FEATURE

Biomimicry of Bacterial Foraging

for Distributed Optimization and Control

By Kevin M. Passino

Natural selection tends to eliminate animals with poor "foraging strategies" (methods for locating, handling, and ingesting food) and favor the propagation of genes of those animals that have successful foraging strategies since they are more likely to enjoy reproductive success (they obtain enough food to enable them to reproduce). After many generations, poor foraging strategies are either eliminated or shaped into good ones (redesigned). Logically, such evolutionary principles have led scientists in the field of "foraging theory" to hypothesize that it is appropriate to model the activity of foraging as an optimization process: A foraging animal takes actions to maximize the energy obtained per unit time spent foraging, in the face of constraints presented by its own physiology (e.g., sensing and cognitive capabilities) and environment (e.g., density of prey, risks from predators, physical characteristics of the search area). Evolution has balanced these constraints and essentially "engineered" what is sometimes referred to as an "optimal foraging policy" (such terminology is especially justified in cases where the models and policies have been ecologically validated). Optimization models are also valid for "social foraging" where groups of animals cooperatively forage.

Here, we explain the biology and physics underlying the chemotactic (foraging) behavior of *E. coli* bacteria (yes, the ones that are living in your intestines). We explain a variety of bacterial swarming and social foraging behaviors and discuss the control system on the *E. coli* that dictates how foraging should proceed. Next, a computer program that emulates the distributed optimization process represented by the activity of social bacterial foraging is presented. To illustrate its operation, we apply it to a simple multiple-extremum function minimization problem and briefly discuss its relationship to some existing optimization algorithms. The article closes with a brief discussion on the potential uses of biomimicry of social foraging to develop adaptive controllers and cooperative control strategies for autonomous vehicles. For this, we provide some basic ideas and invite the reader to explore the concepts further. Hence, this article should be thought of as an introduction to some interesting biological phenomena that suggest new types of

The author (passino@ee.eng.ohio-state.edu) is with the Department of Electrical Engineering, The Ohio State University, 2015 Neil Avenue, Columbus, OH 43210-1272, U.S.A.

0272-1708/02/$17.00©2002IEEE

optimization methods; the relevance to control systems needs to be further investigated, and thorough comparisons to the vast literature on global optimization remain to be done.

Foraging

Elements of Foraging Theory

Foraging theory is based on the assumption that animals search for and obtain nutrients in a way that maximizes their energy intake E per unit time T spent foraging. Hence, they try to maximize a function like

$$\frac{E}{T}$$

(or they maximize their long-term average rate of energy intake). Maximization of such a function provides nutrient sources to survive and additional time for other important activities (e.g., fighting, fleeing, mating, reproducing, sleeping, or shelter building). Shelter-building and mate-finding activities sometimes bear similarities to foraging. Clearly, foraging is very different for different species. Herbivores generally find food easily but must eat a lot of it. Carnivores generally find it difficult to locate food but do not have to eat as much since their food is of high energy value. The "environment" establishes the pattern of nutrients that are available (e.g., via what other organisms are nutrients available, geological constraints such as rivers and mountains, and weather patterns), and it places constraints on obtaining that food (e.g., small portions of food may be separated by large distances). During foraging there can be risks due to predators, the prey may be mobile so it must be chased, and the physiological characteristics of the forager constrain its capabilities and ultimate success.

For many animals, nutrients are distributed in "patches" (e.g., a lake, a meadow, a bush with berries, a group of trees with fruit). Foraging involves finding such patches, deciding whether to enter a patch and search for food, and whether to continue searching for food in the current patch or to go find another patch that hopefully has a higher quality and quantity of nutrients than the current patch. Patches are generally encountered sequentially, and sometimes great effort and risk are needed to travel from one patch to another. Generally, if an animal encounters a nutrient-poor patch, but based on past experience it expects that there should be a better patch elsewhere, then it will consider risks and efforts to find another patch, and if it finds them acceptable, it will seek another patch. Also, if an animal has been in a patch for some time, it can begin to deplete its resources, so there should be an optimal time to leave the patch and venture out to try to find a richer one. It does not want to waste resources that are readily available, but it also does not want to waste time in the face of diminishing energy returns.

Optimal foraging theory formulates the foraging problem as an optimization problem and via computational or analytical methods can provide an optimal foraging "policy" that specifies how foraging decisions are made (traditionally, dynamic programming formulations have been used [1]). There are quantifications of what foraging decisions must be made, measures of currency (the opposite of cost), and constraints on the parameters of the optimization. For instance, researchers have studied how to maximize long-term average rate of energy intake where only certain decisions and constraints are allowed. Constraints due to incomplete information (e.g., due to limited sensing capabilities) and risks (e.g., due to predators) have been considered. Essentially, these optimization approaches seek to construct an optimal controller (policy) for making foraging decisions. Some biologists have questioned the validity of such an approach, arguing that no animal can make optimal decisions (see the references in [1]). However, the optimal foraging formulation is only meant to be a model that explains what optimal behavior would be like. In fact, some researchers have shown that foraging decision *heuristics* are used very effectively by animals to approximate optimal policies, given the physiological (and other) constraints that are imposed on the animal [1].

Search Strategies for Foraging

In one approach to the study of foraging search strategies [2], predation is broken into components that are similar for many animals. First, predators must search for and locate prey. Next, they pursue and attack the prey. Finally, they "handle" and ingest the prey. The importance of various components of foraging behavior depends on the relationship between the predator and the prey. If the prey is larger than the predator, then the pursuit, attack, and handling can be most important. The prey may be easy to find, but the prey's size gives it an advantage. If the prey is smaller than the predator, then generally the search component of foraging is most important. Small size can be an advantage for the prey. Since prey are often smaller than predators for many animals, they must be consumed often and in large numbers; this makes the search time limit other components of the predation cycle. In this article, we consider cases where the searching behavior is the dominant factor in foraging. This is the case for many birds, fish, lizards, and insects.

Some animals are "cruise" or "ambush" searchers. For the cruise approach to searching, the forager moves continuously through the environment, constantly searching for prey at the boundary of the volume being searched (tuna fish and hawks are cruise searchers). In ambush search, the forager (e.g., a rattlesnake) remains stationary and waits for prey to cross into its strike range. The search strategies of many species are actually between the cruise and ambush extremes. In particular, in "saltatory search" strategies, an animal will intermittently cruise and sit and wait, possibly changing direction at various times when it stops and possibly while it moves. To envision this strategy, consider Fig. 1, where distance traveled in searching is plotted versus time.

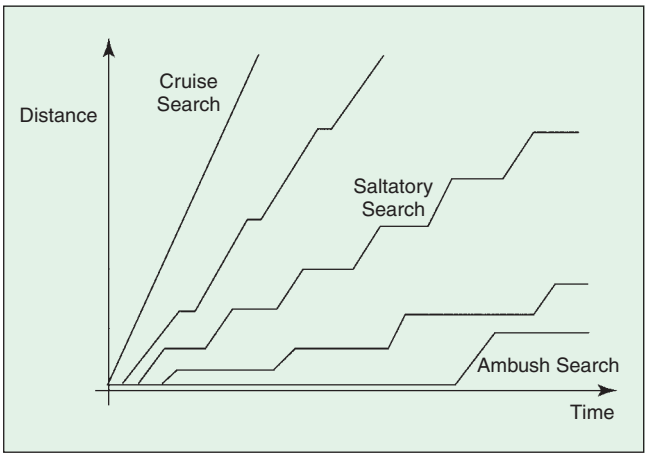

Figure 1. *Search strategies for foraging animals (figure adapted from [2]).*

In cruise search, distance increases at a constant rate dictated by how fast the animal moves in search. At the other extreme, the ambusher sits and waits for a long time and then makes a move to try to obtain a prey. In between are many possible saltatory search strategies that are based on an alternating sequence of cruising and waiting. Many animals' foraging strategies seem to lie somewhere on the continuum between ambush and cruise and hence are saltatory search strategies.

Saltatory search can be adjusted to suit the environment by changing rates of movement during cruises and the lengths of cruises and waits. For instance, some fish are known to pause more briefly and swim farther and faster during repositioning when searching for large prey compared to small ones. This is consistent with foraging theory in that the fish is willing to spend more effort to obtain more energy. In many species the pauses are used for orienting the animal toward prey; that is, they stop and change their direction based on their scan information. Then, for instance, if there is not an abundance of prey, in some species of fish fewer pursuits follow pauses. Also, in some fish the length of the stop and wait generally decreases when they are looking for large, easily located prey. Often, as the difficulty of the search increases, the pauses get longer. In environments where there are few prey, the fish persistently search.

Social and Intelligent Foraging

The foraging and search strategies discussed above were for individual animals. Clearly, however, there can be advantages to group (or social) foraging. Some method of communication is necessary for group foraging. In humans, this could include language. In other animals, it might be certain movements or noises or "trail-laying" mechanisms. The advantages of group foraging include:

- More animals searching for nutrients, so the likelihood of finding nutrients increases. When one animal finds some nutrients, it can tell others in the group where the nutrients are. Joining a group provides access to an "information center" to assist in survival.
- Increased capability to cope with larger prey. The group can "gang up" on a large prey and kill and ingest it.
- Protection from predators can be provided by members of the group (e.g., in some species the members in the middle of the group are protected by the ones at the edges).

Sometimes it is useful to think of a group (swarm) of animals as a single living creature, where via grouping and communication a "collective intelligence" emerges that actually results in more successful foraging for each individual in the group (and the gains can offset effects of food competition within groups; by working together there can be more food than if there is no cooperation).

