

# BIO-INSPIRED COOPERATIVE OPTIMIZATION WITH APPLICATION TO BACTERIA MOTILITY

Jianshu Chen and Ali H. Sayed

Department of Electrical Engineering, University of California, Los Angeles

## ABSTRACT

Inspired by bacterial motility, we propose an algorithm for adaptation over networks with mobile nodes. The nodes have limited abilities and they are allowed to cooperate with their neighbors to optimize a common objective function. In contrast to traditional adaptation formulations, an important consideration in this work is the fact that the nodes do not know the form of the cost function beforehand. The nodes can only sense variations in the values of the objective function as they diffuse through the space, such as sensing the variation in the concentration of nutrients in the environment. We propose a technique for the nodes to pick the search vector as a linear combination of the neighbors' last steps, by attempting to maximize the nutritional gradient. The procedure enables information to flow from "information-rich" nodes to the other nodes.

**Index Terms**— Adaptive network, cooperative optimization, bacterial motility, diffusion.

## 1. INTRODUCTION

Bacteria are single-cell microscopic organisms. They survive by foraging for nutrients in the environment in a manner that maximizes their energy intake per unit time [1]. During the foraging process, bacteria exhibit directed movement in response to chemical stimuli in a behavior known as "chemotaxis" [2].

Bacteria with flagella have two distinct modes of movement: running and tumbling. When, by chance, a bacterium moves up a spatial gradient of a chemical attractant, e.g., nutrients, it runs in that direction at a larger step. Otherwise, it moves the other way and tumbles around, i.e., its direction of motion changes randomly without much translational movement. The bias in the random walk ultimately enables the bacteria to move up gradient directions. Studies on impulsive stimuli indicate that bacteria collect gradient information by comparing the concentration observed over the past 1 second with the concentration observed over the previous 3 second and respond to the difference [3]. When a bacterium is away from the nutrition source, the concentration of nutrition will fall below a threshold that the bacteria can detect. To extend their sensing ability, the bacteria use small molecules for extra- and intracellular signaling to act in a coordinated manner [4].

Motivated by these observations on bacterial motility, and by our earlier study in [4], we propose an algorithm for adaptation over networks of nodes with limited sensing abilities. The nodes cooperate locally with their neighbors to optimize a common objective function. In contrast to conventional adaptation formulations, an important consideration in this work is the fact the nodes do *not* know the form of the cost function beforehand. Instead, the nodes can

only sense variations in the values of the cost function. The analogy with bacteria motility is as follows. The bacteria play the role of the nodes and the concentration of nutrients in the environment plays the role of the objective function. The nodes wish to determine the location of the peak of the concentration profile. The nodes, however, do not know the shape of the concentration profile. Still, they can sense variations in the concentration as they move around. We shall propose a technique for each node to pick its search vector as a linear combination of the neighbors' last steps in a manner that will help maximize the nutritional gradient. The resulting procedure will enable information to flow from the "information-rich" nodes to the other nodes. Simulation results illustrate how the procedure helps extend the sensing ability of each node in the network.

## 2. PROBLEM FORMULATION

Consider a collection of  $N$  nodes, each of which wants to maximize an objective function  $J(w)$  over an  $M$ -dimensional column vector  $w$  and determine the optimal  $w^o$ . One obvious way to accomplish this task is by means of a steepest-descent algorithm, whereby each node  $k$  independently adapts its local estimate,  $w_{k,i-1}$ , at time  $i-1$  along the direction of the the gradient vector,  $\nabla J(w)$ , evaluated at  $w_{k,i-1}$ . In an adaptive implementation, the gradient vector is substituted by an instantaneous approximation computed from the data [5]. Alternatively, the nodes may cooperate with each other through a diffusive process [6–11] to accomplish the same task through a fully decentralized solution to the problem. The performance of these adaptive diffusion algorithms has been studied in detail in [6–11].

The above two implementations, whether distributed [6–11] or non-cooperative [5], rely on the basic assumption that the form of the cost function,  $J(w)$ , is known beforehand by all nodes. This assumption underlies the derivation of most adaptation algorithms. When the form of  $J(w)$  is known, the form of its gradient vector is also known and instantaneous approximations for it can therefore be computed. If we re-examine the example of bacteria foraging for food, the density of the nutrients in the space is not known to the bacteria. The bacteria, however, can sense the concentration of nutrients at their locations. A useful problem would be to show how these localized measurements can be combined together through an adaptive diffusive process to enable the nodes to converge to the peak of the concentration function.

