Diffusion Adaptation over Networks of Particles Subject to Brownian Fluctuations

Ali H. Sayed
University of California, Los Angeles
sayed@ee.ucla.edu

Faten A. Sayed
Yale University
faten.sayed@yale.edu

***Abstract*—This work investigates the influence of diffusion adaptation on the behavior of networks of micro-organisms that are subject to Brownian fluctuations in the motion of their constituent agents. The organisms are assumed to share information, usually through chemical signaling. The information may signal the direction of a target (such as a foreign body) towards which the cells need to migrate. The sharing of information enables the nodes to bias the probabilities of their random walks in favor of the desired direction of motion. It is verified that the adaptive diffusion of direction information enhances the foraging and tracking ability of the cells.**

***Index Terms*—Adaptive networks, diffusion adaptation, Brownian motion, diffusing particles.**

I. Introduction

Small particles and micro-organisms in aqueous environments are subject to random Brownian fluctuations [1]. In the absence of external stimuli or forces, the drift velocity is zero and the particles wander around randomly within the environment. This work describes a mechanism by which the particles interact with their neighbors and share information through signaling. The information is used by the particles to bias the transition probabilities of the random walks. The net result is a network of diffusing particles that move towards a target location more efficiently. A representative example could be a situation corresponding to cells tracking a foreign intruder within a body. Other examples of cooperation and self-organization behavior are abundant in nature and occur in several other biological systems [2]- [4].

A. Random Walks

We start by reviewing some basic facts. Consider a microscopic particle moving along the horizontal $x-$direction. The particle takes one step every Δt seconds and moves a distance Δx either to the right with probability p_x, or to the left with probability $1-p_x$; see Fig. 1. The average one-step displacement by the particle along the horizontal direction is denoted by $\mathsf{E}\,\boldsymbol{r}$ and is given by:

$$\mathsf{E}\,\boldsymbol{r} = (2p_x - 1)\cdot \Delta x \tag{1}$$

The corresponding mean-square displacement is

$$\begin{aligned}\sigma_r^2 &\triangleq \mathsf{E}\,\boldsymbol{r}^2 - (\mathsf{E}\,\boldsymbol{r})^2 \\ &= 4\Delta x^2 \cdot p_x \cdot (1-p_x)\end{aligned} \tag{2}$$

This work was supported in part by NSF grants CCF-0942936 and CCF-1011918.

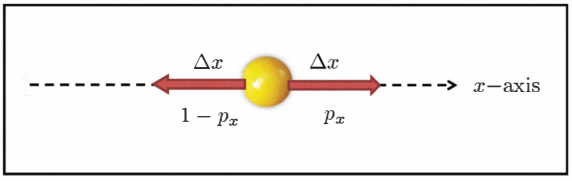

Fig. 1. A particle moves a distance Δx to the right with probability p_x or to the left with probability $1-p_x$.

At time $t = L\Delta t$, after a sequence of L independent random steps $\{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_L\}$ along the x-direction, the particle location is the aggregate sum of the displacements up to that point in time and is denoted by:

$$\boldsymbol{x} = \boldsymbol{r}_1 + \boldsymbol{r}_2 + \boldsymbol{r}_3 + \ldots + \boldsymbol{r}_L \tag{3}$$

The average displacement and mean-square displacement after L steps are then given by:

$$\mathsf{E}\,\boldsymbol{x} = L\cdot \mathsf{E}\,\boldsymbol{r} \triangleq v_x \cdot t \tag{4}$$

$$\sigma_x^2 = L\cdot \sigma_r^2 \triangleq 2D_x \cdot t \tag{5}$$

where the quantities v_x and D_x introduced in (4) and (5) denote the drift velocity and the diffusion coefficient of the particle [1]:

$$v_x \triangleq \frac{\Delta x}{\Delta t}\cdot(2p_x - 1) \tag{6}$$

$$D_x \triangleq 2\cdot\frac{\Delta x^2}{\Delta t}\cdot p_x \cdot (1-p_x) \tag{7}$$

Observe that the drift velocity v_x is nonzero whenever $p_x \neq 1/2$. In this case, the particle gives preference to one direction over the other. Nonzero drift velocities arise due to the presence of external stimuli such as gravity forces or centrifugal forces. In the next section, we describe a mechanism by which the transition probability p_x is biased away from the central value of $1/2$ in order to help move a network of diffusing particles towards a desired target location more efficiently.

