compared with existing algorithms but with lower complexity, which immediately shows the advantage of the proposed general framework.

3) Using the graphical evolutionary game theory, we analyze the information diffusion process over the adaptive network, and derive the diffusion probability of information from good nodes.

4) We prove that the strategy of using information from good nodes is evolutionarily stable strategy either in complete graphs or incomplete graphs.

The rest of this paper is organized as follows. We summarize the existing works in Section II. In Section III, we describe in details how to formulate the distributed adaptive filtering problem as a graphical evolutionary game. We then discuss the information diffusion process over the adaptive network in Section IV, and further analyze the evolutionarily stable strategy in Section V. Simulation results are shown in Section VI. Finally, we draw conclusions in Section VII.

II. RELATED WORKS

Let us consider an adaptive filter network with N nodes. If there is a fusion center that can collect information from all nodes, then global (centralized) optimization methods can be used to derive the optimal updating rule for the parameter $\boldsymbol{w}$, where $\boldsymbol{w}$ is a deterministic but unknown $M \times 1$ vector for estimation, as shown in the left part of Fig. 1. For example, in the global LMS algorithm, the parameter updating rule can be written as [6]

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \mu \sum_{i=1}^{N} \boldsymbol{u}_{i,t}^{*}(d_i(t) - \boldsymbol{u}_{i,t}\boldsymbol{w}_t), \tag{2}$$

where μ is the step size and $\{\cdot\}^{*}$ denotes complex conjugation operation. With (2), we can see that the centralized LMS algorithm requires the information of $\{d_i(t), \boldsymbol{u}_{i,t}\}$ across the whole network, which is generally impractical. Moreover, such a cen-
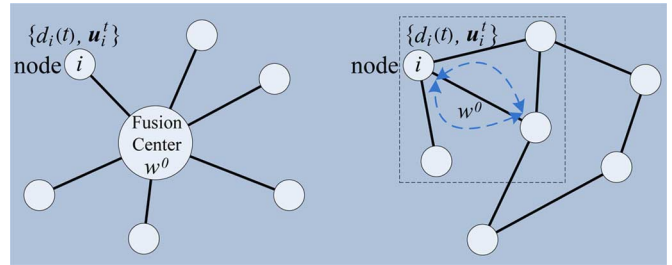

Fig. 1. Left: centralized model. Right: distributed model.

tralized architecture highly relies on the fusion center and will collapse when the fusion center is dysfunctional or some data links are disconnected.

If there is no fusion center in the network, then each node needs to exchange information with the neighbors to update the parameter as shown in the right part of Fig. 1. In the literature, several distributed adaptive filtering algorithms have been introduced, such as distributed incremental algorithms [5], distributed LMS [6], [7], and projection-based algorithms [10], [11]. These distributed algorithms are based on the classical adaptive filtering algorithms, where the difference is that nodes can use information from neighbors to estimate the parameter $\boldsymbol{w}^0$. Taking one of the distributed LMS algorithms, Adapt-then-Combine Diffusion LMS (ATC) [6], as an example, the parameter updating rule for node i is

$$\begin{cases} \boldsymbol{\chi}_{i,t+1} = \boldsymbol{w}_{i,t} + \mu_i \sum_{j \in \mathcal{N}_i} o_{ij} \boldsymbol{u}_{j,t}^{*}(d_j(t) - \boldsymbol{u}_{j,t}\boldsymbol{w}_{j,t}), \\ \boldsymbol{w}_{i,t+1} = \sum_{j \in \mathcal{N}_i} a_{ij} \boldsymbol{\chi}_{j,t+1}, \end{cases} \tag{3}$$

where $\mathcal{N}_i$ denotes the neighboring nodes set of node i (including node i itself), o_{ij} and a_{ij} are linear weights satisfying the following conditions

$$\begin{cases} o_{ij} = a_{ij} = 0, & \text{if } j \notin \mathcal{N}_i, \\ \sum_{j=1}^{N} o_{ij} = 1, & \sum_{j=1}^{N} a_{ij} = 1. \end{cases} \tag{4}$$

In a practical scenario, since the exchange of full raw data $\{d_i(t), \boldsymbol{u}_{i,t}\}$ among neighbors is costly, the weight o_{ij} is usually set as $o_{ij} = 0$, if $j \neq i$, as in [6]. In such a case, for node i with degree n_i (including node i itself, i.e., the cardinality of set $\mathcal{N}_i$) and neighbor set $\{i_1, i_2, \ldots, i_{n_i}\}$, we can write the general parameter updating rule as

$$\begin{aligned} \boldsymbol{w}_{i,t+1} &= A_{i,t+1}\left(F\left(\boldsymbol{w}_{i_1,t}\right), F\left(\boldsymbol{w}_{i_2,t}\right), \ldots, F\left(\boldsymbol{w}_{i_{n_i},t}\right)\right), \\ &= \sum_{j \in \mathcal{N}_i} A_{i,t+1}(j)F(\boldsymbol{w}_{j,t}), \end{aligned} \tag{5}$$

where $F(\cdot)$ can be any adaptive filtering algorithm, e.g., $F(\boldsymbol{w}_{i,t}) = \boldsymbol{w}_{i,t} + \mu \boldsymbol{u}_{i,t}^{*}(d_i(t) - \boldsymbol{u}_{i,t}\boldsymbol{w}_{i,t})$ for the LMS algorithm, $A_{i,t+1}(\cdot)$ represents some specific linear combination rule. The (5) gives a general form of existing distributed adaptive filtering algorithms, where the combination rule $A_{i,t+1}(\cdot)$ mainly determines the performance. Table I summarizes the existing combination rules, where for all rules $A_{i,t+1}(j) = 0$, if $j \notin \mathcal{N}_i$.

TABLE I
DIFFERENT COMBINATION RULES

Name	Rule: $A_i(j) =$
Uniform [11][13]	$\frac{1}{n_i}$, for all $j \in \mathcal{N}_i$
Maximum degree [8][14]	$\begin{cases} \frac{1}{N}, & \text{for } j \neq i, \\ 1 - \frac{n_i-1}{N}, & \text{for } j = i. \end{cases}$
Laplacian [15][16]	$\begin{cases} \frac{1}{n_{\max}}, & \text{for } j \neq i \\ 1 - \frac{n_i-1}{n_{\max}}, & \text{for } j = i. \end{cases}$
Relative degree [8]	$\frac{n_j}{\sum_{k \in \mathcal{N}_i} n_k}$, for all $j \in \mathcal{N}_i$
Relative degree-variance [6]	$\frac{n_j \sigma_j^{-2}}{\sum_{k \in \mathcal{N}_i} n_k \sigma_k^{-2}}$, for all $j \in \mathcal{N}_i$
Metropolis [16][17]	$\begin{cases} \frac{1}{\max\{|\mathcal{N}_i|,|\mathcal{N}_j|\}}, & \text{for } j \neq i, \\ 1 - \sum_{k \neq i} A_i(k), & \text{for } j = i. \end{cases}$
Hastings [17]	$\begin{cases} \frac{\sigma_j^2}{\max\{|\mathcal{N}_i|\sigma_i^2,|\mathcal{N}_j|\sigma_j^2\}}, & \text{for } j \neq i, \\ 1 - \sum_{k \neq i} A_i(k), & \text{for } j = i. \end{cases}$

From Table I, we can see that the weights of the first four combination rules are purely based on the network topology. The disadvantage of such topology-based rules is that, they are sensitive to the spatial variation of signal and noise statistics across the network. The relative degree-variance rule shows better mean-square performance than others, which, however, requires the knowledge of all neighbors' noise variances. As discussed in Section I, all these distributed algorithms are only focusing on designing the combination rules. Nevertheless, a distributed network is just like a natural ecological system and the nodes are just like individuals in the system, which may spontaneously follow some nature evolutionary rules, instead of some specific artificially predefined rules. Besides, although various kinds of combination rules have been developed, there is no general framework which can reveal the unifying fundamentals of distributed adaptive filtering problems. In the sequel, we will use graphical evolutionary game theory to establish a general framework to unify existing algorithms and give insights of the distributed adaptive filtering problem.

III. GRAPHICAL EVOLUTIONARY GAME FORMULATION

A. Introduction of Graphical Evolutionary Game

Evolutionary game theory (EGT) is originated from the study of ecological biology [18], which differs from the classical game theory by emphasizing more on the dynamics and stability of the whole population's strategies [19], instead of only the property of the equilibrium. EGT has been widely used to model users' behaviors in image processing [20], as well as communication and networking area [21], [22], such as congestion control [23], cooperative sensing [24], cooperative peer-to-peer (P2P) streaming [25] and dynamic spectrum access [26]. In these literatures, evolutionary game has been shown to be an effective approach to model the dynamic social interactions among users in a network.

EGT is an effective approach to study how a group of players converges to a stable equilibrium after a period of strategic interactions. Such an equilibrium strategy is defined as the Evolutionarily Stable Strategy (ESS). For an evolutionary game with N players, a strategy profile $\mathbf{a}^* = (a_1^*, \ldots, a_N^*)$, where $a_i^* \in \mathcal{X}$ and $\mathcal{X}$ is the action space, is an ESS if and only if, $\forall \mathbf{a} \neq \mathbf{a}^*$, $\mathbf{a}^*$ satisfies following [19]:

$$1)\ U_i(a_i, \mathbf{a}_{-i}^*) \leq U_i(a_i^*, \mathbf{a}_{-i}^*), \tag{6}$$
$$2)\ \text{if } U_i(a_i, \mathbf{a}_{-i}^*) = U_i(a_i^*, \mathbf{a}_{-i}^*),$$
$$U_i(a_i, \mathbf{a}_{-i}) < U_i(a_i^*, \mathbf{a}_{-i}), \tag{7}$$

where U_i stands for the utility of player i and $\mathbf{a}_{-i}$ denotes the strategies of all players other than player i. We can see that the first condition is the Nash equilibrium (NE) condition, and the second condition guarantees the stability of the strategy. Moreover, we can also see that a strict NE is always an ESS. If all players adopt the ESS, then no mutant strategy could invade the population under the influence of natural selection. Even if a small part of players may not be rational and take out-of-equilibrium strategies, ESS is still a locally stable state.

Let us consider an evolutionary game with m strategies $\mathcal{X} = \{1, 2, \ldots, m\}$. The utility matrix, U, is an $m \times m$ matrix, whose entries, u_{ij}, denote the utility for strategy i versus strategy j. The population fraction of strategy i is given by p_i, where $\sum_{i=1}^{m} p_i = 1$. The fitness of strategy i is given by $f_i = \sum_{j=1}^{m} p_j u_{ij}$. For the average fitness of the whole population, we have $\phi = \sum_{i=1}^{m} p_i f_i$. The Wright-Fisher model has been widely adopted to let a group of players converge to the ESS [27], where the strategy updating equation for each player can be written as

$$p_i(t+1) = \frac{p_i(t)f_i(t)}{\phi(t)}. \tag{8}$$

Note that one assumption in the Wright-Fisher model is that when the total population is sufficiently large, the fraction of players using strategy i is equal to the probability of one individual player using strategy i. From (8), it can be seen that the strategy updating process in the evolutionary game is similar to the parameter updating process in adaptive filter problem. It is intuitive that we can use evolutionary game to formulate the distributed adaptive filter problem.

