Journal of Global Optimization **19:** 229–245, 2001.

On Convergence of a P-Algorithm Based on a Statistical Model of Continuously Differentiable Functions ⋆

JAMES M. CALVIN[1] and ANTANAS ŽILINSKAS[2]
[1]*Department of Computer and Information Science, New Jersey Institute of Technology, Newark, NJ 07102-1982, USA (e-mail: calvin@cis.njit.edu)* [2]*Institute of Mathematics and Informatics, Akademijos str. 4, Vilnius, LT2600, Lithuania*

Abstract. This paper is a study of the one-dimensional global optimization problem for continuously differentiable functions. We propose a variant of the so-called P-algorithm, originally proposed for a Wiener process model of an unknown objective function. The original algorithm has proven to be quite effective for global search, though it is not efficient for the local component of the optimization search if the objective function is smooth near the global minimizer. In this paper we construct a P-algorithm for a stochastic model of continuously differentiable functions, namely the once-integrated Wiener process. This process is continuously differentiable, but nowhere does it have a second derivative. We prove convergence properties of the algorithm.

Key words: Global optimization, Statistical models, Wiener process

1. Introduction

The first statistical model used for construction of global optimization algorithms was the Wiener process (Kushner, 1964; Archetti and Betrò, 1979; Žilinskas, 1985; Ritter, 1990; Locatelli and Schoen, 1995; Locatelli, 1997). This model may be considered as a worst case statistical model because its sample functions, although continuous, are nowhere differentiable with probability one. Many researchers regard the nondifferentiability of the model functions as a drawback. An advantage of the Wiener model is the simplicity of implementation of the so called P-algorithm which was introduced in Kushner (1964) and justified in Žilinskas (1985). The simplicity of the calculations stems from the Markov property of the Wiener process. Experimental testing has demonstrated the efficiency of the global search of the Wiener model-based P-algorithm (Törn and Žilinskas, 1989), but the local search is inefficient for smooth objective functions. The natural idea of extending the P-algorithm to new models was prevented by computational difficulties.

Recently the authors proposed an approximated version of the P-algorithm for a smooth function model, proved its convergence and gave an estimate of the con-

⋆ This project has been funded in part by the National Research Council of the USA under the Collaboration in Basic Science and Engineering Program. The contents of this publication do not necessarily reflect the views or policies of the NRC.

vergence rate (Calvin and Žilinskas, 2000). The latter model is a stationary process whose sample functions are twice continuously differentiable. To understand how much the smoothness of the sample functions of the model influence the features of the P-algorithm it is important to investigate a P-algorithm for an intermediate model of once-continuously differentiable functions.

A natural family of stochastic models for exploring the effect of smoothness on the performance of numerical algorithms is the family of r-times integrated Wiener processes, for $r \geqslant 0$. These processes induce a measure on $C^r([0,1])$, called the r-fold Wiener measure (Novak and Ritter, 1993). In this paper we consider only the case $r = 1$.

The sample functions of the integrated Wiener process are continuously differentiable but are nowhere twice differentiable. The advantage of this model is the Markov property of the vector process whose components are the function value and its derivative. Because of this feature we are able to construct a computationally tractable P-algorithm using derivatives of an objective function.

In Section 2 the P-algorithm for the integrated Wiener process is constructed along with the associated calculations. In Section 3 a simplified version is described. Convergence properties of this simplified algorithm are studied in Sections 4 and 5. Section 6 presents the results of some numerical experiments that complement our theoretical results. The conditional distribution of the integrated Wiener process is derived in the Appendix.

2. The P-Algorithm

In this section we begin by giving a rather general description of a P-algorithm; later we establish the particular form it takes for the stochastic model of this paper.

We suppose the objective function Y is to be minimized over the unit interval. For our setup we start with a priori information that Y is a member of some class of functions, for example Y might be continuous or continuously differentiable. The additional information needed to form an approximation is obtained by sequentially choosing points $\{t_n\} \subset [0, 1]$ at which to observe some information about Y. In this paper the information will be the function value and (sometimes) derivatives. We assume that the information is exact; that is, the observations are not corrupted by noise. We allow each new observation point to depend on all the previous information, so our general algorithm chooses the point t_{n+1} according to some rule

$$t_{n+1} = h_{n+1}\left(t_i, Y(t_i), Y'(t_i), i \leqslant n\right)$$

for some function h_{n+1}.

The method we will describe makes use of derivative information and so is suitable for differentiable objective functions. The simplified version introduced in the next section does not use derivative information and is thus suitable for general continuous functions.

Denote the ordered observations after the first n by $t_0^n < t_1^n < \cdots < t_n^n$, and the corresponding observed function and derivative values by $y_i^n = Y(t_i^n)$ and $x_i^n = Y'(t_i^n)$. Denote by M the global minimum of Y, and let M_n be the minimum of the conditional mean of the function given the first n observations. The general scheme of the P-algorithm is to choose the next point t_{n+1} to maximize the conditional probability that $Y(t_{n+1})$ is less than $M_n - \epsilon_n$ for some sequence of positive numbers ϵ_n, given the information of the first n observations. The outline of the algorithm is given by the following pseudocode.

P-algorithm(n,$\{\epsilon_n\}$, Y, X)

```
1   y_0 ← Y(0), x_0 ← X(0)
2   min ← y_0
3   for k ← 0 to n − 1 do
4       t_{k+1} ← argmax P(Y(t)<min −ε_{k+1}|x_i, y_i, i ⩽ k)
5       y_{k+1} ← Y(t_{k+1}), x_{k+1} ← Y'(t_{k+1})
6       if y_{k+1} < min then min ← y_{k+1}
7   return min
```

For this scheme to be practical, it must be feasible to calculate the conditional distributions for the particular stochastic model under consideration.