In this work, we consider particles that diffuse in the plane, along both the x and y directions, with steps of size Δx and Δy, respectively. We assume that displacements along the x and y directions are independent of each other and occur with probabilities p_x (to the right) and p_y (upwards). We denote

the mean-square displacement of the particle in the plane by σ_p^2 and it is given by:

$$\sigma_p^2 = \sigma_x^2 + \sigma_y^2 = 2(D_x + D_y) \cdot t \tag{8}$$

where D_y is defined similarly to D_x in (7). Equation (8) reveals the well-known result that the mean-square displacement of a diffusing particle increases linearly with time [1]. In the context of locating a target and moving towards it, we would like the diffusive behavior of the particles in the plane to be such that the mean-square displacement of each particle, relative to the center of gravity of the network, is a decreasing function of time.

II. Network of Diffusing Particles

Consider a collection of N particles diffusing within the rectangular domain $[0, X] \times [0, Y]$, where $(0, 0)$ denotes the origin of space. We assume initially that $p_x = 0.5$ and $p_y = 0.5$ so that each particle is equally likely to move right and left, or up and down. A target, whose location is unknown to the particles, is assumed to be present at the coordinates (x_o, y_o). We represent the target location by the column vector:

$$w^o \triangleq \begin{bmatrix} x_o \\ y_o \end{bmatrix}, \qquad (2 \times 1) \tag{9}$$

Let further

$$\ell_{k,i} \triangleq \begin{bmatrix} x_k(i) \\ y_k(i) \end{bmatrix}, \quad (2 \times 1) \tag{10}$$

denote the location of particle k at time instant $t = i\Delta t$ (i.e., we use a discrete representation of the time scale). It is assumed that at every time instant i, each particle k has access to two pieces of information related to w^o– see Fig. 2:

(a) A rough estimate of the distance to the target, corrupted by additive noise, say,

$$\boxed{\boldsymbol{d}_k(i) = d_k^o(i) \; + \; \boldsymbol{v}_k(i)} \tag{11}$$

where $d_k^o(i)$ denotes the actual distance between particle k and the target and is given by

$$\begin{aligned} d_k^o(i) &= \|w^o - \ell_{k,i}\| \\ &= \sqrt{(x_o - x_k(i))^2 \; + \; (y_o - y_k(i))^2} \end{aligned} \tag{12}$$

where the notation $\|\cdot\|$ denotes the Euclidean norm of its vector argument. Moreover, the noise term $\boldsymbol{v}_k(i)$ is assumed to be zero mean and independent of all other random signals. It is further assumed that the variance of $\boldsymbol{v}_k(i)$ depends on the distance to the target: the closer the particle k is to the target, the less noisy the measurement $\boldsymbol{d}_k(i)$ is expected to be. Thus, we assume that the variance of $\boldsymbol{v}_k(i)$ has the form

$$\sigma_{v,k}^2 \; = \; \beta_v^2 \cdot |d_k^o(i)|^2 \tag{13}$$

for some constant β_v.

Notation. Observe that we use boldface letters to refer to random quantities, such as $\boldsymbol{v}_k(i)$ and $\boldsymbol{d}_k(i)$; we likewise use normal font letters to refer to realizations of these random quantities, such as writing $d_k(i)$ to refer to an observation of $\boldsymbol{d}_k(i)$. We also use subscripts to refer to time dependencies of vector quantities, such as $\ell_{k,i}$, and parentheses to refer to time dependencies of scalar quantities, such as $\boldsymbol{v}_k(i)$.