The classical evolutionary game theory considers a population of M individuals in a complete graph. However, in many scenarios, players' spatial locations may lead to an incomplete graph structure. Graphical evolutionary game theory is introduced to study the strategies evolution in such a finite structured population [28], where each vertex represents a player and each edge represents the reproductive relationship between valid neighbors, i.e., θ_{ij} denotes the probability that the strategy of node i will replace that of node j, as shown in Fig. 2. Graphical EGT focuses on analyzing the ability of a mutant gene to overtake a group of finite structured residents. One of the most important research issues in graphical EGT is how to compute the fixation probability, i.e., the probability that the mutant will eventually overtake the whole structured population [29]. In the following, we will use graphical EGT to formulate the dynamic
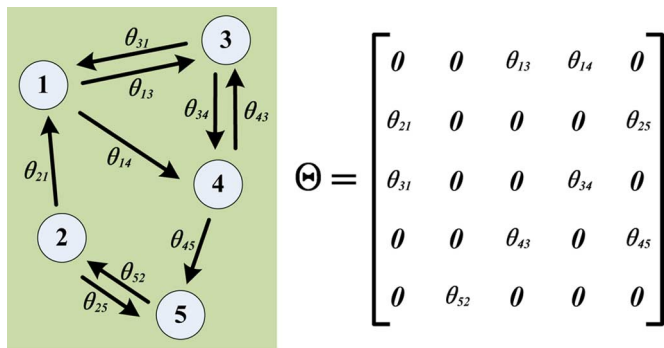
Fig. 2. Graphical evolutionary game model.

TABLE II
CORRESPONDENCE BETWEEN GRAPHICAL EGT AND DISTRIBUTED ADAPTIVE NETWORK

Graphical EGT	Distributed adaptive network
N Players	N Nodes in the network
Pure strategy of player i with n_i neighbors $\{i_1, i_2, ..., i_{n_i}\}$	Node i combines information from one of its neighbors $\{i_1, i_2, ..., i_{n_i}\}$
Mixed strategy of player i with n_i neighbors $\{p_1, p_2, ..., p_{n_i}\}$	Node i's combiner (Weight) $\{A_i(1), A_i(2), ..., A_i(n_i)\}$
Mixed strategy update of player i	Combiner update of node i
Equilibrium	Convergence network state

parameter updating process in a distributed adaptive filter network.

B. Graphical Evolutionary Game Formulation

In graphical EGT, each player updates strategy according to his/her fitness after interacting with neighbors in each round. Similarly, in distributed adaptive filtering, each node updates its parameter $\boldsymbol{w}$ through incorporating the neighbors' information. In such a case, we can treat the nodes in a distributed filter network as players in a graphical evolutionary game. For node i with n_i neighbors, it has n_i pure strategies $\{i_1, i_2, \ldots, i_{n_i}\}$, where strategy j means updating $\boldsymbol{w}_{i,t+1}$ using the updated information from its neighbor j, $A_{i,t+1}(j)$. We can see that (5) represents the adoption of mixed strategy. In such a case, the parameter updating in distributed adaptive filter network can be regarded as the strategy updating in graphical EGT. Table II summarizes the correspondence between the terminologies in graphical EGT and those in distributed adaptive network.

We first discuss how players' strategies are updated in graphical EGT, which is then applied to the parameter updating in distributed adaptive filtering. In graphical EGT, the fitness of a player is locally determined from interactions with all adjacent players, which is defined as [30]

$$f = (1 - \alpha) \cdot B + \alpha \cdot U, \tag{9}$$

where B is the baseline fitness, which represents the player's inherent property. For example, in a distributed adaptive network, a node's baseline fitness can be interpreted as the quality
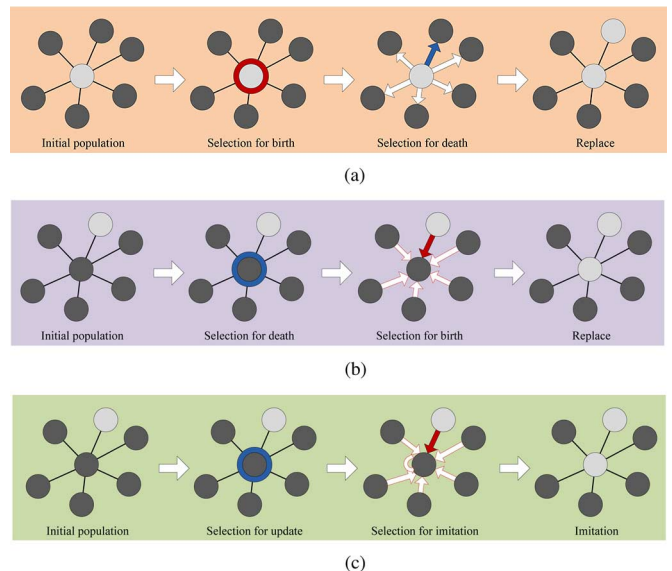

Fig. 3. Three different update rules, where death selections are shown in dark blue and birth selections are shown in red. (a) BD update rule. (b) DB update rule. (c) IM update rule.

of its noise variance. U is the player's utility which is determined by the predefined utility matrix. The parameter α represents the selection intensity, i.e., the relative contribution of the game to fitness. The case $\alpha \to 0$ represents the limit of weak selection [31], while $\alpha = 1$ denotes strong selection, where fitness equals utility. There are three different strategy updating rules for the evolution dynamics, called as birth-death (BD), death-birth (DB) and imitation (IM) [32].

- BD update rule: a player is chosen for reproduction with the probability being proportional to fitness (Birth process). Then, the chosen player's strategy replaces one neighbor's strategy uniformly (Death process), as shown in Fig. 3(a).
- DB update rule: a random player is chosen to abandon his/her current strategy (Death process). Then, the chosen player adopts one of his/her neighbors' strategies with the probability being proportional to their fitness (Birth process), as shown in Fig. 3(b).
- IM update rule: each player either adopts the strategy of one neighbor or remains with his/her current strategy, with the probability being proportional to fitness, as shown in Fig. 3(c).

These three kinds of strategy updating rules can be matched to three different kinds of parameter updating algorithms in distributed adaptive filtering. Suppose that there are N nodes in a structured network, where the degree of node i is n_i. We use $\mathcal{N}$ to denote the set of all nodes and $\mathcal{N}_i$ to denote the neighborhood set of node i, including node i itself.

For the BD update rule, the probability that node i adopts strategy j, i.e., using updated information from its neighbor node j, is

$$P_j = \frac{f_j}{\sum_{k \in \mathcal{N}} f_k} \frac{1}{n_j}, \tag{10}$$

where the first term $\frac{f_j}{\sum_{k \in \mathcal{N}} f_k}$ is the probability that the neighboring node j is chosen to reproduction, which is proportional to its fitness f_j, and the second term $\frac{1}{n_j}$ is the probability that node i is chosen for adopting strategy j. Note that the network topology information (n_j) is required to calculate (10). In such a case, the equivalent parameter updating rule for node i can be written by

$$\boldsymbol{w}_{i,t+1} = \sum_{j \in \mathcal{N}_i \backslash \{i\}} \left(\frac{f_j}{\sum_{k \in \mathcal{N}} f_k} \frac{1}{n_j} \right) F(\boldsymbol{w}_{j,t})$$
$$+ \left(1 - \sum_{j \in \mathcal{N}_i \backslash \{i\}} \left(\frac{f_j}{\sum_{k \in \mathcal{N}} f_k} \frac{1}{n_j} \right) \right) F(\boldsymbol{w}_{i,t}). \quad (11)$$

Similarly, for the DB updating rule, we can obtain the corresponding parameter updating rule for node i as

$$\boldsymbol{w}_{i,t+1} = \frac{1}{n_i} \sum_{j \in \mathcal{N}_i \backslash \{i\}} \left(\frac{f_j}{\sum_{k \in \mathcal{N}_i} f_k} \right) F(\boldsymbol{w}_{j,t})$$
$$+ \left(1 - \frac{1}{n_i} \sum_{j \in \mathcal{N}_i \backslash \{i\}} \left(\frac{f_j}{\sum_{k \in \mathcal{N}_i} f_k} \right) \right) F(\boldsymbol{w}_{i,t}). \quad (12)$$

For the IM updating rule, we have

$$\boldsymbol{w}_{i,t+1} = \sum_{j \in \mathcal{N}_i} \left(\frac{f_j}{\sum_{k \in \mathcal{N}_i} f_k} \right) F(\boldsymbol{w}_{j,t}). \quad (13)$$

Note that (11), (12) and (13) are expected outcome of BD, DB and IM updated rules, which can be referred in [35], [37].

The performance of adaptive filtering algorithm is usually evaluated by two measures: mean-square deviation (MSD) and excess-mean-square error (EMSE), which are defined as

$$\text{MSD} = \text{E} \left\| \boldsymbol{w}_t - \boldsymbol{w}^0 \right\|^2, \quad (14)$$
$$\text{EMSE} = \text{E} \left| \boldsymbol{u}_t (\boldsymbol{w}_{t-1} - \boldsymbol{w}^0) \right|^2. \quad (15)$$

Using (11), (12) and (13), we can calculate the network MSD and EMSE of these three update rules according to [6].