We will now specialize to the model of the integrated Wiener process. Let $X = \{X(t) : 0 \leqslant t \leqslant 1\}$ be a Wiener process, and for $t \in [0, 1]$, set

$$Y(t) = \int_{s=0}^{t} X(s)\, ds.$$

The measure induced on $C^1([0, 1])$ by Y is the one-fold Wiener measure, which we denote by P. Two sample functions of this process are depicted in Figure 1. Since the trajectories of X are continuous and nowhere differentiable with probability one, the paths of Y are continuously differentiable and nowhere twice differentiable. Let t^* be the (almost surely unique) point in $[0, 1]$ where Y attains its global minimum.

The process $Z(t) = (X(t), Y(t))$ is a Markov process. If we observe the value of X and Y at the points $t_1, t_2, \ldots, t_n$, then the distribution of Z at any point depends only on the values observed at the nearest observation points to the left and to the right of the point; i.e., for $t_{i-1}^n < t < t_i^n$,

$$\begin{aligned} &P\left(X(t) \leqslant x, Y(t) \leqslant y \mid X(t_j^n), Y(t_j^n), j \leqslant n\right) \\ &\quad = P\left(X(t) \leqslant x, Y(t) \leqslant y \mid X(t_j^n), Y(t_j^n), j = i-1, i\right). \end{aligned}$$

In the Appendix we work out the conditional distributions. Let us consider the conditional distribution of $(X(t), Y(t))$, for $t \in (t_{i-1}^n, t_i^n)$, given $X(t_{i-1}^n) = x_1$, $Y(t_{i-1}) =$

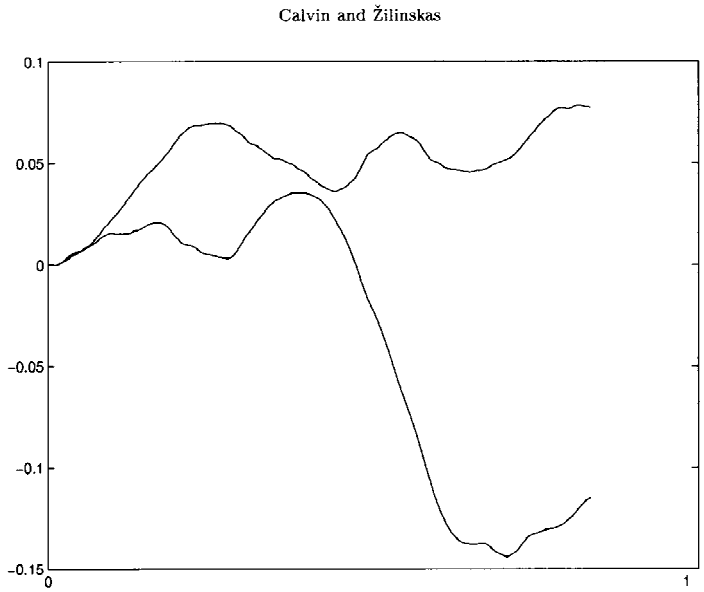

Figure 1. Sample paths of integrated Wiener process.

y_1, and $X(t_i) = x_2$, $Y(t_i) = y_2$. To simplify the expressions, make the substitutions $s = t - t_{i-1}^n$ and $T = t_i^n - t_{i-1}^n$. Setting

$$\begin{aligned}
\mu_x &= \frac{6(s^2 - sT)(y_1 - y_2) + 3Ts^2(x_1 + x_2) - 2sT^2x_2 - 4sT^2x_1 + T^3x_1}{T^3} \\
\mu_y &= \frac{s^2(2s - 3T)(y_1 - y_2) + s^3T(x_1 + x_2)}{T^3} \\
&\qquad - \frac{s^2T^2(2x_1 + x_2) - T^3sx_1 - T^3y_1}{T^3} \\
\sigma_x^2 &= \frac{s(T - s)}{T^3}(3s^2 - 3sT + T^2) \\
\sigma_y^2 &= \frac{s^3(T - s)^3}{3T^3} \\
\rho &= \sqrt{\frac{3(2s - T)^2}{4(3s^2 - 3sT + T^2)}},
\end{aligned}$$

we have that the conditional distribution of $X(t)$ is $N(\mu_x, \sigma_x^2)$, the conditional distribution of $Y(t)$ is $N(\mu_y, \sigma_y^2)$, and the correlation between $X(t)$ and $Y(t)$ is ρ.