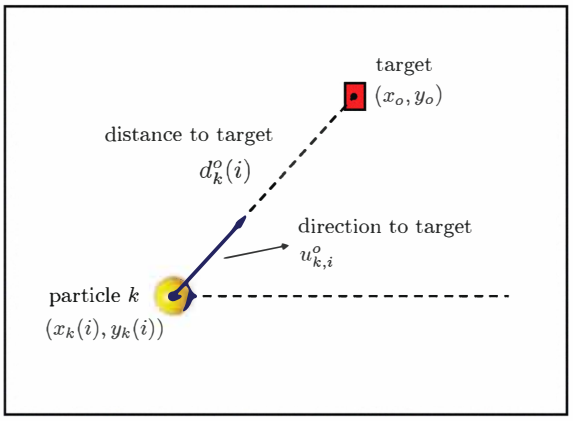

Fig. 2. The actual distance from particle k to the target is denoted by $d_k^o(i)$, and the actual unit-norm direction vector from the same particle to the target is denoted by $u_{k,i}^o$. Particle k is assumed to have access to noisy versions of $\{d_k^o(i), u_{k,i}^o\}$.

(b) A rough estimate of the direction vector towards the target, say,

$$\boxed{\boldsymbol{u}_{k,i} \; = \; u_{k,i}^o \; + \; \boldsymbol{n}_{k,i}} \tag{14}$$

where $u_{k,i}^o$ is a 1×2 (row) unit-vector pointing from the particle's current location towards w^o:

$$\begin{aligned} u_{k,i}^o &= \frac{(w^o - \ell_{k,i})^T}{\|w^o - \ell_{k,i}\|} \\ &= \frac{[x_o - x_k(i) \quad y_o - y_k(i)]}{\sqrt{(x_o - x_k(i))^2 \; + \; (y_o - y_k(i))^2}} \end{aligned} \tag{15}$$

The noise term $\boldsymbol{n}_{k,i}$ is assumed to have zero mean and to be independent of all other random signals. We further assume that the covariance matrix of $\boldsymbol{n}_{k,i}$ is of the form $\sigma_{n,k}^2 I$, where $\sigma_{n,k}^2$ depends on the distance to the target as was the case with $\sigma_{v,k}^2$, say as,

$$\sigma_{n,k}^2 \; = \; \beta_u^2 \cdot |d_k^o(i)|^2 \tag{16}$$

for some constant β_u.

Observe from (12) and (15) that $d_k^o(i)$ is also given by the inner product:

$$d_k^o(i) = u_{k,i}^o (w^o - \ell_{k,i}) \tag{17}$$

so that the available noisy measurements $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ are related to the unknown w^o via a relation of the form:

$$\boxed{\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}(w^o - \ell_{k,i}) \; + \boldsymbol{v}_k'(i)} \tag{18}$$

where the modified noise term $\boldsymbol{v}'_k(i)$ is related to $\boldsymbol{v}_k(i)$ and $\boldsymbol{n}_{k,i}$ via

$$\boldsymbol{v}'_k(i) = \boldsymbol{v}_k(i) - \boldsymbol{n}_{k,i}(w^o - \ell_{k,i}) \tag{19}$$

We now describe three modes of operation for the network of particles to diffuse towards the target location. In the first mode, the particles do not interact with each other. Each particle estimates the location of the target on its own and uses this information to bias the value of its probability parameters $\{p_x, p_y\}$ towards the target, as explained below. In the two other modes of operation, the particles cooperate with their immediate neighbors to improve the foraging and tracking efficiency through a process known as diffusion adaptation [5], [6]. Adaptive diffusion strategies were used before in [8], [11]–[13] to model and study forms of organized behavior arising in bird flight formations, fish schooling, and bacteria motility.

A. No Cooperation Among Particles

In the first mode of operation, each particle acts individually and independent of the other particles in the medium. Each particle uses its noisy data $\{d_k(i), u_{k,i}\}$ and performs a standard adaptation step to estimate the location of the target by means of an LMS-type update as follows [7], [8]:

$$e_k(i) = d_k(i) - u_{k,i}(w_{k,i-1} - \ell_{k,i}) \tag{20}$$

$$w_{k,i} = w_{k,i-1} + \mu u^*_{k,i} e_k(i) \tag{21}$$

starting from the initial condition $w_{k,-1} = 0$ and where μ is a small positive step-size. The vector $w_{k,i}$ denotes the estimate of w^o by particle k at time i. The particle uses its current estimate $w_{k,i-1}$ to assess the quality of the measurement $d_k(i)$ by computing the error signal $e_k(i)$ in (20). The error is subsequently used in (21) to update $w_{k,i-1}$ to $w_{k,i}$ and the process continues. Note that the updating of the successive weight estimates $w_{k,i}$ over time is based solely on data collected by particle k. The star symbol appearing on $u^*_{k,i}$ signifies complex transposition (the vector is transposed and its entries are complex conjugated).

