Bacterial Foraging Optimization

Kevin M. Passino, The Ohio State University, USA

ABSTRACT

The bacterial foraging optimization (BFO) algorithm mimics how bacteria forage over a landscape of nutrients to perform parallel nongradient optimization. In this article, the author provides a tutorial on BFO, including an overview of the biology of bacterial foraging and the pseudo-code that models this process. The algorithms features are briefly compared to those in genetic algorithms, other bio-inspired methods, and nongradient optimization. The applications and future directions of BFO are also presented.

Keywords: *Bacteria Foraging, Bacteria Foraging Optimization, Bacteria Motility, Control, Distributed Control, Optimization*

1 INTRODUCTION. BACTERIAL FORAGING: *E. COLI*

The *E. coli* bacterium has a plasma membrane, cell wall, and capsule that contain, for instance, 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 $1\mu m$ in diameter, and $2\mu m$ in length. The *E. coli* cell only weighs about 1 picogram, and is composed of 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. 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 deg. 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" the population size. For instance, following (Berg, 2000), 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 meters 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 that enables it to search for food and try to avoid noxious substances (the resulting motions are called "taxes"). For instance, it swims away from alkaline and acidic environments, and towards more neutral ones. To explain the motile behavior of *E. coli* bacteria, we will explain its actuator (the flagella), "decision-making," sensors, and closed-loop behavior (i.e., how it moves in various environments—its "motile

DOI: 10.4018/jsir.2010010101

behavior"). You will see that *E. coli* perform a type of "saltatory search."

1.1 Swimming and Tumbling via Flagella

Locomotion is achieved via a set of relatively rigid flagella that enable it 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 flagella 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 towards 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, then 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 being a biological "motor" (a relatively rare contraption in biology even though several types of bacteria use it). The motor is quite efficient in that it rotates a complete revolution using only about 1000 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 independent 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). 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 that when a bacterium stops swimming, it stops within the diameter of a proton. 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 sec.(mean $\pm$ standard deviation, and it is exponentially distributed) (Berg, 1972, 2000). After a tumble, the cell will generally be pointed in a random direction, but there is a slight bias toward being placed in a 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 the 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. On a run, bacteria swim at a rate of about $10-20$ μ meters/sec., or about 10 body lengths per second (assuming the faster speed and an *E. coli* that is 2 μ meters long, a typical length), but in a rich medium they can swim even faster (Lowe, Meister, & Berg, 1987). 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 10 of your body lengths per second. 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 sec.(and it is exponentially distributed) (Berg, 1972, 2000). Also, under these conditions, the mean speed is 14.2 ± 3.4 μm / sec. Runs are not perfectly straight since the cell is subject to Brownian movement that causes it to wander off course by about 30 deg. in 1 sec. in one type of medium, so this is how much it typically

can deviate on a run. In a certain medium, after about 10 sec. it drifts off course more than 90 deg. and hence, essentially forgets the direction it was moving (Berg, 1972). Finally, note that in many bacteria, the motion of the flagella can induce other motions, e.g., rotating the bacteria about an axis.

1.2 Bacterial Motile Behavior: Climbing Nutrient Gradients

The motion patterns (called "taxes") that the bacteria will generate in the presence of chemical attractants and repellents are called "chemotaxes." For *E. coli*, encounters with serine or aspartate result in attractant responses, while repellent 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 simply must 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 period of time (e.g., more than one minute), 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. For instance, in the isotropic homogeneous environment described above, the bacteria alternately tumble and run 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 occur in the characteristics of their motile behavior. For instance, mean run length and mean speed increase and mean tumble time decreases. They do, however, still produce a basic type of searching behavior; even though it has some food, it persistently searches for more. As an example of tumbles and runs in the isotropic homogeneous medium described above, in one trial motility experiment lasting 29.5 sec., there were 26 runs, the maximum run length was 3.6 sec., and the mean speed was about 21 μm / $sec.$ (Berg, 1972, 2000).

