Journal of Global Optimization **10:** 165–184, 1997.

A New Version of the Price's Algorithm for Global Optimization

P. BRACHETTI, M. DE FELICE CICCOLI, G. DI PILLO and S. LUCIDI
Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", via Buonarroti 12, 00185 Roma, Italy; and Gruppo Nazionale per l'Analisi Funzionale e le sue Applicazioni del Consiglio Nazionale delle Ricerche, via S. Marta, 13/A, 50139 Firenze, Italy.

(Received: 1 November 1994; accepted: 17 June 1996)

Abstract. We present an algorithm for finding a global minimum of a multimodal, multivariate function whose evaluation is very expensive, affected by noise and whose derivatives are not available. The proposed algorithm is a new version of the well known Price's algorithm and its distinguishing feature is that it tries to employ as much as possible the information about the objective function obtained at previous iterates. The algorithm has been tested on a large set of standard test problems and it has shown a satisfactory computational behaviour. The proposed algorithm has been used to solve efficiently some difficult optimization problems deriving from the study of eclipsing binary star light curves.

Mathematics Subject Classifications (1991). 65K99, 90C99.

Key words: global optimization, eclipsing binary stars.

1. Introduction

We consider the problem of finding a global solution of the unconstrained optimization problem:

$$\begin{array}{l} \min f(x) \\ x \in \mathbb{R}^n \end{array} \qquad (P)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuous function.

In the literature many algorithms have been proposed to solve unconstrained global optimization problems, see for example [1]–[10]. However, in this paper, we are interested to tackle the particular difficult case of Problem (P) in which:

(i) the evaluation of the objective function is very expensive;
(ii) the values of the objective function can be affected by the presence of noise;
(iii) the derivatives of the objective function are not available.

This class of global optimization problems is particularly important in industrial and scientific applications. In fact, in these applications, we often have to minimize complex functions whose values are determined by measurements made on some complex physical system or by a simulation procedure. In these cases the values of the objective function can be corrupted either by deterministic error (due to

modelling errors, truncation errors or discretization errors) or by stochastic noise (due to inaccurate measurements or rounding error). Moreover, even in the noiseless case, it can be very difficult or impossible to work out the analytical expressions of the partial derivatives of the objective function.

In fact, the motivation of the present work was to solve the global optimization problem deriving from the study of eclipsing binary stars based on light observed curves. The problem of estimating the parameters of the Wilson–Devinney model that describes the behavior of a system of eclipsing binary stars falls in the class described by Problem (P) with features (i), (ii) and (iii), with the additional difficulty that the dimension n is such that even a local minimization without derivatives can be considered a very difficult task.

In [11] a detailed analysis on global optimization methods is reported in order to single out the method which is the most suitable to tackle the particular global optimization problem mentioned before. According to the classification given in [5], the global minimization methods have been divided into three classes: Methods with Guaranteed Accuracy based on covering strategies, Direct Methods based only on local information, Indirect Methods in which the local information is used for building a global model of the level sets of the objective function.

The conclusions of the analysis performed in [11] can be synthesized as follows. Methods with Guaranteed Accuracy are not suitable due to the fact that they are based on estimation of the Lipschitz constant of the objective function or of its gradient and to the fact that their applicability needs that the objective function has limited derivatives; moreover, usually, these methods are efficient only when the function evaluations are not expensive. Indirect Methods have been considered not suitable because the techniques for approximating the level sets of the objective function can lead to large errors, owing to the complexity of the problem to be solved, the correlations among the parameters and the presence of noise. Therefore the selection of the method has been restricted to the second class.

Direct Methods have been, in turn, subdivided into Random Search Methods, Clustering Methods and Generalized Descent Methods. A common feature of these methods is that, in solving the global optimization problem, they tackle, at the same time, two distinct problems:

- the problem of examining all the region of interest in order to locate the subregions "more promising" to contain a global minimum point x^* (the *global search* problem);
- the problem of determining the global minimum x^* by using a local strategy as soon as a "sufficiently small" neighborhood of this point has been located (the *local search* problem).

The peculiarities (i), (ii) and (iii) of the considered global minimization problem restrict very much the choices of the strategies that can be used in the local search. In dealing with local minimization problems which present the difficulties (i), (ii) and (iii), the usually recommended method (see, e.g., [12]) and certainly the most used one (cf. [13]) is the simplex method of Nelder & Mead [14]. This suggests

that, also for solving global minimization problems with the same difficulties, it is convenient a method which, in the local search, draws its inspiration from the strategy of the simplex method. On the other hand, in [11], it has been also pointed out that, in the global search, a clustering strategy has the practical advantage of providing enough information on the features of the problem during the solution process. In fact, the output of the process, showing the evolutions of the clusters, contains a lot of supplementary information that it is very difficult to formalize, but of great importance in the solution of a real problem.