By comparing the updated target location, $w_{k,i}$, to its current location, $\ell_{k,i}$, the particle can estimate the direction towards which it should bias its probabilities of displacement $\{p_x, p_y\}$ in order to get closer to the target. One way to achieve this step is as follows. We denote the individual entries of $w_{k,i}$ by

$$w_{k,i} \triangleq \begin{bmatrix} x_{k,o}(i) \\ y_{k,o}(i) \end{bmatrix} \tag{22}$$

The entries of $w_{k,i}$ denote the estimates by particle k at time i of the target coordinates x_o and y_o. Recall from (10) that the entries of $\ell_{k,i}$ refer to the x and y coordinates of particle k at the same time i. Comparing $x_{k,o}(i)$ to $x_k(i)$, particle k can assess whether the target lies to its right or to its left; likewise, comparing $y_{k,o}(i)$ to $y_k(i)$, particle k can assess whether the target lies above it or below it. The particle then adjusts its transition probabilities $\{p_{k,x}, p_{k,y}\}$ as follows:

$$\begin{cases} x_{k,o}(i) - x_k(i) > 0: \text{ set } p_{k,x} > 0.5 \text{ (e.g., } 0.7) \\ x_{k,o}(i) - x_k(i) < 0: \text{ set } p_{k,x} < 0.5 \text{ (e.g., } 0.3) \\ x_{k,o}(i) - x_k(i) = 0: \text{ set } p_{k,x} = 0.5 \\ \\ y_{k,o}(i) - y_k(i) > 0: \text{ set } p_{k,y} > 0.5 \text{ (e.g., } 0.7) \\ y_{k,o}(i) - y_k(i) < 0: \text{ set } p_{k,y} < 0.5 \text{ (e.g., } 0.3) \\ y_{k,o}(i) - y_k(i) = 0: \text{ set } p_{k,y} = 0.5 \end{cases} \tag{23}$$

The probabilities $\{p_{k,x}, p_{k,y}\}$ change with time i, but we are dropping the time index to simplify the notation rather than write $\{p_{k,x}(i), p_{k,y}(i)\}$. Subsequently, the particle adjusts its location coordinates to:

$$\begin{aligned} \text{if} \quad & |x_{k,o}(i) - x_k(i)| > \Delta x \\ & \text{set} \quad x_k(i+1) = x_k(i) + \alpha_{k,x}\Delta x \\ & \text{otherwise} \quad x_k(i+1) = x_{k,o}(i) \end{aligned} \tag{24}$$

$$\begin{aligned} \text{if} \quad & |y_{k,o}(i) - y_k(i)| > \Delta y \\ & \text{set} \quad y_k(i+1) = y_k(i) + \alpha_{k,y}\Delta y \\ & \text{otherwise} \quad y_k(i+1) = y_{k,o}(i) \end{aligned} \tag{25}$$

where $\alpha_{k,x}$ is a realization of a Bernoulli variable that assumes the value -1 with probability $1 - p_{k,x}$ and the value $+1$ with probability $p_{k,x}$ (similarly, for the Bernoulli variable $\alpha_{k,y}$ whose probability distribution is determined by $p_{k,y}$). The variables $\{\alpha_{k,x}, \alpha_{k,y}\}$ are also dependent on time but we are not indicating the time dependence for ease of notation. The conditions in (24)–(25) ensure that the particle location is updated whenever the particle is sufficiently away from the target; otherwise, the particle simply moves to its estimate of the target location, $w_{k,i}$.