Next, suppose that the bacterium happens to encounter a nutrient gradient (e.g., serine). 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 towards increasing nutrient concentrations) it will tend to lengthen the time it spends swimming (i.e., it runs farther). The directions of movement are "biased" towards 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.

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 sec., but if it swims down a serine gradient, mean run length is 1.10 ± 1.88 sec. (Berg, 2000). 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 to try to climb back up the gradient.

Finally, suppose that the concentration of the nutrient is constant for the region it is in, after it has 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 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 sec. with the concentration observed over the 3 sec. before that and it responds to the difference (Berg, 1972). Hence, it uses the past 4 sec. of nutrient concentration data to decide how long to run (Segall, Block, & Berg, 1986). 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 (Berg, 1993). In effect, the run length results from how much climbing it has done recently. If it has made lots 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 bacteria will try to "chase" after the more favorable environments and run from harmful ones. Clearly, nutrient and noxious substance diffusion and motion will affect the motion patterns of a group of bacteria in complex ways.

1.3 Underlying Sensing and Decision-Making Mechanisms

The sensors are the receptor proteins, which 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 10 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, there is a type of time-averaging that 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 and 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. Even though the sensory and decision-making system in *E. coli* is probably the best understood one in biology, we are ignoring the underlying chemistry that is 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 Yi, Huang, Simon, and Doyle (2000) the authors

show 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 searching 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.

1.4 Elimination and Dispersal Events

It is possible that the local environment where a population of bacteria lives changes either gradually (e.g., via consumption of nutrients) or suddenly due to some other influence. There can be events 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, and so on.

What is the effect of elimination and dispersal events on chemotaxis? It has the effect of possibly destroying chemotactic progress, but it also has the effect of assisting in chemotaxis since dispersal may place bacteria near good food sources. From a broad perspective, elimination and dispersal is part of the population-level motile behavior.

1.5 Evolution of Bacteria

Mutations in *E. coli* occur at a rate of about 10^{-7} per gene, per generation. In addition to mutations that 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. It seems that these gene sequences apparently carry good fitness characteristics in terms of reproductive capability, so conjugation is sometimes thought of as a transmittal of "fertility." To achieve conjugation, a pilus extends to make contact with another bacterium, and the gene sequence transfers through the pilus.

It is important to note that there are some very basic differences in evolution for higher organisms and bacteria. While conjugation apparently spreads "good" gene sequences, the "homogenizing effect" on gene frequency from conjugation is relatively small compared to how sex works in other organisms. This is partly since conjugation is relatively rare, and partly since the rate of reproduction is relatively high, on the order of hours depending on environmental conditions. Due to these characteristics, population genetics for *E. coli* may be dominated by selection sweeps triggered by the acquisition, via sex, of an adaptive allele.

1.6 Taxes in Other Swimming Bacteria

While 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 (Armitage, 1999; Neidhardt, Ingraham, & Schaechter, 1990). For instance, while the proton driven motor on *E. coli* rotates at a few hundred revolutions per second, Na^+-driven motors on some bacteria rotate at speeds up to 1000 revolutions per second, and on some species, the motor can turn in either direction or stop. Different types of bacteria can sense different phenomena and have different underlying

decision-making, so they may search for and try to avoid different phenomena. Some bacteria can sense their own metabolic state and only respond to compounds currently required for growth and their pattern of responses may change based on their environment. Studies of the mechanisms for decision and control in various bacteria do, however, indicate that they have common features and hence, some have suggested that there was a single early evolutionary event that resulted in the swimming capability of bacteria. Swimming generally moves a bacterium to a more favorable environment for growth, or it maintains it in its current position, and hence, it gives the bacteria a survival advantage. Some scientists have suggested that the shapes of motile bacteria developed to allow efficient swimming. Some bacteria even change their shape to reduce the adverse effects of moving through more viscous media. Even though there can be significant differences between species, all swimming bacteria seem to have similar swimming patterns, where there is an alternation between smooth swimming and a change in direction (i.e., a type of saltatory search). Next, several examples of other types of sensing and taxes in swimming bacteria are provided.

