Optimization Based on Bacterial Chemotaxis

Sibylle D. Müller, Jarno Marchetto, Stefano Airaghi, and Petros Koumoutsakos

***Abstract*—We present an optimization algorithm based on a model of bacterial chemotaxis. The original biological model is used to formulate a simple optimization algorithm, which is evaluated on a set of standard test problems. Based on this evaluation, several features are added to the basic algorithm using evolutionary concepts in order to obtain an improved optimization strategy, called the bacteria chemotaxis (BC) algorithm. This strategy is evaluated on a number of test functions for local and global optimization, compared with other optimization techniques, and applied to the problem of inverse airfoil design. The comparisons show that on average, BC performs similar to standard evolution strategies and worse than evolution strategies with enhanced convergence properties.**

***Index Terms*—Bacterial chemotaxis, evolution strategies, stochastic optimization methods.**

I. Introduction

IN THE FIELD of optimization, many researchers have been inspired by biological processes such as evolution [1], [2] or the food-searching behavior of ants [3] to develop new optimization methods such as evolutionary algorithms or ant codes. These techniques have been found to perform better than the classical heuristic or gradient-based optimization methods, especially when faced with the problem of optimizing multimodal, nondifferentiable, or discontinuous functions. Examples of applying these biologically inspired strategies in the field of engineering range from aerodynamic design (e.g., [4]) to job-shop scheduling problems [5].

Another biologically inspired optimization method is the *chemotaxis algorithm*, pioneered by Bremermann [6] and his coworkers [7], [8], proposed by analogy to the way bacteria react to chemoattractants in concentration gradients. This algorithm were analyzed for chemotaxis in a three-dimensional (3-D) gradient by Bremermann [6] and employed for the training of neural networks [7], [8]. A similar algorithm is the guided accelerated random search technique [9], which was applied to optimize parameters in flight control systems [9], [10] and to optimize perceptrons [11].

Manuscript received July 6, 2000; revised September 27, 2000 and February 28, 2001. This work was supported by the Swiss National Science Foundation under Grant 21–54093-98.

S. D. Müller and P. Koumoutsakos are with the Institute of Computational Sciences, Swiss Federal Institute of Technology (ETH), CH-8092 Zürich, Switzerland (e-mail: muellers@inf.ethz.ch; petros@inf.ethz.ch).

S. Airaghi is with the Institute of Fluid Dynamics, Swiss Federal Institute of Technology (ETH), CH-8092 Zürich, Switzerland (e-mail: stefano@eniac.ethz.ch).

J. Marchetto was with the Institute of Fluid Dynamics, Swiss Federal Institute of Technology (ETH), CH-8092 Zürich, Switzerland. He is now with Research and Development, the Fantastic Corporation, 6928 Manno, Switzerland (e-mail: jarno.marchetto@alumni.ethz.ch).

Publisher Item Identifier S 1089-778X(02)02205-1.

Whereas these algorithms contain the fundamental concepts of bacteria chemotaxis (BC), more recent biological findings have provided further details and models of the process. This paper presents a novel optimization strategy whose roots are found in an extended biological model of bacterial chemotaxis [12], [13]. The basic model and the resulting optimization algorithm differ from the *chemotaxis algorithm* as follows.

1) The bacteria from our basic model incorporate information from concentration values.
2) They do not keep on running in the same direction in case of increasing concentrations of the chemoattractants.
3) Beyond studying the biological model, the main part of our work is dedicated to the further development of the basic strategy to yield a powerful, robust optimization technique.

Bacteria are single-cell organisms, one of the simplest form of life developed on earth. Despite their "simplicity," they acquire information about their environment, orient themselves in this environment, and use this information efficiently to survive. This reaction of the organism to its environment has been subject to intensive research in the past decades, see, e.g., [14]–[17]. It is also interesting for scientists in the field of optimization to study the bacterial behavior. Like bacteria, optimization algorithms may be viewed as entities that gather information about a function landscape and then use this information to move to an optimum. At the same time, the simplicity and robustness of the process of bacterial chemotaxis suggest a starting point for constructing an optimization algorithm.

Although it has been found that bacteria do share information among each other [18], not much is known presently about the communication patterns. Generally, bacteria are considered individuals and social interaction is not used in the models. These models differ from the interaction models for the behavior of social insects (such as ants, bees, wasps, or termites) that are viewed as systems with collective intelligence resulting in powerful problem-solving capabilities [19].

The term *taxis* refers to the locomotory response of a cell to its environment. In a taxis, a cell responds such that it changes both its direction and the duration of the next movement step. The tactic response requires some directional information from the environment that bacteria obtain by comparing an environmental property at two different time steps. If the tactic response is related to information about chemical concentrations (that may be either attractants or repellents), it is called chemotaxis.

For optimization purposes, we concentrated on studying microscopic models that consider the chemotaxis of a single bacterium instead of macroscopic models that analyze the movement of bacteria colonies.

Our studies are based on a microscopic model for bacterial chemotaxis proposed by Berg and Brown [12] and Dahlquist

et al. [13]. Berg and Brown [12] analyze the chemotaxis toward amino acids of the bacterium *Escherichia coli*. The analysis provides experimental measurements of parameters in the model. Dahlquist *et al.* [13] study bacterial chemotaxis toward amino acids of the bacterium *Salmonella typhimurium*. They present a mathematical model and present experimental results that validate their model.

In the original model, the 3-D motion of a bacterium is approximated and the model parameters depend on the particular bacterial environment and on the experimental setup. For preliminary studies, an optimization algorithm is formulated based on the biological model. This algorithm is used for optimizing two-dimensional (2-D) test functions and contains model parameters that are selected *a priori*.

After studying the searching behavior of bacteria with this basic model, we enhance it for optimization purposes [20]. The first modification regards the determination of model parameters that will be referred to as strategy parameters in the following. The search for appropriate strategy parameters is inspired by nature. From an evolutionary perspective, bacteria adapt their motility properties so that they have a better chance to survive in changing environments. Our approach finds its counterpart in this evolutionary idea: using an evolution strategy (ES), we optimize the strategy parameters by minimizing the number of iterations that the virtual bacterium needs from the start to the optimum. The second modification regards the extension of the 2-D to an n-dimensional model.

Further improvements include the automatic modification of strategy parameters according to the properties of the optimization function, a possibility of escaping plateaus, and additional features to facilitate the search of global optima. These improvements yield a new optimization algorithm called the BC algorithm.

To assess its properties, BC is evaluated on standard test problems, its performance is compared against other optimization algorithms, and it is finally applied to the optimization of airfoil profile shapes with an inverse design technique.

This paper is organized as follows. Section II describes the basic bacterial chemotaxis model in two dimensions and tests it on various standard optimization problems. How appropriate parameters for the bacteria algorithm are chosen is shown in Section III. Section IV presents the extension of the bacteria algorithm from two to n dimensions and improvements for the n-dimensional strategy, yielding the BC strategy, which is applied on test functions. Convergence properties of the BC strategy are compared with those of other optimization algorithms in Section V. In Section VI, the BC code is further extended for the search of global optima. The application of the BC strategy on the inverse airfoil design is shown in Section VII and conclusions are finally drawn in Section VIII.

II. 2-D Model

In this section, we describe the model of bacterial chemotaxis and we formulate it as an optimization algorithm. This algorithm is evaluated on several test functions to assess its advantages and drawbacks.

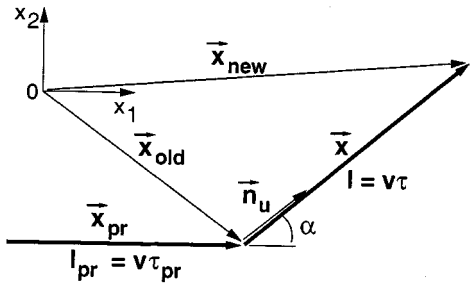

Fig. 1. 2-D path of a bacterium consisting of the previous step $\vec{x}_{\rm pr}$ and the actual step $\vec{x}$ yielding the new position $\vec{x}_{\rm new}$ of the bacterium.

A. Description of the 2-D Model

Dahlquist *et al.* [13] model the motion of a single bacterium in two dimensions by making the following assumptions.

1) The path of a bacterium is a sequence of *straight-line* trajectories joined by instantaneous turns, each trajectory being characterized by *speed, direction, and duration.*
2) All trajectories have the *same constant speed.*
3) When a bacterium turns, its choice of a new direction is governed by a probability distribution, which is azimuthally symmetric about the previous direction. In two dimensions, this means that the probability to turn left or right is the same.
4) The angle between two successive trajectories is governed by a probability distribution.
5) The duration of a trajectory is governed by an exponentially decaying probability distribution.
6) The probability distributions for both the angle and the duration are independent of parameters of the previous trajectory.

These assumptions yield a model that can be described by the algorithm presented below. Every step is characterized by a velocity, a duration, and a direction; the bacterium trajectory consists of a sequence of straight segments of variable length and orientation.

Bacteria algorithm based on the bacterial chemotaxis model: The algorithm is presented below and illustrated in Fig. 1.

1) Compute the velocity v. In the model, the velocity is assumed to be a scalar constant value

$$v = \text{const.} \tag{1}$$

2) Compute the duration of the trajectory τ from the distribution of a random variable with an exponential probability density function

$$P(X = \tau) = \frac{1}{T} e^{-\tau/T} \tag{2}$$

where the expectation value $\mu = E(X) = T$ and the variance $\sigma^2 = \text{Var}(X) = T^2$. The time T is given by

$$T = \begin{cases} T_0, & \text{for } \frac{f_{\rm pr}}{l_{\rm pr}} \geq 0 \\ T_0 \left(1 + b \left| \frac{f_{\rm pr}}{l_{\rm pr}} \right| \right), & \text{for } \frac{f_{\rm pr}}{l_{\rm pr}} < 0 \end{cases} \tag{3}$$

where

T_0	minimal mean time;
$f(x_1, x_2)$	function to minimize (2-D);
$f_{\rm pr}$	difference between the actual and the previous function value;
$l_{\rm pr} = \lvert\vec{x}_{\rm pr}\rvert$, $\vec{x}_{\rm pr}$	vector connecting the previous and the actual position in the parameter space;
b	dimensionless parameter.