For group foraging, you may think of how a pack of wolves hunts or a flock of birds, swarm of bees, colony of ants, or school of fish behave. Connections between optimization, engineering applications, and foraging behavior of colonies of ants have been studied. Bonabeau et al. [3] explain how colonies of ants can solve shortest-path problems, minimum spanning-tree problems, and traveling salesperson problems (all combinatorial optimization problems), among other engineering applications (the resulting computer algorithms are called, for instance, "ant colony optimization" algorithms). These ants use "indirect" communications called "stigmergy," where one ant can modify its environment and later another ant can change its behavior due to that modification. For instance, if an ant goes out foraging, it may search far and wide in a relatively random pattern; however, once it finds a food source, it goes back to the anthill laying a trail of "pheromone" (which can stay in place for up to a few months). Then, when other ants go out foraging, they tend to follow the pheromone trail and find food more easily. You can then think of the first ant as having "recruited" additional foragers and the trails as a type of memory for the whole ant colony (i.e., using communications and working together, they gain the important physiological capability of learning). Communications, memory, and learning result in more efficient foraging for the group. Other social insects use other communication methods. For instance, evidence suggests that after successful foraging, a bee will come back to the hive and communicate the quality and location of the food source via different types of "dances." For studies of group behavior of organisms via computer simulations, see [4]-[6].

Lower life forms can achieve higher forms of foraging intelligence by cooperating in groups (e.g., the ants "learn" where food is located). Lone individuals of higher life forms, however, are naturally endowed with more cognitive and hence foraging capabilities. For instance, in some species an individual forager can pay *attention* to the critical parts of its environment. It may be able to *learn* about the environment and characteristics of its prey (e.g., by developing and storing cognitive maps). It may be able to use this learned information to *plan* its foraging activities. Furthermore, if groups

of foragers can learn and plan their activities, it is possible for even greater success to be obtained. Indeed, it seems that humans often act as group foragers that can collectively learn and plan. Of course, humans are significantly different from many life forms since, rather than simply trying to cope with a given foraging environment, we focus on modifying our environment via farming to improve our foraging success.

In the next section, we will consider individual and group foraging for bacteria, organisms that are much simpler than ants or humans yet can still work together for the benefit of the group. You may think of these as tiny autonomous vehicles or as executing a simple yet effective optimization process during foraging, one that may be useful in the solution to control problems. Or you may simply be interested in the underlying control science (e.g., what type of control algorithm operates to implement their foraging strategy). After explaining how bacteria forage, we will model them via a computer simulation and then briefly discuss their relevance to adaptive control and cooperative control for autonomous vehicles.

Bacterial Foraging: *E. coli*

The *E. coli* bacterium has a plasma membrane, cell wall, and capsule that contains the cytoplasm and nucleoid. The pili (singular, pilus) are used for a type of gene transfer to other *E. coli* bacteria, and flagella (singular, flagellum) are used for locomotion. The cell is about 1 μm in diameter and 2 μm in length. The *E. coli* cell only weighs about 1 picogram and is about 70% water. *Salmonella typhimurium* is a similar type of bacterium.

The *E. coli* bacterium is probably the best understood microorganism. Its entire genome has been sequenced; it contains 4,639,221 of the A, C, G, and T "letters"—adenosine, cytosine, guanine, and thymine—arranged into a total of 4,288 genes. Mutations in *E. coli* occur at a rate of about 10^{-7} per gene, per generation, and can affect its physiological aspects (e.g., reproductive efficiency at different temperatures). *E. coli* bacteria occasionally engage in a type of "sex" called "conjugation" where small gene sequences are unidirectionally transferred from one bacterium to another via an extended pilus.

When *E. coli* grows, it gets longer, then divides in the middle into two "daughters." Given sufficient food and held at the temperature of the human gut (one place where they live) of 37 °C, *E. coli* can synthesize and replicate everything it needs to make a copy of itself in about 20 min; hence growth of a population of bacteria is exponential with a relatively short time to double. For instance, following [8], if at noon today you start with one cell and sufficient food, by noon tomorrow there will be $2^{72} = 4.7 \times 10^{21}$ cells, which is enough to pack a cube 17 m on one side (it should be clear that with enough food, at this reproduction rate, they could quickly cover the entire earth with a knee-deep layer!).

The *E. coli* bacterium has a control system (guidance system) that enables it to search for food and try to avoid noxious substances. For instance, it swims away from alkaline and acidic environments and toward more neutral ones. To explain the behavior of the *E. coli* bacterium, we will explain its actuator (the flagellum), "decision making," sensors, and closed-loop behavior (i.e., how it moves in various environments—its motile behavior). You will see that *E. coli* performs a type of saltatory search. This section is based on the work in [7]-[15].

Swimming and Tumbling via Flagella

Locomotion is achieved via a set of relatively rigid flagella that enable the bacterium to swim via each of them rotating in the same direction at about 100-200 revolutions per second (in control systems terms, we think of the flagellum as providing for actuation). Each flagellum is a left-handed helix configured so that as the base of the flagellum (i.e., where it is connected to the cell) rotates counterclockwise, as viewed from the free end of the flagellum looking toward the cell, it produces a force against the bacterium so it pushes the cell. You may think of each flagellum as a type of propeller. If a flagellum rotates clockwise, it will pull at the cell. From an engineering perspective, the rotating shaft at the base of the flagellum is quite an interesting contraption that seems to use what biologists call a "universal joint" (so the rigid flagellum can "point" in different directions relative to the cell). In addition, the mechanism that creates the rotational forces to spin the flagellum in either direction is described by biologists as a biological motor (a relatively rare contraption in biology, even though several types of bacteria use it) [8], [15]. The motor is quite efficient in that it makes a complete revolution using only about 1,000 protons, and thereby *E. coli* spends less than 1% of its energy budget for motility.

An *E. coli* bacterium can move in two different ways; it can run (swim for a period of time) or it can tumble, and it alternates between these two modes of operation its entire lifetime (i.e., it is rare that the flagella will stop rotating). First, we explain each of these two modes of operation. Following that, we will explain how it decides how long to swim before it tumbles.

If the flagella rotate clockwise, each flagellum pulls on the cell, and the net effect is that each flagellum operates relatively independently of the others, and so the bacterium "tumbles" about (i.e., the bacterium does not have a set direction of movement and there is little displacement). See Fig. 2(a). To tumble after a run, the cell slows down or stops first; since bacteria are so small, they experience almost no inertia, only viscosity, so when a bacterium stops swimming, it stops within the diameter of a proton [14]. Call the time interval during which a tumble occurs a "tumble interval." Under certain experimental conditions (an isotropic, homogeneous medium—one with no nutrient or noxious substance gradients), for a "wild-type" cell (one found in nature), the mean tumble interval is about 0.14 ± 0.19 s (mean ±

standard deviation), and it is exponentially distributed [8], [12]. After a tumble, the cell will generally be pointed in a random direction, but there is a slight bias toward being placed in the direction it was traveling before the tumble.

If the flagella move counterclockwise, their effects accumulate by forming a bundle (it is thought that the bundle is formed due to viscous drag of the medium), and hence they essentially make a composite propeller and push the bacterium so that it runs (swims) in one direction (see Fig. 2(a)). In one type of medium, on a run the bacteria swim at a rate of about 10-20 μm/s, or about ten of its body lengths per second (assuming the faster speed and an *E. coli* that is 2 μm long, a typical length), but in a rich medium they can swim even faster [16]. This is a relatively fast rate for a living organism to travel; consider how fast you could move through water if you could swim at ten of your body lengths per second (you would certainly win a gold medal in the Olympics!). Call the time interval during which a run occurs the "run interval." Under certain experimental conditions (an isotropic, homogeneous medium—the same as the one mentioned above), for a wild-type cell the mean run interval is about 0.86 ±1.18 s (and it is exponentially distributed) [8], [12]. Also, under these conditions, the mean speed is 14.2 ± 3.4 μm/s. Runs are not perfectly straight since the cell is subject to Brownian movement that causes it to wander off course by about 30° in 1 s in one type of medium, so this is how much it typically can deviate on a run. In a certain medium, after about 10 s it drifts off course more than 90° and hence essentially forgets the direction it was moving [8]. Finally, note that in many bacteria and media the motion of the flagella can induce other motions (e.g., rotating the bacteria about an axis).

Bacterial Motile Behavior: Climbing Nutrient Gradients

The motion patterns (called "taxes") that the bacteria will generate in the presence of chemical attractants and repellants are called chemotaxes. For *E. coli*, encounters with serine or aspartate result in attractant responses, whereas repellant responses result from the metal ions Ni and Co, changes in pH, amino acids like leucine, and organic acids like acetate. What is the resulting emergent pattern of behavior for a whole group of *E. coli* bacteria? Generally, as a group, they will try to find food and avoid harmful phenomena, and when viewed under a microscope, you will get a sense that a type of intelligent behavior has emerged, since they will seem to intentionally move as a group.

To explain how chemotaxis motions are generated, we must simply explain how the *E. coli* decides how long to run, since from the above discussion we know what happens during a tumble or run. First, note that if an *E. coli* is in some substance that is neutral in the sense that it does not have food or noxious substances, and if it is in this medium for a long time (e.g., more than 1 min), then the flagella will simultaneously alternate between moving clockwise and counterclockwise so that the bacterium will alternately tumble and run. This alternation between the two modes will move the bacterium, but in random directions, and this enables it to "search" for nutrients (see Fig. 2(b)). For instance, in the isotropic homogeneous environment described above, the bacterium alternately tumbles and runs with the mean tumble and run lengths given above and at the speed that was given. If the bacteria are placed in a homogeneous concentration of serine (i.e., one with a nutrient but no gradients), then a variety of changes occurs in the characteristics of their motile behavior. For instance, mean run length and mean speed increase and mean tumble time decreases. They do still produce, however, a basic type of searching behavior; even though the bacterium has some food, it persistently searches for more. Suppose that we call this its baseline behavior. As an example of tumbles and runs in the isotropic homogeneous medium described above, in one trial motility experiment lasting 29.5 s there were 26 runs, the maximum run length was 3.6 s, and the mean speed was about 21 μm/s [8], [12].