Some bacteria can search for oxygen, and hence their motility behavior is based on "aerotaxis," while 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 deg. to 37 deg. C. Other bacteria, such as *Thiospirillum jenense*, 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 when in the northern hemisphere, they swim towards the north magnetic pole, and in the southern hemisphere, they swim towards the south magnetic pole. (This is due to the presence of a small amount of magnetic material *in the cell* that essentially acts as a compass to passively reorient the cell.)

There are square-shaped bacteria that are propelled either forward or backward via flagella, and when multiple such bacteria naturally collide, their flagella can become "clumped," and this seems to be responsible for their tumbling. Hence, their motility behavior is characterized by forward movement, followed by either forward or backward movement, and an intermittent change in direction via tumbling (Alam, Claviez, Oesterhelt, & Kessel, 1984). *Vibrio alginolyticus* move differently when free-living versus living on a surface. Free-living *Vibrio alginolyticus* swims using a Na^+-driven motor on its flagella but when it is on the surface of a liquid, it senses the increased viscosity via the flagellar motor and then synthesizes many proton-driven flagella, which then allow the cell to move over surfaces (Armitage, 1999). The cells move as groups ("rafts"), since this is thought to help overcome viscous drag and surface tension. In other bacteria, flagella can be synthesized and discarded as they are needed.

1.7 Other Group Phenomena in Bacteria

A particularly interesting group behavior has been demonstrated for several motile species of bacteria, including *E. coli* and *S. typhimurium*, where intricate stable spatio-temporal patterns (swarms)[1] are formed in semi-solid nutrient media (Armitage, 1999; Budrene & Berg, 1995a, 1995b; Woodward, Tyson, Myerscough, Murray, Budrene, & Berg, 1995; Blat & Eisenbach, 1995) . When a group of *E. coli* cells is placed in the center of a semi-solid 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 the nutrient called 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. (Note that many cells in those groups permanently lose motility.) 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 two stimuli (cell-cell signaling and foraging).

The role of these patterns in natural environments is not understood; however, there is evidence that stress to the bacteria results in them releasing chemical signals that other bacteria are chemotactic towards. If enough stress is present, then a whole group can secrete the chemical signal strengthening the total signal, 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 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 there may be other communication methods being employed that are not yet understood.

As another example, there are "biofilms" 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" 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 at a low density in a "polysaccharide matrix." Secreted chemicals provide a mechanism for the cells to sense population density, but motility seems to assist in the early stages of biofilm formation. It is also thought that chemotactic responses are used to drive cells to the outer edges of the biofilm, where nutrient concentrations may be higher.

In a variety of bacteria, including *E. coli*, complex patterns result primarily not from motility, but from reproduction (Shapiro, 1997). In some bacteria, it seems that there is a type of signaling that occurs and results in the formation of regular patterns as the culture of bacteria grows. Formation of such patterns is sometimes thought of as a type of multicellular "morphogenesis." For example, the formation of the "fruiting bodies" by *Myxococcus xanthus* can be viewed as a type of morphogenesis, but one that seems to be primarily based on motility and cell deaths rather than reproduction (Shimkets & Dworkin, 1997).

Other types of bacteria exhibit group behaviors (Losick & Kaiser, 1997). For instance, there are luminous bacteria that will emit no light until the population reaches a certain density. For instance, the bacteria *Vibrio fischeri* lives in the ocean at low concentrations and its secreted "autoinducer" chemical signal is quite dilute. However, the squid *Euprymna scolopes* selects these bacteria to grow in its light organ. When a sufficiently large population is cultivated in its light organ, the autoinducer chemical signals given off by each bacterium effectively add to result in a high concentration of this chemical and, when it reaches a certain threshold, each cell will switch on its luminescence property so that as a group they emit a visible light (Losick & Kaiser, 1997). The squid, which is a nocturnal forager, benefits since the light camouflages it from predators below, since its light resembles moonlight and hence, effectively eliminates its shadow. The bacteria benefit by getting nourishment and shelter. The bacteria and squid are in a symbiont relationship (i.e., they live together to benefit each other).