C. Relationship to Existing Distributed Adaptive Filtering Algorithms

In Section II, we have summarized the existing distributed adaptive filtering algorithms in (5) and Table I. In this subsection, we will show that all these algorithms are the special cases of the IM update rule in our proposed graphical EGT framework. Compare (5) and (13), we can see that different fitness definitions are corresponding to different distributed adaptive filtering algorithms in Table I. For the uniform rule, the fitness can be uniformly defined as $f_i = 1$ and using the IM update rule, we have

$$\boldsymbol{w}_{i,t+1} = \sum_{j \in \mathcal{N}_i} \frac{1}{n_i} F(\boldsymbol{w}_{j,t}), \quad (16)$$

which is equivalent to the uniform rule in Table I. Here, the definition of $f_i = 1$ means the adoption of fixed fitness and

TABLE III
DIFFERENT FITNESS DEFINITIONS

Name	Fitness: $f_j =$
Uniform [11][13]	1, for all $j \in \mathcal{N}_i$
Maximum degree [8][14]	$\begin{cases} 1, & \text{for } j \neq i, \\ N - n_i + 1, & \text{for } j = i. \end{cases}$
Laplacian [15][16]	$\begin{cases} 1, & \text{for } j \neq i \\ n_{\max} - n_i + 1, & \text{for } j = i. \end{cases}$
Relative degree [8]	n_j, for all $j \in \mathcal{N}_i$
Relative degree-variance [6]	$n_j \sigma_j^{-2}$, for all $j \in \mathcal{N}_i$
Metropolis [16][17]	$\begin{pmatrix} 1 - \sum_{k \neq i} A_i(k) & \frac{1}{\max\{|\mathcal{N}_i|, |\mathcal{N}_j|\}} \\ \frac{1}{\max\{|\mathcal{N}_i|, |\mathcal{N}_j|\}} & 1 - \sum_{k \neq j} A_j(k) \end{pmatrix}$
Hastings [17]	$\begin{pmatrix} 1 - \sum_{k \neq i} A_i(k) & \frac{\sigma_{(i,j)}^2}{\max\{|\mathcal{N}_i|\sigma_i^2, |\mathcal{N}_j|\sigma_j^2\}} \\ \frac{\sigma_{(j,i)}^2}{\max\{|\mathcal{N}_i|\sigma_i^2, |\mathcal{N}_j|\sigma_j^2\}} & 1 - \sum_{k \neq j} A_j(k) \end{pmatrix}$

weak selection $(\alpha \ll 1)$. For the Laplacian rule, when updating the parameter of node i, the fitness of nodes in $\mathcal{N}_i$ can be defined as

$$f_j = \begin{cases} 1, & \text{for } j \neq i, \\ n_{\max} - n_i + 1, & \text{for } j = i. \end{cases} \quad (17)$$

From (17), we can see that each node gives more weight to the information from itself through enhancing its own fitness. Similarly, for the Relative-degree-variance rule, the fitness can be defined as

$$f_j = n_j \sigma_j^{-2}, \quad \text{for all } j \in \mathcal{N}_i. \quad (18)$$

For the metropolis rule and Hastings rule, the corresponding fitness definitions are based on strong selection model $(\alpha \to 1)$, where utility plays a dominant role in (9). For the metropolis rule, the utility matrix of nodes can be defined as

$$\begin{array}{cc} \text{Node } i & \text{Node } j \neq i \end{array}$$
$$\begin{array}{c} \text{Node } i \\ \text{Node } j \neq i \end{array} \begin{pmatrix} 1 - \sum_{k \neq i} A_i(k) & \frac{1}{\max\{|\mathcal{N}_i|, |\mathcal{N}_j|\}} \\ \frac{1}{\max\{|\mathcal{N}_i|, |\mathcal{N}_j|\}} & 1 - \sum_{k \neq j} A_j(k) \end{pmatrix}$$
$$(19)$$

For the Hastings rule, the utility matrix can be defined as

$$\begin{array}{cc} \text{Node } i & \text{Node } j \neq i \end{array}$$
$$\begin{array}{c} \text{Node } i \\ \text{Node } j \neq i \end{array} \begin{pmatrix} 1 - \sum_{k \neq i} A_i(k) & \frac{\sigma_{(j,i)}^2}{\max\{|\mathcal{N}_i|\sigma_i^2, |\mathcal{N}_j|\sigma_j^2\}} \\ \frac{\sigma_{(i,j)}^2}{\max\{|\mathcal{N}_i|\sigma_i^2, |\mathcal{N}_j|\sigma_j^2\}} & 1 - \sum_{k \neq j} A_j(k) \end{pmatrix}$$
$$(20)$$

Table III summarizes different fitness definitions corresponding to different combination rules in Table I. Therefore, we can see that the existing algorithms can be summarized into our proposed graphical EGT framework with corresponding fitness definitions.

D. Error-Aware Distributed Adaptive Filtering Algorithm

To illustrate our graphical EGT framework, as examples, we further design two distributed adaptive algorithms by choosing different fitness functions. As discussed in Section II, the existing distributed adaptive filtering algorithms either rely on the prior knowledge of network topology or the requirement of additional network statistics. All of them are not robust to a dynamic network, where a node location may change and the noise variance of each node may also vary with time. Considering these problems, we propose error-aware algorithms based on the intuition that neighbors with low mean-square-error (MSE) should be given more weight while neighbors with high MSE should be given less weight. The instantaneous error of node i, denoted by ϱ_i, can be calculated by

$$\varrho_{i,t} = |d_i(t) - \boldsymbol{u}_{i,t}\boldsymbol{w}_{i,t-1}|^2, \tag{21}$$

where only local data $\{d_i(t), \boldsymbol{u}_{i,t}\}$ are used. The approximated MSE of node i, denoted by β_i, can be estimated by following update rule in each time slot,

$$\beta_{i,t} = (1 - \nu_{i,t})\beta_{i,t-1} + \nu_{i,t}\varrho_{i,t}, \tag{22}$$

where $\nu_{i,t}$ is a positive parameter. We assume that nodes can exchange their instantaneous MSE information with neighbors. Based on the estimated MSE, we design two kinds of fitness: exponential form and power form as follows:

$$\text{Power}: \quad f_i = \beta_i^{-\lambda}, \tag{23}$$
$$\text{Exponential}: \quad f_i = e^{-\lambda\beta_i}, \tag{24}$$

where λ is a positive coefficient. Note that the fitness defined in (23) and (24) are just two examples of our proposed framework, while many other forms of fitness can be considered, e.g., $f_i = \log(\lambda\beta_i^{-1})$. Using the IM update rule, we have

$$\boldsymbol{w}_{i,t+1} = \sum_{j \in \mathcal{N}_i} \frac{\beta_{j,t}^{-\lambda}}{\sum_{k \in \mathcal{N}_i} \beta_{k,t}^{-\lambda}} F(\boldsymbol{w}_{j,t}), \tag{25}$$

$$\boldsymbol{w}_{i,t+1} = \sum_{j \in \mathcal{N}_i} \frac{e^{-\lambda\beta_{j,t}}}{\sum_{k \in \mathcal{N}_i} e^{-\lambda\beta_{k,t}}} F(\boldsymbol{w}_{j,t}). \tag{26}$$

From (25) and (26), we can see that the proposed algorithms do not directly depend on any network topology information. Moreover, they can also adapt to a dynamic environment when the noise variance of nodes are unknown or suddenly change, since the weights can be immediately adjusted accordingly. In [33], a similar algorithm was also proposed based on the instantaneous MSE information, which is a special case of our error-aware algorithm with power form of $\lambda = 2$. Note that the deterministic coefficients are adopted when implementing (25) and (26), instead of using random combining efficient with some probability. However, the algorithm can also be implemented using a random selection with probabilities. There will be no performance loss since the expected outcome is the same, but the efficiency (convergence speed) will be lower. In Section V, we will verify the performance of the proposed algorithm through simulation.
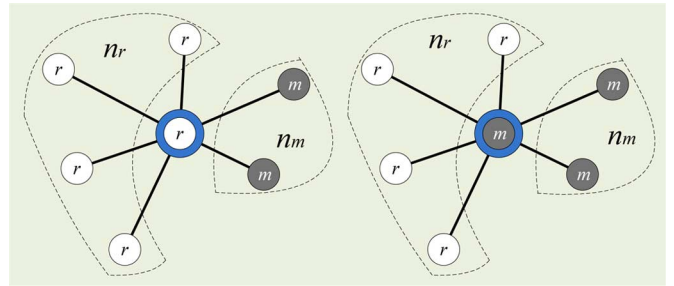

Fig. 4. Graphical evolutionary game model.

IV. DIFFUSION ANALYSIS

In a distributed adaptive filter network, there are nodes with good signals, i.e., lower noise variance, as well as nodes with poor signals. The principal objective of distributed adaptive filtering algorithms is to stimulate the diffusion of good signals to the whole network to enhance the network performances. In this section, we will use the EGT to analyze such a dynamic diffusion process and derive the close-form expression for the diffusion probability. In the following diffusion analysis, we assume that all nodes have the same regressor statistics $\mathbf{R_u}$, but different noise statistics.

In a graphical evolutionary game, the structured population are either residents or mutants. An important concept is the fixation probability, which represents the probability that the mutant will eventually overtake the whole population [34]. Let us consider a local adaptive filter network as shown in Fig. 4, where the hollow points denote common nodes, i.e., nodes with common noise variance σ_r^2; and the solid points denote good nodes, i.e., nodes with a lower noise variance σ_m^2. σ_r^2 and σ_m^2 satisfy that $\sigma_r^2 \gg \sigma_m^2$. Here, we adopt the binary signal model to better reveal the diffusion process of good signals. If we regard the common nodes as residents and the good nodes as mutants, the concept of fixation probability in EGT can be applied to analyze the diffusion of good signals in the network. According to the definition of fixation probability, we define the diffusion probability in a distributed filter network as the probability that a good signal can be adopted by all nodes to update parameters in the network.

A. Strategies and Utility Matrix

As shown in Fig. 4, for the node at the center, its neighbors include both common nodes and good nodes. When the center node updates its parameter $\boldsymbol{w}_i$, it has the following two possible strategies:

$$\begin{cases} \boldsymbol{S}_r, & \text{using information from common nodes,} \\ \boldsymbol{S}_m, & \text{using information from good nodes.} \end{cases} \tag{27}$$

In such a case, we can define the utility matrix as follow:

$$\begin{array}{cc} & \begin{array}{cc} \boldsymbol{S}_r & \quad\quad \boldsymbol{S}_m \end{array} \\ \begin{array}{c} \boldsymbol{S}_r \\ \boldsymbol{S}_m \end{array} & \begin{pmatrix} \pi^{-1}(\sigma_r, \sigma_r) & \pi^{-1}(\sigma_m, \sigma_r) \\ \pi^{-1}(\sigma_r, \sigma_m) & \pi^{-1}(\sigma_m, \sigma_m) \end{pmatrix} = \begin{pmatrix} u_1 & u_2 \\ u_3 & u_4 \end{pmatrix}, \end{array} \tag{28}$$

where $\pi(x, y)$ represents the steady EMSE of node with noise variance x^2 using information from node with noise variance y^2. For example, $\pi(\sigma_r, \sigma_m)$ is the steady EMSE of node with

noise variance σ_r^2 adopting strategy $\boldsymbol{S}_m$, i.e., updating its $\boldsymbol{w}$ using information from node with noise variance σ_m^2 which in turn adopts strategy $\boldsymbol{S}_r$. In our diffusion analysis, we assume that only two players are interacting with each other at one time instant, i.e., there are two nodes exchanging and combining information with each other at one time instant. In such a case, the payoff matrix is two-user case. Note that a node chooses one specific neighbor with some probability, which is equivalent to the weight that the node gives to that neighbor.