We are ready to describe the P-algorithm for the integrated Wiener process. Let $\{\epsilon_n\}$ be a fixed sequence of positive numbers. The $(n + 1)$st point is chosen to maximize the probability

$$P\left(Y(t) < M_n - \epsilon_n \mid X(t_i^n), Y(t_i^n), i \leqslant n\right).$$

To determine how the next point is chosen according to this algorithm, we perform the calculations for an interval $[0, t]$, and choose the best point s in the interval. In light of the Markov property and time-homogeneity, it suffices to do the calculations for the interval $[0, t]$, with endpoint values $X(0) = x_1, Y(0) = y_1, X(t) = x_2, Y(t) = y_2$. Therefore, let us now suppose that we want to choose the point $s \in (0, t)$ to maximize

$$P(Y(s) < c \mid X(0) = x_1, Y(0) = y_1, X(t) = x_2, Y(t) = y_2) =$$
$$\Phi\left(-\frac{st(t-s)^2x_1 - s^2t(t-s)x_2 + (2s^3 - 3s^2t + t^3)y_1 + (3s^2t - 2s^3)y_2 - ct^3}{t^3\sqrt{s^3(t-s)^3/3t^3}}\right),$$

where $c = M_n - \epsilon_n$ and Φ is the normal cumulative distribution function. This is equivalent to choosing $s \in (0, t)$ to maximize

$$\frac{t^3\sqrt{s^3(t-s)^3/3t^3}}{st(t-s)^2x_1 - s^2t(t-s)x_2 + (2s^3 - 3s^2t + t^3)y_1 + (3s^2t - 2s^3)y_2 - ct^3}, \tag{2.1}$$

or, normalizing by the substitution $r = s/t$, choose $r \in (0, 1)$ to maximize

$$\frac{t^{3/2}\sqrt{r^3(1-r)^3}}{t[r(1-r)][(1-r)x_1 - rx_2] + (2r^3 - 3r^2 + 1)y_1 + (3r^2 - 2r^3)y_2 - c}. \tag{2.2}$$

Setting the derivative of this last expression to 0 implies that

$$\begin{aligned} p(r) &= t(x_1 - x_2)r^3 + (tx_2 - 2tx_1 + 3y_1 - 3y_2)r^2 \\ &\qquad + (6c + tx_1 - 6y_1)r - 3c + 3y_1 \\ &= 0. \end{aligned}$$

The maximizer r_0 can be found by comparing the value in (2.2) at the roots of p. (If the maximizer is not unique, choose the maximizer nearest the midpoint of the interval.) We call the maximum of (2.2) the *criterion value* of the interval $[0, t]$.

After these preliminary calculations we can precisely state the general algorithm. For each subinterval $[t_{i-1}^n, t_i^n]$, $i = 1, 2, \ldots, n$, find the r_0^i that maximizes (2.2) (with c replaced by $M_n - \epsilon_n$ and t replaced by $t_i^n - t_{i-1}^n$)), and the criterion value. Choose the interval with the largest criterion value, $[t_{j-1}^n, t_j^n]$, say, and set $t_{n+1} = t_{j-1}^n + r_0^j(t_j^n - t_{j-1}^n)$.

3. Simplified P-Algorithm

As $t \downarrow 0$, the factor of t in the denominator of (2.2) approaches 0. Ignoring that term the criterion becomes

$$\frac{t^{3/2}\sqrt{r^3(1-r)^3}}{(2r^3-3r^2+1)y_1+(3r^2-2r^3)y_2-M_n+\epsilon_n}. \tag{3.3}$$

In this section we study this simplified algorithm, which can be thought of as the original algorithm but ignoring the derivatives. To maximize this simplified criterion we solve for the roots of

$$3(y_1-y_2)r^2-6(y_1-M_n+\epsilon_n)r+3(y_1-M_n+\epsilon_n)=0.$$

The root in $(0, 1)$ is

$$\frac{1}{1+\sqrt{\frac{y_2-M_n+\epsilon_n}{y_1-M_n+\epsilon_n}}}.$$

Substituting this into the expression for the criterion yields

$$\frac{t^{3/2}}{((y_1-M_n+\epsilon_n)(y_2-M_n+\epsilon_n))^{1/4}\left(\sqrt{y_1-M_n+\epsilon_n}+\sqrt{y_2-M_n+\epsilon_n}\right)}.$$

Since we are only concerned with the order of the criterion values in choosing new subintervals, we can apply an increasing function to the criterion values. Raising the values to the power $2/3$ gives

$$\frac{t}{((y_1-M_n+\epsilon_n)(y_2-M_n+\epsilon_n))^{1/6}\left(\sqrt{y_1-M_n+\epsilon_n}+\sqrt{y_2-M_n+\epsilon_n}\right)^{2/3}}.$$

Thus the criterion that we maximize at each step of the algorithm is

$$\gamma_i^n=\frac{(t_i^n-t_{i-1}^n)\left(\sqrt{y_{i-1}^n-M_n+\epsilon_n}+\sqrt{y_i^n-M_n+\epsilon_n}\right)^{-2/3}}{(y_{i-1}^n-M_n+\epsilon_n)^{1/6}(y_i^n-M_n+\epsilon_n)^{1/6}}. \tag{3.4}$$

Let $\gamma^n=\max_{i\leqslant n}\gamma_i^n$. The simplified algorithm works by choosing the subinterval $[t_{i-1}^n, t_i^n]$ with the largest criterion value and then choosing the next observation at the point $t_{i-1}^n+\tau_n(t_i^n-t_{i-1}^n)$, where

$$\tau_n=\frac{1}{1+\sqrt{\frac{y_i^n-M_n+\epsilon_n}{y_{i-1}^n-M_n+\epsilon_n}}}.$$

While the criterion values are different, the formula for τ_n is the same as for the smooth function model in Calvin and Žilinskas (2000).