3) Compute the new direction. The probability density distribution of the angle α between the previous and the new direction is Gaussian and reads, for turning right or left, respectively

$$P(X=\alpha, \nu=\mu) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(\alpha-\nu)^2}{2\sigma^2}\right]$$
$$P(X=\alpha, \nu=-\mu) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(\alpha-\nu)^2}{2\sigma^2}\right] \quad (4)$$

where the expectation value $\mu = E(X) = 62^\circ$ and the standard deviation $\sigma = \sqrt{\mathrm{Var}(X)} = 26^\circ$ referring to the chemotaxis measurements of the bacterium *Escherichia coli* by Berg and Brown $\alpha \in [0^\circ, 180^\circ]$. The choice of a right or left direction as referring to the previous trajectory is determined using a uniform probability density distribution, thereby yielding a probability density distribution for the angle α

$$P(X=\alpha) = \frac{1}{2}\cdot[P(X=\alpha, \nu=\mu) + P(X=\alpha, \nu=-\mu)]. \quad (5)$$

We formulate the following variation of the expectation value and the variance, which is applied only if $f_{\rm pr}/l_{\rm pr} < 0$:

$$\mu = 62^\circ(1-\cos(\theta)) \quad (6)$$
$$\sigma = 26^\circ(1-\cos(\theta)) \quad (7)$$

with

$$\cos(\theta) = e^{-\tau_c \tau_{\rm pr}} \quad (8)$$

where τ_c is the correlation time, and $\tau_{\rm pr}$ the duration of the previous step. Equation (8) is based on the first-order approximation of the average value of the cosine of the angle between successive trajectories at times t and $t+\tau_c$ as a function of t, as shown by Dahlquist *et al.* [13, Fig. 5]. It provides us with a statistical measure of direction-duration dependencies in experimental observations of BC. Note, however, the difference between the formulation in (8) in which the duration of the previous trajectory is used and the formulation

$$\cos(\theta) = e^{-\tau_c \tau} \quad (9)$$

in which the actual trajectory duration is used, as found in [13]. With (8), which is used in the following, the probability distribution of the angle becomes dependent on the duration of the previous step, which is contradictory to the assumption made by Dahlquist *et al.*In Section IV, we discuss results of the application of both algorithms on a number of test problems.

As soon as α is computed, it is possible to obtain the normalized new displacement vector $\vec{n}_u$ with unit length.

4) Compute the new position. The length of the path l is given by

$$l = v\tau. \quad (10)$$

The normalized new direction vector $\vec{n}_u$ with $|\vec{n}_u| = 1$ is multiplied by l to obtain the displacement vector $\vec{x}$

$$\vec{x} = \vec{n}_u l \quad (11)$$

such that the new location of the bacterium is

$$\vec{x}_{\rm new} = \vec{x}_{\rm old} + \vec{n}_u l. \quad (12)$$

In summary, the algorithm contains the following parameters: the minimal mean time T_0, the dimensionless gradient parameter b, the correlation time τ_c, and the velocity v. They constitute the optimization strategy parameters that are adjusted as described in the following sections.

B. Tests of the 2-D Model

The 2-D model presented in the previous subsection is tested to get an insight into the behavior of the "virtual" bacterium. In particular, we test the model on finding optima of simple unimodal test functions, as well as on finding the global optimum of multimodal test functions. Whereas unimodal functions (functions with only one extremum) are used mainly to check the convergence speed of the optimization algorithm, multimodal functions (functions with more than one extremum) are used to check the capability of the algorithm to escape from local minima. In the following, we use test functions suggested in [1], [21], and [22] with known optimum parameters and known optimum function values. All these test functions are to be minimized. The algorithm terminates as soon as a certain convergence criterion is reached. This criterion is based on the difference between the function value f of the current parameters and the function value of the optimum parameters $f_{\rm opt}$. In each iteration step, this difference $|f - f_{\rm opt}|$ is compared to a previously set precision value, ϵ. As soon as the difference becomes smaller than the target precision, ϵ, the algorithm terminates.

Values for the strategy parameters reported in [13] were obtained from experiments with bacteria moving in specific concentration gradients (varying exponentially with distance) and are, therefore, not recommended to use in a general optimization strategy. Only for the preliminary tests presented in this section, a very simple approach to obtain more general parameters is chosen: we define about 20 different random parameter sets and select out of these the set that yields the minimal number of iterations.

1) Tests on Unimodal Functions: The first case is the minimization of the quadratic function

$$F_1(x,y) = (x-1)^2 + (y-1)^2$$

with a single global minimum placed in $(1,1)$ and the minimum value of $F_1(1,1) = 0$. The target precision was set to $\epsilon = 10^{-6}$

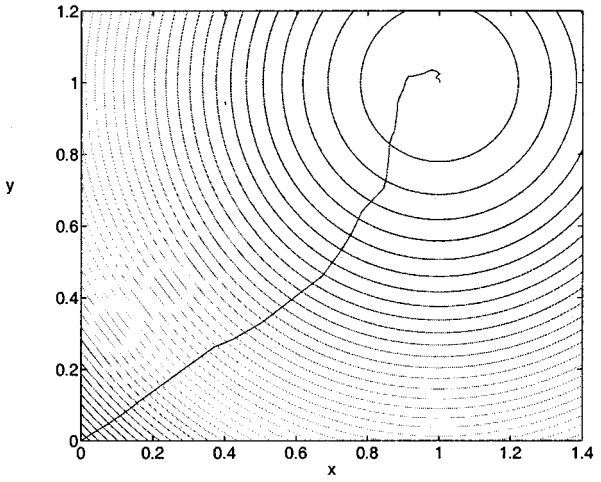

Fig. 2. Path of a single bacterium on $F_1(x,y) = (x-1)^2 + (y-1)^2$. Start point $(0,0)$, $\epsilon = 10^{-10}$, $T_0 = 2.6 \cdot 10^{-5}$, $b = 1.53 \cdot 10^3$, $\tau_c = 3.65 \cdot 10^3$, $v = 1$, number of steps: 389, path length: 1.5727.

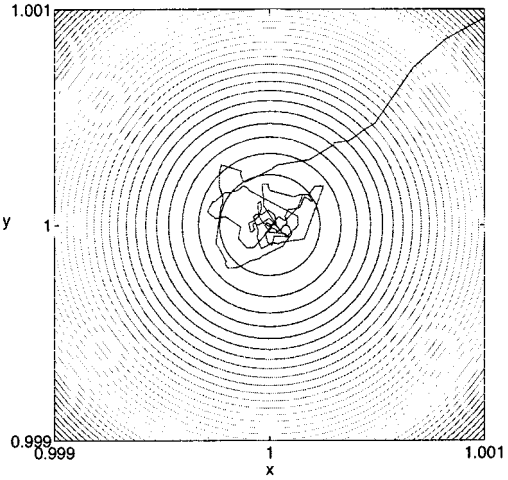

Fig. 3. Zoom of Fig. 2.

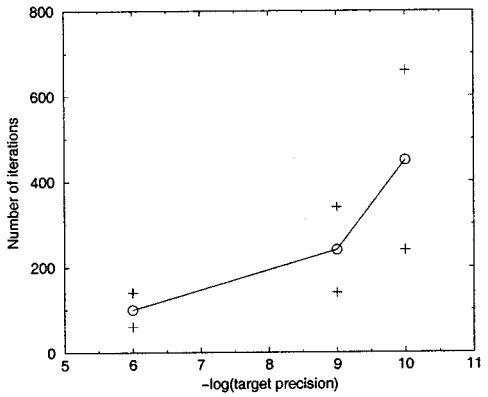

Fig. 4. Average number of iterations plus standard deviation as a function of the target precision for the optimization of $F_1(x,y) = (x-1)^2 + (y-1)^2$. Start point $(0,0)$.

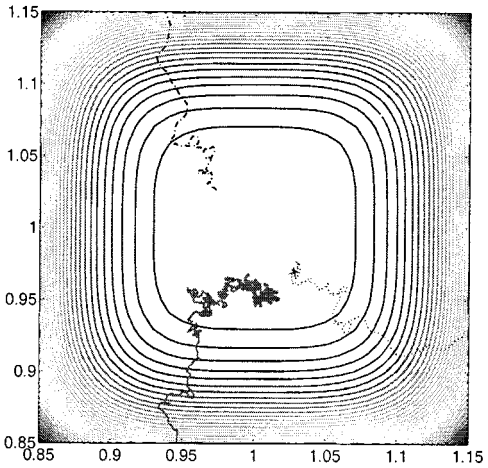

Fig. 5. Paths of three bacteria on $F_2(x,y) = (x-1)^4 + (y-1)^4$ using different values of b. Start point $(0,0)$, $\epsilon = 10^{-6}$, $T_0 = 5.0 \cdot 10^{-4}$, $\tau_c = 7.9 \cdot 10^2$, $v = 1$. $b = 270$ (—), number of steps: 2441, path length: 2.7044; $b = 1000(\cdots)$, number of steps: 495, path length: 2.1017; $b = 2000$ (– . –), number of steps: 232, path length: 3.6102.

with a start point in $(0,0)$. The computation takes 100 steps (average of 10^4 runs) with a standard deviation of 40 steps to reach the minimum of the function F_1. Averaging is necessary because of the stochastic processes involved in the model.

Increasing the precision, more computations are needed to reach the goal starting from the same point. For $\epsilon = 10^{-9}$, the number of iterations is 240 ± 100; for $\epsilon = 10^{-10}$, 450 ± 210 (Fig. 2), averaged on 10^4 runs, as seen in Fig. 4.

Note that in optimizing with a high precision, the bacterium goes straight until an area close to the goal, then it moves in an increasingly random fashion as soon as the function becomes flatter. Thus, the algorithm requires a large number of iterations to arrive finally at the optimum (Fig. 3). This behavior of the virtual bacterium is consistent with that observed in real bacteria: in a uniform isotropic environment, i.e., one without directional information, bacteria will move randomly [17].

