

# Distributed Gradient Descent in Bacterial Food Search

Shashank Singh<sup>\*1</sup>, Sabrina Rashid<sup>\*2</sup>, Saket Navlakha<sup>3</sup>, Ziv Bar-Joseph<sup>4†</sup>

<sup>1</sup> Machine Learning Department and Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213.

<sup>2</sup> Computational Biology Department, Carnegie Mellon University, Pittsburgh, PA 15213.

<sup>3</sup> The Salk Institute for Biological Studies, Center for Integrative Biology, La Jolla, CA 92037.

<sup>4</sup> Machine Learning Department and Computational Biology Department, Carnegie Mellon University, Pittsburgh, PA 15213.

\* These authors contributed equally.

† Contact author. Email: zivbj@cs.cmu.edu

**Abstract.** Communication and coordination play a major role in the ability of bacterial cells to adapt to ever changing environments and conditions. Recent work has shown that such coordination underlies several aspects of bacterial responses including their ability to develop antibiotic resistance. Here we develop a new distributed gradient descent method that helps explain how bacterial cells collectively search for food in harsh environments using extremely limited communication and computational complexity. This method can also be used for computational tasks when agents are facing similarly restricted conditions. We formalize the communication and computation assumptions required for successful coordination and prove that the method we propose leads to convergence even when using a dynamically changing interaction network. The proposed method improves upon prior models suggested for bacterial foraging despite making fewer assumptions. Simulation studies and analysis of experimental data illustrate the ability of the method to explain and further predict several aspects of bacterial swarm food search.

Supporting movies: [https://www.andrew.cmu.edu/user/sabrinar/Bacteria\\_Simulation\\_Movies/](https://www.andrew.cmu.edu/user/sabrinar/Bacteria_Simulation_Movies/)

## 1 Introduction

There are many parallel requirements of computational and biological systems, suggesting that each can learn from the other. Like virtually all large-scale computing platforms, biological systems are mostly distributed consisting of molecules, cells, or organisms that interact, coordinate, and reach decisions without central control [1], [2]. However, unlike most computational methods, biological systems rely on very limited communication protocols, do not assume that the identity of the communicating agents is known and only utilize simple computations [3]. This makes biological systems more robust to interference and environmental noise, and allows them to function efficiently in harsh settings, a property that is desirable in computational systems as well (for example, sensor networks in remote locations or robot swarms in mines [4]). Recent work demonstrated that for certain network based distributed algorithms, our improved ability to study biological processes at the molecular and cellular levels allows us to both improve our understanding of these biological processes as well as suggest novel ways to improve distributed computational algorithms [5].

For example, a number of key machine learning optimization algorithms, including neural networks and non-negative matrix factorization, have been inspired by information processing in the brain [6], [7]. Here we show that a variant of a very commonly used machine learning coordination algorithm, *distributed gradient descent (DGD)*, is also used by large bacterial swarms to effectively search for food. Similar to the regular gradient descent setting [8], [9], by sensing the food gradient, each cell has its own belief about the location of the food source (Figure 1a). However, given potential obstacles in the environment the bacteria operate in, as well as limits on the ability of each cell to accurately detect and move toward the food source in a noisy environment, the individual trajectories may not produce the optimal path to the food source. Thus, in addition to using their own belief each cell also sends and receives messages from other cells (either by secreting specific proteins or by physical interaction; Figure 1(b)), which it then integrates to update its own belief and determine its next movement direction and velocity. The process continues until the swarm converges to the food source.

While it is possible to see (both in experiments and in simulations below) that the group based approach improves upon the time it takes a single cell to reach the food source, and several aspects of the molecular pathways involved in the communication between bacterial cells have been studied [10], the *computations* that the cells perform have not been well characterized. Current models [11],[12] are largely based on differential equation methods. While these can indeed lead to fast coordinated search, they do not fully take into account the dynamically changing topology of cells' interaction network over time. Furthermore, they make unrealistically strong assumptions about the abilities of cells to identify the source(s) of the messages and to utilize a large (effectively continuous valued) set of messages, which is unrealistic given the limited computational powers bacteria cells possess.

Here, we show that a distributed gradient descent model that makes biologically realistic assumptions regarding the dynamics of the agents, the size of the set of the messages, and their ability to identify senders, can solve the bacterial food search problem more efficiently (in terms of the overall complexity of messages sent) and more quickly (in terms of the time it takes the swarm to reach the food source) when compared to current differential equation models. We argue that our model is in fact a distributed pseudogradient descent method,<sup>1</sup> and hence we can adapt proof by [13] to show that our model converges to a local minimum, under reasonable assumptions on

---

<sup>1</sup> *pseudogradient* here refers to the fact that each agent's *expected* direction of movement at each time step is a descent direction, even though it may not be a descent direction in actuality due to stochasticity.

how bacteria communicate. Simulation studies indicate that the solution is feasible and leads to improvements over prior methods and over single cell and single swarm behavior. These efficient and robust bacterial algorithms are thus suitable to distributed sensors or wireless networks that operate under strict communication and computation constraints [2], [14].

## 2 Methods

### 2.1 A computational framework for understanding bacterial food search

While bacterial food search has been extensively studied, at both molecular and cellular levels, most studies have focused on the individual cell, rather than characterizing the collective performance of a bacterial swarm. To formally analyze this process and to derive methods that can also be used for other tasks by collections of severely restricted agents, we cast collective bacterial food search as a distributed gradient descent (DGD) algorithm for determining the direction of movement for each agent. DGD is a classic distributed optimization algorithm for finding a minimum of an objective function. In our case, the minimum corresponds to the food source, and the objective function incorporates the terrain over which the bacteria search. DGD is based on *message passing* between nodes (individual cells or agents) in a graph. These messages contain information that each node uses to update its own movement direction and velocity. The goal of DGD is for all nodes to converge to a single location in the search space.

While DGD has been extensively studied in several different application areas, there are several differences between standard DGD algorithms [15,16] and bacteria food search. First, classical algorithms assume that message passing occurs on a fixed and static network, whereas for bacteria, the edges and their weights are based on a sphere of influence which changes over time based on each bacterium’s current location. Second, classical algorithms typically do not limit message complexity (in terms of message size and number of messages sent), whereas bacteria have a limited message vocabulary [3] and may attempt to minimize the number of messages sent to reduce energy consumption. Third, classical algorithms require significant data aggregation at individual nodes, whereas bacteria are not believed to collect and store data in such a manner. Considering that bacteria have been solving this optimization problem under these constraints for millions of years, we may be able to learn interesting algorithmic strategies from studying this process. Such algorithms may address a new class of problems that may be important in other applications, such as swarm robotics [18].

For bacteria, the movement of an agent  $i$  at time  $n$  is a function  $d_i(n)$  of two quantities: its own sense of direction  $\theta_i(n)$  (based on the chemical gradient), and the locations and movement directions of other cells in the swarm. Nodes in this graph correspond to bacteria, at some current location  $\mathbf{x}_i(n)$ , and edges, weighted by representing physical distance, exist between two bacteria that lie within a sphere of influence of each other. The challenge lies in updating  $d_i(n)$  based on individual belief and the beliefs of neighbors, while using simple messaging (formalized below). For bacteria, messages passed along edges in the graph contains both homophilic components (attraction and orientation) and a heterophilic component (repulsion). Below, we first present prior work by Shklarsh et al. [12] that describes bacteria food search using a differential equation (DE) model and then present our DGD model, which improves upon the Shklarsh DE model in performance while relying on weaker, more biologically realistic, assumptions.