4. Convergence Rate

In this section we consider the convergence rate of the simplified algorithm with constant $\epsilon_n = \epsilon > 0$. From the definition of the criteria at (3.4),

$$\begin{aligned}
&\frac{\gamma_i^n}{t_i^n - t_{i-1}^n}\left(y_{i-1}^n - M_n + \epsilon\right)^{1/3}\left(y_i^n - M_n + \epsilon\right)^{1/3} \\
&= \frac{\left(\sqrt{y_{i-1}^n - M_n + \epsilon} + \sqrt{y_i^n - M_n + \epsilon}\right)^{-2/3}}{(y_{i-1}^n - M_n + \epsilon)^{1/6}(y_i^n - M_n + \epsilon)^{1/6}} \\
&\qquad \cdot \left(y_{i-1}^n - M_n + \epsilon\right)^{1/3}\left(y_i^n - M_n + \epsilon\right)^{1/3} \\
&\to 2^{-2/3}.
\end{aligned}$$

Therefore,

$$\lim_{n\to\infty} \sum_{i=1}^{n} \gamma_i^n = \int_{t=0}^{1} \frac{dt}{(2(Y(t) - M + \epsilon))^{2/3}}.$$

In the limit as $n \to \infty$, the γ_i^n do not change for intervals that do not get a new observation, and for the interval that gets the new observation the two γ values for the two subintervals are about $\gamma^n/2$. This is because $\tau_n \to 1/2$, and since the function values at the endpoints hardly change, the γ values for the two subintervals are about half that of the original interval. Therefore, the γ for a particular interval eventually becomes the maximum γ^n, and a new observation point is added approximately at the midpoint of this interval. This shows that the sequence of observation points is dense in [0, 1], and the method converges.

In order to establish the convergence rate, let γ_s^n be the criterion value for the subinterval containing the global minimizer, and let T_s^n denote the width of that interval, so that

$$\gamma_s^n = \frac{T_s^n\left(\sqrt{y_{i-1}^n - M_n + \epsilon} + \sqrt{y_i^n - M_n + \epsilon}\right)^{-2/3}}{(y_{i-1}^n - M_n + \epsilon)^{1/6}(y_i^n - M_n + \epsilon)^{1/6}}. \tag{4.5}$$

We can stop the search when γ_s^n is approximately equal to the average. More precisely, let n_k be the kth time that γ_s^n crosses the average of the γ_i^n's from below; see (Calvin, 1999) for a more detailed description. Then

$$\lim_{k\to\infty} n_k\, T_s^{n_k} = (2\epsilon)^{2/3} \int_{t=0}^{1} \frac{dt}{(2(Y(t) - M + \epsilon))^{2/3}}. \tag{4.6}$$

For arbitrary random variables $\{U_n\}, \{V_n\}$, we use the notation $U_n = O_P(V_n)$ to indicate that U_n/V_n is bounded in probability; that is, for any $\epsilon > 0$ there exists M_ϵ and N_ϵ such that

$$P\left(-M_\epsilon < U_n/V_n < M_\epsilon\right) > 1 - \epsilon$$

for $n \geqslant N_\epsilon$.

Let t_R^n and t_L^n be the nearest observation points to the right and left of t^*, respectively. By the mean value theorem,

$$Y(t^* + h) - Y(t^*) = X(t^* + \theta h) \cdot h$$

for some $\theta \in [0, 1]$. Since $\max_{0 \leqslant s \leqslant h} X(s) = O_P\left(h^{1/2}\right)$, this implies the bound

$$\begin{aligned} \Delta_{n_k} &= M_{n_k} - M \\ &\leqslant \min\{Y(t_R^n) - Y(t^*), Y(t_L^n) - Y(t^*)\} \\ &\leqslant T_s^{n_k} \max_{|s| \leqslant T_S^{n_k}} X(t^* + s) \\ &= O_P (T_s^{n_k})^{3/2}. \end{aligned}$$

Thus

$$n_k{}^{3/2} \Delta_{n_k} = O_P\left(n_k T_s^{n_k}\right)^{3/2} = O_P\left(\int_{t=0}^1 \left(1 + \frac{Y(t) - M}{\epsilon}\right)^{-2/3}\right)^{3/2} \tag{4.7}$$

by (4.6).

It is interesting to compare (4.7) with similar bounds for P-algorithms based on the Wiener process (Calvin, 1999) and based on a smooth Gaussian process model (Calvin and Žilinskas, 2000). If Y is a Wiener process, then

$$n_k{}^{1/2} \Delta_{n_k} = O_P\left(\int_{t=0}^1 \left(1 + \frac{Y(t) - M}{\epsilon}\right)^{-2}\right)^{1/2}.$$

If Y is twice continuously differentiable with $Y''(t^*) > 0$, then with the algorithm defined in Calvin and Žilinskas (2000),

$$n_k{}^2 \Delta_{n_k} = O_P\left(\int_{t=0}^1 \left(1 + \frac{Y(t) - M}{\epsilon}\right)^{-1/2}\right)^2.$$

(See the references for the exact definitions of the algorithms and the stopping times $\{n_k\}$.)

5. Decreasing Sequence ϵ_n

The basic convergence rate estimated in the previous section is the same (in terms of the number of observations n) as a simple uniform grid of observations (of course, by decreasing ϵ the constant factor in the convergence rate can be made arbitrarily small). Our goal in this section is to investigate the possibility of improving the order of convergence by replacing the fixed ϵ by a positive sequence of numbers ϵ_n converging to 0. We must determine a convergence rate that is slow enough to maintain the global convergence of the algorithm.