Next, suppose that the bacterium happens to encounter a nutrient gradient (e.g., serine), as shown in Fig. 2(c). The *change* in the concentration of the nutrient triggers a reaction such that the bacterium will spend more time swimming and less time tumbling. As long as it travels on a positive concentration gradient (i.e., so that it moves to-

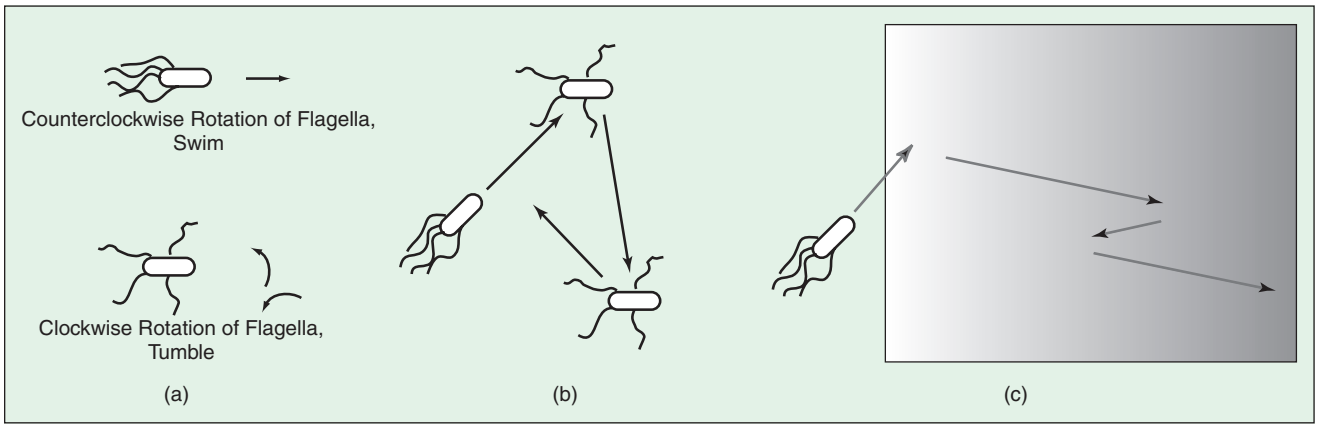

Figure 2. *Swimming, tumbling, and chemotactic behavior of E. coli.*

ward increasing nutrient concentrations) it will tend to lengthen the time it spends swimming (i.e., it runs farther), up to a point. The directions of movement are "biased" toward increasing nutrient gradients. The cell does not change its *direction* on a run due to changes in the gradient—the tumbles basically determine the direction of the run, aside from the Brownian influences mentioned above.

Optimal foraging theory formulates foraging as an optimization problem and via computational or analytical methods can provide an optimal foraging "policy" that specifies how foraging decisions are made.

On the other hand, typically if the bacterium happens to swim down a concentration gradient (or into a positive gradient of noxious substances), it will return to its baseline behavior so that essentially it tries to search for a way to climb back up the gradient (or down the noxious substance gradient). For instance, under certain conditions, for a wild-type cell swimming up serine gradients, the mean run length is 2.19 ±3.43 s, but if it swims down a serine gradient the mean run length is 1.40 ± 1.88 s [12]. Hence, when it moves up the gradient, it lengthens its runs. The mean run length for swimming down the gradient is the one that is expected considering that the bacteria are in this particular type of medium; they act basically the same as in a homogeneous medium, so that they are engaging their search/avoidance behavior (baseline behavior) to try to climb back up the gradient.

Finally, suppose that the bacterium reaches a region with constant nutrient concentration after having been on a positive gradient for some time. In this case, after a period of time (not immediately), the bacterium will return to the same proportion of swimming and tumbling as when it was in the neutral substance, so that it returns to its standard searching (baseline) behavior. It is never satisfied with the amount of surrounding food; it always seeks higher concentrations. Actually, under certain experimental conditions, the cell will compare the concentration observed over the past 1 s with the concentration observed over the 3 s before that, and it responds to the difference [8]. Hence it uses the past 4 s of nutrient concentration data to decide how long to run [13]. Considering the deviations in direction due to Brownian movement discussed above, the bacterium basically uses as much time as it can in making decisions about climbing gradients [14]. In effect, the run length results from how much climbing it has done recently. If it has made a lot of progress and hence has just had a long run, then even if for a little while it is observing a homogeneous medium (without gradients), it will take a longer run. After a certain time period, it will recover and return to its standard behavior in a homogeneous medium.

Basically, the bacterium is trying to swim from places with low concentrations of nutrients to places with high concentrations. An opposite type of behavior is used when it encounters noxious substances. If the various concentrations move with time, then the bacterium will tend to "chase" after the more favorable environments and run from harmful ones.

Underlying Sensing and Decision-Making Mechanisms

The sensors are the receptor proteins that are signaled directly by external substances (e.g., in the case for the pictured amino acids) or via the periplasmic substrate-binding proteins. The sensor is very sensitive, in some cases requiring less than ten molecules of attractant to trigger a reaction, and attractants can trigger a swimming reaction in less than 200 ms. You can then think of the bacterium as having a "high gain" with a small attractant detection threshold (detection of only a small number of molecules can trigger a doubling or tripling of the run length). On the other hand, the corresponding threshold for encountering a homogeneous medium after being in a nutrient-rich one is larger. Also, a type of time averaging is occurring in the sensing process. The receptor proteins then affect signaling molecules inside the bacterium. Also, there is in effect an "adding machine" and an ability to compare values to arrive at an overall decision about which mode the flagella should operate in; essentially, the different sensors add and subtract their effects, and the more active or numerous have a greater influence on the final decision. The sensory and decision-making system in *E. coli* is probably the best understood one in biology; here, we are ignoring the underlying chemistry needed for a full explanation.

It is interesting to note that the "decision-making" system in the *E. coli* bacterium must have some ability to sense a *derivative*, and hence it has a type of memory! At first glance, it may seem possible that the bacterium senses concentrations at both ends of the cell and finds a simple difference to recognize a concentration gradient (a spatial derivative); however, this is not the case. Experiments have shown that it performs a type of sampling, and roughly speaking, it remembers the concentration a moment ago, compares it with a current one, and makes decisions based on the difference (i.e., it computes something like an Euler approximation to a time derivative). Actually, in [17] the authors recently showed how internal bacterial

decision-making processes involve some type of integral feedback control mechanism.

In summary, we see that with memory, a type of addition mechanism, an ability to make comparisons, a few simple internal "control rules," and its chemical sensing and locomotion capabilities, the bacterium is able to achieve a complex type of search and avoidance behavior. Evolution has designed this control system. It is robust and clearly very successful at meeting its goals of survival when viewed from a population perspective.

Elimination and Dispersal Events

It is possible that the local environment where a population of bacteria live changes either gradually (e.g., via consumption of nutrients) or suddenly due to some other influence. Events can occur such that all the bacteria in a region are killed or a group is dispersed into a new part of the environment. For example, local significant increases in heat can kill a population of bacteria that are currently in a region with a high concentration of nutrients (you can think of heat as a type of noxious influence). Or it may be that water or some animal will move populations of bacteria from one place to another in the environment. Over long periods of time, such events have spread various types of bacteria into virtually every part of our environment—from our intestines to hot springs and underground environments.

What is the effect of elimination and dispersal events on chemotaxis? They have the effect of possibly destroying chemotactic progress, but they also have the effect of assisting in chemotaxis, since dispersal may place bacteria near good food sources. From a broad perspective, elimination and dispersal are parts of the population-level long-distance motile behavior.

Bacterial Motility and Swarming

Most bacteria are motile, and many types have analogous taxes capabilities to *E. coli* bacteria. The specific sensing, actuation, and decision-making mechanisms are different [10], [18]. Some bacteria can search for oxygen, and hence their motility behavior is based on aerotaxis, whereas others search for desirable temperatures resulting in thermotaxis. Actually, the *E. coli* is capable of thermotaxis in that it seeks warmer environments with a temperature range of 20 to 37 °C. Other bacteria search for or avoid light of certain wavelengths, and this is called phototaxis. Actually, the *E. coli* tries to avoid intense blue light, so it is also capable of phototaxis. Some bacteria swim along magnetic lines of force that enter the earth, so that in the northern hemisphere they swim toward the north magnetic pole and in the southern hemisphere they swim toward the south magnetic pole.

A particularly interesting group behavior has been demonstrated for several motile species of bacteria, including *E. coli* and *S. typhimurium*, where intricate stable spatiotemporal patterns (swarms) are formed in semisolid nutrient media [18]-[22]. (Microbiologists reserve the term "swarming" for other characteristics of groups of bacteria. Here, we abuse the terminology and favor using the terminology that is used for higher forms of animals such as bees.) When a group of *E. coli* cells is placed in the center of a semisolid agar with a single nutrient chemo-effector (sensor), they move out from the center in a traveling ring of cells by moving up the nutrient gradient created by consumption of the nutrient by the group. Moreover, if high levels of succinate are used as the nutrient, then the cells release the attractant aspartate so that they congregate into groups and hence move as concentric patterns of groups with high bacterial density. The spatial order results from outward movement of the ring and the local releases of the attractant; the cells provide an attraction signal to each other so they swarm together. Pattern formation can be suppressed by a background of aspartate (since it seems that this will in essence scramble the chemical signal by eliminating its directionality). The pattern seems to form based on the dominance of the two stimuli (cell-to-cell signaling and foraging).

The role of these patterns in natural environments is not fully understood; however, there is evidence that stress to the bacteria results in them releasing chemical signals toward which other bacteria are chemotactic. If enough stress is present, then a whole group can secrete the chemical signal, strengthening it, and hence an aggregate of the bacteria forms. It seems that this aggregate forms to protect the group from the stress (e.g., by effectively hiding many cells in the middle of the group). It also seems that the aggregates of the bacteria are not necessarily stationary; under certain conditions they can migrate, split, and fuse. This has led researchers to hypothesize that other communication methods are being employed that are not yet understood.