Subsequently, we studied the influence of higher order polynomials on the number of iteration steps. In the following tests, we analyzed the path on the test function

$$F_2(x,y) = (x-1)^4 + (y-1)^4$$

starting again from the point $(0,0)$ with the goal at $(1,1)$ (Fig. 5). The three simulations differ only in the parameter b. We observe that the larger the b, the smaller the number of steps to reach the goal, although the path seems to be less direct. Simulation with $b > 2000$ (not shown in Fig. 5) becomes unstable in the sense that the bacterium jumps far away from the targeted minimum and is not able to reapproach its goal.

Another test is the generalized Rosenbrock function

$$F_3(x,y) = 100 \cdot (x^2 - y)^2 + (1-x)^2$$

with the minimum $F_3(1,1) = 0$. We note that starting from two different points at $(-2,-2)$ and $(2,-1)$, the bacterium goes relatively straight down in the direction of the minimum, but spends a lot of computation time on the plateau (Fig. 6). Here, we can already note that getting stuck on a plateau seems to be a disadvantage of the current method that needs to be improved.

2) Tests on Multimodal Functions: First, we test the code on Rastrigin's function (see Fig. 7)

$$F_4(x,y) = 20 + (x^2 - 10 \cdot \cos(2\pi x) + y^2 - 10 \cdot \cos(2\pi y))$$

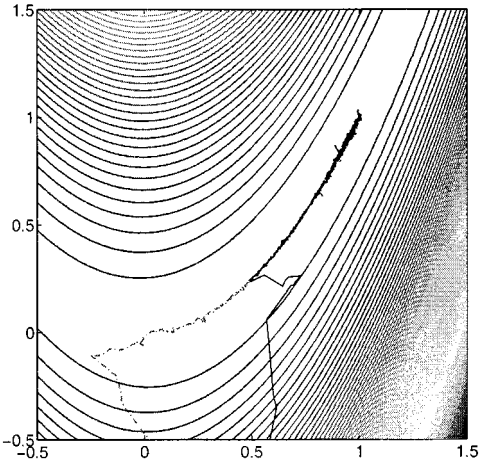

Fig. 6. Paths of two bacteria on $F_3(x,y) = 100 \cdot (x^2 - y)^2 + (1-x)^2$ using different start points. $\epsilon = 10^{-6}, T_0 = 5.0 \cdot 10^{-4}, b = 2.0, \tau_c = 7.9 \cdot 10^2$, $v = 1$. Start point $(2,-1)$(—), number of steps: 17339, path length: 29.144; Start point $(-2,-2)$(- - -), number of steps: 12722, path length: 15.217.

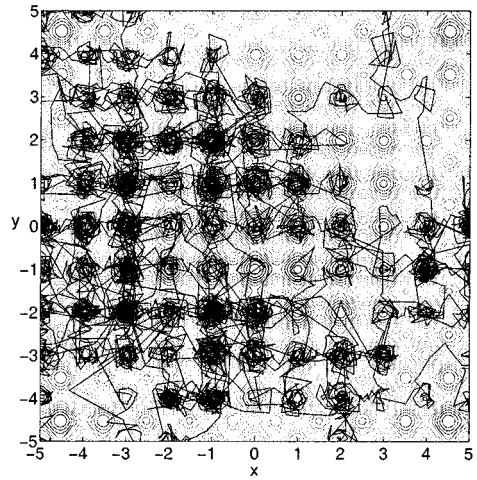

Fig. 8. Path of a single bacterium on $F_4(x,y) = 20 + (x^2 - 10 \cdot \cos(2\pi x) + y^2 - 10 \cdot \cos(2\pi y))$. Start point $(-5,-5)$, $\epsilon = 10^{-6}$, $b = 10.0$, $T_0 = 5.0 \cdot 10^{-4}$, $\tau_c = 7.9 \cdot 10^2$, $v = 1$. Number of steps: $3 \cdot 10^4$. Note that the simulation does not stop at the global minimum.

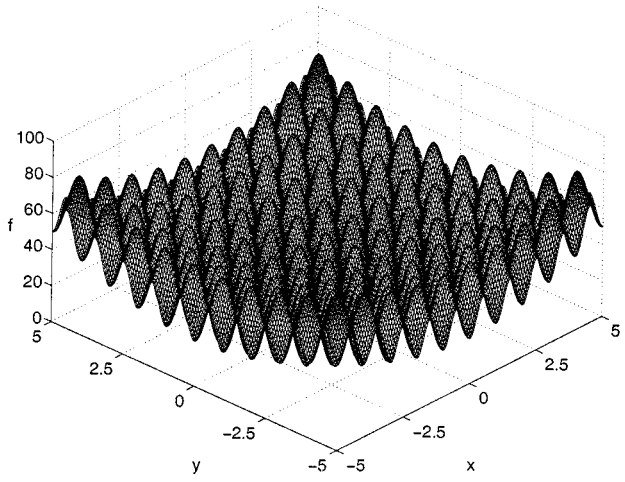

Fig. 7. Function $F_4(x,y) = 20 + (x^2 - 10 \cdot \cos(2\pi x) + y^2 - 10 \cdot \cos(2\pi y))$. Global minimum at $(0,0)$.

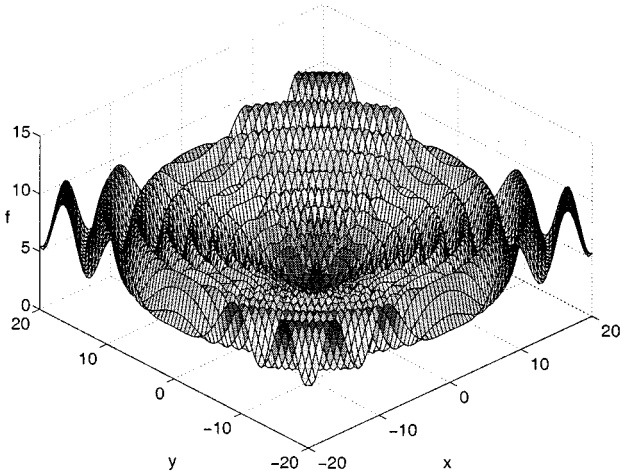

Fig. 9. Function $F_5(x,y) = (x^2+y^2)^{0.25} \cdot (\sin^2(50(x^2+y^2)^{0.1}) + 1.0)$. Global minimum at $(0,0)$.

with the global minimum $F_4(0,0)$. Fig. 8 shows the contour lines of the function together with the path of the bacterium that is characterized by long residence times in areas close to several local optima and short residence times far from local optima. The stochastisity in the chemotactic model allows for escaping local minima. However, using the current methodology, the bacterium visits the global optimum, but is not able to stop there. Ideas to avoid this problem are presented in Section VI.

Another multimodal function with local minima located on concentric circles is (see Fig. 9)

$$F_5(x,y) = (x^2+y^2)^{0.25} \cdot (\sin^2(50(x^2+y^2)^{0.1}) + 1.0).$$

Again, the virtual bacterium finds many local minima and also visits the global one but does not stop there. Fig. 10 illustrates the path on this function, indicating that the bacterium stays longer in regions that are more promising.

The algorithm can escape from local optima due to the stochastic processes inherent in the algorithm when computing the duration and the direction of the trajectory. Therefore, bacterial chemotaxis may provide a basis for the search of a global optimum. In Section VI, we discuss the suitable adjustment of the algorithm in order to effectuate this goal.

III. Optimization of the Strategy Parameters

From the tests of the previous section, we recognize that the quality (measured in terms of number of iterations) of the search of a unimodal function with a bacterial strategy depends strongly on two aspects:

1) the chosen target precision ϵ;
2) the chosen strategy parameters T_0, b, τ_c, and v.

Since one usually has an estimate on the overall desired accuracy of the solution, it is an easy task to define the target precision. However, optimum strategy parameters are not known *a priori* because the values given in [13] depend on the experimental setup and are therefore not generalizable.

To obtain strategy parameters that are independent from these experiments, we study the performance of the bacteria algorithm on a set of quadratic functions in the following way: the strategy

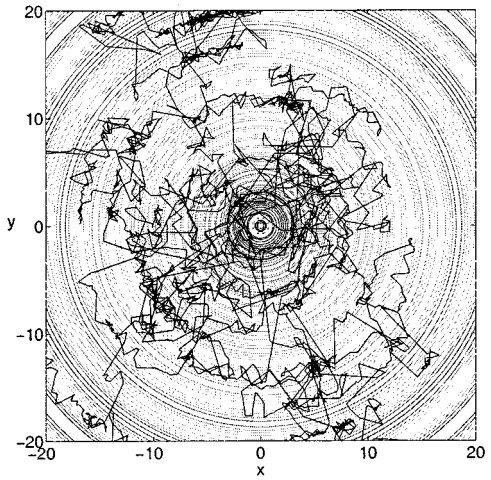

Fig. 10. Path of a single bacterium on $F_5(x,y) = (x^2+y^2)^{0.25} \cdot (\sin^2(50(x^2+y^2)^{0.1}) + 1.0)$. Start point $(-5,-5)$, $\epsilon = 10^{-6}$, $T_0 = 5.0 \cdot 10^{-4}$, $b = 8.0 \cdot 10^2$, $\tau_c = 1.79 \cdot 10^3$, $v = 1$, number of steps: 10^5. Note that the simulation does not stop at the global minimum.

parameters are the optimization parameters and the number of iterations to the optimum is the objective function that is to be minimized. We are aware of the fact that the strategy parameters found by this optimization are only optimal for the class of functions on which the bacteria strategy is applied. However, it is a widely accepted method in the field of evolutionary computation to obtain strategy parameters by investigating the behavior of an optimization algorithm using only certain functions. For example, the well-known 1/5-success rule by Rechenberg [23] is based on his theoretical study of the $(1+1)$ ES, using only the sphere and the corridor functions.

The motivation for optimizing the parameters is consistent with the principles of evolution for realistic bacterial organisms. The motility properties of bacteria tend to adapt to the environment since bacteria that are better adapted to their environment have a bigger chance to survive.