On the basis of the preceding considerations, in [11], the method of Price [15] has been recommended for solving the global optimization problem of the eclipsing binary stars. Then, a computer code for the Wilson–Devinney model has been implemented [16], where the underlying global optimization problem of parameter estimation is tackled by the Price's algorithm [15]. The results obtained by using the code are reported in [17] and [18]. The numerical experiences seem to indicate that the algorithm proposed by Price is efficient enough in the global search while it is not able to perform a sufficient fast local minimization to find the global minimizer when the algorithm has produced an estimate x^k "sufficiently good".

In this paper we describe a new version of the Price's algorithm in which the efficiency of the local search is improved without any significant increase in the number of function evaluations. To this aim, we have drawn our inspiration from the different behaviours of the local minimization algorithms. In fact, in the field of local minimization methods, when we pass from nonderivative methods to gradient-related methods, or from gradient-related methods to Newton-type methods, we get a significant improvement in the efficiency of the minimization process. This clearly points out that the more a method conveys information on the optimization problem (for example by using the information derived from the first or second order derivatives of the objective function) the more the method is effective in locating a local minimum point. Since we are interested to solve minimization problems in which no derivatives are available, the only way to improve the information on the optimization problem is to use the values of the objective function evaluated at the previous iterations. The preceding considerations lead us to define a new version of the Price's algorithm which, unlike the basic version, tries to exploit the information derived from the previous function evaluations in such a way as to improve the efficiency of the algorithm in the local search, without deteriorating the behaviour of the basic version in the global search.

In order to have an evidence of the efficiency of the improved version of the Price's algorithm we have compared it with the basic version on a set of standard test problems. The numerical results obtained on these "easy" test problems convinced us that new algorithm is quite more efficient than the basic one. Therefore we have applied the improved algorithm for solving the "difficult" global optimization problem of estimating the parameters of the model of eclipsing binary stars and, in

this way, we obtained a very significant reduction of the burdensome computing effort needed before by the basic Price's algorithm.

2. The Basic Price's Algorithm

In theory, any unconstrained global minimization algorithm should locate a point x^* such that

$$f(x^*) \leq f(x), \quad \text{for all} \quad x \in R^n.$$

However, for practical purposes, it is necessary to confine the search of the global minimum x^* within a prescribed bounded domain. Therefore, as usual in the field of unconstrained global optimization, we assume to know a compact set D which contains in its interior a global minimum point x^* of $f(x)$.

On the basis of the preceding considerations, the problem to solve becomes, in practice, to find an unconstrained global minimum of the following problem:

$$\begin{array}{l} \min f(x) \\ x \in D \end{array} \qquad (\tilde{P})$$

where D is a given compact set.

In [15] Price has developed an algorithm suitable for tackling the global optimization Problem $(\tilde{P})$, in the case that the derivatives of the objective function are not available. The algorithm of Price is described by the following steps.

THE PRICE'S ALGORITHM

Data. *A positive integer m such that $m \geq n+1$.*

Step 0. *Set $k = 0$; determine the initial set*

$$S^k = \{x_1^k, \ldots, x_m^k\},$$

where the points x_i^k, $i = 1, \ldots, m$ are chosen at random over D; evaluate f at each point x_i^k, $i = 1, \ldots, m$.

Step 1. *Determine the points x_{max}^k, x_{min}^k and the values f_{max}^k, f_{min}^k such that:*

$$\begin{aligned} f_{max}^k &= f(x_{max}^k) = \max_{x \in S^k} f(x) \\ f_{min}^k &= f(x_{min}^k) = \min_{x \in S^k} f(x). \end{aligned}$$

If the stopping criterion is satisfied, then stop.

Step 2. *Choose at random $n+1$ points $x_{i_0}^k, x_{i_1}^k, \ldots, x_{i_n}^k$ over S^k. Determine the centroid c^k of the n points $x_{i_1}^k, \ldots, x_{i_n}^k$ where*

$$c^k = \frac{1}{n} \sum_{j=1}^{n} x_{i_j}^k. \qquad (2.1)$$

Determine the trial point $\tilde{x}^k$ *given by*

$$\tilde{x}^k = c^k - (x^k_{i_0} - c^k). \tag{2.2}$$

If $\tilde{x}^k \notin D$ *go to Step 2; otherwise compute* $f(\tilde{x}^k)$.