As another example, biofilms exist that can be composed of multiple types of bacteria (e.g., *E. coli*) that can coat various objects (e.g., roots of plants or medical implants). It seems that both motility and "quorum sensing" [18], [23] are involved in biofilm formation. A biofilm is a mechanism for keeping a bacterial species in a fixed location, avoiding overcrowding and avoiding nutrient limitation and toxin production by packing them in a low density in a polysaccharide matrix [23]. Secreted chemicals provide a mechanism for the cells to sense population density, but motility seems to assist in the early stages of biofilm formation. Researchers also think that chemotactic responses are used to drive cells to the outer edges of the biofilm where nutrient concentrations may be higher.

Finally, it should be noted that other types of bacteria exhibit swarm behaviors [23]. For instance, the luminous bacteria *Vibrio fischeri* will emit light when its population density reaches a certain threshold. *Streptomycete* colonies can grow a branching network of long fiberlike cells that can penetrate and degrade vegetation and then feed on the resulting decaying matter (in terms of combinatorial optimi-

zation, you may think of finding optimal trees or graphs). *Myxococcus xanthus*, one type of *myxobacterium* (slime bacteria), exhibits relatively exotic foraging and survival behaviors [18], [23]-[27]. For instance, while moving across solid surfaces (via gliding), they secrete slime trails and tend to follow slime trails of each other. Some *myxobacteria* prey on other bacteria (in what has been likened to the behavior of wolves). Mutants of *Myxococcus xanthus* can exhibit "social" and "adventurous" motility (essentially different group foraging behaviors). Under starvation conditions, they form aggregates called fruiting bodies where some cells die and others form spores. These fruiting bodies can then be transported via insects or wind into more favorable environments where the spores germinate and form a new colony. Simulations of *myxobacteria* based on a stochastic cellular automata approach are described in [28] and [29].

It is possible that the local environment where a population of bacteria live changes either gradually (e.g., via consumption of nutrients) or suddenly due to some other influence.

There is a great diversity of strategies for foraging and survival, even at the bacterial level! Clearly, we cannot outline them all here. Our objective is simply to explain how motile behaviors in both individual and groups of bacteria implement foraging and hence optimization.

E. coli Bacterial Swarm Foraging for Optimization

Suppose that we want to find the minimum of $J(\theta)$, $\theta \in \Re^p$, where we do not have measurements or an analytical description of the gradient $\nabla J(\theta)$. Here, we use ideas from bacterial foraging to solve this nongradient optimization problem. First, suppose that θ is the position of a bacterium and $J(\theta)$ represents the combined effects of attractants and repellants from the environment, with, for example, $J(\theta) < 0$, $J(\theta) = 0$, and $J(\theta) > 0$ representing that the bacterium at location θ is in nutrient-rich, neutral, and noxious environments, respectively. Basically, chemotaxis is a foraging behavior that implements a type of optimization where bacteria try to climb up the nutrient concentration (find lower and lower values of $J(\theta)$), avoid noxious substances, and search for ways out of neutral media (avoid being at positions θ where $J(\theta) \geq 0$). It implements a type of biased random walk.

Chemotaxis, Swarming, Reproduction, Elimination, and Dispersal

Define a chemotactic step to be a tumble followed by a tumble or a tumble followed by a run. Let j be the index for the chemotactic step. Let k be the index for the reproduction step. Let l be the index of the elimination-dispersal event. Let

$$P(j,k,l) = \{\theta^i(j,k,l) \mid i = 1,2,\ldots,S\}$$

represent the position of each member in the population of the S bacteria at the jth chemotactic step, kth reproduction step, and lth elimination-dispersal event. Here, let $J(i,j,k,l)$ denote the cost at the location of the ith bacterium $\theta^i(j,k,l) \in \Re^p$ (sometimes we drop the indices and refer to the ith bacterium position as θ^i). Note that we will interchangeably refer to J as being a "cost" (using terminology from optimization theory) and as being a nutrient surface (in reference to the biological connections). For actual bacterial populations, S can be very large (e.g., $S = 10^9$), but $p = 3$. In our computer simulations, we will use much smaller population sizes and will keep the population size fixed. We will allow $p > 3$ so we can apply the method to higher dimensional optimization problems.

Let N_c be the length of the lifetime of the bacteria as measured by the number of chemotactic steps they take during their life. Let $C(i) > 0, i = 1,2,\ldots,S$, denote a basic chemotactic step size that we will use to define the lengths of steps during runs. To represent a tumble, a unit length random direction, say $\phi(j)$, is generated; this will be used to define the direction of movement after a tumble. In particular, we let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\phi(j)$$

so that $C(i)$ is the size of the step taken in the random direction specified by the tumble. If at $\theta^i(j+1,k,l)$ the cost $J(i,j+1,k,l)$ is better (lower) than at $\theta^i(j,k,l)$, then another step of size $C(i)$ in this same direction will be taken, and again, if that step resulted in a position with a better cost value than at the previous step, another step is taken. This swim is continued as long as it continues to reduce the cost, but only up to a maximum number of steps, N_s. This represents that the cell will tend to keep moving if it is headed in the direction of increasingly favorable environments.

The above discussion was for the case where no cell-released attractants are used to signal other cells that they should swarm together. Here, we will also have cell-to-cell signaling via an attractant and will represent that with $J_{cc}^i(\theta, \theta^i(j,k,l))$, $i = 1,2,\ldots,S$, for the ith bacterium. Let

$$d_{\text{attract}} = 0.1$$

be the depth of the attractant released by the cell (a quantification of how much attractant is released) and

$$w_{\text{attract}} = 0.2$$

be a measure of the width of the attractant signal (a quantification of the diffusion rate of the chemical). The cell also repels a nearby cell in the sense that it consumes nearby nutrients and it is not physically possible to have two cells at the same location. To model this, we let

$$h_{\text{repellant}} = d_{\text{attract}}$$

be the height of the repellant effect (magnitude of its effect) and

$$w_{\text{repellant}} = 10$$

be a measure of the width of the repellant. The values for these parameters are simply chosen to illustrate general bacterial behaviors, not to represent a particular bacterial chemical signaling scheme. The particular values of the parameters were chosen with the nutrient profile that we will use later in Fig. 3 in mind. For instance, the depth and width of the attractant is small relative to the nutrient concentrations represented in Fig. 3. Let

$$\begin{aligned} J_{cc}(\theta, P(j,k,l)) &= \sum_{i=1}^{S} J_{cc}^{i}\left(\theta, \theta^{i}(j,k,l)\right) \\ &= \sum_{i=1}^{S}\left[-d_{\text{attract}} \exp\left(-w_{\text{attract}} \sum_{m=1}^{p} (\theta_m - \theta_m^i)^2\right)\right] \\ &\quad + \sum_{i=1}^{S}\left[h_{\text{repellant}} \exp\left(-w_{\text{repellant}} \sum_{m=1}^{p} (\theta_m - \theta_m^i)^2\right)\right] \end{aligned}$$

denote the combined cell-to-cell attraction and repelling effects, where $\theta = [\theta_1, \ldots, \theta_p]^T$ is a point on the optimization domain and θ_m^i is the mth component of the ith bacterium position θ^i (for convenience we omit some of the indices). An example for the case of $S = 2$ and the above parameter values is shown in Fig. 4. Here, note that the two sharp peaks represent the cell locations, and as you move radially away from the cell, the function decreases and then increases (to model the fact that cells far away will tend not to be attracted, whereas cells close by will tend to try to climb down the cell-to-cell nutrient gradient toward each other and hence try to swarm). Note that as each cell moves, so does its $J_{cc}^{i}(\theta, \theta^{i}(j,k,l))$ function, and this represents that it will release chemicals as it moves. Due to the movements of all the cells, the $J_{cc}(\theta, P(j,k,l))$ function is *time varying* in that if many cells come close together there will be a high amount of attractant and hence an increasing likelihood that other cells will move toward the group. This produces the swarming effect. When we want to study swarming, the ith bacterium, $i = 1,2,\ldots,S$, will hill-climb on

$$J(i,j,k,l) + J_{cc}(\theta, P)$$

(rather than the $J(i,j,k,l)$ defined above) so that the cells will try to find nutrients, avoid noxious substances, and at the same time try to move toward other cells, but not too close to them. The $J_{cc}(\theta, P)$ function dynamically deforms the search landscape as the cells move to represent the desire to swarm (i.e., we model mechanisms of swarming as a minimization process).

After N_c chemotactic steps, a reproduction step is taken. Let N_{re} be the number of reproduction steps to be taken. For convenience, we assume that S is a positive even integer. Let

$$S_r = \frac{S}{2} \tag{1}$$

be the number of population members who have had sufficient nutrients so that they will reproduce (split in two) with no mutations. For reproduction, the population is sorted in