B. Combine-Then-Adapt (CTA) Strategy

The second mode of operation we consider is one where the particles are allowed to interact with their immediate neighbors through a so-called combine-then-adapt (CTA) diffusion strategy [5], [6]. The set of neighbors of particle k (including k itself) is denoted by $\mathcal{N}_k$ and is defined as the set of particles that are within a distance r from the particle. Observe that since we are dealing with particles that are in motion, the neighborhood of every particle is likely to change with time and, therefore, writing $\mathcal{N}_{k,i}$ with a time subscript is a more accurate notation. Let $|\mathcal{N}_{k,i}|$ denote the size of the neighborhood of particle k. In order to avoid unnecessarily large neighborhoods, we limit the number of neighbors in each set to some upper limit N_{cta}.

In the CTA strategy, each particle k shares information with its neighbors in order to improve its estimation accuracy of w^o. Each particle first averages the estimates $w_{m,i-1}$ from its neighbors $m \in \mathcal{N}_{k,i}$, and then uses this average estimate and its local data $\{d_k(i), u_{k,i}\}$ to improve its own estimate of w^o via an adaptation step. Specifically, particle k performs the

following steps (compare with the no-cooperation case (20)–(21)):

$$\psi_{k,i-1} = \frac{1}{|\mathcal{N}_{k,i}|}\sum_{m\in\mathcal{N}_{k,i}} w_{m,i-1} \quad (26)$$

$$e_k(i) = d_k(i) - u_{k,i}(\psi_{k,i-1} - \ell_{k,i}) \quad (27)$$

$$w_{k,i} = \psi_{k,i-1} + \mu u_{k,i}^* e_k(i) \quad (28)$$

Step (26) performs simple averaging; it can be replaced by a more general convex combination step where the estimates from different particles are weighted differently [5], [6]. For illustration purposes, it is sufficient for the discussion in this article to proceed with (26). The vector $w_{k,i}$ in (28) continues to denote the estimate for w^o by particle k at time i. This estimate is now based on data collected by particle k and on data received from the neighbors of node k through the sharing of $\{w_{m,i-1}\}$. Using the updated estimate $w_{k,i}$, the particle can now assess its location $\ell_{k,i}$ and determine the direction towards which it should bias its probabilities of displacement as in (23). The particle subsequently updates its location as in (24)–(25).

When cooperation is permitted, as in the CTA strategy, the particles can further use their interaction to improve the cohesion of their motion (see, e.g., [6], [8]–[10]). This can be accomplished by adding a scaled driving term to expressions (24)–(25) as follows. Let

$$\delta \triangleq \frac{1}{|\mathcal{N}_{k,i}|-1}\sum_{m\in\mathcal{N}_{k,i}\setminus k}\left(1-\frac{\epsilon}{\|\ell_{m,i}-\ell_{k,i}\|}\right)\cdot(\ell_{m,i}-\ell_{k,i}) = \begin{bmatrix}\delta_x(i)\\ \delta_y(i)\end{bmatrix} \quad (29)$$

where ϵ is a small positive number related to the distance that we would like the nodes to maintain from each other on average. Then, particle k adjusts its location by replacing (24)–(25) by the following updates:

$$x_k(i+1) = x_k(i) + \alpha_{k,x}\Delta_x + \gamma\cdot\delta_x(i) \quad (30)$$

$$y_k(i+1) = y_k(i) + \alpha_{k,y}\Delta_y + \gamma\cdot\delta_y(i) \quad (31)$$

where γ is a nonnegative scalar; the larger the value of γ, the closer the particles stay to each other. And the $\{\alpha_{k,x}, \alpha_{k,y}\}$ are realizations of Bernoulli variables whose distributions are determined by the probabilities $\{p_{k,x}, p_{k,y}\}$.