Assume that at time  $i-1$ , each node  $k$  has access to measurements of the concentration (or cost) function at times  $i-1$  and  $i-2$ . These measurements determine the local error signal:

$$e_k(i) = J(w_{k,i-1}) - J(w_{k,i-2}) \quad (1)$$

which can be related to the gradient vector (through a first-order Taylor series expansion):

$$e_k(i) \approx [\nabla J(w_{k,i-2})]^T (w_{k,i-1} - w_{k,i-2}) \quad (2)$$

Email: {jshchen, sayed}@ee.ucla.edu. This work was supported in part by NSF grants CCF-1011918, CCF-0942936, and ECS-0725441.

Introduce the row vector

$$u_{k,i}^T \triangleq w_{k,i-1} - w_{k,i-2} \quad (3)$$

and note that it denotes the direction of motion from  $w_{k,i-1}$  to  $w_{k,i-2}$  (assuming we interpret the successive  $w_{k,i}$  as location vectors). In order to move from  $w_{k,i-1}$  to  $w_{k,i}$ , the node needs to estimate the gradient vector at  $w_{k,i-1}$ . Assuming updates with small step-sizes, we can approximate  $\nabla J(w_{k,i-1}) \approx \nabla J(w_{k,i-2})$ . In this way, and using Eq. (2), node  $k$  estimates that the desired gradient direction  $\nabla J(w_{k,i-1})$  essentially lies in the plane:

$$\mathcal{P}_{k,i} = \{\nabla J(w_{k,i-1}) : u_{k,i} \nabla J(w_{k,i-1}) \approx e_k(i)\}. \quad (4)$$

The objective for each node becomes that of determining a good estimate for this gradient vector, say,  $p$ , in order to update  $w_{k,i-1}$  to  $w_{k,i}$  as

$$w_{k,i} = w_{k,i-1} + \mu p \quad (5)$$

where  $\mu$  is the step size parameter. The choice of  $p$  will be determined through cooperation with the neighboring nodes as we proceed to explain.

### 3. COOPERATIVE OPTIMIZATION

#### 3.1. Combination Coefficients

Let us assume that at time  $i - 1$  each node  $k$  can share with its neighbors information about the set  $\mathcal{P}_{k,i}$ , or equivalently, the vector  $u_{k,i}$  and the signal  $e_k(i)$ . We define the neighborhood of node  $k$  at time  $i - 1$  as follows:

$$\mathcal{N}_{k,i-1} = \{\ell : \|w_{\ell,i-1} - w_{k,i-1}\| \leq R_0\} \quad (6)$$

where  $R_0$  is some radius value. Notice that  $\mathcal{N}_{k,i-1}$  also includes node  $k$ . Then the information available to node  $k$  is  $\{u_{\ell,i}, e_{\ell}(i)\}_{\ell \in \mathcal{N}_{k,i-1}}$ . Node  $k$  determines the search vector  $p$  as a linear combination of the motion directions shared by its neighbors, namely,

$$p = \sum_{\ell=1}^{|\mathcal{N}_{k,i-1}|} a_{k\ell} u_{\ell,i}^T \quad (7)$$

where  $|\cdot|$  denotes the cardinality of a set at time  $i - 1$ . The combination coefficients are selected as follows. Through a first-order Taylor series expansion, we have

$$J(w_{k,i}) - J(w_{k,i-1}) \approx [\nabla J(w_{k,i-1})]^T (w_{k,i} - w_{k,i-1}). \quad (8)$$

Substituting (5) into (8) and using (7), we get

$$J(w_{k,i}) - J(w_{k,i-1}) \approx \mu \cdot \sum_{\ell=1}^{|\mathcal{N}_{k,i-1}|} a_{k\ell} u_{\ell,i} \nabla J(w_{k,i-1}) \quad (9)$$

When the neighbors are close to each other, the gradient vectors at their locations are approximately similar so that

$$\nabla J(w_{\ell,i-1}) \approx \nabla J(w_{k,i-1}), \quad \ell = 1, 2, \dots, |\mathcal{N}_{k,i-1}|. \quad (10)$$

Then, (9) becomes

$$J(w_{k,i}) - J(w_{k,i-1}) \approx \mu \cdot \sum_{\ell=1}^{|\mathcal{N}_{k,i-1}|} a_{k\ell} e_{\ell}(i) \quad (11)$$