Since the steady EMSE $\pi(x, y)$ in the utility matrix is determined by the information combining rule, there is no general expressions for $\pi(x, y)$. Nevertheless, by intuition, we know that the steady EMSE of node with variance σ_r^2 should be larger than that of node with variance σ_m^2 since $\sigma_r^2 \gg \sigma_m^2$, and adopting strategy $\boldsymbol{S}_m$ should be more beneficial than adopting strategy $\boldsymbol{S}_r$ since the node can obtain better information from others, i.e., $\pi(\sigma_r, \sigma_r) > \pi(\sigma_r, \sigma_m) > \pi(\sigma_m, \sigma_r) > \pi(\sigma_m, \sigma_m)$. Therefore, we assume that the utility matrix defined in (28) has the quality as follow

$$u_1 < u_3 < u_2 < u_4. \tag{29}$$

Here, we use an example in [17] to have a close-form expression for $\pi(x, y)$ to illustrate and verify this intuition. According to [17], with sufficiently small step size μ, the optimal $\pi(x, y)$ can be calculated by

$$\pi(x, y) = c_1 \sigma_1^2 + c_2 \frac{x^4}{\sigma_2^2}, \tag{30}$$

$$\begin{cases} c_1 = \frac{\mu \text{Tr}(\boldsymbol{R}_u)}{4}, & c_2 = \frac{\mu^2 \|\zeta\|^2}{2}, \\ \sigma_1^2 = \frac{2x^2 y^2}{x^2 + y^2}, & \sigma_2^2 = \frac{x^2 y^2}{2}, \end{cases} \tag{31}$$

where $\zeta = \text{col}\{\zeta_1, \ldots, \zeta_N\}$ consists of the eigenvalues of $\boldsymbol{R}_u$ (recall that $\boldsymbol{R}_u$ is the covariance matrix of the observed regression data $\boldsymbol{u}_t$). According to (30) and (31), we have

$$\pi(\sigma_r, \sigma_r) = c_1 \sigma_r^2 + 2c_2, \tag{32}$$

$$\pi(\sigma_r, \sigma_m) = c_1 \frac{2\sigma_m^2 \sigma_r^2}{\sigma_m^2 + \sigma_r^2} + 2c_2 \frac{\sigma_r^2}{\sigma_m^2}, \tag{33}$$

$$\pi(\sigma_m, \sigma_r) = c_1 \frac{2\sigma_m^2 \sigma_r^2}{\sigma_m^2 + \sigma_r^2} + 2c_2 \frac{\sigma_m^2}{\sigma_r^2}, \tag{34}$$

$$\pi(\sigma_m, \sigma_m) = c_1 \sigma_m^2 + 2c_2. \tag{35}$$

Suppose $\sigma_m^2 = \tau \sigma_r^2$, through comparing (32)–(35), we can derive the condition for $\pi(\sigma_r, \sigma_r) > \pi(\sigma_r, \sigma_m) > \pi(\sigma_m, \sigma_r) > \pi(\sigma_m, \sigma_m)$ as follows

$$\mu < \frac{\tau \text{Tr}(\mathbf{R}_u) \sigma_r^2}{4(1 + \tau)\|\zeta\|^2}. \tag{36}$$

According to [17], the derivation of optimal $\pi(x, y)$ in (30) and (31) is based on the assumption that μ is sufficiently small. Therefore, the condition of μ in (36) holds. In such a case, we can conclude that $\pi(\sigma_r, \sigma_r) > \pi(\sigma_r, \sigma_m) > \pi(\sigma_m, \sigma_r) > \pi(\sigma_m, \sigma_m)$, which implies that $u_1 < u_3 < u_2 < u_4$.

In the following, we will analyze the diffusion process of strategy $\boldsymbol{S}_m$, i.e., the ability of good signals diffusing over the whole network. We consider an adaptive filter network based on a homogenous graph with general degree n and adopt the IM update rule for the parameter update [35]. Let p_r and p_m

denote the percentages of nodes using strategies $\boldsymbol{S}_r$ and $\boldsymbol{S}_m$ in the population, respectively. Let p_{rr}, p_{rm}, p_{mr} and p_{mm} denote the percentages of edge, where p_{rm} means the percentage of edge on which both nodes use strategy $\boldsymbol{S}_r$ and $\boldsymbol{S}_m$. Let $q_{m\,|\,r}$ denote the conditional probability of a node using strategy $\boldsymbol{S}_m$ given that the adjacent node is using strategy $\boldsymbol{S}_r$, similar we have $q_{r\,|\,r}$, $q_{r\,|\,m}$ and $q_{m\,|\,m}$. In such a case, we have

$$p_r + p_m = 1, \quad q_{r\,|\,X} + q_{m\,|\,X} = 1, \tag{37}$$

$$p_{XY} = p_Y \cdot q_{X\,|\,Y}, \quad p_{rm} = p_{mr}, \tag{38}$$

where X and Y are either r or m. The equations (37)–(38) imply that the state of the whole network can be described by only two variables, p_m and $q_{m\,|\,m}$. In the following, we will calculate the dynamics of p_m and $q_{m\,|\,m}$ under the IM update rule.

B. Dynamics of p_m and $q_{m\,|\,m}$

In order to derive the diffusion probability, we first need to analyze the diffusion process of the system. As discussed in the previous subsection, the system dynamics under IM update rule can be represented by parameters p_m and $q_{m\,|\,m}$. Thus, in this subsection, we will first analyze the dynamics of p_m and $q_{m\,|\,m}$ to understand the dynamic diffusion process of the adaptive network. According to the IM update rule, a node using strategy $\boldsymbol{S}_r$ is selected for imitation with probability p_r. As shown in the left part of Fig. 4, among its n neighbors (not including itself), there are n_r nodes using strategy $\boldsymbol{S}_r$ and n_m nodes using strategy $\boldsymbol{S}_m$, respectively, where $n_r + n_m = n$. The percentage of such a configuration is $\binom{n}{n_m} q_{m\,|\,r}^{n_m} q_{r\,|\,r}^{n_r}$. In such a case, the fitness of this node is

$$f_0 = (1 - \alpha) + \alpha(n_r u_1 + n_m u_2), \tag{39}$$

where the baseline fitness is normalized as 1. We can see that (39) includes the normalized baseline fitness and also the fitness from utility, which is the standard definition of fitness used in the EGT filed, as shown in (9). Among those n neighbors, the fitness of node using strategy $\boldsymbol{S}_m$ is

$$f_m = (1 - \alpha) + \alpha([(n-1)q_{r\,|\,m} + 1]u_3 \\ + (n-1)q_{m\,|\,m}u_4), \tag{40}$$

and the fitness of node using strategy $\boldsymbol{S}_r$ is

$$f_r = (1 - \alpha) + \alpha([(n-1)q_{r\,|\,r} + 1]u_1 \\ + (n-1)q_{m\,|\,r}u_2). \tag{41}$$

In such a case, the probability that the node using strategy $\boldsymbol{S}_r$ is replaced by $\boldsymbol{S}_m$ is

$$P_{r \to m} = \frac{n_m f_m}{n_m f_m + n_r f_r + f_0}. \tag{42}$$

Therefore, the percentage of nodes using strategy $\boldsymbol{S}_m$, p_m, increases by $1/N$ with probability

$$\text{Prob}\left(\Delta p_m = \frac{1}{N}\right) = p_r \sum_{n_r + n_m = n} \binom{n}{n_m} q_{m\,|\,r}^{n_m} q_{r\,|\,r}^{n_r} \\ \cdot \frac{n_m f_m}{n_m f_m + n_r f_r + f_0}. \tag{43}$$

Meanwhile, the edges that both nodes use strategy $\boldsymbol{S}_m$ increase by n_m, thus, we have

$$
\text{Prob}\left(\Delta p_{mm} = \frac{2n_m}{nN}\right) = p_r \binom{n}{n_m} q_{m\,|\,r}^{n_m} q_{r\,|\,r}^{n_r}
$$
$$
\cdot \frac{n_m f_m}{n_m f_m + n_r f_r + f_0}. \quad (44)
$$

Similar analysis can be applied to the node using strategy $\boldsymbol{S}_m$. According to the IM update rule, a node using strategy $\boldsymbol{S}_m$ is selected for imitation with probability p_m. As shown in the right part of Fig. 4, we also assume that there are n_r nodes using strategy $\boldsymbol{S}_r$ and n_m nodes using strategy $\boldsymbol{S}_m$ among its n neighbors. The percentage of such a phenomenon is $\binom{n}{n_m} q_{m\,|\,m}^{n_m} q_{r\,|\,m}^{n_r}$. Thus, the fitness of this node is

$$
g_0 = (1 - \alpha) + \alpha(n_r u_2 + n_m u_3). \quad (45)
$$

Among those n neighbors, the fitness of node using strategy $\boldsymbol{S}_m$ is

$$
g_m = (1 - \alpha) + \alpha((n-1)q_{r\,|\,m} u_3
$$
$$
+ [(n-1)q_{m\,|\,m} + 1]u_4), \quad (46)
$$

and the fitness of node using strategy $\boldsymbol{S}_r$ is

$$
g_r = (1 - \alpha) + \alpha((n-1)q_{r\,|\,r} u_1
$$
$$
+ [(n-1)q_{m\,|\,r} + 1]u_2). \quad (47)
$$

In such a case, the probability that the node using strategy $\boldsymbol{S}_m$ is replaced by $\boldsymbol{S}_r$ is

$$
P_{m \to r} = \frac{n_r g_r}{n_m g_m + n_r g_r + g_0}. \quad (48)
$$

Therefore, the percentage of nodes using strategy $\boldsymbol{S}_m$, p_m, decreases by $1/N$ with probability

$$
\text{Prob}\left(\Delta p_m = -\frac{1}{N}\right) = p_m \sum_{n_r + n_m = n} \binom{n}{n_m} q_{m\,|\,m}^{n_m} q_{r\,|\,m}^{n_r}
$$
$$
\cdot \frac{n_r g_r}{n_m g_m + n_r g_r + g_0}. \quad (49)
$$

Meanwhile, the edges that both nodes use strategy $\boldsymbol{S}_m$ decrease by n_m, thus, we have

$$
\text{Prob}\left(\Delta p_{mm} = -\frac{2n_m}{nN}\right) = p_m \binom{n}{n_m} q_{m\,|\,m}^{n_m} q_{r\,|\,m}^{n_r}
$$
$$
\cdot \frac{n_r g_r}{n_m g_m + n_r g_r + g_0}. \quad (50)
$$