C. Adapt-Then-Combine (ATC) Strategy

The third mode of operation we consider is one where the particles are still allowed to interact with their neighbors except that now the adaptation step precedes the combination (averaging) step [5]. Specifically, in the ATC implementation, each node k first processes its local data to update its weight estimate $w_{k,i-1}$ and subsequently averages the updated estimates from its neighbors:

$$e_k(i) = d_k(i) - u_{k,i}(w_{k,i-1} - \ell_{k,i}) \quad (32)$$

$$\psi_{k,i} = w_{k,i-1} + \mu u_{k,i}^* e_k(i) \quad (33)$$

$$w_{k,i} = \frac{1}{|\mathcal{N}_{k,i}|}\sum_{m\in\mathcal{N}_{k,i}} \psi_{m,i} \quad (34)$$

Again, step (34) performs simple averaging; it can be replaced by a more general convex combination step [5], [6]. The vector $w_{k,i}$ in (34) continues to denote the estimate for w^o by particle k at time i. Using $w_{k,i}$, the particle can assess its location $\ell_{k,i}$ and determine the direction towards which it should bias its probabilities of displacement as in (23). The particle subsequently updates its location as in (24)–(25) or (30)–(31).

III. Simulations Results

The table below lists the numerical values of the parameters used in the simulations. The target is located at coordinates $(x_o, y_o) = (90, 90)$ and $N = 50$ particles are placed randomly inside the rectangular region $[0, 10]\times[0, 10]$. Subsequently, the positions of the particles start to evolve according to one of the strategies described in the prior sections: no-cooperation, CTA cooperation, and ATC cooperation. We limited the number of neighbors to 5 for CTA ad ATC.

TABLE I
Simulation parameters.

N	μ	$\beta_v = \beta_u$	$\Delta x = \Delta y$	γ	ϵ	r
50	0.1	0.01	1	0.25	1	$4\sqrt{\Delta x^2 + \Delta y^2}$

Regardless of the strategy employed, the coordinates of the center of gravity of the network at any particular time instant are defined by averaging the x and y coordinates of the particles, namely,

$$\mathsf{cg}_x(i) \triangleq \frac{1}{N}\sum_{k=1}^{N} x_k(i), \quad \mathsf{cg}_y(i) \triangleq \frac{1}{N}\sum_{k=1}^{N} y_k(i) \quad (35)$$

Figure 3 shows a snapshot of the evolving network. Over repeated simulations, experiments indicate that the center of gravity of the network for the ATC strategy generally stays ahead of the centers of gravity for the CTA and no-cooperation strategies. At higher noise levels, (say, at higher values for β_v and β_u), the center of gravity of CTA generally outruns the center of gravity for the no-cooperation strategy. Moreover, since cooperation enables cohesion, the particles end up moving more cohesively in ATC and CTA than in the no-cooperation case.

Figure 4 plots the evolution of the mean-square-error curve that measures the squared distance from the center of gravity of the network to the target location (x_o, y_o):

$$\mathrm{MSE}(i) = (x_o - \mathsf{cg}_x(i))^2 + (y_o - \mathsf{cg}_y(i))^2 \quad (36)$$

The curves shown in the figure correspond to a single simulation (one could consider averaging the results over multiple

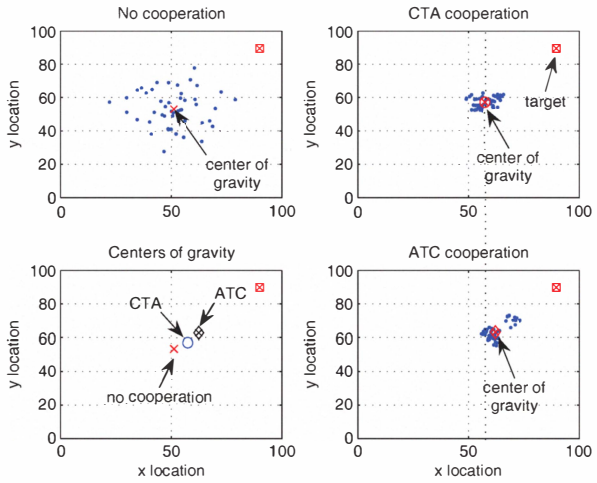

Fig. 3. A typical snapshot over one experiment of the evolving network under three operation modes: no cooperation (top left), CTA (top right), and ATC (bottom right). The red symbols within each of these three subplots indicate the locations of the centers of gravity of the respective networks; these locations are extracted and shown separately in the bottom-left plot with the $\diamond$ symbol corresponding to ATC, the $\circ$ symbol corresponding to CTA, and the $\times$ symbol corresponding to no cooperation. In all cases, the target location is at coordinates $(90, 90)$.

experiments). It is seen that cooperation improves the estimation accuracy with the center of gravity moving closer to the target. It is also seen that the ATC strategy delivers superior performance than the CTA strategy; this is because ATC allows the particles to process and filter the data prior to combination. Figure 5 shows three typical trajectories for one of the particles under the three strategies.