### 2.2 The Shklarsh model: Cell based differential equation computation

Initial models for bacterial swarm movements assume that cells solve a system of differential equations (DE) to determine their next move [12]. We briefly review this model below. The

model assumes that individual cells follow a chemical gradient of food source by decreasing their (random) tumble frequency in high concentration and thus largely move in the direction of the food. Specifically, the bacteria perpetually moves in a direction which it repeatedly perturbs randomly. The frequency and magnitude of these perturbations are inversely related to the change in the food concentration between iterations, with the rough effect that the agent continues to move in directions along the gradient. Formally, under these assumptions, at time  $n$ , the bacterium changes its direction by an angle  $\theta(n)$ , which is a function of  $\Delta c(n)$ , the difference in food concentration between the current and previous time steps. Specifically, the new tumbling angle  $\theta(n)$  is sampled randomly from a Gaussian distribution  $\theta(n) \sim N(\theta(n - \Delta n), \sigma(\Delta c(n))^2)$  centered at the previous angle  $\theta(n - \Delta n)$ , with the variance  $\sigma(\Delta c(n))^2$  given as:

$$\sigma(\Delta c(n))^2 = \begin{cases} 0; & \Delta c(n) \geq 0 \\ \pi; & \Delta c(n) < 0 \end{cases}. \quad (1)$$

Thus, based only on its own perception of the food gradient, the  $i^{\text{th}}$  agent updates its location  $\mathbf{x}_i(n)$  according to

$$\mathbf{x}_i(n + \Delta n) = \mathbf{x}_i(n) + \mathbf{v}_i(n) \cdot \Delta n, \quad (2)$$

where  $\mathbf{v}_i(n)$  is the unit vector in the direction of the movement.

$$\mathbf{v}_i(n) = (\cos(\theta(n)), \sin(\theta(n))). \quad (3)$$

So far we have discussed movements based on sensing by individual cells (agents). However, in almost all cases, cells move in large swarms. Communication among swarm members, and between swarms, improves the ability of individual cells to handle obstacles in the direction of the food source leading to faster and more efficient ways to reach the food. The communication among agents is divided into three components: (1) *repulsion* from very close agents to avoid collisions; (2) *orientation* to match the directions of neighboring cells, and (3) *attraction* to to distant agents to keep the swarm unified (Figure 1(b)). The model of Shklarsh et al. assumes that cells align their trajectory with the direction of the other cells if they are close enough, while being attracted toward cells that are relatively far away.

Denote by  $\mathbf{u}_i(n)$  the vector agent  $i$  computes using the messages from the other cells (in this model, movements are of fixed length and so the only variable in each iteration is the direction). Then, if any other agents  $j$  are within the physical interaction (repulsion) range  $RR$  as shown in Figure 1(b), the Shklarsh model sets:

$$\mathbf{u}_i(n) = - \sum_{j \in B_{RR}} \frac{\mathbf{x}_i(n) - \mathbf{x}_j(n)}{\|\mathbf{x}_i(n) - \mathbf{x}_j(n)\|}. \quad (4)$$

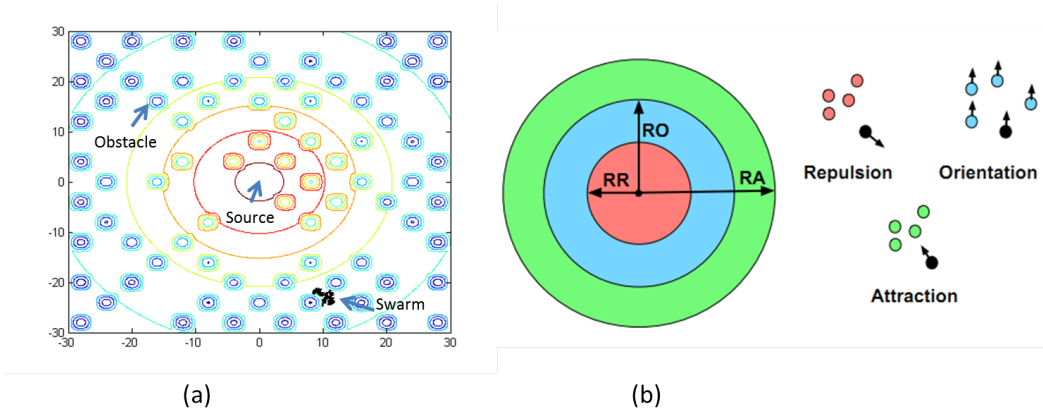
Otherwise, the Shklarsh model sets:

$$\mathbf{u}_i(n) = \sum_{j \in B_{RO}} \mathbf{v}_j(n) + \sum_{j \in B_{RA}, j \notin B_{RO}} \frac{\mathbf{x}_i(n) - \mathbf{x}_j(n)}{\|\mathbf{x}_i(n) - \mathbf{x}_j(n)\|}, \quad (5)$$

where the first sum is over all agents  $j$  in the orientation range  $RO$  and the second sum is over all agents in the attraction range  $RA$  (but not in the range  $RO$ ). Next, the agent combines the messages it received with its own observation, resulting in the following modification of equation (2):

$$\mathbf{x}_i(n + \Delta n) = \mathbf{x}_i(n) + \left( \frac{\mathbf{u}_i(n)}{\|\mathbf{u}_i(n)\|} + w\mathbf{v}_i(n) \right) \cdot \Delta n, \quad (6)$$

where  $w$  is a scalar weighting factor.



**Fig. 1. (a):** Terrain model for bacterial food search. Obstacles are randomly placed and the food source is at the center of the region. Contours display the diffusion of the food source gradient. **(b):** Dynamics of repulsion, orientation, and attraction for a single bacterial cell in the Shklarsh et al. model. RR = radius of repulsion, RO = radius of orientation, and RA = radius of attraction. While we still maintain the physical (RR) versus communication (RO, RA) split between the repulsion and attraction / orientation information, our model does not assume that the identity of the sender is known and so it does not distinguish cells in the RO and RA locations.

### 2.3 A distributed gradient descent model

While the model presented by Shklarsh et al. [12] captures the basics of bacterial swarm movements, it suffers from several problems which make it unrealistic and, as we show in the Results section, less efficient. First, the model assumes that cells can determine their exact distances from *each* other cell (as can be seen by the summations over neighbors cells in the orientation and attraction ranges). This implicitly assumes that cells can *identify* individual senders, which is very unlikely due to the large and dynamic nature of bacterial swarms. In addition, the model assumes that cells can interpret complex (real-valued) messages in the forms of other cells locations and orientations, which is again unrealistic [3]. Finally, the model assumes that, within each of the ranges above, each cell exerts the same influence regardless of their distance from the receiving cell, which is again unrealistic due to the nature of the communication channel (diffusion of a secreted protein). We have thus modeled bacteria food search using a DGD model that relaxes many of these assumptions while still allowing cells to (provably) reach an agreement regarding the direction of movement and eventually the location of the food source, as observed in nature.

Our model still distinguishes between physical interactions (leading to repulsion) and messages (secreted proteins). The former is handled by summing up the number of cells that are in physical proximity without relying on their exact location. However, we make several changes to Equation (5). First, we remove the requirement that cells can identify the distance and direction to each other cell (and thus determine whether to use the attraction or orientation terms). Instead, we simply sum over all cells, taking into account their relative influences under the assumption that message strength decays exponentially with distance [19]. Second, we discretize the messages that cells receive, resulting in simple messages with finitely many possible values. The changes lead to the following modification of equation (5):

$$\mathbf{u}_i(n) = D_{L,T} \left( \sum_j \exp(-(C_o \|\mathbf{x}_i(n) - \mathbf{x}_j(n)\|)) \mathbf{v}_j(n) + \sum_j \frac{\exp(-(C_a \|\mathbf{x}_i(n) - \mathbf{x}_j(n)\|) (\mathbf{x}_i(n) - \mathbf{x}_j(n)))}{\|\mathbf{x}_i(n) - \mathbf{x}_j(n)\|} \right). \quad (7)$$

Here,  $D_{L,T}$  is a discrete thresholding operator parametrized by  $L$ , a positive integer denoting the number of possible messages, and  $T$ , an upper bound above which all messages as treated as the highest value possible (see [20] for the exact construction of this “stone-age computing” threshold which has been used in ant models).  $C_a$  and  $C_o$  are positive diffusion constants, determining how

quickly the attraction and orientation signals diffuse from the source agent. Typically,  $C_o > C_a$ , in which case nearby agents are influenced more by the orientation component and faraway agents are influenced more by the attraction component. Under this model, bacteria communicate orientation and attraction information using only  $3 + \log_2 L$  bits to communicate ( $\log_2 8$  bits for direction and  $\log_2 L$  bits for magnitude). In addition, we also add a Gaussian noise component with a small variance  $\sigma$  to make this process stochastic. Note that the individual component of the agents' movement (based on the chemical gradient they perceive) is identical to that of the Shklarsh model, and that we are modifying only the communication model. See appendix for the complete combined model.