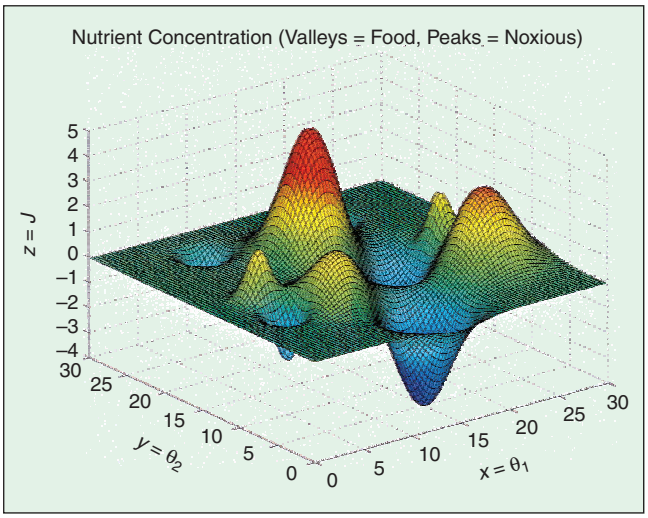

Figure 3. *Nutrient landscape.*

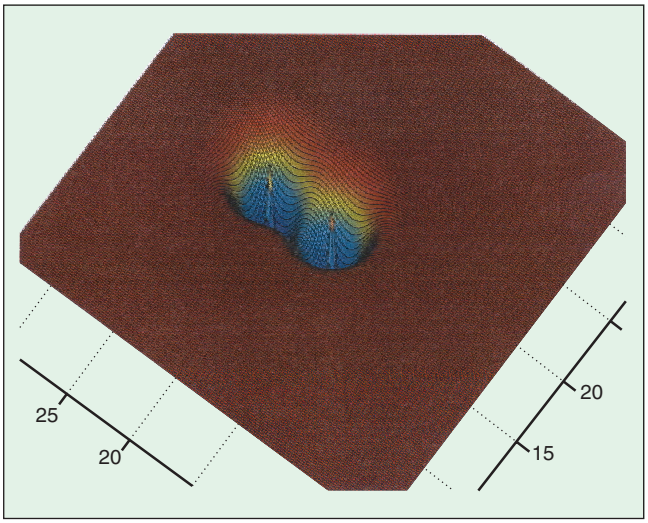

Figure 4. *Cell-to-cell chemical attractant model, $S = 2$.*

Bacterial Foraging Optimization Algorithm

For initialization, you must choose p, S, N_c, N_s, N_{re}, N_{ed}, p_{ed}, and the $C(i)$, $i = 1,2,\ldots,S$. If you use swarming, you will also have to pick the parameters of the cell-to-cell attractant functions; here we will use the parameters given above. Also, initial values for the θ^i, $i = 1,2,\ldots,S$, must be chosen. Choosing these to be in areas where an optimum value is likely to exist is a good choice. Alternatively, you may want to simply randomly distribute them across the domain of the optimization problem. The algorithm that models bacterial population chemotaxis, swarming, reproduction, elimination, and dispersal is given here (initially, $j = k = l = 0$). For the algorithm, note that updates to the θ^i automatically result in updates to P. Clearly, we could have added a more sophisticated termination test than simply specifying a maximum number of iterations.

1) Elimination-dispersal loop: $l = l + 1$
2) Reproduction loop: $k = k + 1$
3) Chemotaxis loop: $j = j + 1$
   a) For $i = 1,2,\ldots,S$, take a chemotactic step for bacterium i as follows.
   b) Compute $J(i,j,k,l)$. Let $J(i,j,k,l) = J(i,j,k,l) + J_{cc}(\theta^i(j,k,l), P(j,k,l))$ (i.e., add on the cell-to-cell attractant effect to the nutrient concentration).
   c) Let $J_{\text{last}} = J(i,j,k,l)$ to save this value since we may find a better cost via a run.
   d) Tumble: Generate a random vector $\Delta(i) \in \Re^p$ with each element $\Delta_m(i)$, $m = 1,2,\ldots,p$, a random number on [-1,1].
   e) Move: Let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}.$$

   This results in a step of size $C(i)$ in the direction of the tumble for bacterium i.
   f) Compute $J(i,j+1,k,l)$, and then let $J(i,j+1,k,l) = J(i,j+1,k,l) + J_{cc}(\theta^i(j+1,k,l), P(j+1,k,l))$.
   g) Swim (note that we use an approximation since we decide swimming behavior of each cell as if the bacteria numbered $\{1,2,\ldots,i\}$ have moved and $\{i+1, i+2,\ldots,S\}$ have not; this is much simpler to simulate than simultaneous decisions about swimming and tumbling by all bacteria at the same time):
      i) Let $m = 0$ (counter for swim length).
      ii) While $m < N_s$ (if have not climbed down too long)
      - Let $m = m + 1$.
      - If $J(i,j+1,k,l) < J_{\text{last}}$ (if doing better), let $J_{\text{last}} = J(i,j+1,k,l)$ and let

$$\theta^i(j+1,k,l) = \theta^i(j+1,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

      and use this $\theta^i(j+1,k,l)$ to compute the *new* $J(i,j+1,k,l)$ as we did in f).
      - Else, let $m = N_s$. This is the end of the while statement.
   h) Go to next bacterium $(i+1)$ if $i \neq S$ (i.e., go to b) to process the next bacterium).
4) If $j < N_c$, go to step 3. In this case, continue chemotaxis, since the life of the bacteria is not over.
5) Reproduction:
   a) For the given k and l, and for each $i = 1,2,\ldots,S$, let

$$J^i_{\text{health}} = \sum_{j=1}^{N_c+1} J(i,j,k,l)$$

   be the health of bacterium i (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters $C(i)$ in order of ascending cost J_{health} (higher cost means lower health).
   b) The S_r bacteria with the highest J_{health} values die and the other S_r bacteria with the best values split (and the copies that are made are placed at the same location as their parent).
6) If $k < N_{re}$, go to step 2. In this case, we have not reached the number of specified reproduction steps, so we start the next generation in the chemotactic loop.
7) Elimination-dispersal: For $i = 1,2,\ldots,S$, with probability p_{ed}, eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if you eliminate a bacterium, simply disperse one to a random location on the optimization domain.
8) If $l < N_{ed}$, then go to step 1; otherwise end.

order of ascending accumulated cost (higher accumulated cost represents that a bacterium did not get as many nutrients during its lifetime of foraging and hence is not as "healthy" and thus unlikely to reproduce); then the S_r least healthy bacteria die and the other S_r healthiest bacteria each split into two bacteria, which are placed at the same location. Other fractions or approaches could be used in place of (1); this method rewards bacteria that have encountered a lot of nutrients and allows us to keep a constant population size, which is convenient in coding the algorithm.

Let N_{ed} be the number of elimination-dispersal events, and for each elimination-dispersal event each bacterium in the population is subjected to elimination-dispersal with probability p_{ed}. We assume that the frequency of chemotactic steps is greater than the frequency of reproduction steps, which is in turn greater in frequency than

Our objective is to explain how motile behaviors in both individual and groups of bacteria implement foraging and hence optimization.

elimination-dispersal events (e.g., a bacterium will take many chemotactic steps before reproduction, and several generations may take place before an elimination-dispersal event).

Clearly, we are ignoring many characteristics of the actual biological optimization process in favor of simplicity and capturing the gross characteristics of chemotactic hill-climbing and swarming. For instance, we ignore many characteristics of the chemical medium and we assume that consumption does not affect the nutrient surface (e.g., while a bacterium is in a nutrient-rich environment, we do not increase the value of J near where it has consumed nutrients), where clearly in nature bacteria modify the nutrient concentrations via consumption. A tumble does not result in a perfectly random new direction for movement; however, here we assume that it does. Brownian effects buffet the cell so that after moving a small distance, it is within a pie-shaped region with its start point at the tip of the piece of pie. Basically, we assume that swims are straight, whereas in nature they are not. Tumble and run lengths are exponentially distributed random variables, not constant, as we assume. Run-length decisions are actually based on the past 4 s of concentrations, whereas here we assume that at each tumble, older information about nutrient concentrations is lost. Although naturally asynchronous, we force synchronicity by requiring, for instance, chemotactic steps of different bacteria to occur at the same time, all bacteria to reproduce at the same time instant, and all bacteria that are subjected to elimination and dispersal to do so at the same time. We assume a constant population size, even if there are many nutrients and generations. We assume that the cells respond to nutrients in the environment in the same way that they respond to ones released by other cells for the purpose of signaling the desire to swarm (a more biologically accurate model of the swarming behavior of certain bacteria is given in [22]). Clearly, other choices for the criterion of which bacteria should split could be used (e.g., based only on the concentration at the end of a cell's lifetime, or on the quantity of noxious substances that were encountered). We are also ignoring conjugation and other evolutionary characteristics. For instance, we assume that $C(i)$, N_s, and N_c remain the same for each generation. In nature it seems likely that these parameters could evolve for different environments to maximize population growth rates. The intent here was simply to come up with a simple model that only represents certain aspects of the foraging behavior of bacteria.

Guidelines for Algorithm Parameter Choices

The bacterial foraging optimization algorithm requires specification of a variety of parameters. First, you can pick the size of the population, S. Clearly, increasing the size of S can significantly increase the computational complexity of the algorithm. However, for larger values of S, if you choose to randomly distribute the initial population, it is more likely that you will start at least some bacteria near an optimum point, and over time, it is then more likely that many bacterium will be in that region, due to either chemotaxis or reproduction.

What should the values of the $C(i)$, $i=1,2,\ldots,S$, be? You can choose a biologically motivated value; however, such values may not be the best for an engineering application. If the $C(i)$ values are too large, then if the optimum value lies in a valley with steep edges, the search will tend to jump out of the valley, or it may simply miss possible local minima by swimming through them without stopping. On the other hand, if the $C(i)$ values are too small, convergence can be slow, but if the search finds a local minimum it will typically not deviate too far from it. You should think of the $C(i)$ as a type of "step size" for the optimization algorithm.

The size of the values of the parameters that define the cell-to-cell attractant functions J_{cc}^{i} will define the characteristics of swarming. If the attractant width is high and very deep, the cells will have a strong tendency to swarm (they may even avoid going after nutrients and favor swarming). On the other hand, if the attractant width is small and the depth shallow, there will be little tendency to swarm and each cell will search on its own. Social versus independent foraging is then dictated by the balance between the strengths of the cell-to-cell attractant signals and nutrient concentrations.

Next, large values for N_c result in many chemotactic steps, and hopefully more optimization progress, but of course more computational complexity. If the size of N_c is chosen to be too short, the algorithm will generally rely more on luck and reproduction, and in some cases, it could more easily get trapped in a local minimum (premature convergence). You should think of N_s as creating a bias in the random walk (which would not occur if $N_s=0$), with large values tending to bias the walk more in the direction of climbing down the hill.