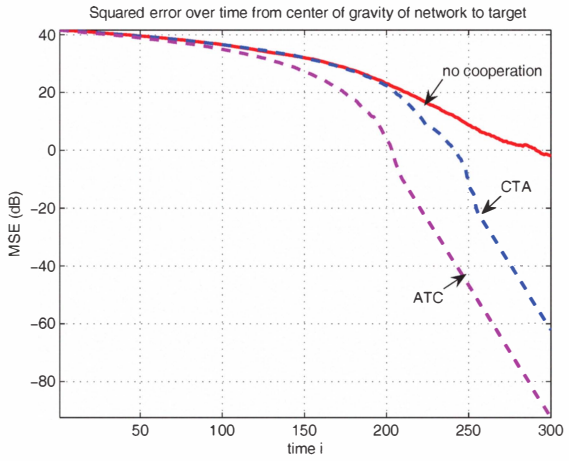

Fig. 4. A typical evolution of the squared error over time from the center of gravity of the network to the location of the target over one experiment. It is observed that as time progresses, the center of gravity of the network gets closer to the target under the ATC strategy.

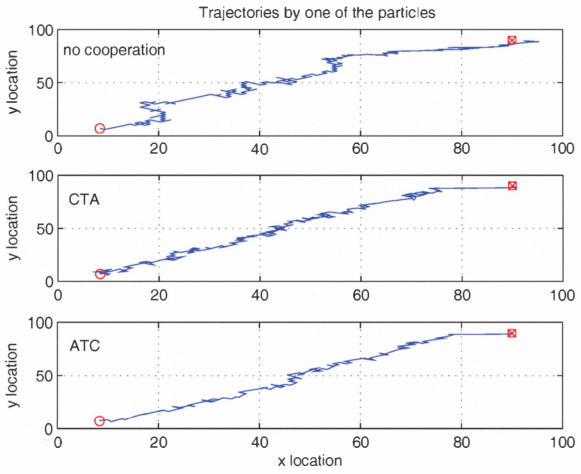

Fig. 5. Typical trajectories by one of the particles over one experiment from its initial location until the end of the simulation for the three modes of operation under consideration.

Figure 6 illustrates the evolution of the variance of the particle locations relative to the center of gravity, along the horizontal and vertical directions. The plots provide an indication about how the particles are scattered around the centers of gravity. It is seen that ATC and CTA lead to more cohesive networks.

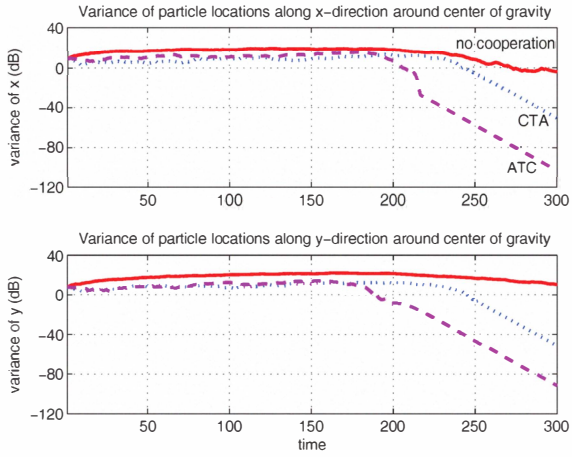

Fig. 6. Typical evolution over one experiment of the variance of the distribution of the particles around the center of gravity for both directions: horizontal and vertical. It is observed that the particles are better clustered under the ATC strategy.