Combining (43) and (49), we have the dynamic of p_m as

$$
\dot{p}_m = \frac{1}{N}\text{Prob}\left(\Delta p_m = \frac{1}{N}\right)
$$
$$
- \frac{1}{N}\text{Prob}\left(\Delta p_m = -\frac{1}{N}\right)
$$
$$
= \frac{\alpha n(n-1)p_{rm}}{N(n+1)^2}(\gamma_1 u_1 + \gamma_2 u_2 + \gamma_3 u_3 + \gamma_4 u_4) + O(\alpha^2),
$$
$$
(51)
$$

where the second equality is according to Taylor's Theorem and weak selection assumption with α goes to zero [36], and the parameters γ_1, γ_2, γ_3 and γ_4 are given as follows:

$$
\gamma_1 = -q_{r\,|\,r}[(n-1)(q_{r\,|\,r} + q_{m\,|\,m}) + 3], \quad (52)
$$

$$
\gamma_2 = -q_{m\,|\,m} - q_{m\,|\,r}[(n-1)(q_{r\,|\,r} + q_{m\,|\,m}) + 2] - \frac{2}{n-1}, \quad (53)
$$

$$
\gamma_3 = q_{r\,|\,r} + q_{r\,|\,m}[(n-1)(q_{r\,|\,r} + q_{m\,|\,m}) + 2] + \frac{2}{n-1}, \quad (54)
$$

$$
\gamma_4 = q_{m\,|\,m}[(n-1)(q_{r\,|\,r} + q_{m\,|\,m}) + 3]. \quad (55)
$$

In (51), the dot notation $\dot{p}_m$ represents the dynamic of p_m, i.e., the variation of p_m within a tiny period of time. In such a case, the utility obtained from the interactions is considered as limited contribution to the overall fitness of each player. On one hand, the results derived from weak selection often remain as valid approximations for larger selection strength [31]. On the other hand, the weak selection limit has a long tradition in theoretical biology [37]. Moreover, the weak selection assumption can help achieve a close-form analysis of diffusion process and better reveal how the strategy diffuses over the network. Similarly, by combining (44) and (50), we have the dynamics of p_{mm} as

$$
\dot{p}_{mm} = \sum_{n_m=0}^{n} \frac{2n_m}{nN}\text{Prob}\left(\Delta p_{mm} = \frac{2n_m}{nN}\right)
$$
$$
- \sum_{n_m=0}^{n} \frac{2n_m}{nN}\text{Prob}\left(\Delta p_{mm} = -\frac{2n_m}{nN}\right)
$$
$$
= \frac{2p_{rm}}{(n+1)N}(1 + (n-1)(q_{m\,|\,r} - q_{m\,|\,m})) + O(\alpha). \quad (56)
$$

Besides, we can also have the dynamics of $q_{m\,|\,m}$ as

$$
\dot{q}_{m\,|\,m} = \frac{d}{dt}\left(\frac{p_{mm}}{p_m}\right)
$$
$$
= \frac{2}{(n+1)N}\frac{p_{rm}}{p_m}(1 + (n-1)(q_{m\,|\,r} - q_{m\,|\,m})) + O(\alpha). \quad (57)
$$

C. Diffusion Probability Analysis

The dynamic equation of p_m in (51) reflects the dynamic of nodes updating $\boldsymbol{w}$ using information from good nodes, i.e., the diffusion status of good signals in the network. A positive $\dot{p}_m$ means that good signals are diffusing over the network, while a negative $\dot{p}_m$ means that good signals have not been well adopted. The diffusion probability of good signals is closely related to the noise variance of good nodes σ_m. Intuitively, the lower σ_m, the higher probability that good signals can spread the whole network. In this subsection, we will analyze the close-form expression for the diffusion probability.

As discussed at the beginning of Section IV, the state of whole network can be described by only p_m and $q_{m\,|\,m}$. In such a case, (51) and (57) can be re-written as functions of p_m and $q_{m\,|\,m}$

$$
\dot{p}_m = \alpha \cdot G_1(p_m, q_{m\,|\,m}) + O(\alpha^2), \quad (58)
$$
$$
\dot{q}_{m\,|\,m} = G_2(p_m, q_{m\,|\,m}) + O(\alpha). \quad (59)
$$

From (58) and (59), we can see that $q_{m\,|\,m}$ converges to equilibrium in a much faster rate than p_m under the assumption of weak selection. At the steady state of $q_{m\,|\,m}$, i.e., $\dot{q}_{m\,|\,m} = 0$, we have

$$
q_{m\,|\,m} - q_{m\,|\,r} = \frac{1}{n-1}. \quad (60)
$$

In such a case, the dynamic network will rapidly converge onto the slow manifold, defined by $G_2(p_m, q_{m\,|\,m}) = 0$. Therefore, we can assume that (60) holds in the whole convergence process of p_m. According to (37)–(38) and (60), we have

$$q_{m\,|\,m} = p_m + \frac{1}{n-1}(1 - p_m), \tag{61}$$

$$q_{m\,|\,r} = \frac{n-2}{n-1}p_m, \tag{62}$$

$$q_{r\,|\,m} = \frac{n-2}{n-1}(1 - p_m), \tag{63}$$

$$q_{r\,|\,r} = 1 - \frac{n-2}{n-1}p_m. \tag{64}$$

Therefore, the diffusion process can be characterized by only p_m. Thus, we can focus on the dynamics of p_m to derive the diffusion probability, which is given by following *Theorem 1*.

Theorem 1: In a distributed adaptive filter network which can be characterized by a N-node regular graph with degree n, suppose there are common nodes with noise variance σ_r and good nodes with noise variance σ_m, where each common node has connection edge with only one good node. If each node updates its parameter $\boldsymbol{w}$ using the IM update rule, the diffusion probability of the good signal can be approximated by

$$P_{\text{diff}} = \frac{1}{n+1} + \frac{\alpha nN}{6(n+1)^3}(\xi_1 u_1 + \xi_2 u_2 + \xi_3 u_3 + \xi_4 u_4), \tag{65}$$

where the parameters ξ_1, ξ_2, ξ_3 and ξ_4 are as follows:

$$\xi_1 = -2n^2 - 5n + 3, \quad \xi_2 = -n^2 - n - 3, \tag{66}$$

$$\xi_3 = 2n^2 + 2n - 3, \quad \xi_4 = n^2 + 4n + 3. \tag{67}$$

Proof: See Appendix. ∎

Using *Theorem 1*, we can calculate the diffusion probability of the good signals over the network, which can be used to evaluate the performance of an adaptive filter network. Similarly, the diffusion dynamics and probabilities under BD and DB update rules can also be derived using the same analysis. The following theorem shows an interesting result, which is based on an important theorem in [29], stating that evolutionary dynamics under BD, DB, and IM are equivalent for undirected regular graphs.

Theorem 2: In a distributed adaptive filter network which can be characterized by a N-node regular graph with degree n, suppose there are common nodes with noise variance σ_r and good nodes with noise variance σ_m, where each common node has connection edge with only one good node. If each node updates its parameter $\boldsymbol{w}$ using the IM update rule, the diffusion probabilities of good signals under BD and DB update rules are same with that under the IM update rule.

V. EVOLUTIONARILY STABLE STRATEGY

In the last section, we have analyzed the information diffusion process in an adaptive network under the IM update rule, and derived the diffusion probability of strategy $\boldsymbol{S}_m$ that using information from good nodes. On the other hand, considering that if the whole network has already chosen to adopt this favorable strategy $\boldsymbol{S}_m$, is the current state a stable network state, even though a small fraction of nodes adopt the other strategy $\boldsymbol{S}_r$? In the following, we will answer these questions using the concept of evolutionarily stable strategy (ESS) in evolutionary

game theory. As discussed in Section III-A, the ESS ensures that one strategy is resistant against invasion of another strategy [38]. In our system model, it is obvious that $\boldsymbol{S}_m$, i.e., using information from good nodes, is the favorable strategy and a desired ESS in the network. In this section, we will check whether strategy $\boldsymbol{S}_m$ is evolutionarily stable.

A. ESS in Complete Graphs

We first discuss whether strategy $\boldsymbol{S}_m$ is an ESS in complete graphs, which is shown by the following theorem.

Theorem 3: In a distributed adaptive filter network that can be characterized by complete graphs, strategy $\boldsymbol{S}_m$ is always an ESS strategy.

Proof: In a complete graph, each node meets every other node equally likely. In such a case, according to the utility matrix in (28), the average utilities of using strategies $\boldsymbol{S}_r$ and $\boldsymbol{S}_m$ are given by

$$U_r = p_r u_1 + p_m u_2, \tag{68}$$

$$U_m = p_r u_3 + p_m u_4, \tag{69}$$

where p_r and p_m are the percentages of population using strategies $\boldsymbol{S}_r$ and $\boldsymbol{S}_m$, respectively. Consider the scenario that the majority of the population adopt strategy $\boldsymbol{S}_m$, while a small fraction of the population adopt $\boldsymbol{S}_r$ which is considered as invasion, $p_r = \epsilon$. In such a case, according to the definition of ESS in (7), strategy $\boldsymbol{S}_m$ is evolutionary stable if $U_m > U_r$ for $(p_r, p_m) = (\epsilon, 1 - \epsilon)$, i.e.,

$$\epsilon(u_3 - u_1) + (1 - \epsilon)(u_4 - u_2) > 0. \tag{70}$$

For $\epsilon \to 0$, the left hand side of (70) is positive if and only if

$$``u_4 > u_2" \quad \text{or} \quad ``u_4 = u_2 \text{ and } u_3 > u_1". \tag{71}$$

The (71) gives the sufficient evolutionary stable condition of strategy $\boldsymbol{S}_m$. In our system, we have $u_4 > u_2 > u_3 > u_1$, which means that (71) always holds. Therefore, strategy $\boldsymbol{S}_m$ is always an ESS if the adaptive filter network is a complete graph. ∎

B. ESS in Incomplete Graphs

Let us consider an adaptive filter network which can be characterized by an incomplete regular graph with degree n. The following theorem shows that strategy $\boldsymbol{S}_m$ is always an ESS in such an incomplete graph.

Theorem 4: In a distributed adaptive filter network which can be characterized by a regular graph with degree n, strategy $\boldsymbol{S}_m$ is always an ESS strategy.