### 3 Convergence theorem

Unlike standard distributed gradient descent algorithms, the model described above does not rely on a fixed network. Instead, in each iteration, the topology of the network (and thus the weights placed on neighbors) changes with their movement. To prove that using such network (and the computation we assume) indeed leads to convergence of the swarm as seen in nature (regardless of agents' starting locations), we adapt a convergence theorem for distributed pseudogradient descent from [13]. The convergence proof below is only focused on the attraction component of the model (see Discussion). Note that this theorem holds for both synchronous and asynchronous message passing, so that the model could be generalized to the case where agents' messages themselves travel stochastically.

**Convergence Theorem:** Let  $x_i(n) \in \mathbb{R}^2$  denote the position of the  $i^{\text{th}}$  agent at time  $n$  and let  $J : \mathbb{R}^2 \rightarrow \mathbb{R}$  denotes the concentration of the food source (the objective function). Under assumptions A1-A6 (see Appendix), the following hold:

1. There exists a constant  $\gamma^* > 0$  such that, if each step size is at most  $\gamma^*$  (i.e.,  $\sup_{i,n} \gamma_i(n) \leq \gamma^*$ ), then, with probability 1, for all distinct agents  $i$  and  $j$ ,

$$\lim_{n \rightarrow \infty} (x_i(n) - x_j(n)) = 0.$$

2. If, in addition, the level sets of  $J$  are compact, then, with probability 1, for all agents  $i$ ,

$$\lim_{n \rightarrow \infty} \|\nabla_x J(x_i(n))\|_2 = 0.$$

3. Finally, if, in addition, all stationary points of  $J$  are minima (e.g., if  $J$  is convex), then, with probability 1, for all agents  $i$ ,

$$\lim_{n \rightarrow \infty} J(x_i(n)) = \inf_x J(x).$$

The three components of the above theorem gives sufficient conditions for the agents to converge spatially, to converge to a (connected set of) stationary points, and to converge to a (connected set of minima), respectively. Assumptions A1-A4 (stated precisely in the Appendix) are that  $J$  is sufficiently smooth (its gradient is Lipschitz continuous) and that the agents' communication network is sufficiently dense so that information from any agent eventually propagates through the swarm. Assumptions A5 and A6 correspond to Lemmas 1 and 2 which we explicitly discuss below.

In the case of distributed pseudogradient descent, the main idea of the proof is that there exists a sequence  $\{y(n)\}_{n=1}^{\infty}$  such that, if, at time  $n$ , all agents were to cease to follow the gradient

while continuing to move according to the attraction term (i.e., weighted averaging), then all agents would converge to  $y(n)$ . For example, in the case that all agents communicate with equal weights,  $y(n)$  is simply the mean of the agents' positions at time  $n$ . Under reasonable assumptions on the edge weights, information from each agent is likely to propagate throughout the swarm and so the agents will converge on  $y(n)$  (plus individual noise), regardless of their starting position. Furthermore, once the agents are sufficiently close, the change in  $y(n)$  (which is a weighted average of the gradients perceived by each agent) in each iteration becomes approximately proportional to  $J(y(n))$ , so that the swarm collectively behaves as a traditional gradient descent. The formal proofs rely on martingale convergence results and can be found in [13].

Two lemmas are needed to establish that our algorithm is indeed a distributed pseudogradient algorithm before we can apply the results of [13]. The lemmas essentially state that (a) the agents expect to move in a descent direction and (b) the variance in the agent's movements is at most proportional to the magnitude of the true gradient (so that, for example, it vanishes near stationary points and hence agents cease to explore once they have located an optima).

**Lemma 1:** The expected gradient sensed by each agent, conditioned on the history of the algorithm, is in a descent direction. That is, if  $\{F_n\}_{n=1}^\infty$  is the filtration underlying the sequence of agents' movements, then, for all times  $n$  and agents  $i$ ,

$$\mathbb{E} [\nabla_x J(x_i(n)) \cdot s^i(n) | F_n] \leq 0. \quad (8)$$

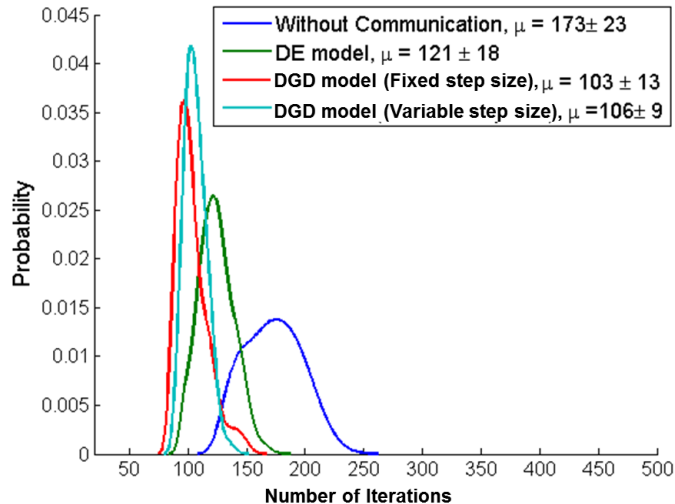
**Lemma 2:** The variance in the gradient sensed by each agent is at most proportional to the true gradient at that agent's location. That is, there exists  $K_0 > 0$  such that,

$$\mathbb{E} [\|s^i(n)\|_2^2 | F_n] \leq -K_0 \mathbb{E} [\nabla_x J(x_i(n)) \cdot s^i(n) | F_n]. \quad (9)$$

for all times  $n$  and agents  $i$ . Here,  $s^i(n)$  denotes the gradient perceived by the agent (i.e., the agent's movement due to tumbling but not due to communication). The proof of Lemma 1 is based on the fact that agents continue moving in the correct descent direction once they have detected an increase in food gradient in one of their tumbles (until that direction ceases to be a descent direction). As we show, to achieve the expected decrease required in the lemma for each step, we need to change either the tumbling distribution assumed in the original model, or the number of tumbles per step (i.e. prior to communicating a new location). While the former (a uniform tumbling distribution) is less likely in practice, the latter (communicating only when a new descent direction is established) both makes biological sense [21] and, as we show empirically in the Results section, reduces the time it takes the swarm to reach the food source. To prove the second lemma we again need to modify the original algorithm somewhat, so that the step size (amount of progress made in the direction computed) is proportional to the detected gradient, causing the agents to slow as they approach the food source. This may be implemented in practice via a feedback loop used by bacterial cells [22]. Detailed assumptions and rigorous proofs of both lemmas and the convergence theorem are given in the Appendix.

## 4 Empirical Results

To determine whether the restricted communication model we assume can indeed lead to efficient convergence we performed several simulations of bacterial food search. First, we compare the search efficiency of bacterial food search with and without communication, and between our model and the Shklarsh et al. model. Second, we introduce multiple swarms and test how their trajectories affect each other. Third, we evaluate our method using real experimental data. Finally, we explore the predictions of the model for a setting in which a fraction of the bacteria are



**Fig. 2.** Comparison between the original Shklarsh model and the DGD model.  $\mu$  denotes the mean number of iterations per agent, plus or minus the standard deviation.

behaving differently than the others. This latter point is of great current interest since ‘cheaters’ (cells that receive messages, but do not spend the energy on sending them) may be responsible for a form of antibiotic resistance that has been recently observed [23]. Thus, the results presented below are of interest to both the computational part of this work (efficient and robust DGD model) and the biological aspects (models of bacterial coordination).