Also, the soil-dwelling *streptomycete* colonies can grow a branching network of long fiber-like cells that can penetrate and degrade vegetation and then feed on the resulting decaying matter. (In terms of combinatorial optimization, you may think of finding optimal trees or graphs.) Under starvation conditions, they can cooperate to produce spores on a structure called an "aerial mycelium" that may be carried away.

As another example, in *Proteus mirabilis* the rod-shaped cells exist as "swimmers" that are driven by fewer than 10 flagella when they are in liquid media and they have chemotactic responses analogous to those of *E. coli*. If,

however, these swimmers are placed on a solid surface, the swimmer cell "differentiates" (changes) into a "swarmer cell" that is an elongated rod (of roughly the same diameter) with more than 10,000 flagella. On solid surfaces, the cells aggregate and exhibit swarm behavior in foraging via group chemotaxis. If they are then placed back in a liquid medium, there is a process of "consolidation" where swarmer cells split into swimmer cells. Moreover, when swarming they exhibit the "Dienes phenomenon," where swarms of the same type of bacteria try to avoid each other. (The mechanisms of this apparent territorial behavior are not well-understood.)

2 *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 repellents 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 $J(\theta)$) and avoid noxious substances and search for ways out of neutral media (avoid being at positions θ where $J(\theta) \geq 0$).

2.1 An Optimization Model for *E. coli* Bacterial Foraging

To define our optimization model of *E. coli* bacterial foraging, we need to define a population (set) of bacteria, and then model how they execute chemotaxis, swarming, reproduction, and elimination/dispersal. After doing this, we will highlight the limitations (inaccuracies) in our model.

2.1.1 Population and Chemotaxis

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

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

represent the positions of each member in the population of the S bacteria at the j^{th} chemotactic step, k^{th} reproduction step, and ℓ^{th} elimination-dispersal event. Here, let $J(i,j,k,\ell)$ denote the cost at the location of the i^{th} bacterium $\theta^i(j,k,\ell) \in \Re^p$ (sometimes we drop the indices and refer to the i^{th} 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 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

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

so that $C(i)$ is the size of the step taken in the random direction specified by the tumble. If at $\theta^i(j+1,k,\ell)$ the cost $J(i,j+1,k,\ell)$ is better (lower) than at $\theta^i(j,k,\ell)$, 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.

2.1.2 Swarming Mechanisms

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 $J_{cc}^i(\theta,\theta^i(j,k,\ell))$, $i=1,2,\ldots,S$, for the i^{th} bacterium. Let

$d_{attract} = 0.1$

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

$w_{attract} = 0.2$

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

$h_{repellent} = d_{attract}$

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

$w_{repellent} = 10$

be a measure of the width of the repellent. 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 are chosen with the nutrient profile in mind. For instance, the depth and width of the attractant is small relative to the nutrient concentrations represented in the cost function. Let

```
J_cc(,P(j,k,)) = _i=1^SJ_cc^i(,^i(j,k,))
= _i=1^S[-d_attract(-w_attract_m=1^p(_m-
m^i)^2)]
+ _i=1^S [ h_repellent (-w_repellent
m=1^p(_m-_m^i)^2)]
```

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 m^{th} component of the i^{th} bacterium position θ^i (for convenience, we omit some of the indices). Note that as each cell moves, so does its $J_{cc}^i(\theta,\theta^i(j,k,\ell))$ function, and this represents that it will release chemicals as it moves. Due to the movements of all the cells, the $J_{cc}(\theta,P(j,k,\ell))$ 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 towards the group. This produces the swarming effect. When we want to study swarming, the i^{th} bacterium, $i=1,2,\ldots,S$, will hill-climb on

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

(rather than the $J(i,j,k,\ell)$ defined above) so that the cells will try to find nutrients, avoid noxious substances, and at the same time try to move towards 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).

2.1.3 Reproduction and Elimination/Dispersal

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=S2 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 order of ascending accumulated cost (higher accumulated cost represents that it 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 Equation (2.1.3); 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 such 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 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).