This relation suggests a criterion for selecting the combination coefficients  $\{a_{k\ell}\}$  in order to maximize the increment in  $J(w)$  as  $w$  goes from  $w_{k,i-1}$  to  $w_{k,i}$ . Though our ultimate goal is to find  $w$  that maximizes  $J(w)$ , we cannot do this directly because we do not know its explicit expression. However, we can expect that, by sequentially updating the estimate of  $w$  so that, at each iteration,  $J(w)$  is increased by a maximum amount, the estimate is able to approach the peak point of  $J(w)$ . Specifically, introduce the column vectors

$$a = [a_{k1} \quad a_{k2} \quad \cdots \quad a_{k|\mathcal{N}_{k,i-1}|}]^T \quad (12)$$

$$e = [e_1(i) \quad e_2(i) \quad \cdots \quad e_{|\mathcal{N}_{k,i-1}|}(i)]^T \quad (13)$$

Then, node  $k$  picks its combination coefficients by solving:

$$\begin{aligned} \max \quad & e^T a \\ \text{s.t.} \quad & a^T a \leq 1 \text{ and } a_{k\ell} \geq 0 \end{aligned}$$

We limit the norm of the vector  $a$  to one to avoid the possibility of an unbounded solution. The above optimization is equivalent to

$$\begin{aligned} \min \quad & -e^T a \\ \text{s.t.} \quad & a^T a \leq 1 \text{ and } a_{k\ell} \geq 0 \end{aligned} \quad (14)$$

Using Lagrange multipliers, we introduce the function:

$$L = -e^T a + \gamma(a^T a - 1) - \lambda^T a \quad (15)$$

where

$$\lambda = [\lambda_{k1} \quad \lambda_{k2} \quad \cdots \quad \lambda_{k|\mathcal{N}_{k,i-1}|}]. \quad (16)$$

The optimal solution  $a$  is the unique solution to the following Karush-Kuhn-Tucker (KKT) conditions [12]:

$$\nabla_a L = -e + 2\gamma a - \lambda = 0 \quad (17)$$

$$a^T a \leq 1, \quad a \geq 0 \quad (18)$$

$$\lambda \geq 0, \quad \gamma \geq 0 \quad (19)$$

$$\lambda_{k\ell} a_{k\ell} = 0, \quad \ell = 1, 2, \dots, |\mathcal{N}_{k,i-1}| \quad (20)$$

$$\gamma \cdot [a^T a - 1] = 0 \quad (21)$$

From (17) we have

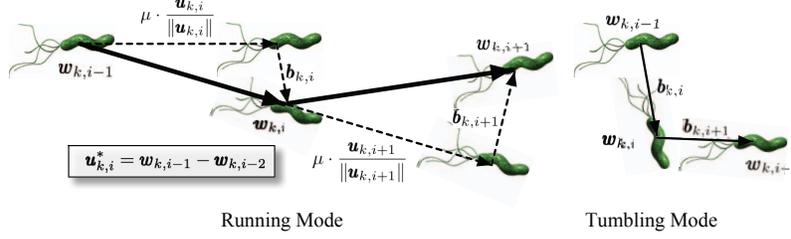
$$2\gamma a_{k\ell} = e_{\ell} + \lambda_{k\ell}. \quad (22)$$

Multiplying (22) by  $\lambda_{k\ell}$  and using (20), we get

$$0 = \lambda_{k\ell}(e_{\ell} + \lambda_{k\ell}) \quad (23)$$

This means that either  $\lambda_{k\ell} = 0$  or  $\lambda_{k\ell} = -e_{\ell}$  for each  $\ell = 1, \dots, |\mathcal{N}_{k,i-1}|$ . Let us examine the possibilities:

1.  $e_{\ell} > 0$ :  
Then  $\lambda_{k\ell} = 0$ . Otherwise,  $\lambda_{k\ell} = -e_{\ell} < 0$ , contradicting (19). Substituting  $\lambda_{k\ell} = 0$  into (22), we get  $2\gamma a_{k\ell} = e_{\ell} > 0$ . Therefore,  $\gamma > 0$  and  $a_{k\ell} = e_{\ell}/2\gamma$ .
2.  $e_{\ell} < 0$ :  
Then  $\lambda_{k\ell} = -e_{\ell}$ . Otherwise,  $\lambda_{k\ell} = 0$  and (22) becomes  $2\gamma a_{k\ell} = e_{\ell} < 0$ , contradicting  $\gamma \geq 0$  in (19) and  $a_{k\ell} \geq 0$  in (18). Using  $\lambda_{k\ell} = -e_{\ell} > 0$  in (20) gives  $a_{k\ell} = 0$ .
3.  $e_{\ell} = 0$ :  
Then  $\lambda_{k\ell} = 0$ . Eq. (22) implies either  $\gamma = 0$  or  $a_{k\ell} = 0$ .