To save space, in the remainder of this section we refer to the Shklarsh et al. model as the ‘DE’ model and our model as the ‘DGD’ model (even though both rely on updating magnitude and direction in time steps).

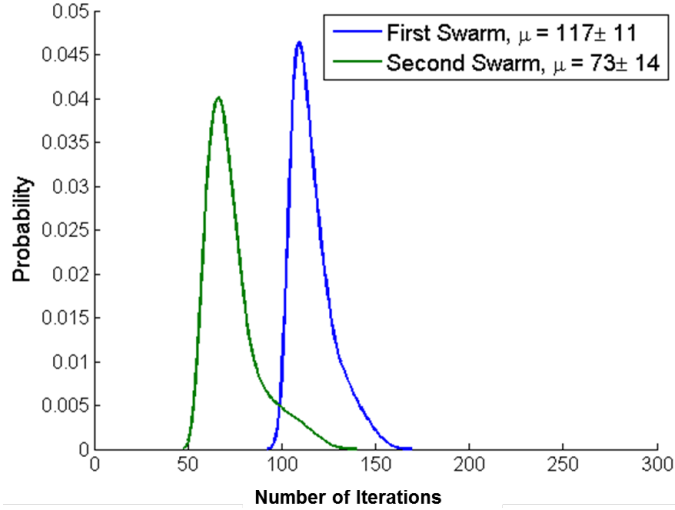
### Performance on a realistic food search simulation

To evaluate the performance of our method we first tested it using the terrain and obstacle model from Shklarsh et al. [12] (see Appendix). In this settings we varied the number of agents, the communication between agents and the number of swarms.

The quantity we compared across all these comparisons was the time it took cells to reach the food source (in terms of steps, since both algorithms can be run synchronously). Figure 2 presents the distribution of the number of steps it takes cells to reach the food source under several different models. The first is a model without communication (i.e. each cell can only sense the food gradient, but does not receive or send any message). The second is the Shklarsh et al. model with adaptive weighting, and the other two models are our DGD model with fixed or variable step size (the latter is required for convergence proof above while the former is usually used in bacterial models). As can be seen, communication greatly improves the time it takes these cells to reach the food source, which may explain why such a secretion based communication system has evolved in these cells. As for the specific communication model, the DGD model improves the results when compared to the DE model, even though our DGD model severely restricts the set of messages that are used. In fact, the discretization of messages decreases both the mean and variance of the distribution. This is likely due to the fact that, by thresholding, the discretization step is effectively reducing noise, which can be rather large for each individual message. In addition, the distance-weighted edges (corresponding to the diffusion rates of secreted communication proteins) also improve the performance of the method. Simulation movies of the with communication and without communication setting can be observed [here](#).



Figure 3 displays the effect on the time it takes cells to reach the food source if another swarm is added to the simulation. In this sequential set up, the second swarm starts 50 iterations after the first swarm (see [movie](#)). As can be seen, the fact that the first swarm was already able to successfully navigate to the food source enables the second swarm to utilize (at least partially) the trajectory they identified to further reduce the time it takes to reach the food. Interestingly, while the improvement for the second swarm is indeed large, we also see a *decrease* in the performance of the first swarm compared to the single swarm result presented in Figure 2. This is due to the negative influence that the second swarm has on the first when it enters the region of influence. Since the second swarm starts in the opposite direction of the food source, the first swarm is (partially) adjusting its direction in the wrong way (based on attraction to cells in the second swarm) which increases the number of iterations it takes cells to reach the food source.



**Fig. 3.** Distribution of number of iterations for two sequential swarms using variable step sizes.  $\mu$  denotes the mean number of iterations per agent.

### Model evaluation and comparison using real data

To further test our model and compare it with the Shklarsh et al. model we used experimental data from Taylor and Welch [24]. In their experiments, Taylor and Welch focus on the effects of mobility on swarm performance when a food gradient exists. Note that immobility of a large fraction of cells directly affects swarms coordination since those that are still mobile are receiving much fewer accurate messages impairing their ability to move in the correct direction. Thus, as the fraction of immobile cells increases, we expect several of the mobile cells to make at least some moves in the wrong direction. Indeed, this is what Taylor and Welch observed in their experiments. To quantify the impact of immobile agents they measured the ratio between cells moving *towards* the food source and those moving *away* from the food source at a specific time point. Formally, we define this as:

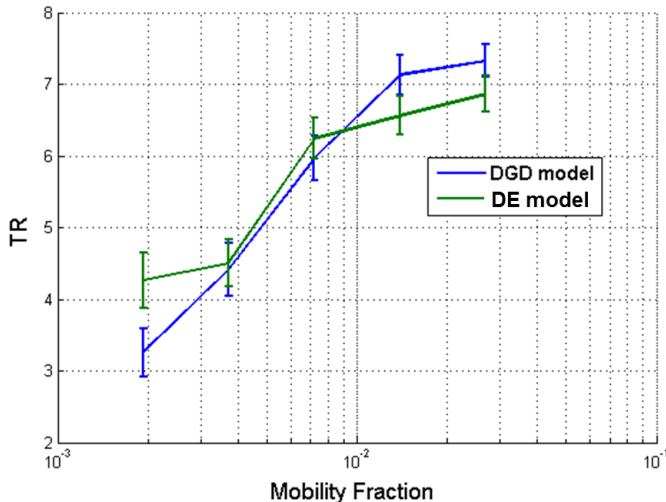
$$TR_i = \frac{\max \left\{ d^S - d_i^j : d^S \geq d_i^j \right\}}{\left| \min \left\{ d^S - d_i^j : d^S \leq d_i^j \right\} \right|} \quad (10)$$

where  $TR_i$  denotes the tracking ratio during iteration  $i$ ,  $d^S$  is the initial distance of the swarm center from the food source and  $d_i^j$  is the distance of cell  $j$  from the food source at iteration

*i.* In their experiments, cells start far enough from the food source such that the impact of the food gradient is minimal early on and so cells depend on swarm communication to move in the correct direction. Thus, when a large fraction of cells is immobile the swarm would expand in both direction leading to a TR that is close to 1. TR greater than 1 represents chemotactic (or coordinated) movement.

When mobility is drastically impaired (only 1% of cells can move) Taylor and Welch report a TR of close to 1 which. The TR increases as the fraction of immobile cells decreases. These results were then used to identify mutants associated with movement since their TR was reduced when compared to WT cells.

To test our model in this setting we have simulated the immobilization of different fractions of cells and for each computed the resulting TR after a set number of iterations. Fig. 4 presents the resulting tracking ratios when using 10000 cells Note that while in our simulation we observe a TR that is higher than 1, even when only 0.5% are mobile, this can be attributed to the much smaller number of cells in our simulations since for a such a small set all mobile cells can likely communicate with all other mobile cells whereas when the number of cells is larger this ability may be reduced. However, as was observed in experimental results we also see a rise in TR as the fraction of mobile cells increases. Further, while the TR obtained by our DGD method for the least constrained settings (3% mobile) is slightly higher than the Shklarsh et al. TR for the same setting, the initial TR is significantly lower than the TR obtained by the Shklarsh model (and closer to the real result) indicating that our model can still capture the communication capabilities of bacterial cells with weaker assumptions and less computational power.



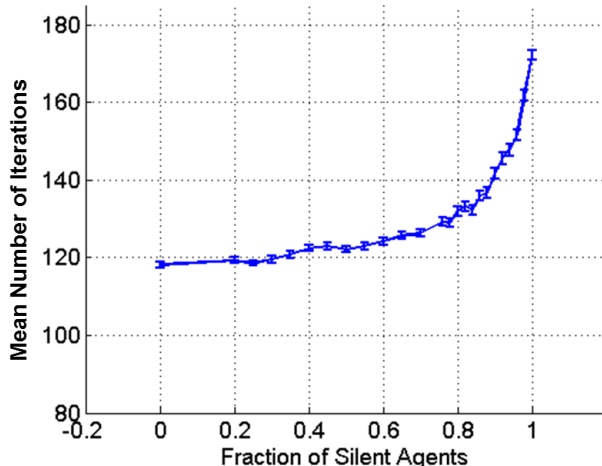
**Fig. 4.** Plot of TR vs mobility fraction. The results are generated with 300 independent trials using swarm population of 10000.