2.1.4 Foraging Model Limitations

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 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 of 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 sec. 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 Woodward et al., 1995.) 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.

2.2 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. 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 below (initially, $j = k = \ell = 0$). 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.

Elimination-dispersal loop: $\ell = \ell + 1$ **Reproduction loop:** $k = k + 1$ **Chemotaxis loop:** $j = j + 1$ For $i = 1,2,\ldots,S$, take a chemotactic step for bacterium i as follows. Compute $J(i,j,k,\ell)$. Let

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

(i.e., add on the cell-to-cell attractant effect to the nutrient concentration). Let $J_{last} = J(i,j,k,\ell)$ to save this value, since we may find a better cost via a run. 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 $[-1,1]$. Move: let

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

This results in a step of size $C(i)$ in the direction of the tumble for bacterium i. Compute $J(i,j+1,k,\ell)$, and then let

$$J(i,j+1,k,\ell) = J(i,j+1,k,\ell) + J_{cc}(\theta^i(j+1,k,\ell), P(j+1,k,\ell))$$

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): Let $m = 0$ (counter for swim length). While $m < N_s$ (if have not climbed down too long)

Let $m = m + 1$. If $J(i,j+1,k,\ell) < J_{last}$ (if doing better), let $J_{last} = J(i,j+1,k,\ell)$ and let

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

and use this $\theta^i(j+1,k,\ell)$ to compute the *new* $J(i,j+1,k,\ell)$ as we did in (f) above. Else, let $m = N_s$. This is the end of the while statement.

Go to next bacterium $(i+1)$ if $i \neq S$ (i.e., go to (b) above to process the next bacterium).

If $j < N_c$, go to step 3. In this case, continue chemotaxis, since the life of the bacteria is not over.

Reproduction: For the given k and ℓ, and for each $i = 1,2,\ldots,S$, let

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

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). 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 mother).

If $k < N_{re}$, go to step 2. In this case, we have not reached the number of specified repro-

duction steps, so we start the next generation in the chemotactic loop.

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.

If $\ell < N_{ed}$, then go to step 1; otherwise end.

Matlab code for this can be obtained at: http://www.ece.osu.edu/passino/

2.3 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 bacterium 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, it 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, then convergence can be slow, but if it 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} 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.

2.4 Relations to Other Nongradient Optimization Methods

There are *algorithmic analogies* between the genetic algorithm and the above optimization model for foraging. There are 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 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 for other foragers predator/prey characteristics). 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 an environment.

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 their 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).

You probably also recognize some similarities between certain features of the foraging algorithm and simultaneous perturbation stochastic approximation algorithm (SPSA) (Spall, Hill, & Stark, 2000). What are they? What are the relationships to other nongradient methods (pattern search methods)? 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 Spall et al., 2000, for work along these lines), and then proceed to consider foraging algorithms in this context. It also seems useful to consider how well the foraging algorithms will perform for time-varying nutrient landscapes, which occurs in the underlying biological problem and many engineering problems.

3 CONCLUSION. BFO APPLICATIONS AND DIRECTIONS

Since its initial development and introduction and popularization via the book (Passino, 2002, 2004), BFO has been used in a number of applications:

Optimization over continuous surfaces (cost functions) (Passino, 2002); Algorithmic extensions: Hybrid approach (Kim, Abrahamb, & Choa, 2007); Comparative analysis with other methods, in particular particle swarm optimization (PSO) (Biswas, Dasgupta, Das, & Abrahamb, 2008); Adaptive control: Introduction of the idea and application to liquid level control (Passino, 2002); proportional-integral-derivative (PID) controller tuning (Kim & Cho, 2005); Harmonic estimation (Mishra, 2005); Active power filter for load optimization (Mishra & Bhende, 2007); Transmission loss reduction: Application in power systems (Tripathy, Mishra, Lai, & Zhang, 2006); Optimizing power loss and voltage stability limits: Application in power systems (Tripathy & Mishra, 2007). These are the most popular applications as measured by the number of citations to them on Google. Applications to fuzzy controller construction/tuning, neural network training, job-shop scheduling, electromagnetics, stock market predication, optimal power flow, motor control, temperature control, system identification, and others have also been studied but apparently have not received as much attention to-date. The reader is encouraged to search the internet since more such applications seem likely in the coming years.