In order to optimize the strategy parameters, we implement a covariance-matrix adaptation evolution strategy (CMA-ES) [22]. This strategy adapts the covariance matrix of the mutation distribution efficiently to the topology of badly scaled and/or nonseparable functions. Moreover, the CMA-ES is invariant against linear transformations of the object parameter space. All strategy parameters are chosen according to [22].

For every test, the start point is placed randomly on a circle with center in $(0.5, 0.5)$ and radius $r = \sqrt{0.5}$. The mean and variance were computed on 10^4 values from simulations with the same start point. Averaging over more than 10^4 simulations does not influence mean and variance significantly.

Note that for all the considered test functions the velocity v always tended to a constant value, namely, $v = 1$. For this reason, the value $v = 1$ is used as a constant in the algorithm.

Relations among the optimum strategy parameters and the required precision are identified using the following algorithm.

1) Define the needed precision ϵ for the computation.
2) From the given precision compute a value for T_0 using

$$T_0 = \epsilon^{0.30} \cdot 10^{-1.73}. \tag{13}$$

3) From the computed T_0, find a value for b using

$$b = T_0 \cdot (T_0^{-1.54} \cdot 10^{0.60}). \tag{14}$$

4) From the values of T_0 and b, compute τ_c using

$$\tau_c = \left(\frac{b}{T_0}\right)^{0.31} \cdot 10^{1.16}. \tag{15}$$

The coefficients in the above equations were determined using a linear regression on the whole set of data obtained from the considered test functions.

The relation between T_0 and b is dictated by the following reasons.

1) If T_0 is too big, the needed precision ϵ may not be reached. In fact, T_0 is the shortest time step, which should be small enough to let the bacterium reach the goal with the needed precision.
2) If the bacterium sees a big gradient and if b is not small enough, the bacterium can jump far away from the area of attraction and may not come back. This is unstable behavior.

IV. Extension to an n-Dimensional Strategy

In this section, we show the extension of the current algorithm from two to n dimensions, improvements to the algorithm, yielding the BC strategy. The BC code is then tested on a set of standard optimization functions.

A. Development of an n-Dimensional Algorithm

Extending the 2-D algorithm to n dimensions amounts merely to changes of geometrical nature. We define a position $(x_1, \ldots, x_n)$ in an n-dimensional space with a radius r and $n-1$ angles $(\varphi_1, \ldots, \varphi_{n-1})$. The position in n dimensions is defined as follows:

$$\begin{aligned} x_1 &= r \cdot \prod_{k=1}^{n-1} \cos(\varphi_k) \\ x_i &= r \cdot \sin(\varphi_{i-1}) \cdot \prod_{k=i}^{n-1} \cos(\varphi_k), \quad i = 2, \ldots, n-1 \\ x_n &= r \cdot \sin(\varphi_{n-1}). \end{aligned} \tag{16}$$

Applying (16) to the bacterial chemotaxis model, we can extend the algorithm described in Section II to n dimensions. The 2-D and the n-dimensional model differ only in the computation of the angles and of the normalized direction vector.

The new direction is computed by using the Gaussian probability density distribution of the angle φ_i in the (x_i, x_{i+1}) plane, where φ_i is measured from the the axis x_i and reads for turning left or right, respectively

$$\begin{aligned} P(X_i = \varphi_i, \nu_i = \mu_i) &= \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left[-\frac{(\varphi_i - \nu_i)^2}{2\sigma_i^2}\right] \\ P(X_i = \varphi_i, \nu_i = -\mu_i) &= \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left[-\frac{(\varphi_i - \nu_i)^2}{2\sigma_i^2}\right] \end{aligned} \tag{17}$$

where $\mu_i = E(X_i) = 62°$ and $\sigma_i = \sqrt{\mathrm{Var}(X_i)} = 26°$ and $\varphi_i \in [0°, 180°]$. The left or right direction is determined using a uniform probability density distribution, thereby yielding a probability density distribution of the angle φ_i

$$P(X_i = \varphi_i) = \frac{1}{2} \cdot \Big[P(X_i = \varphi_i, \nu_i = \mu_i) + P(X_i = \varphi_i, \nu_i = -\mu_i) \Big]. \quad (18)$$

As for the 2-D algorithm, the expectation and the variance are modified so that the previous and the new directions are correlated, if the bacterium is moving toward a decreasing value of the function (this implies a negative gradient) and if the previous computed trajectory duration is short [13]. These two conditions imply that

$$\mu_i = 62°(1 - \cos(\theta)) \quad (19)$$

$$\sigma_i = 26°(1 - \cos(\theta)). \quad (20)$$

As soon as φ_i are computed, it is possible to obtain the normalized new displacement vector $\vec{n}_u$. To obtain this vector, we sum the new computed angles $(\varphi_1 \ldots \varphi_{n-1})$ to the old ones and, using (16), the radius r is set to one. The new position in Cartesian coordinates is then found.

B. Improvements of the n-Dimensional Algorithm to Yield the BC Strategy

In this section, we present features of the n-dimensional algorithm that were added to obtain an improved optimization strategy. In particular, we address the following extensions.

1) Adapt the parameter b automatically to functions with different slopes and values.
2) Change all strategy parameters during the local search in order to refine them for an increasing precision requirement.
3) Avoid spending much computation time on plateaus.

1) Automatic Adaptation of the Parameter b: One of the critical points in the formulation of the current bacteria strategy is the definition of the time T [see (3)]

$$T = \begin{cases} T_0, & \text{for } \frac{f_{\mathrm{pr}}}{l_{\mathrm{pr}}} \geq 0 \\ T_0 \left(1 + b \left| \frac{f_{\mathrm{pr}}}{l_{\mathrm{pr}}} \right| \right), & \text{for } \frac{f_{\mathrm{pr}}}{l_{\mathrm{pr}}} < 0. \end{cases} \quad (21)$$

The influence of the gradient on the time T is controlled only by the strategy parameter b. The optimal strategy parameter b was found for certain types of functions, as described in (14). However, functions may have completely different slopes than the ones considered in the tests. For example, if the gradient becomes too big, T may become so large that the bacterium jumps out of the area of attraction and may never be able to get back. We refer to this behavior as instability. To prevent the bacteria strategy from leaving promising regions, we modify the algorithm such that it becomes less dependent on the function that is optimized. The following approach is, of course, not completely independent of the function, but increases the stability significantly. Hence, (3) now reads

$$T = \begin{cases} T_0, & \text{for } \frac{f_{\mathrm{pr}}}{l_{\mathrm{pr}}} \geq 0 \\ T_0 \left(1 + b_{\mathrm{corr}} \left| \frac{f_{\mathrm{pr}}}{l_{\mathrm{pr}}} \right| \right), & \text{for } \frac{f_{\mathrm{pr}}}{l_{\mathrm{pr}}} < 0 \end{cases} \quad (22)$$

with

$$b_{\mathrm{corr}} = \begin{cases} b \cdot \frac{1}{|\frac{\Delta f_{\mathrm{pr}}}{\Delta l_{\mathrm{pr}}}| + 1}, & \text{at each iteration step} \\ b \cdot \frac{1}{|\frac{\Delta f_{\mathrm{pr}}}{\Delta l_{\mathrm{pr}}}| + 1} \cdot \frac{1}{|f_{\mathrm{pr}}| + 1}, & \text{once in the beginning and once after each parameter change} \end{cases} \quad (23)$$

where b is computed as described in (14). f is the function to be minimized, f_{pr} is the difference between the actual and the previous function value, $l_{\mathrm{pr}} = |\vec{x}_{\mathrm{pr}}|$, $\vec{x}_{\mathrm{pr}}$ is the vector connecting the previous and the actual position in the parameter space, and we have

$$\frac{\Delta f_{\mathrm{pr}}}{\Delta l_{\mathrm{pr}}} = \frac{1}{n_{\mathrm{c}}} \sum_{i=1}^{n_{\mathrm{c}}} \frac{f_{\mathrm{pr}}(i)}{l_{\mathrm{pr}}(i)} \quad (24)$$

for the n_{c} (user-defined) preceding subsequent values of $f_{\mathrm{pr}}/l_{\mathrm{pr}}$.

The first factor of the correction $1/(|\Delta f_{\mathrm{pr}}/\Delta l_{\mathrm{pr}}|+1)$ controls the parameter b_{corr} depending on the change of the slope of the function. If the change in slope tends to zero, the term does not influence b_{corr}. If the change in slope tends to infinity, the parameter b_{corr} tends to zero. This means that the bacterium moves with little steps. This correction should avoid that the bacterium skips a domain with a minimum by jumping far away from it and not being able to come back.

The second factor of the correction, $1/(|f_{\mathrm{pr}}| + 1)$ considers the function values at the actual position and the previous position. Assuming a slowly varying gradient, the values of the function f at the previous and at the actual position tend to be the same (this means that the bacterium is on the same level), the parameter b_{corr} is not corrected. If the difference of the two values tends to infinity, b_{corr} tends to zero and the bacterium moves with small steps.

These two terms, which have a similar influence on the correction, should be well distinguished. In fact, if we consider, for example, a quadratic function, the value of $(\Delta f_{\mathrm{pr}})/(\Delta l_{\mathrm{pr}})$ could be nearly the same on the whole domain, but the bacterium must slow down close to the goal. For this reason the term $1/(|f_{\mathrm{pr}}| + 1)$ must also be introduced for correction. This confines the bacterium in the domain of the minimum although the change of slope is equal to zero.

The second factor is computed only in "critical" situations, i.e., in the beginning of the optimization and after changes of the parameters (see next subsection), where the function values are especially difficult to control. For all other instances, we prefer to leave out the second factor since we do not want to reduce the parameter b_{corr} to extremely small values that may cause the algorithm to work inefficiently.