THEOREM 5.1. *Let* $\{\epsilon_n\}$ *be a sequence of positive numbers converging to* 0 *such that* $n^{3/2}\epsilon_n \to \infty$. *Then the algorithm described above will converge for any continuous objective function.*

Proof The algorithm converges for any continuous function if and only if the observation sequence is dense in the unit interval. We will construct a subsequence $\{n_k\}$ such that $\gamma^{n_k} \to 0$. It can be easily seen that the existence of such a subsequence implies the density of the observation sequence.

Let $\omega_n = \min_{i \leqslant n} t_i^n - t_{i-1}^n$ denote the length of the shortest interval formed by the observations 1 through n (so $\omega_n \leqslant 1/n$), and let n_k be the kth time that a new observation results in a new smallest interval; that is, at time n_k an interval of width, say, $\widetilde{\omega}_{n_k}$ is to be split, with its smallest child then having width ω_{n_k+1}. Let y_l and y_r denote the function values at the left and right endpoints, respectively (we suppress the index n_k). Using this notation, the maximum criterion at time n_k is

$$\gamma^{n_k} = \frac{\widetilde{\omega}_{n_k}\left(\sqrt{y_l - M_{n_k} + \epsilon_{n_k}} + \sqrt{y_r - M_{n_k} + \epsilon_{n_k}}\right)^{-2/3}}{(y_l - M_{n_k} + \epsilon_{n_k})^{1/6}(y_r - M_{n_k} + \epsilon_{n_k})^{1/6}}. \tag{5.8}$$

Let $\overline{y} = \max(y_r, y_l)$ and $\underline{y} = \min(y_r, y_l)$. By straightforward calculation,

$$\min\{\tau_{n_k}, 1 - \tau_{n_k}\} = \frac{1}{1 + \sqrt{\frac{\overline{y} - M_{n_k} + \epsilon_{n_k}}{\underline{y} - M_{n_k} + \epsilon_{n_k}}}}.$$

Since a new minimum interval is being formed,

$$\omega_{n_k+1} = \widetilde{\omega}_{n_k} \min\{\tau_{n_k}, 1 - \tau_{n_k}\} = \widetilde{\omega}_{n_k} \frac{1}{1 + \sqrt{\frac{\overline{y} - M_{n_k} + \epsilon_{n_k}}{\underline{y} - M_{n_k} + \epsilon_{n_k}}}} \leqslant \frac{1}{n_k}, \tag{5.9}$$

which yields the inequality

$$\widetilde{\omega}_{n_k} \leqslant \frac{1}{n_k}\left(1 + \sqrt{\frac{\overline{y} - M_{n_k} + \epsilon_{n_k}}{\underline{y} - M_{n_k} + \epsilon_{n_k}}}\right).$$

Substituting this into the expression for the criterion (5.8) gives

$$
\begin{aligned}
& \gamma^{n_k} \\
\leqslant\ & \frac{\frac{1}{n_k}\left(1+\sqrt{\frac{\overline{y}-M_{n_k}+\epsilon_{n_k}}{\underline{y}-M_{n_k}+\epsilon_{n_k}}}\right)\left(\sqrt{\underline{y}-M_{n_k}+\epsilon_{n_k}}+\sqrt{\overline{y}-M_{n_k}+\epsilon_{n_k}}\right)^{-2/3}}{(\underline{y}-M_{n_k}+\epsilon_{n_k})^{1/6}(\overline{y}-M_{n_k}+\epsilon_{n_k})^{1/6}} \\
=\ & \frac{\frac{1}{n_k}\left(\sqrt{\underline{y}-M_{n_k}+\epsilon_{n_k}}+\sqrt{\overline{y}-M_{n_k}+\epsilon_{n_k}}\right)^{1/3}}{(\underline{y}-M_{n_k}+\epsilon_{n_k})^{2/3}(\overline{y}-M_{n_k}+\epsilon_{n_k})^{1/6}} \\
=\ & \frac{1}{n_k}\left(\frac{\sqrt{\underline{y}-M_{n_k}+\epsilon_{n_k}}+\sqrt{\overline{y}-M_{n_k}+\epsilon_{n_k}}}{(\underline{y}-M_{n_k}+\epsilon_{n_k})^{2}(\overline{y}-M_{n_k}+\epsilon_{n_k})^{1/2}}\right)^{1/3} \\
=\ & \frac{1}{n_k}\left(\frac{1}{(\underline{y}-M_{n_k}+\epsilon_{n_k})^{3/2}(\overline{y}-M_{n_k}+\epsilon_{n_k})^{1/2}}\right. \\
& \left.+\frac{1}{(\underline{y}-M_{n_k}+\epsilon_{n_k})^{2}}\right)^{1/3} \\
\leqslant\ & \frac{1}{n_k}\left(\frac{2}{\epsilon_{n_k}^2}\right)^{1/3} \to 0
\end{aligned}
$$

if $n_k\epsilon_{n_k}^{2/3} \to \infty$. This completes the proof.

There are numerous technical difficulties in estimating the convergence rate for the algorithm with decreasing sequence $\{\epsilon_n\}$. In the next section we examine the efficiency of the approach with numerical examples.

6. Numerical Experiments

The integrated Wiener process is a well-suited statistical model for the construction of global optimization algorithms which involve derivatives in addition to objective function values. Theoretically the efficiency of the constructed algorithms may be compared with the efficiency of other algorithms by means of investigation of their convergence rate. However, from the practical point of view it is no less interesting to have some assessments for a moderate number of observations. Such assessments may be performed experimentally. The goal of our experimental testing was to evaluate the benefit of derivatives in construction of global optimization algorithms based on statistical models. It might be expected that derivative information would improve the efficiency of the search at least in the vicinity of local minima. The computational resources saved in this way would be used for the global search, thus improving the overall efficiency of the algorithm.