Finally, additional applications and studies of the method still holds potential: Optimization: There is still a wide variety of domains in which BFO could be useful for. For instance, it would be useful to study its use in energy efficiency optimization for buildings and distributed energy generation. Comparative analysis: There is a need for a comprehensive Monte-Carlo-based evaluation of its performance relative to other nongradient methods (e.g., the genetic algorithm). This should include evaluation for a large data base of cost functions. Adaptive control: The method holds potential to solve more challenging adaptive control problems, yet it needs to be compared to "genetic adaptive control methods" (see Passino, 2004, or the publications at: http://www.ece.osu.edu/passino/).

REFERENCES

Alam, M., Claviez, M., Oesterhelt, D., & Kessel, M. (1984). Flagella and motility behavior of square bacteria. *The EMBO Journal*, *13*(12), 2899–2903.

Armitage, J. P. (1999). Bacterial Tactic Responses. *Advances in Microbial Physiology*, *41*, 229–290. doi:10.1016/S0065-2911(08)60168-X

Berg, H. C. (1993). *Random Walks in Biology. New Expanded Edition*. Princeton, NJ: Princeton University Press.

Berg, H. C. (2000). Motile Behavior of Bacteria. *Physics Today*, 24–29. doi:10.1063/1.882934

Berg, H. C., & Brown, D. A. (1972). Chemotaxis in *Escherichia coli* Analysed by Three-Dimensional Tracking. *Nature*, *239*, 500–504. doi:10.1038/239500a0

Biswas, A., Dasgupta, S., Das, S., & Abraham, A. (2008). Synergy of PSO and Bacterial Foraging Optimization--A Comparative Study on Numerical Benchmarks. *Advances in Soft Computing . Innovations in Hybrid Intelligent Systems*, *44*, 255–263. doi:10.1007/978-3-540-74972-1_34

Blat, Y., & Eisenbach, M. (1995). Tar-Dependent and Independent Pattern Formation by *Salmonella typhimurium*. *Journal of Bacteriology*, *177*, 1683–1691.

Budrene, E. O., & Berg, H. C. (1995a). Complex Patterns Formed by Motile Cells of *Escherichia coli*. *Nature*, *349*, 630–633. doi:10.1038/349630a0

Budrene, E. O., & Berg, H. C. (1995b). Dynamics of formation of symmetrical patterns by chemotactic bacteria. *Nature*, *376*, 49–53. doi:10.1038/376049a0

Kim, D. H., Abrahamb, A., & Choa, J. H. (2007). A hybrid genetic algorithm and bacterial foraging approach for global optimization. *Information Sciences*, *177*(18), 3918–3937. doi:10.1016/j.ins.2007.04.002

Kim, D. H., & Cho, J. H. (2005). Adaptive Tuning of PID Controller for Multivariable System Using Bacterial Foraging Based Optimization. In *Advances in Web Intelligence* (LNCS 3528, pp. 231-235).

Losick, R., & Kaiser, D. (1997). Why and How Bacteria Communicate. *Scientific American, 276*(2), 68–73. doi:10.1038/scientificamerican0297-68

Lowe, G., Meister, M., & Berg, H. (1987). Rapid Rotation of Flagellar Bundles in Swimming Bacteria. *Nature, 325*, 637–640. doi:10.1038/325637a0

Mishra, S. (2005). A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. *IEEE Transactions on Evolutionary Computation, 9*(1), 61–73. doi:10.1109/TEVC.2004.840144

Mishra, S., & Bhende, C. N. (2007). Bacterial Foraging Technique-Based Optimized Active Power Filter for Load Compensation. *IEEE Transactions on Power Delivery, 22*(1), 457–465. doi:10.1109/TPWRD.2006.876651

Neidhardt, F. C., Ingraham, J. L., & Schaechter, M. (1990). *Physiology of the Bacterial Cell: A Molecular Approach*. Sunderland, MA: Sinauer Associates.