**Fig. 1.** Bacterial motion model with two modes of locomotion: running and tumbling.

We therefore see that if  $e$  has positive components, then  $\gamma > 0$  and

$$a_{k\ell} = \frac{1}{2\gamma} I(e_\ell) e_\ell \Rightarrow \|a\|^2 = \frac{1}{4\gamma^2} \left[ \sum_{\ell=1}^{|\mathcal{N}_{k,i-1}|} I(e_\ell) e_\ell^2 \right] \quad (24)$$

where  $I(x)$  is the indicator function: it is equal to one when  $x > 0$  and zero otherwise. Using  $\gamma > 0$  in (21), and setting  $\|a\|^2 = 1$ , we have

$$\gamma = \frac{1}{2} \sqrt{\sum_{\ell=1}^{|\mathcal{N}_{k,i-1}|} I(e_\ell) e_\ell^2}. \quad (25)$$

When  $e = 0$ , then from (14), we know that  $-e^T a = 0$  and  $a_{k\ell}$  can be any value that satisfies the constraints. We set  $a_{k\ell} = 0$  in this case. In summary, the general expression for  $a_{k\ell}$  is

$$a_{k\ell}(i) = \begin{cases} \frac{I(e_\ell(i)) e_\ell(i)}{\sqrt{\sum_{\ell=1}^{|\mathcal{N}_{k,i-1}|} I(e_\ell(i)) e_\ell^2(i)}} & e_\ell(i) \neq 0 \\ 0 & e_\ell(i) = 0 \end{cases} \quad (26)$$

Eq. (26) implies that when neighbor  $\ell$  moved in the wrong direction resulting in  $e_\ell(i) \leq 0$ , its direction  $u_{\ell,i}$  is ignored. On the other hand, nodes moving in the right direction will have their motion directions averaged in proportion to their error signals,  $e_\ell(i)$ . When implementing the algorithm, we add a random perturbation to (5) to model the random perturbation component that occurs in bacterial motility:

$$w_{k,i} = w_{k,i-1} + \mu p + b_{k,i} \quad (27)$$

### 3.2. Reliability of Error Signal

By examining (11), each individual term  $a_{k\ell} e_\ell(i)$  inside the sum is the increment gained by assigning  $a_{k\ell}$  to  $u_{\ell,i}$ . In practice, however, we do not know the  $\{e_\ell(i)\}$ ; instead we have access to noisy measurements of these signals:

$$y_\ell(i) = e_\ell(i) + v_\ell(i) \quad (28)$$

where  $v_\ell(i)$  is the measurement noise process modeled as independent and identically distributed Gaussian random process with mean zero and power  $\sigma_v^2$ . We may optimize (14) by replacing  $e$  with

$$y = [y_1(i) \quad y_2(i) \quad \cdots \quad y_{|\mathcal{N}_{k,i-1}|}(i)]^T \quad (29)$$

However, we need to account for the effect of noise on performance. Let

$$\text{SNR}_\ell \triangleq \frac{E\{|e_\ell(i)|^2\}}{E\{|v_\ell(i)|^2\}} = \frac{r_\ell^2}{\sigma_v^2} \quad (30)$$

When the SNR at node  $\ell$  is sufficiently high, then  $y_\ell(i)$  is a good estimate for  $e_\ell(i)$  and the corresponding increment  $a_{k\ell} y_\ell(i)$  due to  $u_{\ell,i}$  will be a reliable estimate for  $a_{k\ell} e_\ell(i)$ . Based on this argument, we should place larger weights on measurements  $y_\ell(i)$  with higher SNRs. We therefore consider the following criterion. Node  $k$  would select the combination coefficients  $\{a_{k\ell}\}$  such that the SNR that results from using the combination is larger than the SNR that results from using only its update direction,  $u_{k,i}$ .

$$\text{SNR} \triangleq \frac{E \left| \sum_{\ell=1}^{|\mathcal{N}_{k,i-1}|} a_{k\ell} e_\ell(i) \right|^2}{E |a_{k\ell} v_\ell(i)|^2} = \frac{\sum_{\ell=1}^{|\mathcal{N}_{k,i-1}|} a_{k\ell}^2 r_\ell^2}{\sum_{\ell=1}^{|\mathcal{N}_{k,i-1}|} a_{k\ell}^2 \sigma_v^2} \geq \frac{r_k^2}{\sigma_v^2} \quad (31)$$

$$\Leftrightarrow \sum_{\ell=1}^{|\mathcal{N}_{k,i-1}|} a_{k\ell}^2 (r_\ell^2 - r_k^2) \geq 0 \quad (32)$$