Two versions of the algorithm were considered: the basic version with derivatives (which we call the *exact* algorithm), and the simplified version without

derivatives (which we call the *approximate* algorithm). Versions of the P-algorithm with fixed and decreasing ϵ_n have been implemented for both models.

For the first experiment the sample functions of a discrete-time version of the process $Y(t)$ were used as the test functions. The values of $X(i/N)$, $i = 0, 1, \cdots, N$, $N = 1000$, were generated using independent Gaussian increments. The values of $Y(t)$ at the same points were obtained by means of numerical integration of X. The values of $X(t)$, $Y(t)$, $t_i \leqslant t < t_{i+1}$, $t_i = i/N$ were substituted by the conditional means of $X(t)$, $Y(t)$, given $(X(t_i), Y(t_i), X(t_{i+1}), Y(t_{i+1}))$; such a substitution is justified by the Markov property of $(X(t), Y(t))$.

One hundred randomly generated sample functions were minimized over the feasible interval $0 \leqslant t \leqslant 1$. Each function was minimized by both versions of the algorithm with constant value $\epsilon = 0.01$. The average error after k function (and derivatives for the exact algorithm) evaluations is presented in Table 1. The average value of the sample function minima was -0.2319.

Table 1.

n	10	20	30	40
Exact	0.0001	0.0000	0.0000	0.0000
Approx.	0.0004	0.0001	0.0001	0.0000

Since the global minimum is identified to reasonable precision after 20 to 40 function (and derivative) evaluations these functions may be considered easy to optimize. For this model the use of derivatives seems to give little benefit.

Let us now consider a different set of random test functions corresponding to the stationary Gaussian process $\xi(t)$ with correlation function $\rho(\tau) = \exp\{-(\tau/30)^2\}$; this process has mean 0 and unit variance. The sample functions were generated using an approximate spectral expansion

$$Y(t) = \sum_{k=1}^{20} (U_k \cos(30\,(\omega_k - 0.1)\cdot t) + V_k \sin(30\,(\omega_k - 0.1)\cdot t))\,, \qquad 0 \leqslant t \leqslant 1, \tag{6.10}$$

where $\omega_k = k/5$, U_k and V_k are independent Gaussian random variables with mean 0 and variance $D_k = 0.2 \exp\{-(\omega_k - 0.1)^2/4\}$. An example of a sample function is presented in Figure 2. One hundred functions generated according to (6.10) were minimized by means of both versions of the algorithm. The constant value $\varepsilon_n = 0.01$ was used. The sample mean and sample variance of the minima of the generated functions were -2.1344 and 0.2646, respectively. The average error after n function (for exact also derivative) evaluations is multiplied by 100 and presented in Table 2.

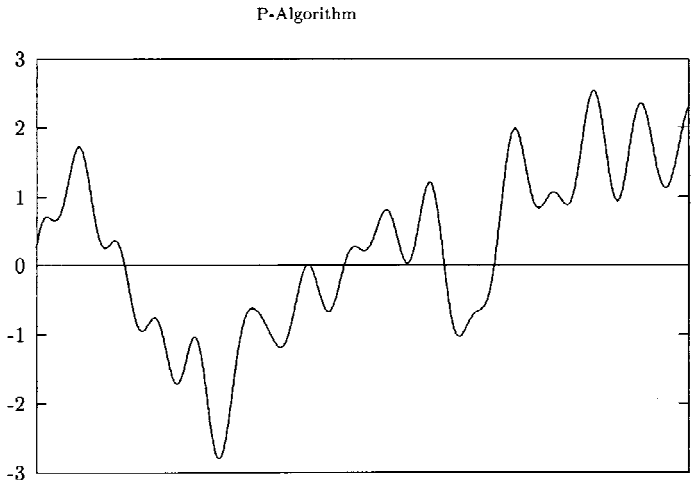

Figure 2. Representative sample path.

Table 2.

n	10	20	30	50	70	100	150	200
Exact	52.02	15.57	5.222	0.0193	0.0086	0.0031	0.0016	0.0007
Approx	50.35	15.44	2.445	0.2326	0.0100	0.0046	0.0018	0.0011

For these more complicated functions the global minimum with four decimal digits precision is found after 50–70 function (and derivative) evaluations. Also for these objective functions the use of derivatives does not seem helpful.

Common sense arguments recommend starting the search with the most global strategy. During the search the globality should be reduced to complete the search with a local refinement of the best found local minima. The effect of localization may be achieved by means of a decreasing sequence $\{\epsilon_n\}$ (Žilinskas, 1985). To evaluate the practical efficiency of such a decreasing sequence the following experiment was carried out. The same previous one-hundred random functions generated according to (6.10) were minimized with $\epsilon = 0.05/\sqrt{n}$. The average errors multiplied by 100 are presented in Table 3.

Table 3.

n	10	20	30	50	70	100	150	200
Exact	45.73	15.34	3.427	0.0109	0.0048	0.0016	0.0006	0.0003
Approx	55.54	14.92	4.209	0.0218	0.0056	0.0020	0.0007	0.0004

Comparing the results of Tables 2 and 3 we may conclude that the use of a decreasing sequence ϵ_n may more significantly improve the efficiency than the use of derivatives. The advantage of a decreasing sequence ϵ_n is more significant for the approximate model based algorithm than for the exact model based algorithm. The experiments were carried out with different constant ϵ and decreasing sequences ϵ_n. The best results are obtained choosing ϵ in the interval 0.5–5 per cent of the range of function values (which is generally unknown and must be estimated). The average error is almost constant with respect to changes of ϵ in such an interval. The empirically justified sequence $\epsilon_n = 5\epsilon/\sqrt{n}$ has a similar robustness property.