Proof: Using the pair approximation method [32], the replicator dynamics of strategies $\boldsymbol{S}_m$ and $\boldsymbol{S}_r$ on a regular graph of degree n can be approximated simply by

$$\dot{p}_r = p_r(p_r u_1' + p_m u_2' - \phi), \tag{72}$$

$$\dot{p}_m = p_m(p_r u_3' + p_m u_4' - \phi), \tag{73}$$

where $\phi = p_r p_r u_1' + p_r p_m(u_2' + u_3') + p_m p_m u_4'$ is the average utility, and u_1', u_2', u_3' and u_4' are given as follows:

$$\begin{cases} u_1' = u_1, \\ u_2' = u_2 + u', \\ u_3' = u_3 - u', \\ u_4' = u_4. \end{cases} \tag{74}$$

The parameter u' depends on the three update rules (IM, BD and DB), which is given by [32]

$$\text{IM}: \quad u' = \frac{(n+3)u_1 + u_2 - u_3 - (n+3)u_4}{(n+3)(n-2)}, \quad (75)$$

$$\text{BD}: \quad u' = \frac{(n+1)u_1 + u_2 - u_3 - (n+1)u_4}{(n+1)(n-2)}, \quad (76)$$

$$\text{DB}: \quad u' = \frac{u_1 + u_2 - u_3 - u_4}{n-2}. \quad (77)$$

In such a case, the equivalent utility matrix is

$$
\begin{array}{cc}
 & \boldsymbol{S}_r \qquad \boldsymbol{S}_m \\
\begin{array}{c} \boldsymbol{S}_r \\ \boldsymbol{S}_m \end{array} & \begin{pmatrix} u_1 & u_2 + u' \\ u_3 - u' & u_4 \end{pmatrix}.
\end{array} \quad (78)
$$

According to (71), the evolutionary stable condition for strategy $\boldsymbol{S}_m$ is

$$u_4 > u_2 + u'. \quad (79)$$

Since $u_1 < u_3 < u_2 < u_4$, we have $u' < 0$ for all three update rules. In such a case, (79) always holds, which means that strategy $\boldsymbol{S}_m$ is always an ESS strategy. This completes the proof of the theorem. ∎

VI. SIMULATION RESULTS

In this section, we develop simulations to compare the performances of different adaptive filtering algorithms, as well as to verify the derivation of information diffusion probability and the analysis of ESS.

A. Mean-Square Performances

The network topology used for simulation is shown in the left part of Fig. 5, where 20 randomly nodes are randomly located. The signal and noise power information of each node are also shown in the right part of Fig. 5, respectively. In the simulation, we assume that the regressors with size $M = 5$, are zero-mean Gaussian and independent in time and space. The unknown vector is set to be $\boldsymbol{w}^0 = \mathbb{1}_5/\sqrt{2}$ and the step size of the LMS algorithm at each node i is set as $\mu_i = 0.01$. All the simulation results are averaged over 500 independent runnings. All the performance comparisons are conducted among six different kinds of distributed adaptive filtering algorithms as follows:

- Relative degree algorithm [8];
- Hastings algorithm [17];
- Adaptive combiner algorithm [7];
- Relative degree-variance algorithm [6];
- Proposed error-aware algorithm with power form;
- Proposed error-aware algorithm with exponential form.

Among these algorithms, the adaptive combiner algorithm [7] and our proposed error-aware algorithm are based on dynamic combiners (weights), which are updated in each time slot. The difference is the updating rule, where the adaptive combiner algorithm in [7] uses optimization and projection method, and our proposed algorithms use the approximated EMSE information.

In the first comparison, we assume that the noise variance of each node is known by the Hastings and relative degree-variance algorithms. Fig. 6 shows the transient network-performance comparison results among six kinds of algorithms in
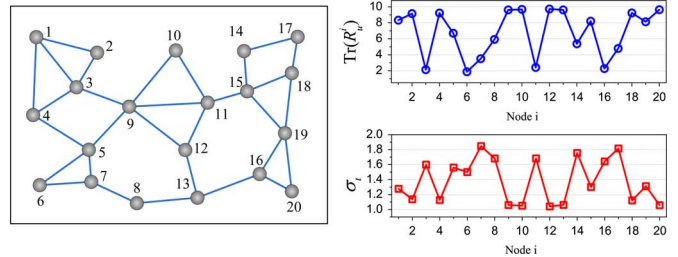

Fig. 5. Network information for simulation, including network topology for 20 nodes (left), trace of regressor covariance $\text{Tr}(\boldsymbol{R}_u)$ (right top) and noise variance σ_i (right bottom).
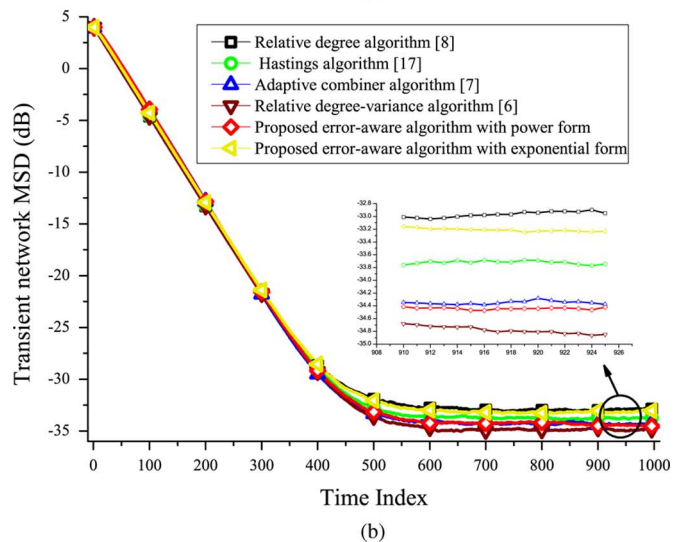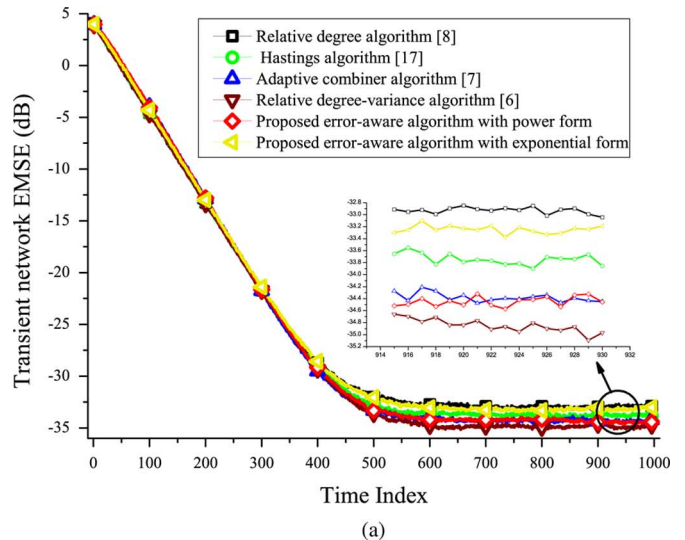

Fig. 6. Transient performances comparison with known noise variances. (a) Network EMSE. (b) Network MSD.

terms of EMSE and MSD. Under the similar convergence rate, we can see that the relative degree-variance algorithm performs the best. The proposed algorithm with exponential form performs better than the relative degree algorithm. With the power form fitness, the proposed algorithm can achieve similar performance, if not better than, compared with adaptive combiner algorithm, and both algorithms performs better than all other algorithms except the relative degree-variance algorithm. However, as discussed in Section 2, the relative degree-variance algorithm
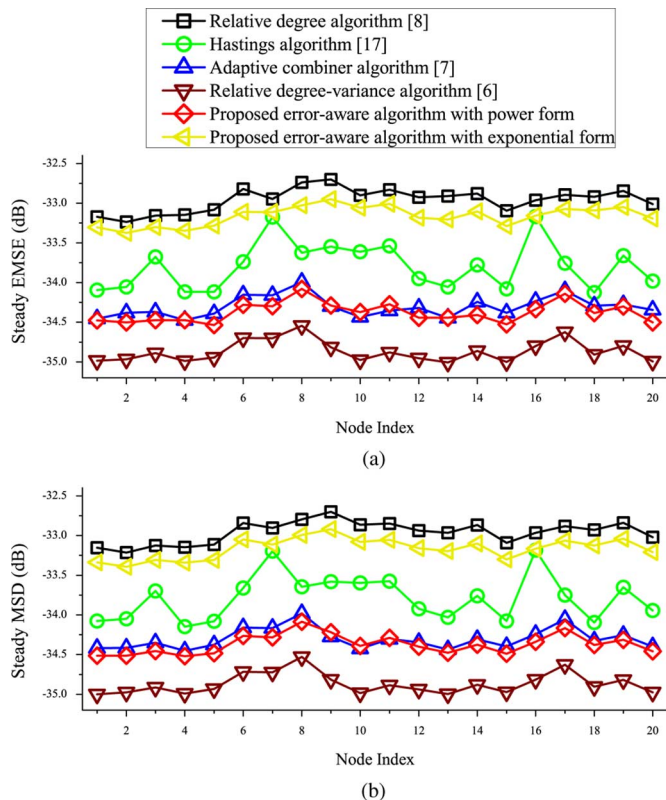
Fig. 7. Steady performances comparison with known noise variances. (a) Node's EMSE. (b) Node's MSD.

requires noise variance information of each node, while our proposed algorithm does not. Fig. 7 shows the corresponding steady-state performances of each node for six kinds of distributed adaptive filtering algorithms in terms of EMSE and MSD. Since the steady-state result is for each node, besides averaging over 500 independent runnings, we average at each node over 100 time slots after the convergence. We can see that the comparison results of steady-state performances are similar to those of the transient performances.

In the second comparison, we assume that the noise variance of each node is unknown, but can be estimated by the method proposed in [17]. Figs. 8 and 9 show the transient and steady-state performances for six kinds of algorithms in terms of EMSE and MSD under similar convergence rate. Since the noise variance estimation requires additional complexity, we also simulate the Hastings and relative degree-variance algorithms without variance estimation for fair comparison, where the noise variance is set as the network average variance, which is assumed to be prior information. Comparing with Fig. 7, we can see that when the noise variance information is not available, the performance degradation of relative degree-variance algorithm is significant, about 0.5 dB (12% more error) even with noise variance estimation, while the performance of Hastings algorithm degrades only a little since it relies less on the noise variance information. From Fig. 8(b), we can clearly see that when the variance estimation method is not adopted, our proposed algorithm with power form achieves the best performance. When the variance estimation method is adopted, the performances of our proposed algorithm with power form, the relative degree-variance and the adaptive combiner algorithm
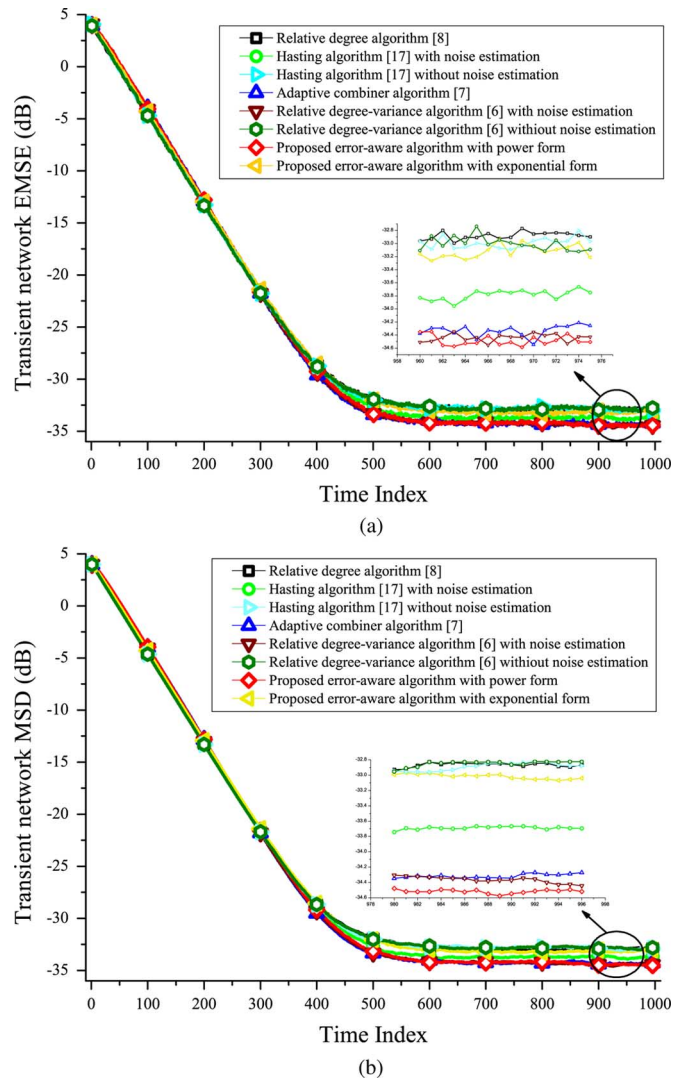