### Sensitivity to silent agents

Previous work has shown that some cells in a population become ‘silent’ [25],[26], [27]. These cells receive messages from the other cells but do not send messages themselves. While such behavior is beneficial from the individual standpoint (less energy required to secrete the proteins) it may be harmful for the population as a whole since if these ‘silent’ cells proliferate the population will lose its ability to utilize communication to improve food search. Recent work has shown that stochastic activation of such silencing and other individual behavior mechanisms can explain how they can be advantageously used (for example, for antibiotics resistance) without affecting the overall ability

to utilize population based coordination. We have thus used the obstacle model again to study the sensitivity of swarm performance to the fraction of silent cells in the population. For this, we varied the fraction of silent agents from 0 to 1. As can be seen in Figure 5 up to a certain threshold we do not see a large impact for the increase in silent cells which supports the recent findings of [23]. Specifically, based on our models the performance of the population is only 20% less than the optimal even if 85% of them are silent. These results provide the first quantitative analysis of this phenomena and can be directly translated to an experimental hypothesis.

We have further analyzed various aspects of the communication model to determine the roles of each type of the message being sent (orientation, attraction) or sensed (physical proximity), in terms of the impact on the time taken to reach the food source. See Appendix for details.



**Fig. 5.** Mean path lengths over silent fractions, based on 100 trials. More silent bins were taken in the range of .7 to 1 to illustrate the phase transition. Error bars indicate standard deviations.

## 5 Discussion and conclusions

We have shown, both theoretically and using simulations, that a distributed gradient descent model can efficiently be implemented under severe limitations on the way the agents communicate. These include limits on the complexity of the messages and the identifiability of the senders, while at the same time assuming a dynamic environment where neighbors’ locations (and their influence) constantly change. We have proved that, under reasonable biological assumptions, the communication algorithm discussed is likely to converge, which helps explain how bacteria can efficiently coordinate food search in harsh environments, and which improves upon prior models suggested for this process. Our convergence proof only holds for the attraction information and does not hold for the repulsion and direction component of the computation performed by each agent. Since the orientation information is also a vector being averaged (over all neighbors), we believe the proof can be extended to include this communication term as well, though a key challenge would be to understand the sometimes competing goals of local versus global improvement near certain obstacles. Repulsion is more difficult to analyze, since it is not based on averages. However, as we show in the Appendix, the attraction term has by far the most significant effect on the performance of the swarm, with the other two communication terms having lesser impacts on the time it takes cells to reach the food source. Hence, for computational implementation of the method, it may suffice to use the attraction term, in which case convergence is guaranteed.

## References

1. Seeley T. D. (2002). When is self-organization used in biological systems?. *Biol. Bull.*, 202(3),314-318.
2. Babaoglu O. et al (2006). Design Patterns from Biology for Distributed Computing. *ACM Transactions on Autonomous and Adaptive Systems*, 1:26-66.
3. Cheong, R., Rhee, A., Wang, C. J., Nemenman, I., Levchenko, A. (2011). Information transduction capacity of noisy biochemical signaling networks. *Science*, 334(6054),354-358.
4. Ong, K. G. et al. (2004). A wireless sensor network for long-term monitoring of aquatic environments: Design and implementation. *Sensor Letters*, 2(1), 48-57.
5. Navlakha, S., Bar-Joseph, Z. (2014). Distributed information processing in biological and computational systems. *Communications of the ACM*, 58(1), 94-102.
6. Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. of the Nat. Academy of Sciences*, 79(8), 2554-2558.
7. Bishop, C. M. (1996). Neural Networks for Pattern Recognition, 1st edn. USA: *Oxford University Press*
8. Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., Tibshirani, R. (2009). *The elements of statistical learning*, 2(1). New York: springer.
9. Mezard, M., Montanari, A. (2009). *Information, physics, and computation*. Oxford University Press.
10. Waters, C. M., Bassler, B. L. (2005). Quorum sensing: cell-to-cell communication in bacteria. *Annu. Rev. Cell Dev. Biol.*, 21, 319-346.
11. Torney, C., Neufeld, Z., Couzin, I. D. (2009). Context-dependent interaction leads to emergent search behavior in social aggregates. *Proc. of the Nat. Academy of Sciences*, 106(52), 22055-60.
12. Shklarsh, A., Ariel, G., Schneidman, E., Ben-Jacob, E. (2011). Smart swarms of bacteria-inspired agents with performance adaptable interactions. *PLoS comp. bio.*, 7(9), e1002177.
13. Tsitsiklis, J. N., Bertsekas, D. P., Athans, M. (1986). Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE trans. on auto. control*, 31(9), 803-812.
14. Anastasi, G., Conti, M., Di Francesco, M., and Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad hoc networks*, 7(3), 537-568.
15. Zinkevich, M., Weimer, M., Li, L., and Smola, A. J. (2010). Parallelized stochastic gradient descent. In *Advances in neural information processing systems* (pp. 2595-2603).
16. Li, F., Wu, B., Xu, L., Shi, C., and Shi, J. (2014). A fast distributed stochastic gradient descent algorithm for matrix factorization. In *Proceedings of the 3rd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications* (pp. 77-87).
17. Koutra, D., Ke, T. Y., Kang, U., Chau, D. H. P., Pao, H. K. K., Faloutsos, C. (2011). Unifying guilt-by-association approaches: Theorems and fast algorithms. In *Machine Learning and Knowledge Discovery in Databases*, 245-260. Springer Berlin Heidelberg.
18. Rubenstein, M., Cornejo, A., Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198), 795-799.
19. Wartlick, O., Kicheva, A., Gonzalez-Gaitn, M. (2009). Morphogen gradient formation. *Cold Spring Harbor perspectives in biology*, 1(3), a001255.
20. Emek, Y., Wattenhofer, R. (2013, July). Stone age distributed computing. In *Proc. of the 2013 ACM symp. on Principles of distributed computing*, 137-146.
21. Hense, B. A., Kuttler, C., Mller, J., Rothballer, M., Hartmann, A., Kreft, J. U. (2007). Does efficiency sensing unify diffusion and quorum sensing?. *Nature Rev. Microbiology*, 5(3), 230-239.
22. Vladimirov, N., Løvdok, L., Lebedz, D., Sourjik, V. (2008). Dependence of bacterial chemotaxis on gradient shape and adaptation rate. *PLoS computational biology*, 4(12), e1000242.
23. Yurtsev, E. A. et al (2013). Bacterial cheating drives the population dynamics of cooperative antibiotic resistance plasmids. *Molecular systems biology*, 9(1).
24. Taylor, Rion G., and Roy D. Welch. 'Chemotaxis as an emergent property of a swarm.' *Journal of bacteriology* 190.20 (2008): 6811-6816.
25. Velicer, G. J., Kroos, L., Lenski, R. E. (2000). Developmental cheating in the social bacterium *Myxococcus xanthus*. *Nature*, 404(6778), 598-601.
26. Hammerschmidt, K., Rose, C. J., Kerr, B., Rainey, P. B. (2014). Life cycles, fitness decoupling and the evolution of multicellularity. *Nature*, 515(7525), 75-79.
27. Strassmann, J. E., Zhu, Y., Queller, D. C. (2000). Altruism and social cheating in the social amoeba *Dicystostelium discoideum*. *Nature*, 408(6815), 965-967.

# Appendix

## A Methods

### A.1 Complete update rule for a single cell

Below we present the complete update rule performed by each cell in each of the steps of the algorithm to compute its new trajectory. At each iteration, there are three communication components in the BP model, i) repulsion, ii) orientation, and iii) attraction.