To illustrate the behavior of the algorithm on familiar test functions the following functions (Locatelli, 1997), (Törn and Žilinskas, 1989) were minimized:

$$f_1(x) = -\sum_{i=1}^{5} \sin((i+1)x + i), \ -10 \leqslant x \leqslant 10,$$

$$f_2(x) = (x + \sin(x))e^{-t^2}, \ -10 \leqslant x \leqslant 10.$$

Both functions represent specific difficulties discussed for example in Törn and Žilinskas (1989). The range of values of the first test function is 20, and choosing one per cent of this value, $\epsilon = 0.2$ was chosen for the version of the algorithm with fixed ϵ. According to the recommendations given above the sequence $\epsilon_n = \epsilon/\sqrt{n}$ was chosen. For the second function the range of function values was estimated as 1, implying $\epsilon = 0.01$ and $\epsilon_n = 0.05\epsilon/\sqrt{n}$ correspondingly. Minimization errors of both functions after n observations are presented in Tables 4 and 5 respectively. These results confirm that the use of derivatives in the algorithms considered in this paper does not seem prospective. A decreasing sequence $\{\epsilon_n\}$ normally grants better results than the constant ϵ.

Table 4.

n	10	20	30	40	50
Exact, fixed ϵ	2.7538	2.7638	0.0003	0.0003	0.0003
Exact, variable ϵ_n	4.4163	3.6931	1.0723	0.0003	0.0003
Approx., fixed ϵ	9.4007	2.5703	2.5703	0.2601	0.0002
Approx., variable ϵ_n	0.3709	0.2561	0.2561	0.0000	0.0000

An alternative to considering the error after a fixed number of function evaluations is to fix an error tolerance and then consider the number of function evaluations needed to attain the tolerance. We tested the same algorithms using this criterion with the same two test functions used previously as well as two new test functions, taken from Törn and Žilinskas, (1989), defined by

$$f_3(x) = \sin(x) + \sin(10x/3) + \log(x) - 0.84x + 3, \quad 2.7 \leqslant x \leqslant 7.5,$$

$$f_4(x) = \sin(x) + \sin(2x/3), \quad 3.1 \leqslant x \leqslant 20.4.$$

Table 5.

n	10	20	30	40	50
Exact, fixed ϵ	0.8242	0.8175	0.2196	0.0015	0.0000
Exact, variable ϵ_n	0.8241	0.2917	0.0000	0.0000	0.0000
Approx., fixed ϵ	0.8183	0.4850	0.0395	0.0000	0.0000
Approx., variable ϵ_n	0.8183	0.4268	0.0011	0.0011	0.0001

In Tables 6–9 we report the number of function evaluations needed to reach the given error tolerance for each of the test functions f_1–f_4. In each case we used $\epsilon = 0.01$ and $\epsilon_n = 0.05/\sqrt{n}$. Taken as a whole, the experiments seem to indicate that the exact algorithm performs a bit better than the approximate, and using a decreasing sequence $\{\epsilon_n\}$ is better than the fixed ϵ (though these conclusions do not hold in all examples).

Table 6. Number of evaluations to reach tolerance for f_1

Tolerance	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
Exact, fixed ϵ	25	29	34	81	81
Exact, variable ϵ_n	23	26	29	58	58
Approx., fixed ϵ	17	30	30	75	75
Approx., variable ϵ_n	15	17	17	47	47

Table 7. Number of evaluations to reach tolerance for f_2

Tolerance	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
Exact, fixed ϵ	25	28	28	47	47
Exact, variable ϵ_n	22	26	27	27	27
Approx., fixed ϵ	28	30	33	33	33
Approx., variable ϵ_n	24	26	41	41	90

Appendix

In this appendix we derive the conditional distribution of the integrated Wiener process Y given function and derivative values at a set of points.

Table 8. Number of evaluations to reach tolerance for f_3

Tolerance	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
Exact, fixed ϵ	11	11	11	11	11
Exact, variable ϵ_n	8	8	8	42	42
Approx., fixed ϵ	9	12	36	36	120
Approx., variable ϵ_n	9	11	18	18	82

Table 9. Number of evaluations to reach tolerance for f_4

Tolerance	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
Exact, fixed ϵ	13	13	20	20	105
Exact, variable ϵ_n	10	12	12	12	57
Approx., fixed ϵ	12	14	19	19	92
Approx., variable ϵ_n	10	11	16	16	69

Recall the definition of the Markov process $Z(t) = (X(t), Y(t)), 0 \leqslant t \leqslant 1$. Denote its (homogeneous) transition density by p_h:

$$p_h((x_1, y_1), (x_2, y_2)) = P(X(t+h) \in dx_2, Y(t+h) \in dy_2 \mid X(t) = x_1, Y(t) = y_1)/dx_2\, dy_2.$$