Step 3. *If* $f(\tilde{x}^k) \geq f^k_{max}$ *then take*

$$S^{k+1} = S^k.$$

Set $k = k + 1$ *and go to Step 2.*

Step 4. *If* $f(\tilde{x}^k) < f(x^k_{max})$ *then take*

$$S^{k+1} = S^k \cup \{\tilde{x}^k\} - \{x^k_{max}\}$$

Set $k = k + 1$ *and go to Step 1.*

The stopping criterion used at Step 1 is $f^k_{max} - f^k_{min} < \epsilon$, where ϵ is a suitable small value; $\epsilon = 10^{-6}$ has been used in the numerical experiments reported here. As concerns the value of m, we have used $m = 25n$, as suggested by Price in [15].

The numerical results show that, as we said in the introduction, the Price's algorithm is efficient enough in exploring uniformly the region of interest and in identifying a neighbourhood of a global minimum point. However, it becomes slow in determining exactly the global minimum point starting from this neighbourhood.

3. The Improved Price's Algorithm

In order to overcome the inefficiency of the Price's algorithm we propose some improvements and extensions. More in particular, we propose a new algorithm that tries to gather the information about the objective function making better use of the values of the objective function already evaluated than in the basic Price's algorithm. This is done by means of the following three simple heuristic tools:

- the use of a weighted centroid;
- the use of a weighted reflection;
- the use of a quadratic model of the objective function.

3.1. THE USE OF A WEIGHTED CENTROID

Instead of the centroid c^k given by (2.1), the new algorithm makes use of the *weighted centroid* c^k_w defined by:

$$c^k_w = \sum_{j=1}^{n} w^k_j x^k_{i_j}, \tag{3.3}$$

where

$$w_j^k = \frac{\eta_j^k}{\sum_{j=1}^n \eta_j^k}, \tag{3.4}$$

$$\eta_j^k = \frac{1}{f(x_{i_j}^k) - f_{min}^k + \phi^k}, \tag{3.5}$$

$$f_{min}^k = \min_{x \in S^k} f(x). \tag{3.6}$$

The sequence $\{\phi^k\}$ is any sequence of positive numbers such that $\phi^k \gg f(x_{i_j}^k) - f_{min}^k, j = 1, \ldots, n$, at the initial iterations and such that $\phi^k \to 0$ for $k \to \infty$. A suitable choice is

$$\phi^k = \omega \frac{(f_{max}^k - f_{min}^k)^2}{f_{max}^0 - f_{min}^0}, \tag{3.7}$$

where ω is a positive constant sufficiently large (for instance $\omega = 10^3$), and $f_{max}^k = f(x_{max}^k)$. This choice was implemented in our numerical experiments reported in Section 4.

At the initial iterations the weighted centroid is practically the same as the centroid used in the Price's algorithm; in fact for such iterations we have $w_j^k \simeq \frac{1}{n}$. This guarantees that the new algorithm inherits the good behaviour of the Price's algorithm in the global search phase. When the number of iterations is sufficiently large and, hence, the values of ϕ^k are sufficiently small, the coefficients w_j^k weight more the points $x_{i_j}^k$ whose function values are close to f_{min}^k. The motivation of this choice is the following: when the number of iterations grows the points belonging to the set S^k are better approximations of local minimum points, so that the weighted centroid, given by (3.3), allows us to explore a region which is more promising in order to locate the global minimum point.

3.2. The use of a weighted reflection

Instead of the trial point $\tilde{x}^k$ given by (2.2), the new algorithm gives a point $\tilde{x}^k$ by using the following *weighted reflection*. Let

$$f_w^k = \sum_{j=1}^n w_j^k f(x_{i_j}^k), \tag{3.8}$$

where w_j^k is given by (3.4). Then we take

$$\tilde{x}^k = \begin{cases} c_w^k - \alpha^k (x_{i_0}^k - c_w^k), & \text{if} \quad f_w^k \le f(x_{i_0}^k); \\ x_{i_0}^k - \alpha^k (c_w^k - x_{i_0}^k), & \text{if} \quad f_w^k > f(x_{i_0}^k); \end{cases} \tag{3.9}$$

where

$$\alpha^k = \begin{cases} 1 - \dfrac{f(x_{i_0}^k) - f_w^k}{f_{max}^k - f_{min}^k + \psi^k}, \text{ if } \quad f_w^k \le f(x_{i_0}^k); \\ \\ 1 - \dfrac{f_w^k - f(x_{i_0}^k)}{f_{max}^k - f_{min}^k + \psi^k}, \text{ if } \quad f_w^k > f(x_{i_0}^k); \end{cases} \tag{3.10}$$

and the sequence $\{\psi^k\}$ is any sequence of positive numbers such that $\psi^k \gg f_{max}^k - f_{min}^k$, at the initial iterations and such that $\psi^k \to 0$ for $k \to \infty$; in particular in our numerical experience we chose for $\{\psi^k\}$ the same sequence $\{\phi^k\}$ given by means of (3.7).