Figure 7 plots the evolution of the network mean-square-deviation (MSD), which is computed as follows (again, the figure shows curves that resulted from a single simulation):

$$\text{MSD}(i) = \frac{1}{N}\sum_{k=1}^{N} \|w_{k,i} - w^o\|^2 \qquad (37)$$

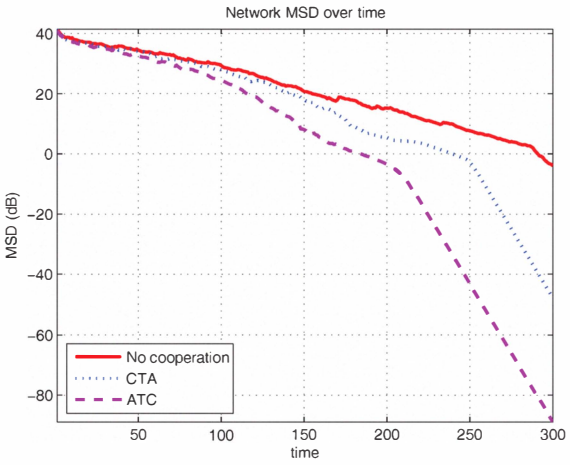

Fig. 7. Evolution of the network MSD over time.

A. Mean-Square Analysis

Introduce the error vectors

$$\tilde{\boldsymbol{w}}_{k,i} \triangleq w^o - \boldsymbol{w}_{k,i}, \quad \tilde{\boldsymbol{\ell}}_{k,i} \triangleq w^o - \boldsymbol{\ell}_{k,i} \tag{38}$$

The first error measures how accurate the estimate of w^o is by node k at time i, while the second error measures how far the location of node k at time i is from the target location. Following arguments similar to those presented in [5], [6], one can study the mean-square behavior of the ATC and CTA diffusion algorithms. Under reasonable assumptions on the data and for sufficiently small step-sizes, one can derive recursions that describe the evolution of these error quantities over time, as well obtain approximate expressions for the steady-state value of the mean-squared error measure, $E\|\tilde{\boldsymbol{w}}_{k,i}\|^2$. To do so, one should extend the argument of [6] to incorporate the presence of the Bernoulli variables $\{\boldsymbol{\alpha}_{k,x}, \boldsymbol{\alpha}_{k,y}\}$. Due to space limitations, we forgo the details here.

REFERENCES

[1] H. C. Berg, *Random Walks in Biology*, expanded edition, Princeton University Press, 1993.

[2] S. J. Shettleworth, *Cognition, Evolution, and Behavior*, Oxford University Press, 2010.

[3] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau, *Self-Organization in Biological Systems*, Princeton University Press, 2003.

[4] I. D. Couzin, "Collective cognition in animal groups," *Trends in Cognitive Sciences*, vol. 13, pp. 36–43, Jan. 2009.

[5] F. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, March 2010.

[6] S-Y. Tu and A. H. Sayed, "Mobile adaptive networks," *IEEE J. Selected Topics on Signal Processing*, vol. 5, no. 4, pp. 649-664, August 2011.

[7] A. H. Sayed, *Adaptive Filters*, Wiley, NJ, 2008.

[8] S-Y. Tu and A. H. Sayed, "Cooperative prey herding based on diffusion adaptation," *Proc. IEEE ICASSP*, Prague, Czech Republic, pp. 3752–3755, May 2011.

[9] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. on Automatic Control*, vol. 51, pp. 401-420, Mar. 2006.

[10] S. Janson, M. Middendorf, and M. Beekman, "Honeybee swarms: How do scouts guide a swarm of uninformed bees?" *Anim. Behav,* vol. 70, pp. 349-358, 2005.

[11] F. Cattivelli and A. H. Sayed, "Modeling bird flight formations using diffusion adaptation," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2038–2051, May 2011.

[12] J. Chen and A. H. Sayed, "Bio-inspired cooperative optimization with application to bacteria motility," *Proc. IEEE ICASSP*, Prague, Czech Republic, pp. 5788–5791, May 2011.

[13] J. Chen, X. Zhao, and A. H. Sayed, "Bacterial motility via diffusion adaptation," *Proc. 44th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, pp. 1930–1934, Nov. 2010.