2) Strategy Parameter Adaptation: In order to further improve our algorithm, we introduce the automatic change of all strategy parameters during the local search in order to refine

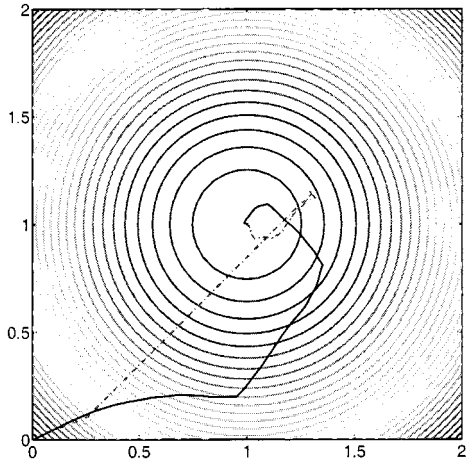

Fig. 11. Paths of two bacteria on $F_1(x, y)$ with and without strategy parameter refinement. Start point $(0, 0)$, $\epsilon_{\rm end} = 10^{-10}$. Without strategy parameters refinement (—): $\epsilon_{\rm init} = 10^{-10}$, $n_{\rm pc} = 0$, number of function evaluations: 10 806. With strategy parameters refinement $(- \cdot -)$: $\epsilon_{\rm init} = 10^{-2}$, $n_{\rm pc} = 10$, number of function evaluations: 1172.

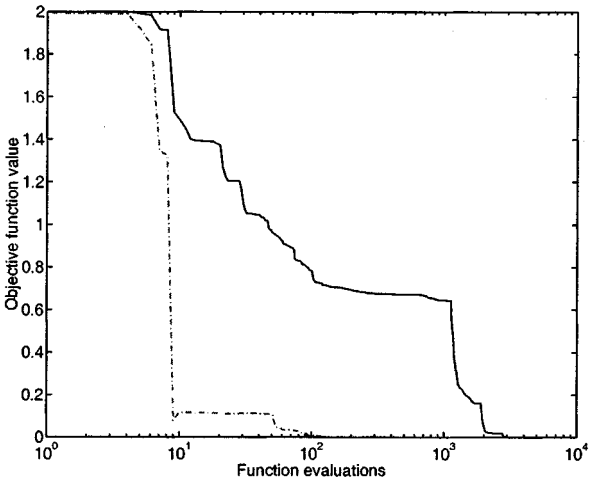

Fig. 12. Convergence plot for the two bacteria paths shown in Fig. 11. Without parameter refinement (—). With parameter refinement $(- \cdot -)$.

them for increased precision. Using the relationships given in (13), (14), and (15), the optimal parameters for a given precision are known. We adapt the parameters during the optimization as follows.

1) Define initial and final precisions $\epsilon_{\rm init}$ and $\epsilon_{\rm end}$ with $\epsilon_{\rm init} > \epsilon_{\rm end}$. The initial precision determines the starting parameters using the relations in (13), (14), and (15).
2) Define the maximum number of parameter changes.
3) Start the computation searching for the minimum of the function. As the initial precision is reached, the parameters are adapted to another precision (defined by the number of parameter changes) and the search continues, until this new precision is reached. This adaptation is computed until the bacterium finds the minimum with the final needed precision $\epsilon_{\rm end}$.

A given precision ϵ is reached if the difference between a given number (user-defined) of *subsequent* function values found by the bacterium is smaller than ϵ

$$|f_{\rm pr}| < \epsilon$$

for a given number $n_{\rm pc}$ of subsequent values. This parameter adaptation is important because it permits refining the search adaptively. In this way, the bacterium does not waste much time at the beginning because of parameters that constrain small steps and it can proceed toward the minimum directly (Figs. 11 and 12). On the other hand, by adapting the parameters, the bacterium slows down near the minimum so that it does not skip the minimum domain.

3) Escape from Plateaus: As discussed earlier, the bacterial algorithm faces a difficulty in escaping from plateaus. To remedy this, we propose the following algorithm: if the absolute value of the slope $(|f_{\rm pr}/l_{\rm pr}|)$ is smaller than a given value for a subsequent number of steps, then the value of $b_{\rm corr}$ must be multiplied by a constant

$$b_{\rm plateau} = b_{\rm corr} \cdot B \text{ if } \left|\frac{f_{\rm pr}}{l_{\rm pr}}\right| < \epsilon_{\rm actual} \cdot A$$

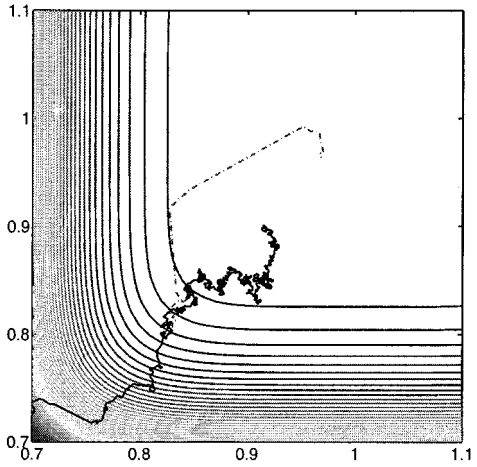

Fig. 13. Paths of two bacteria on $F_6 = (x-1)^6 + (y-1)^6$. Start point $(0, 0)$, $n_{\rm plateau} = 3$, $A = 10^6$, $B = 4.0$, $\epsilon_{\rm end} = 10^{-7}$. Without plateau speedup (—), number of steps: 2156. With plateau speedup $(- \cdot -)$, number of steps: 365.

for a given number $n_{\rm plateau}$ of subsequent numbers (user-defined). Here, $\epsilon_{\rm actual}$ is the actual required precision and A and B are user-defined constants. This strategy should give the bacterium the capability to speed up in plateau areas. Fig. 13 compares two simulations with and without the plateau feature. The test function is 2-D

$$F_6 = (x-1)^6 + (y-1)^6.$$

In this case, we can see one order of magnitude improvement in the required number of iterations.

In this example, all strategy parameters were set to the same values; with the plateau speedup feature, it takes 365 iterations to reach the goal, whereas without the plateau speedup feature, 2156 iterations are required.

C. Tests of the BC Strategy on Different Test Functions

The test functions are $F_7 = \sum_{i=1}^{n}(x_i - 1)^2$ (n-dimensional "sphere") and $F_8 = \sum_{i=1}^{n-1}(100 \cdot (x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$ (generalized Rosenbrock's function) the latter chosen particularly for the plateau problem. The start point of every test is the

point $x_{i,\text{start}} = 0, i = 1, \ldots, n$, the goal is in $x_{i,\text{goal}} = 1, i = 1, \ldots, n$. The final required precision is set to $\epsilon_{\text{end}} = 10^{-10}$ and the plateau speedup factor is set to $B = 1.1$. We test these functions in different dimensions $n = 5, 20$ in order to analyze how the number of steps for reaching the goal increases with respect to the dimension. The number of steps to convergence was averaged on 100 tests because of the stochastic processes contained in the model.

Testing the n-dimensional "sphere" in five dimensions, the bacterium converged after $12\,900 \pm 7700$ steps. The standard deviation represents 60% of the average. The parameters T_0, b, and τ_c were refined 200 times. In 20 dimensions, it needed $725\,000 \pm 383\,000$ steps to the goal and the standard deviation was 53% of the average. The strategy parameters listed above were refined 2000 times. Jumping from five to 20 dimensions, the number of steps to convergence increased by a factor of 56.2, i.e., for a four-fold increase in dimensions, the steps to converge increases 56 fold.

In the tests on Rosenbrock's function, the bacterium needed $238\,000 \pm 35\,000$ steps in five dimensions (with 2000 parameter changes) and $430\,000 \pm 221\,000$ steps in 20 dimensions (with 8000 parameter changes) to reach the goal. In the first case, the standard deviation represented 15% of the average and in the second it is 51%.

From these tests we can see that the stochastic processes contained in the model have a large influence on the results.

D. Dependency of Current on Previous Steps

As discussed in Section II, our BC optimization algorithm employs a dependency of the current step on the previous duration τ_{pr} [see (8)], whereas Dahlquist *et al.* [13] assume no dependency [see (9)]. To assess the differences between both algorithms, they are evaluated on two test functions $F_7 = \sum_{i=1}^{n}(x_i - 1)^2$ (n-dimensional "sphere") and $F_9 = \sum_{i=1}^{n}(x_i - 1)^4$, where $n = 3, 5$. For F_7, the plateau speedup feature is not used whereas for F_9 it is activated ($B = 1.2$). The starting point lies in $x_{i,\text{ start}} = 0, i = 1, \ldots, n$, the goal in $x_{i,\text{ goal}} = 1, i = 1, \ldots, n$ for both functions. Starting from the initial precision $\epsilon_{\text{init}} = 10^{-2}$, the strategy parameters may be either held constant ($T_0 = 2.6 \cdot 10^{-5}$, $b = 1.52 \cdot 10^3$, $\tau_c = 3.64 \cdot 10^3$) or refined 200 times until the final precision $\epsilon_{\text{end}} = 10^{-10}$ is reached. The number of iterations and the standard deviation obtained from 100 runs are listed in Table I.

Table I shows that the algorithm without dependency on previous steps needs fewer function evaluations than the algorithm with dependency for F_9 and for the five-dimensional (5-D) F_7 function with parameter adaptation. On the other hand, it takes longer to converge for the algorithm without dependency for the 5-D F_7 function without parameter refinement and for the 3-D F_7 function both with and without parameter adaptation. From these results, we cannot conclude that either algorithm is superior for the test problems examined here.