Fig. 8. Transient performances comparison with unknown noise variances. (a) Network EMSE. (b) Network MSD.

are similar, all of which perform better than other algorithms. Nevertheless, the complexity of both relative degree-variance algorithm with variance estimation and the adaptive combiner algorithm are higher than that of our proposed algorithm with power form. Such results immediately show the advantage of the proposed general framework. We should notice that more algorithms with better performances under certain criteria can be designed based on the proposed framework by choosing more proper fitness functions.

B. Diffusion Probability

In this subsection, we develop simulation to verify the diffusion probability analysis in Section IV. For the simulation setup, three types of regular graphs are generated with degree $n = 3$, 4 and 6, respectively, as shown in Fig. 10(a). All these three types of graphs are with $N = 100$ nodes, where each node's trace of regressor covariance is set to be $\mathrm{Tr}(\boldsymbol{R}_u) = 10$, the common nodes's noise variance is set as $\sigma_r^2 = 1.5$ and the good node's noise variance is set as $\sigma_m^2 \in [0.2, 0.8]$. In the simulation, the network is initialized with the state that all common nodes choosing strategy $\boldsymbol{S}_r$. Then, at each time
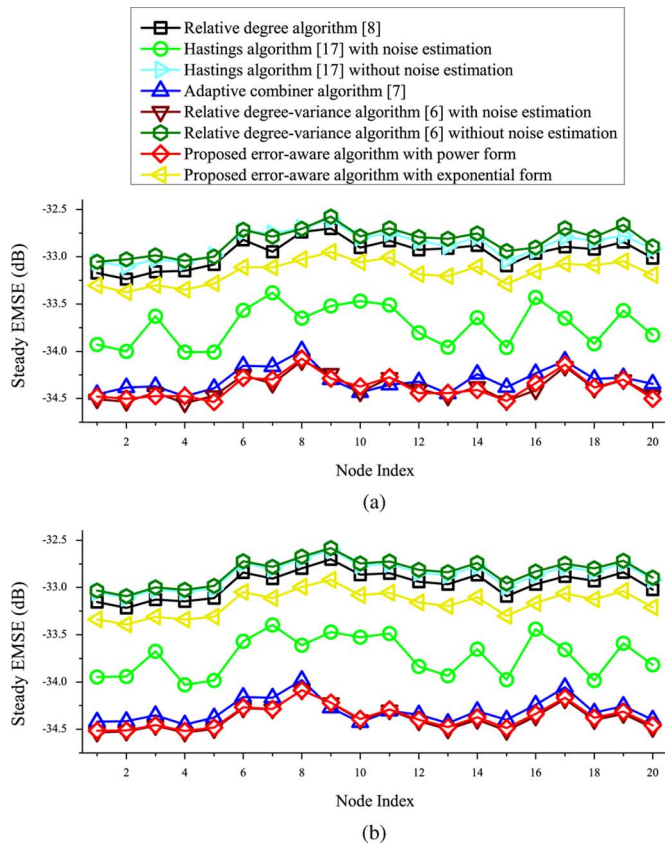
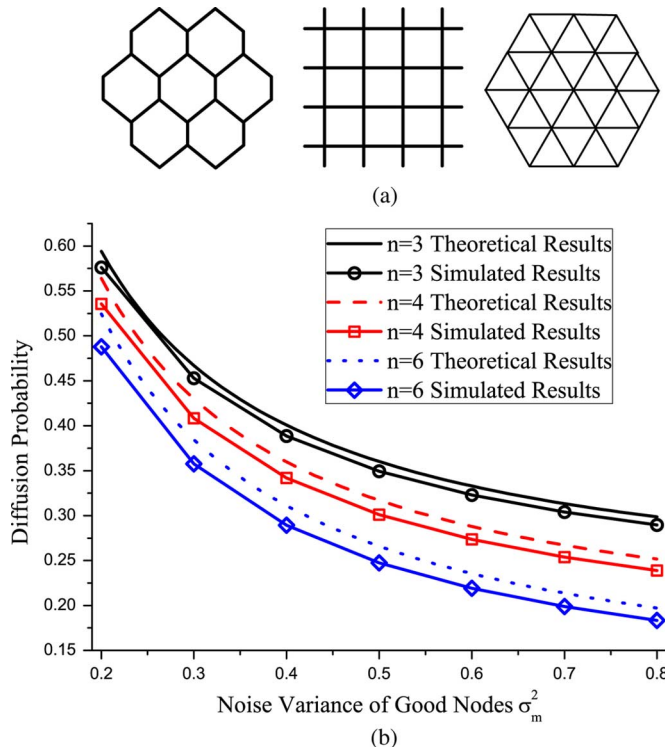Fig. 9. Steady performances comparison with unknown noise variances. (a) Node's EMSE. (b) Node's MSD.

Fig. 10. Diffusion probabilities under three types of regular graphs. (a) Regular graph structures with degree $n = 3$, 4 and 6. (b) Diffusion probability.
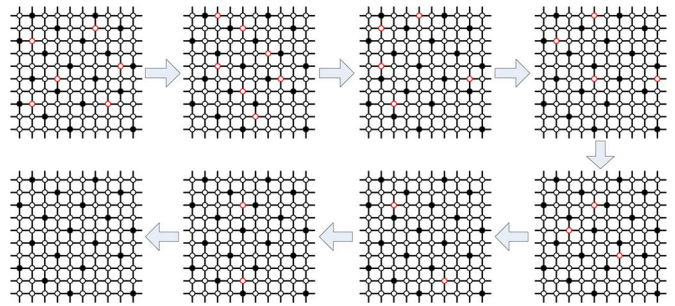

Fig. 11. Strategy updating process in a 10×10 grid network with degree $n = 4$ and number of nodes $N = 100$.

step, a randomly chosen node's strategy is updated according to the IM rules under weak selection ($w = 0.01$), as illustrated in Section III-B. The update steps are repeated until either strategy S_m has reached fixation or the number of steps has reach the limit. The diffusion probability is calculated by the fraction of runs where strategy S_m reached fixation out of 10^6 runs. Fig. 10(b) shows the simulation results, from which we can see that all the simulated results are basically accord with the corresponding theoretical results and the gaps are due to the approximation during the derivations. Moreover, we can see that the diffusion probability of good signal decreases along with the increase of its noise variance, i.e., better signal has better diffusion capability.

C. Evolutionarily Stable Strategy

To verify that strategy S_m is an ESS in the adaptive network, we further simulate the IM update rule on a 10×10 grid network with degree $n = 4$ and number of nodes $N = 100$, as shown in Fig. 11 where the hollow points represent common nodes and the solid nodes represent good nodes. In the simulation, all the settings are same with those in the simulation of diffusion probability in Section VI.B, except the initial network setting. The initial network state is set that the majority of nodes adopt strategy S_m denoted with black color (including both hollow and solid nodes) in Fig. 11, and only a very small percentage of nodes use strategy S_r denoted with red color. From the strategy updating process of the whole network illustrated in Fig. 11, we can see that the network finally abandons the unfavorable strategy S_r, which verifies the stability of strategy S_m.

VII. CONCLUSION

In this paper, we proposed an evolutionary game theoretic framework to offer a very general view of the distributed adaptive filtering problems and unify existing algorithms. Based on this framework, as examples, we further designed two error-aware adaptive filtering algorithms. Using the graphical evolutionary game theory, we analyzed the information diffusion process in the network under the IM update rule, and proved that the strategy of using information from nodes with good signal is always an ESS. We would like to emphasize that, unlike the traditional bottom-up approaches, the proposed graphical evolutionary game theoretic framework provides a top-down design

philosophy to understand the fundamentals of distributed adaptive algorithms. Such a top-down design philosophy is very important to the field of distributed adaptive signal process, since it offers a unified view of the formulation and can inspire more new distributed adaptive algorithms to be designed in the future.

APPENDIX
PROOF OF THEOREM 1

Proof: First, let us define $m(p_m)$ as the mean of the increment of p_m per unit time given as follows

$$m(p_m) = \frac{\dot{p}_m}{1/N}$$
$$\simeq \frac{\alpha n(n-2)}{(n-1)(n+1)^2} p_m(1-p_m)(ap_m + b). \quad (80)$$

where the second step is derived by substituting (60)–(64) into (51) and the parameters a and b are given as follows:

$$a = (n-2)(n+3)(u_1 - u_2 - u_3 + u_4), \quad (81)$$
$$b = -(n-1)(n+3)u_1 - 3u_2$$
$$+ (n^2 + n - 3)u_3 + (n+3)u_4. \quad (82)$$

We then define $v(p_m)$ as the variance of the increment of p_m per unit time, which can be calculated by

$$v(p_m) = \frac{\dot{p_m^2} - (\dot{p}_m)^2}{1/N}, \quad (83)$$

where $\dot{p_m^2}$ can be computed by

$$\dot{p_m^2} = \frac{1}{N^2} \left(\text{Prob}\left(\Delta p_m = \frac{1}{N} \right) \right.$$
$$\left. + \text{Prob}\left(\Delta p_m = -\frac{1}{N} \right) \right)$$
$$= \frac{2}{N^2} \frac{n(n-2)}{(n-1)(n+1)} p_m(1-p_m) + O(\alpha). \quad (84)$$