If N_c is large enough, the value of N_{re} affects how the algorithm ignores bad regions and focuses on good ones, since bacteria in relatively nutrient-poor regions die (this models, with a fixed population size, the characteristic where bacteria will tend to reproduce at higher rates in favorable environments). If N_{re} is too small, the algorithm may converge prematurely; however, larger values of N_{re} clearly increase computational complexity.

A low value for N_{ed} dictates that the algorithm will not rely on random elimination-dispersal events to try to find favorable regions. A high value increases computational complexity but allows the bacteria to look in more regions to find good nutrient concentrations. Clearly, if p_{ed} is large, the algorithm can degrade to random exhaustive search. If, however, it is chosen appropriately, it can help the algorithm jump out of local optima and into a global optimum.

Connections with other Nongradient Global Optimization Methods

The reader already familiar with genetic algorithms (or evolutionary programming) may recognize the *algorithmic analogies* between the fitness function and the nutrient concentration function (both a type of landscape), selection and bacterial reproduction (bacteria in the most favorable environments gain a selective advantage for reproduction), crossover and bacterial splitting (the children are at the same concentration, whereas with crossover they generally end up in a region around their parents on the fitness landscape), and mutation and elimination and dispersal. However, the algorithms are certainly not equivalent, and neither is a special case of the other. Each has its own distinguishing features. The fitness function and nutrient concentration functions are not the same (one represents likelihood of survival for given phenotypic characteristics, whereas the other represents nutrient/noxious substance concentrations, or perhaps other environmental influences such as heat or light). Crossover represents mating and resulting differences in offspring, something we ignore in the bacterial foraging algorithm (we could, however, have made less than perfect copies of the bacteria to represent their splitting). Moreover, mutation represents gene mutation and the resulting phenotypical changes, not physical dispersal in a geographical area. From one perspective, note that all the typical features of genetic algorithms could augment the bacterial foraging algorithm by representing evolutionary characteristics of a forager in its environment. From another perspective, foraging algorithms can be integrated into evolutionary algorithms and thereby model some key survival activities that occur during the lifetime of the population that is evolving (i.e., foraging success can help define fitness, mating characteristics, etc.). For the bacteria studied here, foraging happens to entail hill-climbing via a type of biased random walk, and hence the foraging algorithm can be viewed as a method to integrate a type of approximate stochastic gradient search (where only an approximation to the gradient is used, not analytical gradient information) into evolutionary algorithms. Of course, standard gradient methods, quasi-Newton methods, etc., depend on the use of an explicit analytical representation of the gradient, something that is not needed by a foraging or genetic algorithm. Lack of dependence on analytical gradient information can be viewed as an advantage (fewer assumptions) or a disadvantage (e.g., since if gradient information is available then the foraging or genetic algorithm may not exploit it properly).

There are in fact many approaches to "global optimization" when there is no explicit gradient information available; however, it is beyond the scope of this article to evaluate the relative merits of foraging algorithms to the vast array of such methods that have been studied for many years. To start such a study, it makes sense to begin by considering the theoretical convergence guarantees for certain types of evolutionary algorithms, stochastic approximation methods, and pattern search methods (e.g., see [30] for

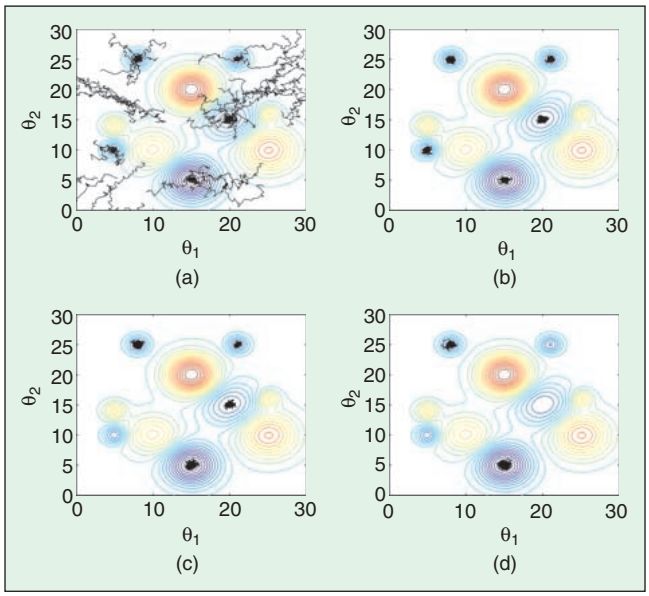

Figure 5. *Bacterial motion trajectories, generations 1-4, on contour plots. (a) Generation 1, (b) generation 2, (c) generation 3, and (d) generation 4.*

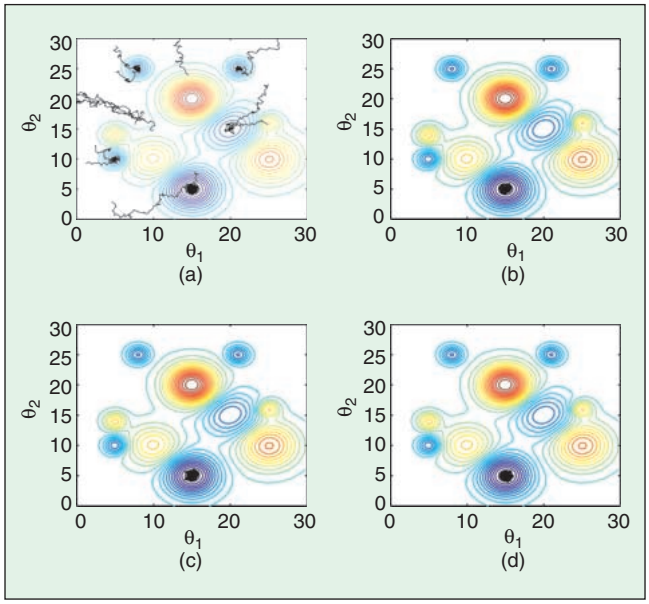

Figure 6. *Bacterial motion trajectories, generations 1-4, on contour plots, after an elimination-dispersal event. (a) Generation 1, (b) generation 2, (c) generation 3, and (d) generation 4.*

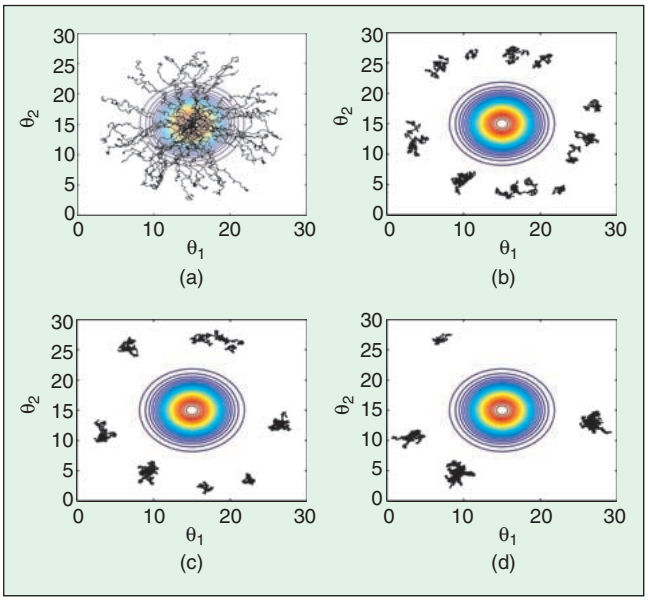

Figure 7. *Swarm behavior of E. coli on a test function. (a) Generation 1, (b) generation 2, (c) generation 3, and (d) generation 4.*

work along these lines) and then proceed to consider foraging algorithms in this context. Finally, note that evolution designed the bacteria to forage in a time-varying and noisy environment (i.e., to achieve robust optimization for noisy time-varying cost functions). Can we exploit the characteristics of such an optimization approach for engineering problems?

Example: Function Optimization via Bacterial Foraging

As a simple illustrative example, we use the algorithm to try to find the minimum of the function in Fig. 3 (note that the point $[15,5]^T$ is the global minimum point). To gain more insight into the operation of the algorithm, it is recommended that you run the algorithm yourself (see the sidebar on p. 61 for a Web address from which you can obtain the code).

Nutrient Hill-Climbing: No Swarming

According to the above guidelines, choose $S = 50$, $N_c = 100$, $N_s = 4$ (a biologically motivated choice), $N_{re} = 4$, $N_{ed} = 2$, $p_{ed} = 0.25$, and the $C(i) = 0.1$, $i = 1,2,\ldots,S$. The bacteria are initially spread randomly over the optimization domain. The results of the simulation are illustrated by motion trajectories of the bacteria on the contour plot of Fig. 3, as shown in Fig. 5. In the first generation, starting from their random initial positions, searching occurred in many parts of the optimization domain, and you can see the chemotactic motions of the bacteria as the black trajectories where the peaks are avoided and the valleys are pursued. Reproduction picks the 25 healthiest bacteria and copies them, and then as shown in Fig. 5 in generation 2, all the chemotactic steps are in five local minima. This again happens in going to generations 3 and 4, but bacteria die in some of the local minima (due essentially to our requirement that the population size stay constant), so that in generation 3 there are four groups of bacteria in four local minima, whereas in generation 4 there are two groups in two local minima.

Next, with the above choice of parameters there is an elimination-dispersal event, and we get the *next* four generations shown in Fig. 6. Notice that elimination and dispersal shift the locations of several of the bacteria, and thereby the algorithm explores other regions of the optimization domain. However, qualitatively we find a similar pattern to the previous four generations where chemotaxis and reproduction work together to find the global minimum; this time, however, due to the large number of bacteria that were placed near the global minimum, after one reproduction step, all the bacteria are close to it (and remain this way). In this way, the bacterial population has found the global minimum.