TABLE I
AVERAGE NUMBER OF FUNCTION EVALUATIONS TO REACH THE OPTIMIZATION GOAL FOR THE TWO ALGORITHMS WITH AND WITHOUT DEPENDENCY ON THE PREVIOUS STEP

Function	Dimension	Parameter refinement	Nr. of iterations Dependency (our algorithm)	Nr. of iterations No dependency (Dahlquist *et al.* [13])
F_7	5	yes	18700 ± 7200	11800 ± 3700
F_7	5	no	59700 ± 50000	94000 ± 78500
F_7	3	yes	20400 ± 36500	37300 ± 45600
F_7	3	no	37800 ± 34200	52500 ± 47200
F_9	5	yes	13100 ± 10600	8100 ± 3100
F_9	5	no	8000 ± 800	7700 ± 800
F_9	3	yes	4800 ± 1300	4100 ± 700
F_9	3	no	5700 ± 600	5500 ± 400

TABLE II
AVERAGE NUMBER OF FUNCTION EVALUATIONS TO REACH THE OPTIMIZATION GOAL FOR THE DIFFERENT METHODS

	Sphere 5-D	Sphere 20-D	Rosenbrock 5-D	Rosenbrock 20-D
L-BFGS	18	63	270	$3.0 \cdot 10^3$
DE	$8 \cdot 10^2$	$5 \cdot 10^3$	$4 \cdot 10^3$	10^5
CMA-ES	$7 \cdot 10^2$	$3 \cdot 10^3$	10^3	$2 \cdot 10^4$
ES	$8 \cdot 10^2$	$3 \cdot 10^3$	10^5	10^6
BC	$1.3 \cdot 10^4$	$7.3 \cdot 10^5$	$2.4 \cdot 10^5$	$4.3 \cdot 10^5$

V. COMPARISON WITH OTHER OPTIMIZATION STRATEGIES

In this section, we compare the proposed BC code (BC) with:

1) a quasi-Newton optimization algorithm (L-BFGS);
2) differential evolution (DE), described in [24], using strategy parameters that we optimized for the particular test functions, also see [25];
3) a simple ES with a global step size adaptation as described in [22];
4) an ES with CMA-ES using strategy parameters proposed in [22].

Since the quasi-Newton optimizer requires not only function evaluations but also its derivatives, we choose test functions where the gradient can be computed analytically. The two test functions are a simple sphere function in five and 20 dimensions

$$F_7 = \sum_{i=1}^{n}(x_i - 1)^2 \tag{25}$$

and Rosenbrock's function

$$F_8 = \sum_{i=1}^{n-1}\left(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right) \tag{26}$$

also in five and 20 dimensions.

Note that the start points and the precision ($\epsilon = 10^{-10}$) in the tests were the same for all the strategies. The average number of function evaluations of all the tests are summarized in Table II. For the L-BFGS, the table shows the sum of the number of function and gradient evaluations.

For all the test functions, the quasi-Newton method clearly outperforms the other stochastic techniques as we would expect.

The evolutionary computation methods rely on function information only, whereas the quasi-Newton method uses both function and gradient information. The additional use of gradients yields a higher convergence rate of the L-BFGS. However, this comes at the expense of requiring gradient information, which can be difficult to obtain in several realistic applications.

All the evolutionary computation methods (DE, ES, and CMA-ES) have similar convergence properties on the sphere function in five and 20 dimensions. BC performs worse than all the other methods for the sphere functions. The behavior of the bacterium on the sphere can also be interpreted in a qualitative way. The n-dimensional spherical test function presents specific difficulties to the bacterial strategy; in fact, the slope of this function changes from point to point, but the numerical approximation of the change of slope remains to a large extent constant on the whole domain. This implies that the correction presented in Section IV cannot give different values from point to point and so does not influence the step lengths. The plateau correction presented in Section IV-B3 cannot be applied either because the slope is not constant on the whole domain. In conclusion, both model improvements cannot be applied on this function and the bacterium moves with very small steps to the goal. For this specific problem, a modification of the BC needs to be performed.

On Rosenbrock's function in five dimensions, CMA and DE are faster than ES and BC. For the 20-dimensional case, CMA-ES is the fastest method followed by DE, BC, and finally ES.

Summarizing these results, BC performs worse than the evolutionary computation techniques for the sphere function for the reasons given but comparable to existing techniques for Rosenbrock's problem.

VI. Search of Global Minima

In this section, we present further extensions of the BC optimization strategy with respect to the search of global minima as well as results from test runs.

A. Strategies to Adapt the BC Strategy for the Search of Global Minima

From the previous sections, we saw that the bacterial chemotactic strategy can be valuable for the search of the global minimum on a multimodal function. Analyzing the results of the bacterium behavior on such functions, we define a strategy to perform a global minimum search into a given domain. The algorithms encompasses the following features.

1) Search the whole domain of the multimodal function with a significantly reduced precision and mark all the points visited by the bacterium. This can be interpreted as the deposition of a marker (or a "pheromone") by the bacterium [3], [18]. The points with the highest frequency of passages are identified by the concentration of this marker and are considered as "potential" global minima.
2) Analyze the concentration of the marker deposited over the whole domain. The neighborhood of the point with the highest concentration is considered as a candidate for the global minimum.
3) Start a local search with refined parameters from this candidate point and run until the needed final precision is reached. The final value is chosen as the global minimum of the function, reached by the algorithm.

This concept can be realized in terms of the following steps.

1) The global search on the whole domain is performed as described in Section IV. Note that the features for the plateau correction and for the automatic parameter adaptation reported in Section IV are not activated during this first search, as these are special improvements for the local search. The parameters are computed as described in Section III from a fixed precision ϵ_{global} (user-defined).
2) To avoid spending too much time or even being blocked in a local minimum without succeeding in jumping from one local minimum to another, the parameter b is increased (only one time) after the bacterium has found a subsequent number of values below a given precision ϵ_{jump}

$$\text{If } |f_{\text{pr}}| < \epsilon_{\text{jump}} \text{ for } n_{\text{jump}} \text{ subsequent times (user-defined)} \quad \text{then } b_{\text{jump}} = b_{\text{global}} \cdot C \text{ (only for the } n_{\text{jump}}\text{th step)} \tag{27}$$

where

ϵ_{jump}	user-defined precision;
$b_{\text{global}} = b_{\text{corr}}$	[see (23)];
C	user-defined constant.

This gives the bacterium the needed "push" for one step to jump out from a local minumum and to return to the global search.

3) At every point where it passes, the bacterium puts a marker, whose concentration is given by the exponential decaying function

$$c(x_i) = c_0 \cdot \exp\left[\frac{\sum_{i=1}^{n}(x_i - x_{i,\text{actual}})^2}{\sigma_{\text{marker}}^2}\right] \tag{28}$$

where

$c(x_i)$	n-dimensional concentration of the marker on the whole domain;
$c_0 = c_0(f_{\text{actual}}), x_{i,\text{actual}}$	actual position of the bacterium;
f_{actual}	function value in this point;
x_i	coordinates of a point in the n-dimensional space;
σ_{marker}	constant (for the decay control of the concentration function).

Note that c_0 is a function value of the stopping point of the bacterium, so that not only the frequency of passes in a local minimum but also the function value found there influences the choice of the "best" local minimum. To avoid spending time computing the values of the marker concentration far away from the position of the bacterium, where these are close to zero (to mark the local minima the exponential function must have a rapid decay), we define a radius r_{conc} within which this function must be

evaluated. The values are put on a grid (with a user-defined grid spacing) to decrease the computational time and find easily the maximal marker concentration.

4) The concentration of the marker deposited on the whole domain is analyzed. The point with the highest concentration represents the point with the highest frequency of passages. This point is assumed to be the nearest point to the global minimum.
5) Start a local search from this point as described in Section IV until the final precision ϵ_{end} is reached. The reached point is considered the global minimum of the function in the considered domain.

There are two disadvantages to this tactic. First, because of the stochastic processes involved, the frequency of passes could yield a minimum that is not the global minimum. This disadvantage can be circumvented by expressing the parameter c_0 as a function of the actual function value f_{actual} as follows. The lower the value of the function, the stronger the concentration trace, as this point would be more likely to be the global minimum. Second, it is not assured that the bacterium will pass from the global minimum at least one time. By increasing the number of steps for the global search on the whole domain, one can increase the possibility of getting into a region of the global minimum, but only at cost of an increase in computational time. Moreover, one has to find good values for ϵ_{jump} and C (contained in $b_{\text{jump}} = b_{\text{global}} \cdot C$), so that the bacterium visits as many local minima as possible on the given domain during the local search, but without spending much time there and without jumping far away from the domain (this can happen because of the multiplication of b_{jump} by C).

B. Comparison of the Global Properties With Other Optimization Strategies

In this section, we compare the global search properties of the proposed BC code with other stochastic optimization techniques as before: DE, a simple ES, and a CMA-ES. In addition, the performance of a pure random search (PRS) method is shown.

The two test functions are:

1) Rastrigin's function in two dimensions

$$F_4(x,y) = 20 + (x^2 - 10 \cdot \cos(2\pi x) + y^2 - 10 \cdot \cos(2\pi y)) \tag{29}$$

with a global minimum $F_4(0,0) = 0$;

2) the modified Rosenbrock function in two dimensions

$$\begin{aligned} F_{10}(x,y) =& 74 + 100(y - x^2)^2 + (1-x)^2 \\ & - 400 \exp\left(10(-\left((x+1)^2 + (y+1)^2\right)\right) \end{aligned} \tag{30}$$

with the global minimum $F_{10}(-0.909\,553\,7, -0.950\,571\,7) = 34.040\,242\,5$ and a local one $F_{10}(1,1) = 74$.

The optimization parameter ranges are $(x,y) \in [-600,600]$ for F_4 and $(x,y) \in [-2,2]$ for F_{10}. Within these domains, the start points were placed randomly and the termination criteria were set to $F_4(x,y) < 0.9$ and $F_{10}(x,y) < 74$ in order to check if the area of attraction of the global minimum was found. If the method failed to meet this criterion, it was stopped after 50 000 iterations.