In such a case, $v(p_m)$ can be approximated by

$$v(p_m) \simeq \frac{2}{N} \frac{n(n-2)}{(n-1)(n+1)} p_m(1-p_m). \quad (85)$$

Suppose the initial percentage of good nodes in the network is p_{m0}. Let us define $H(p_{m0})$ as the probability that these good signals can finally be adopted by the whole network, i.e., all nodes can update their own $\boldsymbol{w}$ using information from good nodes. According to the backward Kolmogorov equation [39], $H(p_{m0})$ satisfies following differential equation

$$0 = m(p_{m0}) \frac{dH(p_{m0})}{dp_{m0}} + \frac{v(p_{m0})}{2} \frac{d^2 H(p_{m0})}{dp_{m0}^2}. \quad (86)$$

With the weak selection assumption, we can have the approximate solution of $H(p_{m0})$ as

$$H(p_{m0}) = p_{m0} + \frac{\alpha N}{6(n+1)} p_{m0}$$
$$\times (1 - p_{m0})((a + 3b) + ap_{m0}). \quad (87)$$

Let us consider the worst initial system state that each common node has connection with only one good node, i.e., $p_{m0} = \frac{1}{n+1}$, we have

$$H\left(\frac{1}{n+1} \right) \simeq \frac{1}{n+1} + \frac{\alpha n N}{6(n+1)^3}(a + 3b). \quad (88)$$

By substituting (81) and (82) into (88), we can have the close-form expression for the diffusion probability in (65). This completes the proof of the theorem. ∎

Remark: From (87), we can see that there are two terms constituting the expression of diffusion probability: the initial percentage of strategy $\boldsymbol{S}_m$, p_{m_0} (the initial system state) and the second term representing the changes of system state after beginning, in which $a + 3b$ determines whether p_m is increasing or decreasing along with the system updating. If $a + 3b < 0$, i.e., the diffusion probability is even lower than the initial percentage of strategy $\boldsymbol{S}_m$, the information from good nodes are shrinking over the network, instead of spreading. Therefore, $a + 3b > 0$ is more favorable for the improvement of the adaptive network performance.

REFERENCES

[1] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Distributed processing over adaptive networks," in *Proc. Adapt. Sens. Array Process. Workshop*, Lexington, MA, USA, Jun. 2006, pp. 1–3.

[2] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayed, "Detection, classification, and tracking of targets," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 17–29, 2002.

[3] F. C. R. Jr, M. L. R. de Campos, and S. Werner, "Distributed cooperative spectrum sensing with selective updating," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Bucharest, Romania, Aug. 2012, pp. 474–478.

[4] S. Haykin and K. J. R. Liu, *Handbook on Array Processing and Sensor Networks*. New York, NY, USA: IEEE-Wiley, 2009.

[5] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4064–4077, 2007.

[6] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, 2010.

[7] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4795–4810, 2010.

[8] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865–1877, 2008.

[9] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 97–123, 2011.

[10] R. L. G. Cavalcante, I. Yamada, and B. Mulgrew, "An adaptive projected subgradient approach to learning in diffusion networks," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2762–2774, 2009.

[11] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4692–4707, 2011.

[12] S.-Y. Tu and A. H. Sayed, "Mobile adaptive networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 649–664, 2011.

[13] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. Joint 44th IEEE Conf. Decision Contr. Eur. Contr. Conf. (CDC-ECC)*, Seville, Spain, Dec. 2005, pp. 2996–3000.

[14] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. Inf. Process. Sens. Netw. (IPSN)*, Los Angeles, CA, USA, Apr. 2005, pp. 63–70.

[15] D. S. Scherber and H. C. Papadopoulos, "Locally constructed algorithms for distributed computations in Ad Hoc networks," in *Proc. Inf. Process. Sens. Netw. (IPSN)*, Berkeley, CA, USA, Apr. 2004, pp. 11–19.

[16] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Contr. Lett.*, vol. 53, no. 1, pp. 65–78, 2004.

[17] X. Zhao and A. H. Sayed, "Performance limits for distributed estimation over LMS adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5107–5124, 2012.

[18] J. M. Smith, *Evolution and the Theory of Games*. Cambridge, U.K.: Cambridge Univ. Press, 1982.

[19] R. Cressman, *Evolutionary Dynamics and Extensive Form Games*. Cambridge, MA, USA: MIT Press, 2003.

[20] Y. Chen, Y. Gao, and K. J. R. Liu, "An evolutionary game-theoretic approach for image interpolation," in *Proc. IEEE ICASSP*, 2011, pp. 989–992.

[21] K. J. R. Liu and B. Wang, *Cognitive Radio Networking and Security: A Game Theoretical View*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[22] B. Wang, Y. Wu, and K. J. R. Liu, "Game theory for cognitive radio networks: An overview," *Comput. Netw.*, vol. 54, no. 14, pp. 2537–2561, 2010.

[23] E. H. Watanabe, D. Menasché, E. Silva, and R. M. Leão, "Modeling resource sharing dynamics of VoIP users over a WLAN using a game-theoretic approach," in *Proc. IEEE INFOCOM*, 2008, pp. 915–923.

[24] B. Wang, K. J. R. Liu, and T. C. Clancy, "Evolutionary cooperative spectrum sensing game: How to collaborate?," *IEEE Trans. Commun.*, vol. 58, no. 3, pp. 890–900, 2010.

[25] Y. Chen, B. Wang, W. S. Lin, Y. Wu, and K. J. R. Liu, "Cooperative peer-to-peer streaming: An evolutionary game-theoretic approach," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 20, no. 10, pp. 1346–1357, 2010.

[26] C. Jiang, Y. Chen, Y. Gao, and K. J. R. Liu, "Joint spectrum sensing and access evolutionary game in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 5, pp. 2470–2483, 2013.

[27] R. Fisher, *The Genetical Theory of Natural Selection*. New York, NY, USA: Clarendon, 1930.

[28] E. Lieberman, C. Hauert, and M. A. Nowak, "Evolutionary dynamics on graphs," *Nature*, vol. 433, pp. 312–316, 2005.

[29] P. Shakarian, P. Roos, and A. Johnson, "A review of evolutionary graph theory with applications to game theory," *Biosyst.*, vol. 107, no. 2, pp. 66–80, 2012.

[30] M. A. Nowak and K. Sigmund, "Evolutionary dynamics of biological games," *Science*, vol. 303, pp. 793–799, 2004.

[31] H. Ohtsuki, M. A. Nowak, and J. M. Pacheco, "Breaking the symmetry between interaction and replacement in evolutionary dynamics on graphs," *Phys. Rev. Lett.*, vol. 98, no. 10, p. 108106, 2007.

[32] H. Ohtsukia and M. A. Nowak, "The replicator equation on graphs," *J. Theor. Biol.*, vol. 243, pp. 86–97, 2006.

[33] X. Zhao and A. H. Sayed, "Clustering via diffusion adaptation over networks," in *Proc. Int. Workshop Cognitive Inf. Process.*, Spain, May 2012, pp. 1–6.

[34] M. Slatkin, "Fixation probabilities and fixation times in a subdivided population," *J. Theor. Biol.*, vol. 35, no. 3, pp. 477–488, 1981.

[35] H. Ohtsuki, C. Hauert, E. Lieberman, and M. A. Nowak, "A simple rule for the evolution of cooperation on graphs and social networks," *Nature*, vol. 441, pp. 502–505, 2006.

[36] F. Fu, L. Wang, M. A. Nowak, and C. Hauert, "Evolutionary dynamics on graphs: Efficient method for weak selection," *Phys. Rev.*, vol. 79, no. 4, p. 046707, 2009.

[37] G. Wild and A. Traulsen, "The different limits of weak selection and the evolutionary dynamics of finite populations," *J. Theor. Biol.*, vol. 247, no. 2, pp. 382–390, 2007.

[38] H. Ohtsukia and M. A. Nowak, "Evolutionary stability on graphs," *J. Theor. Biol.*, vol. 251, no. 4, pp. 698–707, 2008.

[39] W. J. Ewense, *Mathematical Population Genetics: Theoretical Introduction*. New York, NY, USA: Springer, 2004.

Chunxiao Jiang (S'09–M'13) received the B.S. degree in information engineering from Beijing University of Aeronautics and Astronautics (Beihang University) in 2008 and the Ph.D. degree from Tsinghua University (THU), Beijing, in 2013, both with the highest honors. During 2011–2012, he visited the Signals and Information Group (SIG) at the Department of Electrical and Computer Engineering (ECE), University of Maryland (UMD), supported by China Scholarship Council (CSC) for one year.

He is currently a Research Associate with the ECE Department, UMD, with Prof. K. J. Ray Liu, and also a postdoctoral researcher with the EE Department, THU. His research interests include the applications of game theory and queuing theory in wireless communication and networking and social networks.

Dr. Jiang received the Beijing Distinguished Graduated Student Award, Chinese National Fellowship and Tsinghua Outstanding Distinguished Doctoral Dissertation in 2013.

Yan Chen (S'06–M'11) received the Bachelor's degree from University of Science and Technology of China in 2004, the M.Phil. degree from Hong Kong University of Science and Technology (HKUST) in 2007, and the Ph.D. degree from University of Maryland, College Park, in 2011. From 2011 to 2013, he has been a Postdoctoral research associate with the Department of Electrical and Computer Engineering, University of Maryland, College Park.

Currently, he is a Principal Technologist at Origin Wireless Communications. He is also affiliated with Signal and Information Group of the University of Maryland, College Park. His current research interests are in social learning and networking, behavior analysis and mechanism design for network systems, multimedia signal processing, and communication.

Dr. Chen received the University of Maryland Future Faculty Fellowship in 2010, Chinese Government Award for outstanding students abroad in 2011, University of Maryland ECE Distinguished Dissertation Fellowship Honorable Mention in 2011, and was the Finalist of A. James Clark School of Engineering Deans Doctoral Research Award in 2011.

K. J. Ray Liu (F'03) was named a Distinguished Scholar-Teacher of University of Maryland, College Park, in 2007, where he is Christine Kim Eminent Professor of Information Technology. He leads the Maryland Signals and Information Group conducting research encompassing broad areas of signal processing and communications with recent focus on cooperative and cognitive communications, social learning and network science, information forensics and security, and green information and communications technology.

Dr. Liu is the recipient of numerous honors and awards including the IEEE Signal Processing Society Technical Achievement Award and Distinguished Lecturer. He also received various teaching and research recognitions from the University of Maryland including the university-level Invention of the Year Award; and Poole and Kent Senior Faculty Teaching Award, Outstanding Faculty Research Award, and Outstanding Faculty Service Award, all from A. James Clark School of Engineering. An ISI Highly Cited Author, he is a Fellow of AAAS. He is President of IEEE Signal Processing Society where he has served as Vice President—publications and Board of Governors. He was the Editor-in-Chief of the IEEE SIGNAL PROCESSING MAGAZINE and the founding Editor-in-Chief of EURASIP *Journal on Advances in Signal Processing*.