As discussed in the main text, we divide the update into two cases. If there are *any* cells within the Repulsion Radius (RR, determined by physical interactions) then the cell moves as follows:

$$\mathbf{u}_i(n) = - \sum_{j \in B_{RR}} \frac{\mathbf{x}_i(n) - \mathbf{x}_j(n)}{\|\mathbf{x}_i(n) - \mathbf{x}_j(n)\|}, \quad (11)$$

where  $x_i(n)$  is the location of agent  $i$  at time  $n$ . Otherwise (if there are no cells in RR) we set:

$$\mathbf{u}_i(n) = D_{L,T} \left( \sum_j \exp(-C_o \|\mathbf{x}_i(n) - \mathbf{x}_j(n)\|) \mathbf{v}_j(n) + \sum_j \frac{\exp(-C_a \|\mathbf{x}_i(n) - \mathbf{x}_j(n)\|) (\mathbf{x}_i(n) - \mathbf{x}_j(n))}{\|\mathbf{x}_i(n) - \mathbf{x}_j(n)\|} \right). \quad (12)$$

In either case, the cell updates its own position by moving according to

$$x_i(n+1) = x_i(n) + \gamma_i(n)(u_i(n) + s^i(n)),$$

where  $s^i(n)$  is the belief of cell  $i$  at time  $n$  based on its sensing of the food gradient ( $s^i(n)$  is a pseudogradient, as shown in Lemma 1 below).  $\gamma_i(n)$  is the step size for agent  $i$  at time  $n$ , which, as discussed in the main text, is a function of the food gradient at the cell's location (a detailed formulation of  $\gamma_i(n)$  is in the proof of Lemma 2 below).

### A.2 Convergence Theorem

In a simplified model that incorporates only the attraction term of communication, our convergence theorem follows from a general convergence theorem proven by Tsitsiklis et al. for a broad class of distributed pseudogradient descent algorithms (Theorem 3.1 and Corollary 3.1 of [13]). Work is needed only to show that our setting satisfies the six assumptions therein (listed and justified below). In order to match the notation of [13], we note that the update equation (using only the attraction term) can be written in the form

$$x_i(n+1) = \sum_j^M a_{ij}(n) x_j(n) + \gamma_i(n) s^i(n) \quad (13)$$

where  $a_{ij}(n)$  is the edge weight between cells  $i$  and  $j$  at time  $n$ .

**Assumptions** The assumptions needed for the general convergence proof are the following:

- A1** There is some  $\alpha > 0$  such that  $a_{ij}(n) \geq \alpha$ , for all times  $n$  and distinct agents  $i$  and  $j$ .
- A2** There exists a constant  $B$  such that the time between consecutive communications between any pair of distinct agents  $i$  and  $j$  is at most  $B_1$ .
- A3** The number of messages communications between any two distinct agents during any duration of length  $B_1$  is bounded by some constant  $B_2$ .

**A4**  $J$  is continuously differentiable and its gradient  $\nabla_x J$  is Lipschitz continuous.

**A5** The conclusion of Lemma 1.

**A6** The conclusion of Lemma 2.

Below we discuss why each of these holds in our setting.

**A1** Since the edge weight  $a_{ij}(n)$  is a positive, strictly decreasing function of the distance  $\|x_i(n) - x_j(n)\|_2$  and, almost surely, the agents move only within a bounded region of  $\mathbb{R}^2$ , such an  $\alpha = \inf_{i \neq j, n} a_{ij}(n) > 0$  exists almost surely.

**A2, A3** This point is somewhat subtle, and quite important for the generality of our work. In our model, each agent regularly (after a fixed number of rounds on individual movement, as discussed in the proof of Lemma 1 below) measures the *weighted average* of the signals from all other agents in the swarm, through the update equation (13). The key observation here is that, in [13] (which only considers communication as being between pairs of agents, and hence phrases the assumption as above), each agent’s update depends *only on the weighted average* of the messages received. Thus, this assumption is *weaker than* the assumption that the weighted average in the update equation incorporates *all* other agents (with weights satisfying the constraints of assumption A1). By construction, our model satisfies this.<sup>2</sup>

**A4** Food is assumed to diffuse smoothly from its source (usually according to a Bessel, Gaussian, or exponential decay function of distance).

**Lemma 1** Conditioned on the history of the algorithm, the expected gradient perceived directly by each agent (not accounting for information from the swarm) is a descent direction. That is,

$$E \left[ \frac{dJ}{dx}(x_i(n)) s^i(n) \right] \leq 0.$$

In order to guarantee this we will allow agents to tumble a fixed number of times between rounds of communication, so that, in expectation, they will find and move along a descent direction.

*Proof of Lemma 1:* We first show the case when the direction after a tumble is chosen uniformly at random, in which case, the proof is straightforward. For sake of generality, we then show the case where the new angle is chosen according to any distribution (such a Gaussian) satisfying a certain weak uniformity condition, allowing multiple tumbles within each round of communication.

*Uniform Case:* In each iteration  $n > 1$ , if the previous direction  $s^i(n-1)$  is not a descent direction (i.e., if  $J(x_i(n)) \leq J(x_i(n-1))$ ), then  $\frac{s^i(n)}{\|s^i(n)\|_2} \sim \text{Unif}(\partial B_1(0))$ ,<sup>3</sup> and the speed  $\|s^i(n)\|_2$  is deterministic given  $s^i(n-1)$  and  $x_i(n-1)$ . Since the function  $s \mapsto \nabla_x J(x_i(n)) \cdot s$  is linear, its expectation over a uniformly distributed variable is 0, and so

$$\mathbb{E} \left[ \nabla_x J(x_i(n)) \cdot s^i(n) \mid F_n \right] = \|s^i(n)\|_2 \mathbb{E} \left[ \nabla_x J(x_i(n)) \cdot \frac{s^i(n)}{\|s^i(n)\|_2} \mid F_n \right] = 0.$$

If the previous direction is a descent direction, the agent retains its previous direction.  $\square$

*General Case:* Now consider a more general algorithm parameterized by a *tumbling distribution* with density  $D$  on  $[-\pi, \pi]$  (above,  $D$  is uniform). In general, a single tumble may be insufficient to ensure that the expected resulting direction is a descent direction, and so multiple

<sup>2</sup> Compare *Example V* of [13], which considers a similar communication pattern in a more abstract setting.

<sup>3</sup>  $B_1(0)$  denotes the unit ball centered at 0, and  $\partial B_1(0)$  denotes its boundary.

tumbles may be necessary. We assume that  $D$  is somewhat uniform in the following sense:

$$0 < c := \inf_{\theta \in [-\pi, \pi]} \int_{\theta}^{\theta + \pi/2} D(\phi_{2\pi}) d\phi, \quad 4$$

i.e., the agent has probability at least  $c \in (0, 1/4]$  of tumbling toward any particular quadrant. This assumption certainly holds for the Gaussian tumbling distribution we use in the main paper, but also allows a broad range of other possibilities, including any distribution whose density is lower bounded away from 0 on  $[-\pi, \pi]$ .

To prove the lemma, let  $\theta_0 = \theta(\nabla_x J(x_i(n)), s^i)$ ,  $\theta_{\ell+1} = \theta_{\ell} + \Delta\theta_{\ell}$ , where each  $\Delta\theta_{\ell} \sim D$  denotes the change in angle due to the  $\ell^{\text{th}}$  tumble.<sup>5</sup> With probability in  $(c, 1-3c)$ , any particular  $|\theta_{\ell}| \leq \frac{\pi}{4}$ , in which case

$$\nabla_x J(x_i(n)) \cdot s^i(n) \geq \frac{1}{\sqrt{2}} \|J(x_i(n))\|_2 \|s^i(n)\|_2.$$

Also, with probability in  $(2c, 1-2c)$ ,  $|\theta_{\ell}| \geq \frac{\pi}{2}$  for all  $\ell \in \{1, \dots, k\}$ , in which case, by the Cauchy-Schwarz inequality,  $\nabla_x J(x_i(n)) \cdot s^i(n) \geq -\|J(x_i(n))\|_2 \|s^i(n)\|_2$ . Otherwise,  $|\theta_{\ell}| \in (\frac{\pi}{4}, \frac{\pi}{2})$ , so  $\nabla_x J(x_i(n)) \cdot s^i(n) \geq 0$ .