If the variances  $\{r_\ell^2\}$  are available, then we could add (32) as a new constraint to problem (14). In practice, however, the  $\{r_\ell^2\}$  are usually unknown. Instead, it is easier to obtain information about their ordering (i.e., which neighbors have more or less noisy data or which neighbors are more or less confident about their measurements). Then node  $k$  would ignore measurements from neighbors whose data are noisier than node  $k$ . In this way, we can solve the same optimization problem (14) with the vector  $e$  replaced by a new vector containing only the measurements  $y_\ell(i)$  from the nodes that are more reliable than node  $k$ .

The remaining problem is to construct a reliability measure  $\{r_\ell^2\}$  that can capture well the ordering of the nodes' reliability. One intuition is that the closer the node is to the optimal location  $w^o$ , the more reliable its information is. However, the nodes are generally unable to access their distance to  $w^o$ . Thus we propose to use:

$$r_k^2 = \begin{cases} 1, & \tilde{J}(w_{k,i}) > J_T \\ \max_{\ell \in \mathcal{N}_{\ell,i-1}, \ell \neq k} \frac{1}{2} r_\ell^2, & \text{otherwise} \end{cases} \quad (33)$$

where  $\tilde{J}(w_{k,i})$  is the noisy measurement of  $J(w_{k,i})$ , and  $J_T$  is a threshold. This method actually corresponds to using dynamic programming to find a shortest path towards the node with reliability 1. And  $r_k^2$  measures the distance between the nearest node with reliability 1 and node  $k$ . Actually,  $-\log_2 r_k^2$  is the minimum number of hops between them.

### 3.3. Application to Bacterial Motility

Since bacteria have limited sensing ability, they cannot evaluate  $e_k(i)$  as in (1). Instead, we assume that they are only aware of its sign; that is, we assume that the nodes are only aware whether

the concentration (cost) function is increasing or decreasing while moving from  $w_{k,i-2}$  to  $w_{k,i-1}$ . Then it is reasonable to simplify  $a_{k\ell}$  in (26) into the following form:

$$a_{k\ell}(i) = \begin{cases} \frac{I(e_\ell(i))}{\sqrt{\sum_{\ell=1}^{|\mathcal{N}_{k,i-1}|} I(e_\ell(i))}} & e_\ell(i) \neq 0 \\ 0 & e_\ell(i) = 0 \end{cases} \quad (34)$$

Then we update the position of bacterium  $k$  according to (27).

From (34), we see that if a neighbor of node  $k$  (including  $k$ ) moved in the wrong direction during the last step ( $e_\ell(i) \leq 0$ ), then node  $k$  will ignore its direction  $u_{k,i}$  in (7). For the neighbors with  $e_\ell(i) > 0$ , node  $k$  will average their directions using equal weights. When none of the neighbors (including  $k$ ) are in the right direction, then node  $k$  will tumble, i.e., will set  $p = 0$ . For the special case of non-cooperation, (34) is reduced to  $a_{k\ell}(i) = I[e_\ell(i)]$  and  $p$  becomes

$$p = I[e_\ell(i)] \frac{u_{k,i}}{\|u_{k,i}\|}. \quad (35)$$

It follows that when the nutritional concentration increases during the last step, the bacterium will run in that direction. Otherwise, it will stop running, and influenced by the random perturbation  $b_{k,i}$  shown in (27), it will tumble. Then, the bacterium clears its memory of the previous running direction, and searches for a new running direction, along which  $J(w)$  increases. The analogy between the above formulation and bacterial motility is illustrated in Fig. 1; more details about bacteria motility can be found in [4]. The cost function  $J(w)$  corresponds to the density of nutrition. If a bacterium senses that the value of  $J(w)$  is increasing as it moves from  $w_{k,i-1}$  to  $w_{k,i}$ , it will continue to move in a running mode along the direction of  $u_{k,i}$ . Otherwise, the bacterium will stop running, and switch to the tumbling mode.