Passino, K. M. (2002). Biomimicry of Bacterial Foraging for Distributed Optimization and Control. *IEEE Control Systems Magazine, 22*(3), 52–67. doi:10.1109/MCS.2002.1004010

Passino, K. M. (2004). *Biomimicry for Optimization, Control, and Automation*. London: Springer-Verlag.

Segall, J. E., Block, S. M., & Berg, H. C. (1986). Temporal Comparisons in Bacterial Chemotaxis. *Proceedings of the National Academy of Sciences of the United States of America, 83*, 8987–8991. doi:10.1073/pnas.83.23.8987

Shapiro, J. A. (1997). Multicellularity: The Rule, Not the Exception. In J. A. Shapiro & M. Dworkin (Eds.), *Bacteria as Multicellular Organisms* (pp. 14-49). New York: Oxford University Press.

Shimkets, L. J., & Dworkin, M. (1997). Myxobacterial Multicellularity. In J. A. Shapiro & M. Dworkin (Eds.), *Bacteria as Multicellular Organisms* (pp. 220-244). New York: Oxford University Press.

Spall, J. C., Hill, S. D., & Stark, D. R. (2000). Some Theoretical Comparisons of Stochastic Optimization Approaches. In *Proceedings of the American Control Conference*, Chicago.

Tripathy, M., & Mishra, S. (2007). Bacteria Foraging-Based Solution to Optimize Both Real Power Loss and Voltage Stability Limit. *IEEE Transactions on Power Systems, 22*(1), 240–248. doi:10.1109/TPWRS.2006.887968

Tripathy, M., Mishra, S., Lai, L. L., & Zhang, Q. P. (2006). Transmission Loss Reduction Based on FACTS and Bacteria Foraging Algorithm. In *Parallel Problem Solving from Nature - PPSN IX* (LNCS 4193, pp. 222-231).

Woodward, D. E., Tyson, R., Myerscough, M. R., Murray, J. D., Budrene, E. O., & Berg, H. C. (1995). Spatio-temporal patterns generated by Salmonella typhimurium. *Biophysical Journal, 68*, 2181–2189. doi:10.1016/S0006-3495(95)80400-5

Yi, T.-M., Huang, Y., Simon, M. I., & Doyle, J. C. (2000). Robust Perfect Adaptation in Bacterial Chemotaxis Through Integral Feedback Control. *Proceedings of the National Academy of Sciences of the United States of America, 97*(9), 4649–4653. doi:10.1073/pnas.97.9.4649

ENDNOTE

1 Actually, 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.

Kevin M. Passino received his PhD in electrical engineering from the University of Notre Dame in 1989. He is currently a professor of electrical and computer engineering at The Ohio State University. He leads the education component of the IEEE/UN Humanitarian Technology Challenge. 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 the program chair of the 2001 IEEE Conf. on Decision and Control; and is currently a Distinguished Lecturer for the IEEE Control Systems Society. He is co-editor (with P.J. Antsaklis) of the book An Introduction to Intelligent and Autonomous Control, *Kluwer Academic Press, 1993; co-author (with S. Yurkovich) of the book* Fuzzy Control, *Addison Wesley Longman Pub., 1998; co-author (with K.L. Burgess) of the book* Stability Analysis of Discrete Event Systems, *John Wiley and Sons, 1998; co-author (with V. Gazi, M.L. Moore, W. Shackleford, F. Proctor, and J.S. Albus) of the book* The RCS Handbook: Tools for Real Time Control Systems Software Development, *John Wiley and Sons, NY, 2001; co-author (with J.T. Spooner, M. Maggiore, R. Ordonez) of the book* Stable Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximator Techniques, *John Wiley and Sons, NY, 2002; and author of* Biomimicry for Optimization, Control, and Automation, *Springer-Verlag, London, UK, 2005. For more information, see: http://www.ece.osu.edu/~passino/*