Given $X(t) = x_1, Y(t) = y_1$, the distribution of $X(t+h)$ is normal with mean x_1 and variance h. Conditional on $X(t+h) = x_2$,

$$\begin{aligned} Y(t+h) &= y_1 + \int_{s=0}^{h} X(t+s)\, ds \\ &= y_1 + \int_{s=0}^{h} \left(x_1 + \frac{s}{h}(x_2 - x_1) + B(s) \right) ds \\ &= y_1 + \frac{h}{2}(x_1 + x_2) + \int_{s=0}^{h} B(s)\, ds, \end{aligned}$$

where B is a Brownian bridge of duration h (that is, a Brownian motion conditioned to take the value 0 at time h). Now

$$\int_{s=0}^{h} B(s)\, ds$$

is normally distributed with mean 0 and variance $h^3/12$. Therefore

$$
\begin{aligned}
&p_h\left((x_1, y_1), (x_2, y_2)\right) \\
&= \frac{\sqrt{3}}{\pi h^2} \exp\left(-\frac{4h^2x_1^2 + 4h^2x_2^2 + 4h^2x_1x_2 + 12(y_2 - y_1)^2}{2h^3} \right. \\
&\qquad \left. + \frac{12h(x_1 + x_2)(y_2 - y_1)}{2h^3} \right).
\end{aligned}
$$

We will need the conditional density of $(X(s), Y(s))$, conditional on $X(0) = x_1, Y(0) = y_1$ and $X(t) = x_2, Y(t) = y_2$, which is given by

$$
\begin{aligned}
&\frac{p_s\left((x_1, y_1), (x, y)\right) p_{t-s}\left((x, y), (x_2, y_2)\right)}{p_t\left((x_1, y_1), (x_2, y_2)\right)} \\
&= \frac{\sqrt{3}t^2}{\pi s^2(t-s)^2} \exp\Bigg(\\
&-\frac{2t}{s(t-s)} \left(x - \frac{6(s^2 - st)(y_1 - y_2) + 3ts^2(x_1 + x_2) - 2st^2x_2 - 4st^2x_1 + t^3x_1}{t^3} \right)^2 \\
&+\frac{6t(t-2s)}{s^2(t-s)2} \left(x - \frac{6(s^2 - st)(y_1 - y_2) + 3ts^2(x_1 + x_2) - 2st^2x_2 - 4st^2x_1 + t^3x_1}{t^3} \right) \\
&\left(y - \frac{2s^3(y_1 - y_2) + s^3t(x_1 + x_2) - s^2t^2(2x_1 + x_2) - 3s^2t(y_1 - y_2) + t^3sx_1 + t^3y_1}{t^3} \right) \\
&-\frac{6t(3s^2 - 3st + t^2)}{s^3(t-s)^3} \\
&\cdot \left(y - \frac{s^2(2s - 3t)(y_1 - y_2) + s^3t(x_1 + x_2) - s^2t^2(2x_1 + x_2) + t^3sx_1 + t^3y_1}{t^3} \right)^2 \Bigg).
\end{aligned}
$$

Letting

$$
\begin{aligned}
\mu_x &= \frac{6(s^2 - st)(y_1 - y_2) + 3ts^2(x_1 + x_2) - 2st^2x_2 - 4st^2x_1 + t^3x_1}{t^3} \\
\mu_y &= \frac{s^2(2s - 3t)(y_1 - y_2) + s^3t(x_1 + x_2) - s^2t^2(2x_1 + x_2) + t^3sx_1 + t^3y_1}{t^3} \\
\sigma_x^2 &= \frac{s(t-s)}{t^3}(3s^2 - 3st + t^2) \\
\sigma_y^2 &= \frac{s^3(t-s)^3}{3t^3} \\
\rho &= \sqrt{\frac{3(2s - t)^2}{4(3s^2 - 3st + t^2)}},
\end{aligned}
$$

we have that conditional density of $X(s)$ is $N(\mu_x, \sigma_x^2)$ and the conditional density of $Y(s)$ is $N(\mu_y, \sigma_y^2)$, and the correlation coefficient is ρ. Thus

$$\operatorname{cov}(X, Y) = \frac{(2s - t)s^2(t - s)^2}{2t^3}.$$

Acknowledgements

The idea of constructing a P-algorithm for the integrated Wiener process model was suggested to us by F. Schoen.

The suggestions of two anonymous referees significantly improved the paper.

References

Archetti, F. and Betrò, B. (1979), A probabilistic algorithm for global optimization. *Calcolo*, 16, 335–343.

Calvin, J. (1999), Convergence rate of the P-algorithm for optimization of continuous functions. In Pardalos, P.M. (ed.), *Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems*, Kluwer Academic Publishers, Boston.

Calvin, J. and Žilinskas, A. (1999), On convergence of the P-algorithm for one dimensional global optimization of smooth functions. *Journal of Optimization Theory and Application* 102(3): 479–495.

Kushner, H. (1964), A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86: 97–106.

Locatelli, M. (1997), Bayesian algorithms for one-dimensional global optimization. *Journal of Global Optimization*, 10: 57–76.

Locatelli, M. and Schoen, F. (1995), An adaptive stochastic global optimization algorithm for one-dimensional functions. *Annals of Operations Research* 58: 263–278.

Novak, E. and Ritter, K. (1993), Some complexity results for zero finding for univariate functions. *Journal of Complexity*, 9: 15–40.

Ritter, K. (1990), Approximation and optimization on the Wiener space. *Journal of Complexity*, 6: 337–364.

Törn, A. and Žilinskas, A. (1989). Global Optimization. Springer-Verlag, Berlin.

Žilinskas, A. (1985), Axiomatic characterization of global optimization algorithm and investigation of its search strategy. *OR Letters* 4: 35–39.