#### 4. SIMULATION RESULTS

In this section, we use the algorithm to simulate bacteria motility. The source is placed at the coordinates (40, 40); it generates a nutrition field in the shape of a two-dimensional Gaussian distribution, as shown in Fig. 2. The peak value of the nutritional density is 15. At the beginning, a total of 500 bacteria are randomly and uniformly distributed over a  $40 \times 40$  rectangular region centered at (20, 20). Their random perturbation  $b_{k,i}$  is modeled as an i.i.d. two-dimensional Gaussian random variable with zero mean and 0.1 standard deviation. The step size  $\mu$  is 0.5. The radius of the neighborhood  $\mathcal{N}_{k,i}$  for node  $k$  is  $R_0 = 5$ . And the variance  $\sigma_v^2$  for the measurement noise  $v_\ell^n(i)$  is 0.1.

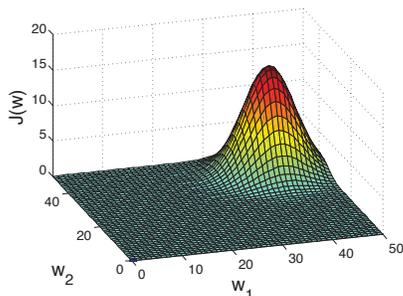


Fig. 2. Distribution of nutrients in the environment.

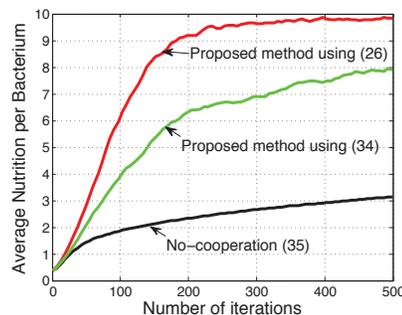


Fig. 3. Average nutrition per bacterium vs. number of iterations.

Fig. 3 shows the average nutrition density per bacterium defined as  $\frac{1}{N} \sum_{k=1}^N J(w_{k,i})$  against  $i$ . The methods we compare are optimal search with  $a_{k\ell}$  given by (26), bacterial motility whose  $a_{k\ell}$  is given by (34) and the noncooperative case in (35). The result is obtained by averaging 100 independent realizations. We see that methods (26) and (34) converge fast. Both methods can reach their steady state within 200 steps, and the performance loss of the bacterial motility method is not large. Furthermore, without cooperation, only the bacteria close to the food source are able to reach nutrition.

#### 5. CONCLUSION

We proposed a dynamic adaptive network based on bacterial motility. Our results show that cooperation can extend the sensing ability of each node as long as the direction of information flow is oriented from the “information-rich” nodes towards the other nodes. As a result, the nodes without local information can still learn from the neighbors, and the network can swarm towards the the nutritional sources.

#### 6. REFERENCES

- [1] K. Passino, “Biomimicry of bacterial foraging for distributed optimization and control,” *IEEE Control Systems Magazine*, vol. 22, no. 6, pp. 52–67, 2002.
- [2] E. Martin, *Macmillan Dictionary of Life Sciences*. Macmillan, 1985, pp. 362.
- [3] H. Berg, “Motile behavior of bacteria,” *Physics Today*, vol. 53, no. 1, pp. 24–29, 2000.
- [4] J. Chen, X. Zhao and A. H. Sayed, “Bacterial motility via diffusion adaptation,” in *Proc. 44th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2010.
- [5] A.H. Sayed, *Adaptive Filters*, Wiley, NJ, 2008.
- [6] C.G. Lopes and A.H. Sayed, “Diffusion least-mean squares over adaptive networks: Formulation and performance analysis,” *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, 2008.
- [7] F. S. Cattivelli and A. H. Sayed, “Diffusion LMS strategies for distributed estimation,” vol. 58, no. 3, pp. 1035–1048, March 2010.
- [8] F. Cattivelli and A. H. Sayed, “Self-organization in bird flight formations using diffusion adaptation,” in *Proc. 3rd International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, Aruba, Dutch Antilles, Dec. 2009, pp. 49–52.
- [9] S-Y. Tu and A. H. Sayed, “Tracking behavior of mobile adaptive networks,” in *Proc. 44th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2010.
- [10] S-Y. Tu and A. H. Sayed, “Mobile adaptive networks with self-organization abilities,” in *Proc. 7th International Symposium on Wireless Communication Systems*, York, United Kingdom, Sep. 2010, pp. 379–383.
- [11] S-Y. Tu and A. H. Sayed, “Foraging behavior of fish schools via diffusion adaptation,” in *Proc. International Workshop on Cognitive Information Processing*, Elba Island, Italy, June 2010, pp. 63–68.
- [12] S.P. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge Univ Pr, 2004, pp. 243–244.