Swarming Effects

Here we use the parameters defined earlier to define the cell-to-cell attraction function. Also, we choose $S = 50$, $N_c = 100$, $N_s = 4$, $N_{re} = 4$, $N_{ed} = 1$, $p_{ed} = 0.25$, and the $C(i) = 0.1$, $i = 1,2,\ldots,S$. We will first consider swarming effects on the nutrient concentration function with the contour map shown on Fig. 7, which has a zero value at $[15,15]^T$ and decreases to successively more negative values as you move away from that point; hence, the cells should tend to swim away from the peak. We will initialize the bacterial positions by placing all the cells at the peak $[15,15]^T$. Using these conditions, we get the result in Fig. 7. Notice that in the first generation, the cells swim radially outward, and then in the second and third generations, swarms are formed in a concentric pattern of groups. Notice also that with our simple method of simulating health of the bacteria and reproduction, some of the swarms are destroyed by the fourth generation. We omit additional simulations that show the behavior of the swarm on the surface in Fig. 3 since qualitatively the behavior is as one would expect from the above simulations. The interested reader can obtain the code mentioned above and further study the behavior of the algorithm. Note, however, that simulation of swarming mechanisms is somewhat delicate and stretches the simple inaccurate model that we are using for the bacteria.

Biomimicry of Foraging for Control: Challenges and Directions

It is certainly impossible to explore all the potential uses of foraging algorithms in this single article, even if we only focused on the field of control. To conclude this article, however, we next point to some ideas on the potential uses of foraging algorithms in control to give the reader a flavor of their potential applicability. Even from this brief discussion, it should seem at least plausible that there are applications of the methods to optimization, optimal control, model predictive control, adaptive estimation and control, and computer-aided control system design.

Foraging for Adaptive Control

Optimization methods such as gradient algorithms and least squares are used to implement estimation methods, which are used to estimate models or controllers in adaptive control. Hence, we can use foraging algorithms as the basis for adaptive control. To see this, suppose that we consider indirect adaptive control where we seek to learn a plant model during the operation of a system. Suppose that we view learning as foraging for good model information (i.e., information that is truthful and useful for meeting goals). Suppose that we use an "identifier model," which is a parametrized model of the plant, and think of the foraging algorithm as searching in the parameter space of that model. By moving parameters, it searches for regions in the parameter space that correspond to finding nutrients; hence, we need to define what we mean by nutrients. Suppose that we use the standard definition of identifier error as the error between the model output and the plant output, but squared and summed over the past several samples to define J. Moreover, we will think of having multiple identifier models as in [31] and social foraging (i.e., multiple models being tuned simultaneously, with foragers possibly sharing information to try to improve foraging success) as shown in Fig. 8. There, S foraging algorithms search in the parameter space for locations that correspond to getting low identification errors between the model and plant. Then, according to the sum of the squared identifier errors, we simply choose at each time instant the model that is the best and use it in a standard certainty equivalence approach to specify a controller. Such a strategy is similar to indirect genetic adaptive control strategies [32], and direct adaptive control strategies can be developed using the approach in [33], [34] (combined direct/indirect strategies in [33] and [34] are adaptive model predictive control approaches).

Consider a "surge tank" liquid level control problem where we use a discretized version of the model

$$\frac{dh(t)}{dt} = \frac{-\bar{d}\sqrt{2gh(t)}}{A(h(t))} + \frac{\bar{c}}{A(h(t))}u(t)$$

where $h(t)$ is liquid level (saturated so that it cannot go below zero), $u(t)$ is the input (also saturated), $\bar{c}$ and $\bar{d}$ are the constants, and $A(h(t)) = |\bar{a}h(t) + \bar{b}|$, with $\bar{a}$ and $\bar{b}$ also constants, is the *unknown* tank cross-sectional area (creating the need for estimation of plant dynamics for compensation). We let each identifier model be an an affine mapping (linear plus constant) to match plant nonlinearities. The identifier model parameters represent the forager's position. We choose to have $S = 10$ foragers. The cost function for each forager, which defines the nutrient profile, is defined to be the sum of squares of $N = 100$ past identifier error ($e(t)$) values for each identifier model. For parameter adjustment, we use a foraging algorithm that is based on *E. coli* chemotactic behavior as defined earlier but no forager-forager communication. We interleave movements in the parameter space (i.e., estimator updates) with chemotactic steps as you would for many adaptive control algorithms for discrete time systems. The tracking performance and the best cost and index of the best forager are shown in Fig. 9. Note that it takes time before the controller adapts, but that as the cost index decreases over time (representing that at least the best forager has found nutrients), the tracking error decreases. Also, notice that there is a lot of switching between which forager is the best, and hence which controller is chosen to select the control input. You can obtain the MATLAB code for this simulation at http://eewww.eng.ohio-state.edu/~passino.

While the above approach may hold some promise, it also raises the following questions: How well can the method perform relative to standard adaptive control ap-

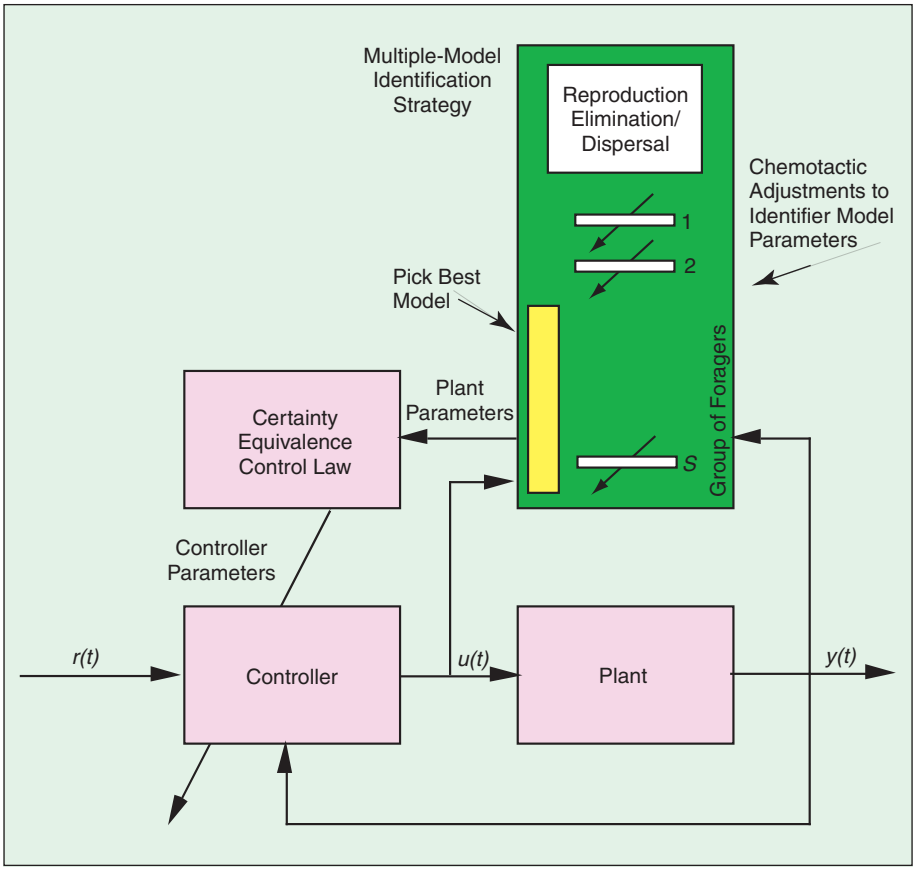

Figure 8. *Swarm foraging in adaptive control (r(t): reference or desired plant output).*

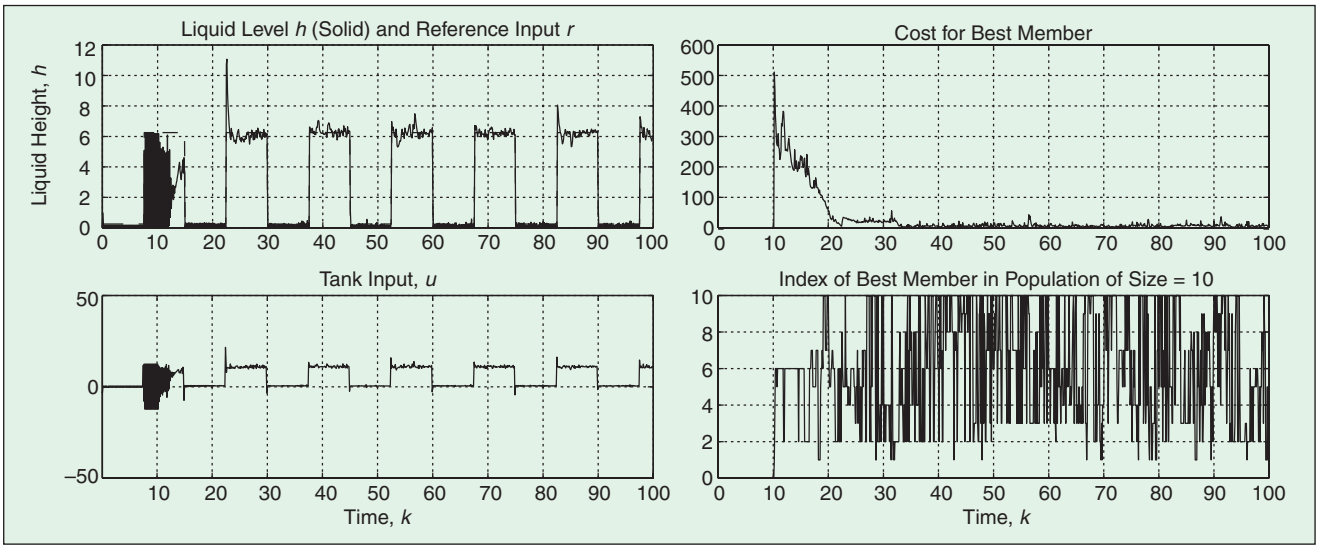

Figure 9. *Closed-loop response of adaptive controller.*

proaches? For example, could we get better tracking performance using other methods (or by tuning this one better)? Which approach is easier to develop so that it can achieve acceptable performance? Which is more computationally complex and more difficult to implement? Can we mathematically prove that the tracking error will converge? What class of problems might be particularly appropriate for this type of approach? While we did not illustrate it above, compared to standard multiple-model adaptive control approaches, the social foraging approach suggests ideas for sharing information between different parameter update mechanisms for different identifier models (as do genetic adaptive control methods). How fruitful will such ideas be for applications? It is beyond the scope of this article to address these questions.