TABLE III
SUCCESS RATES FOR THE DIFFERENT METHODS

	Rastrigin 2-D	Mod. Rosenbrock 2-D
DE	100%	52%
CMA-ES	33%	7%
ES	37%	10%
BC	10%	100%
PRS	0.1%	100%

The strategy parameters of the BC were given as follows. For the marker concentration function, it was $c_0(f_{\text{actual}}) = 1/(100 \cdot f_{\text{actual}} + 0.1)$, $\sigma^2_{\text{marker}} = 0.05$, $r_{\text{conc}} = 0.5$. The number of function evaluations for the global search was set to $5 \cdot 10^4$ and for the local search, $5 \cdot 10^3$. The parameters for the global search were chosen as $\epsilon_{\text{global}} = 10^{-1}$, $n_{\text{jump}} = 10$, $C = 1.1$; the parameters for the local search were $\epsilon_{\text{init}} = 10^{-2}$, $n_{\text{pc}} = 3$, $\epsilon_{\text{end}} = 10^{-6}$, $n_{\text{end}} = 5$. Based on an evaluation of all previous results, these strategy parameters are chosen since they fit a wide range of different optimization problems in terms of stability and convergence speed.

For the ESs, a population of two parents and ten offspring was chosen and the initial global step size is 10^{-2}. Other strategy parameters are defined in [22]. For DE, we optimized the strategy parameters for the individual test cases. Note that the choice of the parameters strongly influences the capabilities of the algorithms of finding the global optimum.

To compare the global properties of the various strategies, the success rate is measured. The success rate is the ratio of the number of successful trials in which the convergence criterion is met to the total number of trials. The success rates of the different optimization methods, summarized in Table III, are determined using 30 trials for the stochastic optimization strategies and 10^3 trials for the PRS.

As seen from Table III, the PRS technique performs very poorly on Rastrigin's function. The reason that the evolutionary methods find the global optimum with a higher success rate lies in the fact that they can ignore the superimposed sinusoidal noise on the quadratic function using appropriate step sizes. DE works perfectly on this function, whereas ES, CMA-ES, and BC find the global optimum less likely.

On the modified Rosenbrock function, the behavior is different. Here, the PRS technique succeeds in all trials. The number of maximum iterations was likely chosen too large. Also BC performs perfectly whereas the other evolutionary algorithms DE, ES, and CMA-ES are less likely to succeed.

It is noteworthy that the above values for the success rates may differ significantly when other strategy parameters are chosen or when the number of allowed iterations is modified. This analysis should give a rough idea about the limitations of the algorithms. The improvement of the strategy parameters for global optimization in the BC is a subject of ongoing investigation.

VII. Application of the BC Strategy to Inverse Design of Airfoils

This section contains a general description of optimizaton of airfoil profiles with inverse design and also the application of the bacteria strategy to this problem. In addition, it provides a further comparison of the BC with ESs, showing that BC can be a viable alternative.

A. Theory of the Airfoil Optimization Using the Inverse Design Method

The design cycle of airfoil profiles involves the calculation of the pressure distribution associated with the airfoil geometry. When the geometry of the airfoil is given, the surface pressure distribution is computed from the solution of the flow field. This computation that involves the implementation of a potential or Euler flow solver is referred to as *analysis* or *direct technique*. However, it is common practice to design the airfoil geometry based on a desired wall-pressure distribution. This inverse design process can be achieved by applying an inverse algorithm such as an adjoint procedure [26] to the governing flow equations or by iteratively modifying the geometry of the airfoil, using an optimization algorithm in order to achieve the desired pressure distribution. Inverse algorithms exhibit high convergence rates, but they may not be easily available for computation of certain flows or for configurations that involve combinations of empirical models and equations as is often the case in industrial applications. Here, we implement an iterative approach along with the chemotactic bacterial strategy. We carry out the inverse design procedure for the problem of potential flow past a nonlifting configuration. It should be emphasized that this flow model was selected for its simplicity and that the optimization algorithm is not inherently linked to the flow solver. The flow is solved using a panel-method approach with a distribution of constant-strength vortices along the profile [27]. The problem amounts to determining the strength of the vortex sheet in order to enforce the boundary condition of no-through flow. This is achieved by solving a well-posed system of linear equations for the panel strengths requiring that the normal velocity component vanishes at the centerpoint of the panel.

The parameters are variables defining the shape of the airfoil. The optimization problem requires the minimization of the measure of the distance between the achieved and desired pressure distribution. In our case, the objective function is defined by

$$f = \sum_{i=1}^{N_{\mathrm{p}}} \left(C_{\mathrm{p}}(x_i) - C_{\mathrm{p,goal}}(x_i)\right)^2$$

where

C_{p}	pressure coefficient;
$C_{\mathrm{p,goal}}$	target pressure coefficient;
x_i	location on the chordwise coordinate;
N_{p}	total number of panels.

For the optimization, our target pressure distribution corresponds to potential flow past wing sections of a standardized family of airfoil geometries, the so-called NACA four-digit airfoil series. The thickness distribution nondimensionalized by the chordlength can be expressed as a fraction of the maximum thickness t[28]. We introduced a sinusoidal term in the equation for the thickness distribution as follows:

$$\frac{y_t}{0.12} = \pm 5(a_1\sqrt{x} + a_2 x + a_3 x^2 + a_4 x^3 + a_5 x^4 + a_6 x(1-x)\sin(4\pi x));$$
$$x \leq 0.992\,404.$$

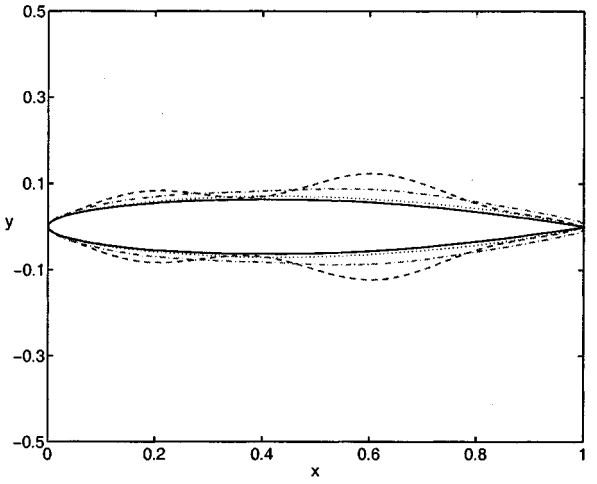

Fig. 14. Evolution of the airfoil profile after 0 (– – –), 200 (– · –), 1000 (· · ·), and 2400 (—) objective function evaluations for a single optimization run.

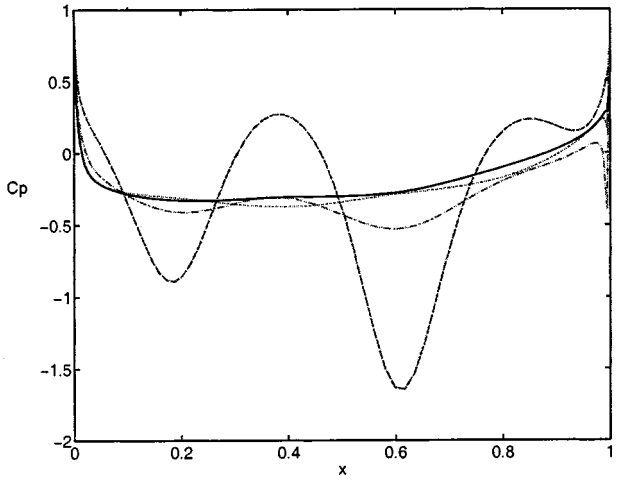

Fig. 15. Evolution of pressure distribution (C_{p}) on the profile after 0 (– – –), 200(– · –), 1000 (· · ·), and 2400 (—) objective function evaluations for a single optimization run.

B. Airfoil Optimization With the Bacterial Chemotaxis Algorithm

We apply the BC strategy on this six-dimensional problem, using the following parameters: initial precision $\epsilon_{\mathrm{init}} = 10^{-2}$, final precision $\epsilon_{\mathrm{end}} = 10^{-7}$, number of parameter changes $n_{\mathrm{pc}} = 5$, $A = 5 \cdot 10^5$, $B = 1.35$, $n_{\mathrm{plateau}} = 3$. Again, parameters are chosen based on our experience. For the initial parameters $a_1 = 0.2$, $a_2 = 0.2$, $a_3 = -0.2$, $a_4 = -0.2$, $a_5 = 0.0$, $a_6 = 0.2$, the objective function value is $f = 6.132\,73$. We stopped the simulation after 10^5 obtaining a final objective function value of $f = 0.136\,364$.

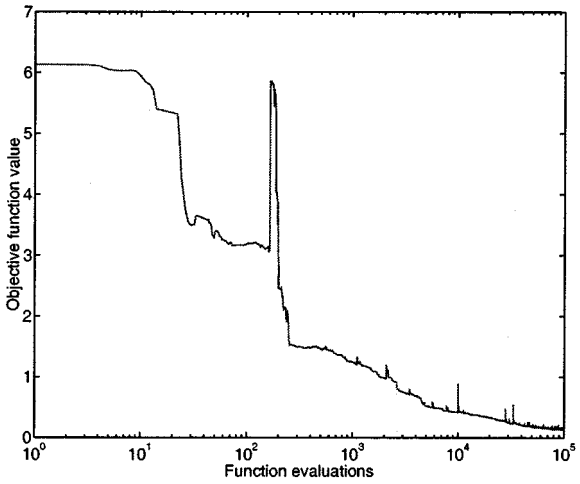

Fig. 16. Convergence of the BC for a single airfoil optimization run.

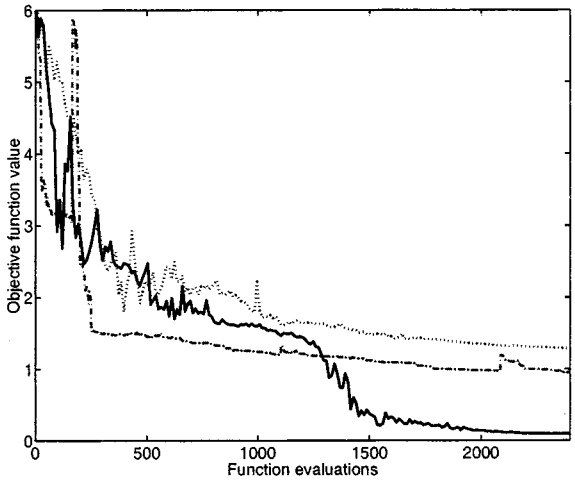

Fig. 17. Convergence plot of the airfoil problem. Comparison of the bacterial chemotaxis strategy $(- \cdot \; -)$ with a (3/3,12)-ES $(\cdots)$ and a (3/3,12)-CMA-ES (—).