Let  $L \in \{0, \dots, k\}$  denote the last  $\ell$  at which  $\theta_{\ell} > \frac{\pi}{2}$  (i.e., after which, the agent maintains its current direction for the remaining  $k - \ell$  tumbles). Since  $L = \ell$  if and only if  $|\theta_0, \dots, \theta_{\ell}| \geq \frac{\pi}{2}$

$$\begin{aligned} \mathbb{E} [\nabla_x J(x_i(n)) \cdot s^i(n) | F_n] &= \sum_{\ell=0}^k \mathbb{E} [\nabla_x J(x_i(n)) \cdot s^i(n) | F_n, L = \ell] \mathbb{P}[L = \ell] \\ &\geq \sum_{\ell=0}^k \|\nabla_x J(x_i(n))\|_2 \|s^i(n)\|_2 \left( \frac{1}{\sqrt{2}} (k - \ell) - \ell \right) 2^{\ell} c^{\ell+1} \\ &= \frac{c \|\nabla_x J(x_i(n))\|_2 \|s^i(n)\|_2}{\sqrt{2}} \left( \sum_{\ell=0}^k (2c)^{\ell} \left( k - (1 + \sqrt{2}) \ell \right) \right). \end{aligned}$$

It is easy to see that this quantity is positive for sufficiently large  $k$ , since, as  $k \rightarrow \infty$ ,  $\sum_{\ell=0}^k \ell (2c)^{\ell}$  converges while  $k \sum_{\ell=0}^k (2c)^{\ell}$  diverges (recalling  $c \in (0, 1/4)$ ).  $\square$

**Lemma 2** The variance of the updates goes to zero as the cost function goes to zero. That is, for some  $K_0 > 0$ ,

$$\mathbb{E} [\|s^i(n)\|^2] \leq -K_0 \mathbb{E} \left[ \frac{dJ}{dx}(x_i(n)) s^i(n) \right].$$

*Proof of Lemma 2:* We assume here that the food follows an isotropic Gaussian distribution centered at the origin. That is, for some  $\sigma > 0$ ,  $\forall x \in \mathbb{R}^2$ ,

$$J(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\|x\|_2^2}{2\sigma^2}\right).$$

Then, for all agents  $i$  and times  $n$ ,

$$\frac{dJ}{dx}(x_i(n)) = -\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\|x_i(n)\|_2^2}{2\sigma^2}\right) x_i(n)$$

<sup>4</sup> For  $x \in \mathbb{R}$ ,  $x_{\pi} = ((x + \pi) \bmod 2\pi) - \pi$  denotes the angle in  $[-\pi, \pi]$  equivalent to  $x$ .

<sup>5</sup> For two vectors  $u$  and  $v$ ,  $\theta(u, v) = \cos^{-1}\left(\frac{u \cdot v}{\|u\|_2 \|v\|_2}\right)$  denotes the (smallest) angle between  $u$  and  $v$ . For notational convenience, we measure angles as lying in  $[-\pi, \pi]$ , with 0 denoting the direction of  $\nabla_x J(x_i(n))$ .

Plugging in

$$x_i(n) = \sum_j a_{ij}(n)x_j(n) + \gamma_i(n)s^i(n)$$

gives

$$\frac{dJ}{dx}(x_i(n)) = -\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\|x_i(n)\|^2}{2\sigma^2}\right) \sum_j a_{ij}(n)x_j(n) - \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{\|x_i(n)\|^2}{2\sigma^2}\right) \gamma_i(n)s^i(n)$$

Now the right-hand side of the lemma:

$$\begin{aligned} & -K_0 \mathbb{E} \left[ \frac{dJ}{dx}(x_i(n))s^i(n) \right] \\ &= K_0 \mathbb{E} \left[ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\|x_i(n)\|^2}{2\sigma^2}\right) s^i(n) \sum_j a_{ij}(n)x_j(n) + \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\|x_i(n)\|^2}{2\sigma^2}\right) \gamma_i(n)s^i(n) \cdot s^i(n) \right] \\ &\geq K_0 \mathbb{E} \left[ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\|x_i(n)\|^2}{2\sigma^2}\right) \gamma_i(n)\|s^i(n)\|^2 \right] \\ &= \mathbb{E} \left[ \exp\left(-\frac{\|x_i(n)\|^2}{2\sigma^2}\right) K' \gamma_i(n)\|s^i(n)\|^2 \right] \geq \mathbb{E} [\|s^i(n)\|^2] \end{aligned}$$

when the step size  $\gamma_i(n)$  satisfies  $K' \gamma_i(n) \geq \exp\left(\frac{\|x_i(n)\|^2}{2\sigma^2}\right)$ .  $\square$

### A.3 Terrain Modeling and Swarm Initialization

We use a similar terrain model to the one used in [12]. Food density and terrain are stationary over time. Food is assumed to be diffuse through the terrain, with a global maximum at the source. Specifically, we modeled the food density as an isotropic Gaussian function:

$$J(r) = \frac{K}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{r^2}{2\sigma^2}\right). \quad (14)$$

In addition, we introduce random obstacles in the form of local minima. In particular, we use the half cosine function to generate field of obstacles:

$$g(r) = \min(0, -4(\cos(\pi r/4) + 0.5)),$$

from which we randomly remove a small number of obstacles to increase trial-to-trial variability.

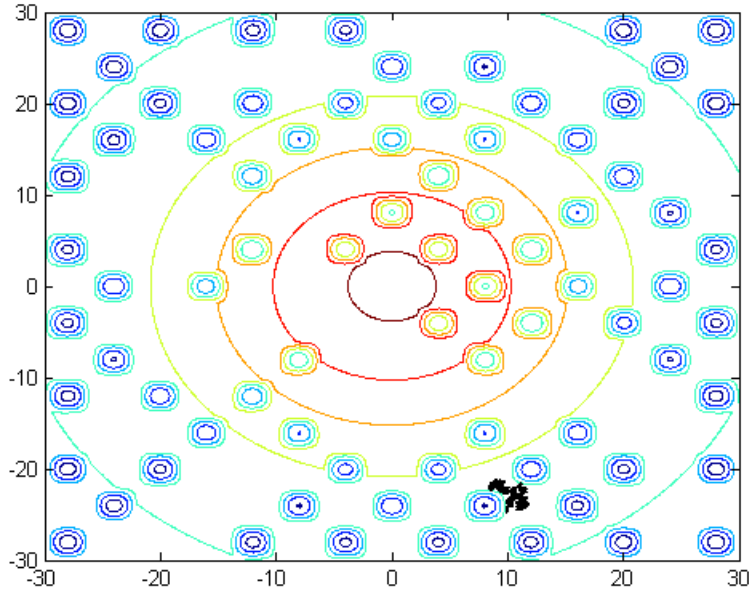
Each swarm is generated by choosing *swarm center* at a certain fixed distance from the food source, with the angle along this distance circle chosen uniformly at random. Agents are placed uniformly in a small square centered at the swarm center, with the constraint that no agent starts on an obstacle. See Figure 6 for an example of a single trial initialization.

## B Simulation parameters

The results presented in the paper are produced with swarms consisting of 30 agents (unless otherwise stated). For the simulation we use a Repulsion Radius of 0.1 and for the [12] model we use 4.0 for Orientation Radius and 4.3 for the Attraction Radius. The discretized thresholding operator  $D_{L,T}$  in our model used  $L = 4$  and  $T = 3$  (see [20]).