Autonomous Vehicle Guidance: What Can Nature Teach Us?

The "artificial potential field method" in autonomous vehicle guidance bears some similarities to bacterial foraging algorithms. There are clear analogies between foraging and cooperative control of groups of uninhabited autonomous vehicles (UAVs) that are used in military (or commercial) applications: i) Animals, organisms = UAVs, ii) social foragers = group of cooperating UAVs that can communicate with each other, iii) prey, nutrients = targets, iv) predators, noxious substances = threats, and v) environment = battlefield. Are these analogies useful? Biomimicry of social foraging of ants [3] has provided some concepts for UAV problems [35], [36]. What future research directions does this article suggest along these lines? Some are as follows: First, the utility of further development of the social foraging metaphor needs more study. For example, do bacteria employ cooperative control methods that would also be effective strategies for military applications? Do other social foraging animals? For example, ones that operate with a similar "physiology" as a UAV and in a similar environment to the one the animal is found in and hence optimized for via evolution? We need to understand the social foraging strategies of several organisms, try to understand why these are good for the environment they live in, and then seek to develop analogous implementable strategies for groups of UAVs. Second, it would be interesting to characterize the physiological and environmental aspects that drove evolution to "design" a specific foraging strategy and optimize its operation. This would help us understand how vehicular constraints and tactical situations affect the design and operation of the cooperative strategy. Perhaps it would also suggest ideas for how to optimize the design of cooperative control strategies for UAVs.

The warfare strategies (foraging) employed by many animals have been optimized via evolution for millions of years; it seems logical that they may suggest some novel approaches to the design of guidance strategies for UAVs. Can we find engineering applications that call for functionalities that are similar to what evolution has fine-tuned for an organism, then exploit the bio-design for solution to practical technological problems? Or, put another way, can we find biological systems that can solve technological problems that are beyond our current engineering capabilities?

Acknowledgments

The author would like to thank Prof. H. Berg of Harvard University for his helpful comments on the description of the motile behavior of *E. coli* bacteria. The author would also like to thank the reviewers and editors for helpful suggestions on how to improve the article. The author would like to thank DAGSI/AFRL for financial support.

References

[1] D. Stephens and J. Krebs, *Foraging Theory*. Princeton, NJ: Princeton Univ. Press, 1986.

[2] W. O'Brien, H. Browman, and B. Evans, "Search strategies of foraging animals," *Amer. Scientist*, vol. 78, pp. 152-160, Mar./Apr. 1990.

[3] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems.* New York: Oxford Univ. Press, 1999.

[4] C. Adami, *Introduction to Artificial Life*. New York: Springer-Verlag, 1998.

[5] M. Resnick, *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*. Cambridge, MA: MIT Press, 1994.

[6] S. Levy, *Artificial Life: A Report from the Frontier where Computers Meet Biology.* New York: Vintage Books, 1992.

[7] T. Audesirk and G. Audesirk, *Biology: Life on Earth*, 5th ed. Englewood Cliffs, NJ: Prentice Hall, 1999.

[8] H. Berg, "Motile behavior of bacteria," *Phys. Today*, pp. 24-29, Jan. 2000.

[9] M. Madigan, J. Martinko, and J. Parker, *Biology of Microorganisms*, 8th ed. Englewood Cliffs, NJ: Prentice Hall, 1997.

[10] F. Neidhardt, J. Ingraham, and M. Schaechter, *Physiology of the Bacterial Cell: A Molecular Approach*. Sunderland, MA: Sinauer, 1990.

[11] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. Watson, *Molecular Biology of the Cell,* 2nd ed. New York: Garland Publishing, 1989.

[12] H. Berg and D. Brown, "Chemotaxis in escherichia coli analysed by three-dimensional tracking," *Nature*, vol. 239, pp. 500-504, Oct. 1972.

[13] J. Segall, S. Block, and H. Berg, "Temporal comparisons in bacterial chemotaxis," *Proc. Nat. Acad. Sci.*, vol. 83, pp. 8987-8991, Dec. 1986.

[14] H. Berg, *Random Walks in Biology*. Princeton, NJ: Princeton Univ. Press, 1993.

[15] D. DeRosier, "The turn of the screw: The bacterial flagellar motor," *Cell*, vol. 93, pp. 17-20, 1998.

[16] G. Lowe, M. Meister, and H. Berg, "Rapid rotation of flagellar bundles in swimming bacteria," *Nature*, vol. 325, pp. 637-640, Oct. 1987.

[17] T.-M. Yi, Y. Huang, M. Simon, and J. Doyle, "Robust perfect adaptation in bacterial chemotaxis through integral feedback control," *PNAS*, vol. 97, pp. 4649-4653, April 25, 2000.

[18] J. Armitage, "Bacterial tactic responses," *Adv. Microbial Phys.*, vol. 41, pp. 229-290, 1999.

[19] E. Budrene and H. Berg, "Dynamics of formation of symmetrical patterns by chemotactic bacteria," *Nature*, vol. 376, pp. 49-53, 1995.

[20] Y. Blat and M. Eisenbach, "Tar-dependent and -independent pattern formation by salmonella typhimurium," *J. Bacteriology*, vol. 177, pp. 1683-1691, Apr. 1995.

[21] E. Budrene and H. Berg, "Complex patterns formed by motile cells of escherichia coli," *Nature*, vol. 349, pp. 630-633, Feb. 1991.

[22] D. Woodward, R. Tyson, M. Myerscough, J. Murray, E. Budrene, and H. Berg, "Spatio-temporal patterns generated by salmonella typhimurium," *Biophysi. J.*, vol. 68, pp. 2181-2189, 1995.

[23] R. Losick and D. Kaiser, "Why and how bacteria communicate," *Sci. Amer.*, vol. 276, no. 2, pp. 68-73, 1997.

[24] M. McBride, P. Hartzell, and D. Zusman, "Motility and tactic behavior of myxococcus xanthus," in *Myxobacteria II*, M. Dworkin and D. Kaiser, Eds. Washington, DC: American Society for Microbiology, 1993, pp. 285-305.

[25] L. Shimkets and M. Dworkin, "Myxobacterial multicellularity," in *Bacteria as Multicellular Organisms*, J. Shapiro and M. Dworkin, Eds., New York: Oxford Univ. Press, 1997, pp. 220-244.

[26] H. Reichenbach, "Biology of the myxobacteria: Ecology and taxonomy," in *Myxobacteria II*, M. Dworkin and D. Kaiser, Eds. Washington, DC: American Society for Microbiology, 1993, pp. 13-62.

[27] J. Shapiro, "Bacteria as multicellular organisms," *Sci. Amer.*, vol. 258, pp. 62-69, 1988.

[28] A. Stevens, "Simulations of the gliding behavior and aggregation of myxobacteria," in *Biological Motion* (Lecture Notes in Biomathematics, vol. 89), W. Alt and G. Hoffmann, Eds. Berlin: Springer-Verlag, 1990, pp. 548-555.

[29] A. Stevens, "A stochastic cellular automaton, modeling gliding and aggregation of myxobacteria," *SIAM J. Appli. Math.*, vol. 61, no. 1, pp. 172-182, 2000.

[30] J. Spall, S. Hill, and D. Stark, "Some theoretical comparisons of stochastic optimization approaches," in *Proc. American Control Conf.*, Chicago, IL, June 2000, pp. 1904-1908.

[31] R. Murray-Smith and T.A. Johansen, Eds., *Multiple Model Approaches to Nonlinear Modeling and Control*. London, UK: Taylor and Francis, 1996.

[32] K. Kristinsson and G. Dumont, "System identification and control using genetic algorithms," *IEEE Trans. Syst., Man, Cybernet.*, vol. 22, no. 5, pp. 1033-1046, 1992.

[33] L.L. Porter and K. Passino, "Genetic adaptive and supervisory control," *Int. J. Intell. Contr. Syst.*, vol. 12, no. 1, pp. 1-41, 1998.

[34] W. Lennon and K. Passino, "Genetic adaptive identification and control," *Eng. Applicat. Artif. Intell.*, vol. 12, pp. 185-200, Apr. 1999.

[35] S. Brueckner and H. Parunak, "Multiple pheromones for improved guidance," in *Proceedings AEC*, 2000 [Online]. Available: www.erim.org/cec/projects/adaptiv/

[36] M. Krieger, J.-B. Billeter, and L. Keller, "Ant-like task allocation and recruitment in cooperative robots," *Nature*, vol. 406, pp. 992-995, Aug. 2000.

Kevin M. Passino received his Ph.D. in electrical engineering from the University of Notre Dame in 1989. He has worked on control systems research at Magnavox Electronic Systems Co. and McDonnell Aircraft Co. He is currently a Professor in the Department of Electrical Engineering at The Ohio State University and Director of the Collaborative Center of Control Science that is supported by AFRL/VA and AFOSR. He was the Program Chair of the 2001 IEEE Conference on Decision and Control. He has served as the Vice President of Technical Activities of the IEEE Control Systems Society (CSS); was an elected member of the IEEE Control Systems Society Board of Governors; was Chair for the IEEE CSS Technical Committee on Intelligent Control; has been an Associate Editor for *IEEE Transactions on Automatic Control* and *IEEE Transactions on Fuzzy Systems*; has served as a Guest Editor for *IEEE Control Systems Magazine* and *IEEE Expert Magazine*; and was on the Editorial Board of the *International Journal for Engineering Applications of Artificial Intelligence.* He was the Program Chair for the 8th IEEE International Symposium on Intelligent Control, 1993, and the General Chair for the 11th IEEE International Symposium on Intelligent Control. He is coeditor or coauthor of five books. He is a Senior Member of the IEEE.