Figs. 14 and 15 show, respectively, the evolution of the profile geometry and of the pressure distribution (given in terms of the pressure coefficient C_{p} on the profile). Fig. 16 shows the convergence behavior of the bacterial strategy on the airfoil problem (note the logarithmic scale).

Starting from the initial values, the BC algorithm converges to the coefficients $a_1 = 0.3097$, $a_2 = -0.1810$, $a_3 = -0.1622$, $a_4 = 0.0046$, $a_5 = 0.0289$, $a_6 = -0.0008$. The converged coefficients are different from the NACA0012 values that are $a_1 = 0.2969$, $a_2 = -0.1260$, $a_3 = -0.3516$, $a_4 = 0.2843$, $a_5 = -0.1015$, and $a_6 = 0.0$. However, different sets of parameters (representing the coefficients of a polynomial) may yield similar geometric shapes and pressure distributions of the airfoil, as it is the case here.

We compare this result with those given by a single optimization run using (3/3,12)-ES and a single run with (3/3,12)-CMA-ES (Fig. 17). After 2400 steps, BC reaches a value of f around 1, ES around 1.3, and CMA-ES around 0.1. At this point, the BC strategy is better than the ES, but worse than CMA-ES. This behavior does not change until 10^5 iterations when the optimization stops.

VIII. Conclusion

The purpose of this paper was to present a class of optimization algorithms based on a model of bacterial chemotaxis. Starting from a simple 2-D chemotaxis, we developed an n-dimensional model with additional features to improve its performance in situations that may cause a loss of computational time, e.g., the run on plateaus or the need of high precision. The improved optimization strategy, which we call BC strategy, is stochastic and relatively simple. The stochasticity gives an important feature not contained in other optimization strategies, particularly the possibility of finding the global optimum of a multimodal function, a common feature of many engineering problems.

The performance of the new optimization strategy is worse than that of ESs for quadratic functions but comparable for the Rosenbrock function. Also in case of the inverse design of airfoils, the bacteria strategy performs better than standard ESs and is comparable to ESs with improved convergence properties. In the future, strategy parameters for the additional features of the algorithm should be determined adaptively depending on the optimization problem. It would be interesting to add communication features into an optimization strategy based on BC as they may correspond to actual biological behavior of certain types of bacteria populations.

Acknowledgment

The authors would like to thank the anonymous reviewers and Dr. Fogel for their valuable comments.

References

[1] H.-P Schwefel, *Evolution and Optimum Seeking*. New York: Wiley, 1995.

[2] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. of Michigan Press, 1975.

[3] M. Dorigo, "Learning and Natural Algorithms," Ph.D. dissertation (in Italian), Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy, 1992.

[4] S. Obayashi, "Pareto genetic algorithm for aerodynamic design using the Navier-Stokes equations," in *Genetic Algorithms in Engineering and Computer Science*. New York: Wiley, 1997.

[5] A. Colorni, M. Dorigo, V. Maniezzo, and M. Trubian, "Ant system for job-shop scheduling," *Belg. J. Oper. Res. Stat. Comput. Sci.*, vol. 34, no. 1, pp. 39–53, 1994.

[6] H. J. Bremermann, "Chemotaxis and optimization," *J. Franklin Inst.*, vol. 297, pp. 397–404, 1974.

[7] H. J. Bremermann and R. W. Anderson, "How the brain adjusts synapses-maybe," in *Automated Reasoning: Essays in Honor of Woody Bledsoe*, R. S. Boyer, Ed. Norwell, MA: Kluwer, 1991, pp. 119–147.

[8] R. W. Anderson, "Biased random-walk learning: A neurobiological correlate to trial-and-error," in *Neural Networks and Pattern Recognition*, O. M. Omidvar and J. Dayhoff, Eds. New York: Academic, 1998, pp. 221–244.

[9] R. L. Barron, "Self-organizing and learning control systems," in *Cybernetic Problems in Bionics—Bionics Symposium, Dayton, May 1966*. New York: Gordon and Breach, 1968, pp. 147–203.

[10] ——, "Neuromine nets as the basis for predictive component of robot brains," in *Cybernetics, Artificial Intelligence, and Ecology—Fourth Annual Symposium American Society of Cybernetics*, H. W. Robinson and D. E. Knight, Eds. Washington, DC: Spartan, 1972, pp. 159–193.

[11] A. N. Mucciardi, "Adaptive flight control systems," in *Principles and Practise of Bionics—NATO AGARD Bionics Symp.*, Sept. 1968, pp. 119–167.

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

[13] F. W. Dahlquist, R. A. Elwell, and P. S. Lovely, "Studies of bacterial chemotaxis in defined concentration gradients—A model for chemotaxis toward l-serine," *J. Supramolecular Structure*, vol. 4, pp. 329(289)–342(302), 1976.
[14] J. Armitage and J. M. Lackie, Eds., *Biology of the Chemotactic Response—Forty-Sixth Symposium of the Society for General Microbiology Jointly Organised with the British Society for Cell Biology*. Cambridge, U.K.: Cambridge Univ. Press, 1990.
[15] D. E. Koshland Jr, *Bacterial Chemotaxis as a Model Behavioral System*. New York: Raven, 1980.
[16] J. M. Lackie and P. C. Wilkinson, *Biology of the Chemotactic Reponse*. Cambridge, U.K.: Cambridge Univ. Press, 1981.
[17] J. M. Lackie, *Cell Movement and Cell Behavior*. London, U.K.: Allen and Unwin, 1986.
[18] D. G. Davies, M. R. Parsek, J. P. Pearson, B. H. Iglewski, J. W. Costerton, and E. P. Greenberg, "The involvement of cell-to-cell signals in the development of a bacterial biofilm," *Science*, vol. 280, pp. 295–298, Apr. 1998.
[19] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence—From Natural to Artificial Systems*. Oxford, U.K.: Oxford Univ. Press, 1999, Santa Fe Inst. Studies in the Sciences of Complexity.
[20] S. Müller, S. Airaghi, J. Marchetto, and P. Koumoutsakos, "Optimization algorithms based on a model of bacterial chemotaxis," in *Proc. 6th Int. Conf. Simulation of Adaptive Behavior: From Animals to Animats,* , SAB 2000 Proc. Suppl., Sept. 2000, pp. 375–384.
[21] D. Whitley, K. Mathias, S. Rana, and J. Dzubera, "Building better test functions," in *Proceedings of the Sixth International Conference on Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann, 1995.
[22] N. Hansen and A. Ostermeier, "Convergence properties of evolution strategies with the derandomized covariance matrix adaptation: The $(\mu/\mu_I, \lambda)$-CMA-ES ," in *Proc. 5th Eur. Congr. Intelligent Techniques and Soft Computing*, 1997, pp. 650–654.
[23] I. Rechenberg, *Evolutionsstrategie '94*. Stuttgart, Germany: Fromann-Holzboog, 1994.
[24] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, pp. 341–359, 1997.
[25] R. Storn, "System design by constraint adaptation and differential evolution," *IEEE Trans. Evol. Comput.*, vol. 3, pp. 22–34, Apr. 1999.
[26] J. J. Reuther, A. Jameson, J. J. Alonso, M. L. Rimlinger, and D. Saunders, "Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, Part 2," *J. Aircraft*, vol. 36, no. 1, pp. 51–60, 1999.
[27] J. L. Hess, "Panel methods in computational fluid dynamics," *Annu. Rev. Fluid Mechan.*, vol. 22, pp. 255–274, 1990.
[28] I. H. Abbott and A. E. von Doenhoff, *Theory of Wing Sections*. New York: Dover, 1949.

Sibylle D. Müller received the Dipl. degree in mechanical engineering from the University of Stuttgart, Germany, in 1998. She is working toward the Ph.D. degree at the Institute of Computational Sciences, Swiss Federal Institute of Technology (ETH), Zürich, Switzerland.

Her graduate studies at the University of Wisconsin, Madison, from September 1997 to May 1998 were granted by the German Academic Exchange Service. Her current research interests include development of biologically inspired optimization algorithms and their application on various engineering problems.

Jarno Marchetto received the Dipl. degree in mechanical engineering from the Swiss Federal Institute of Technology (ETH), Zürich, Switzerland, in 2000.

He investigated optimization algorithms, in particular those based on bacterial chemotaxis, as a project at the Institute of Fluid Dynamics, ETH. Currently, he is a Software Engineer in the areas telecommunication and broadcasting technologies and broadband multimedia platforms with Research and Development, the Fantastic Corporation, Manno, Switzerland.

Stefano Airaghi received the Dipl. degree in mechanical engineering from the Swiss Federal Institute of Technology (ETH), Zürich, Switzerland, in 2000. He is currently working toward the Ph.D. degree at the Institute of Fluid Dynamics, ETH.

In his diploma thesis, he studied optimization algorithms based on bacteria chemotaxis among other biologically inspired algorithms.

Petros Koumoutsakos received the Dipl. degree in naval architecture and mechanical engineering from the National Technical University of Athens, Greece, in 1986, the Dipl. degree in naval architecture from the University of Michigan, Ann Arbor, in 1987, and the M.Sc. degree in aeronautics and the Ph.D. degree in aeronautics and applied mathematics from the California Institute of Technology, Pasadena, in 1988 and 1992, respectively.

From 1992 to 1994, he was a National Science Foundation Postdoctoral Fellow in parallel supercomputing with the California Institute of Technology. Since 1994, he has been a Senior Research Associate with the Center for Turbulence Research, NASA Ames/Stanford University. From September 1997 to June 2000, he was an Assistant Professor of Computational Fluid Dynamics, Swiss Federal Institute of Technology (ETH), Zürich, Switzerland, where he has been a Full Professor of Computational Sciences since July 2000. His current research interests are in the areas of particle methods, machine learning, and biologically inspired computation and the application of these techniques to problems of interest in the areas of engineering and life sciences.