All the probability distributions are generated from 300 independent trials. For each trial the number of iterations required for at least 75% of the agents to reach within a fixed radius of the food source was measured. We have set this source radius at 2.5. For modeling the food source we have used  $\sigma^2 = 1000$  and  $K = 200$  in equation (14).





**Fig. 6.** A single swarm in our modelled terrain, immediately after initialization. Contour lines illustrate food concentration (maximized at the origin) and obstacles. Black dots indicate individual agents.

## C Additional Results

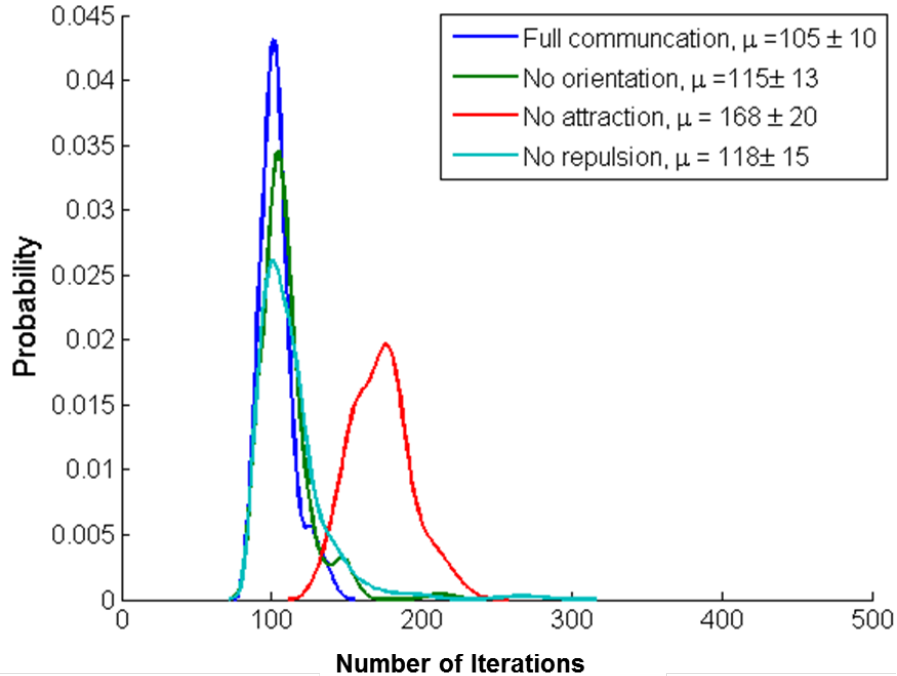
### Effect of different communication components

In addition to the results presented in the main paper we also tested the effects of each of the signals the agent / cell utilizes as part of the DGD model. These include: i) repulsion information, ii) orientation, and iii) attraction. We have evaluated the behavior of the swarms in the absence of either of these component to determine their impact of the ability of cells to efficiently reach the food source.

Fig. 7 presents the simulation results for this analysis. As can be seen, while performance decreases when the orientation and repulsion components are disabled (average number of iterations required to reach the food source increases from 105 to 118 for repulsion and 105 to 115 for orientation) the effect is not large. In contrast the attraction component has a large effect on performance. Specifically, removing this component almost doubles the mean path length (168 iterations). One reason for this observation is that the decay in the weights of attraction components is one fourth of that of orientation (see methods section in the paper), providing opportunity for the swarms to have a longer range of interactions which dominates the swarm movement.

### Effect of using distances to weight messages

To test the impact of taking the distance into account when computing the local descent function we have run our method with and without using distance based weights, keeping all the other parameter settings same. Fig. 8 presents the results of this analysis using two sequential swarms.

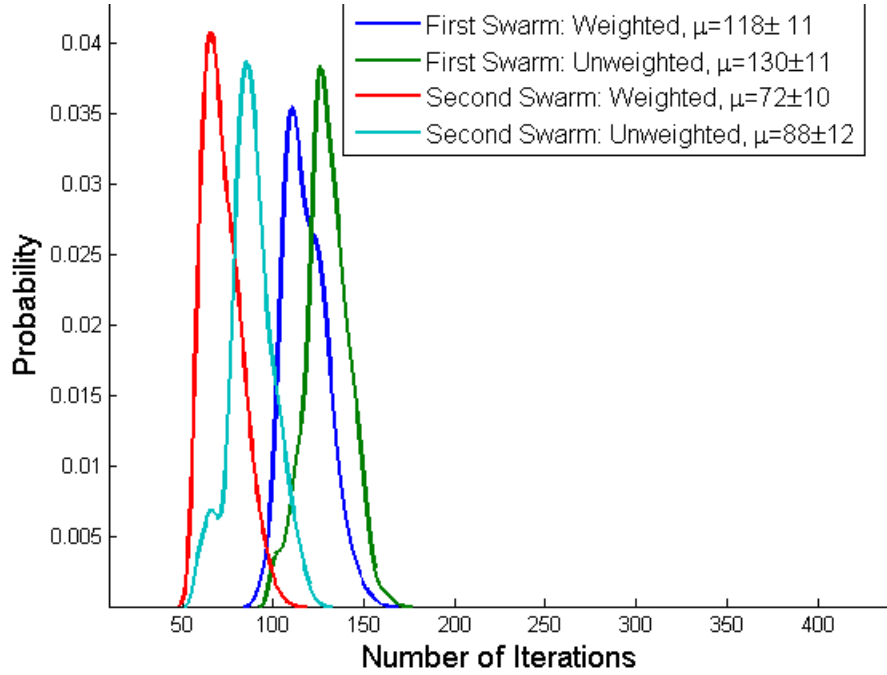


**Fig. 7.** Effect of the removal of different communication components.  $\mu$  denotes the average number of iterations required for each agent to reach the objective.

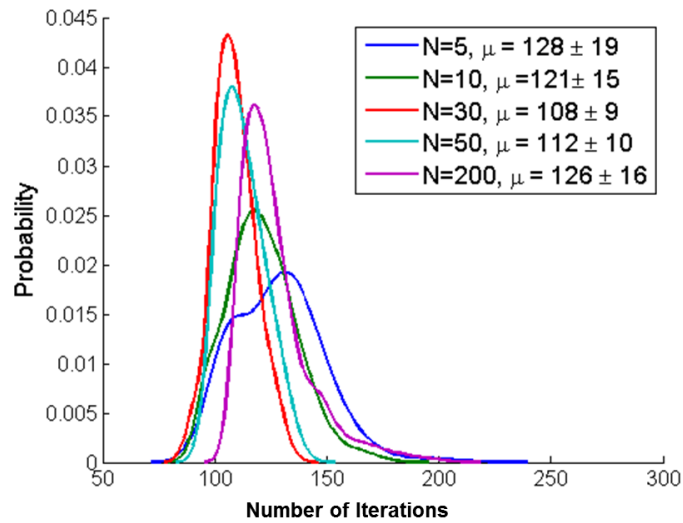
As can be seen, for both the first and second swarm, weighted communication leads to better performance than unweighted communication. Specifically, without using weights we see 10% and 22% increase in the number of iterations cells need to reach the food source for the first and second swarm, respectively. Thus, using the weighted version, which is also likely biologically correct, improves performance of such food search.

### Effect of swarm size

Another issue we tested is the impact of swarm size. As can be seen in Fig. 9), overall performance improves with the increase in swarm size indicating that communication helps cells reach their goal faster. However, depending on how we model the repulsion distance, at some point such increase can lead to crowding. When the agents are too close to each other, their movement is highly constrained and mostly dominated by the repulsion effect. Therefore it takes much longer for a larger swarm to reach the food source. From our simulations we find that swarm size 30 to 50 gives the optimal performance.



**Fig. 8.** Comparison of swarm performance using weighted and unweighted communication.  $\mu$  denotes the average number of iterations required for each agent to reach the objective.



**Fig. 9.** Distribution of path length over different swarm sizes.  $N$  denotes the number of agents in the swarm and  $\mu$  denotes the average number of iterations required for each agent to reach the objective.