Again, in producing the new point $\tilde{x}^k$ we exploit as much as possible the values of the objective function already computed. In particular we try also to evaluate the "goodness" of the centroid c_w^k. This should be done by computing the objective function in c_w^k, but in order to avoid this additional function evaluation we estimate this value by f_w^k, given by (3.8), which is a weighted mean of the values $f(x_{i_j}^k), j = 1, \ldots, m$, where the weights are the same used in the definition of the centroid c_w^k. By comparing the value f_w^k with the function value $f(x_{i_0}^k)$ we consider the direction $d^k = c_w^k - x_{i_0}^k$ a "descent" direction if $f_w^k \le f(x_{i_0}^k)$ and, in this case, the algorithm gives a new trial point along this direction. Otherwise, if $f_w^k > f(x_{i_0}^k)$, a new point is taken along the opposite direction $d^k = x_{i_0}^k - c_w^k$.

As regards the choice of steplength α^k along the direction d^k, this is determined by formula (3.10). This formula, using the sequence $\{\psi^k\}$, yields values $\alpha^k \approx 1$ at the initial iterations, so that for such iterations we nearly perform a simple reflection as it is done in the Price's algorithm. When the number of iterations increases, the same formula yields values α^k which are strictly smaller than one if the variation $|f_w^k - f(x_{i_0}^k)|$ is sufficiently large with respect to the difference $f_{max}^k - f_{min}^k$. In this way, when the point around which the reflection is performed can be considered a good estimation of the minimum point of the objective function, the new point produced by (3.8–3.10) is closer to the point around which the reflection is performed than the point that would be produced by the Price's algorithm.

3.3. THE USE OF A QUADRATIC MODEL OF THE OBJECTIVE FUNCTION

This tool can be put in action if the value chosen for the parameter m is such that $m \ge 2n + 1$. In this case it is possible to build a quadratic approximation of the objective function given by:

$$q(x) = \frac{1}{2} x' Q x + c' x + d,$$

where Q is an $n \times n$ diagonal matrix, such that $q(x)$ interpolates the values of the objective function computed at $2n + 1$ different points.

Let $m \geq 2n+1$. At each iteration we check if

$$f(\tilde{x}^k) < f^k_{min}. \tag{3.11}$$

In this case we select a subset S_{min} of $2n+1$ points of the set $S^k \cup \{\tilde{x}^k\} - \{x^k_{max}\}$ corresponding to the smallest values of the objective function. Then we determine Q, c and d by imposing that the quadratic function $q(x)$ interpolates these function values. If the diagonal matrix Q results to be positive definite we evaluate the minimizer $\tilde{x}^k_q = -Q^{-1}c$ and we check the condition

$$f(\tilde{x}^k_q) < \min_{x \in S_{min}} f(x). \tag{3.12}$$

If condition (3.12) holds the point $\tilde{x}^k_q$ will substitute in the new set S^{k+1} the point $\tilde{x}^k_{max}$ that gives the maximum of f in the set S_{min}, that is the point such that

$$f(\tilde{x}^k_{max}) = \max_{x \in S_{min}} f(x). \tag{3.13}$$

Otherwise we assume $S^{k+1} = S^k \cup \{\tilde{x}^k\} - \{x^k_{max}\}$.

3.4. THE IMPROVED ALGORITHM

On the basis of the points described above we can define the new algorithm.

THE IMPROVED ALGORITHM

Data. *A positive integer m such that $m \geq 2n+1$.*

Step 0. *Set $k = 0$; determine the initial set*

$$S^k = \{x^k_1, \ldots, x^k_m\},$$

where the points x^k_i, $i = 1, \ldots, m$ are chosen at random over D; evaluate f at each point x^k_i, $i = 1, \ldots, m$.

Step 1. *Determine the points x^k_{max}, x^k_{min} and the values f^k_{max}, f^k_{min} such that:*

$$f^k_{max} = f(x^k_{max}) = \max_{x \in S^k} f(x)$$
$$f^k_{min} = f(x^k_{min}) = \min_{x \in S^k} f(x).$$

If the stopping criterion is satisfied, then stop.

Step 2. *Choose at random $n+1$ points $x^k_{i_0}, x^k_{i_1}, \ldots, x^k_{i_n}$ over S^k. Determine the centroid c^k_w of the n points $x^k_{i_1}, \ldots, x^k_{i_n}$:*

$$c^k_w = \sum_{j=1}^{n} w^k_j x^k_{i_j}$$

where

$$w_j^k = \frac{\eta_j^k}{\sum_{j=1}^n \eta_j^k}$$

and η_j^k *are given by (3.5) and (3.7).*

Step 3. *Determine the trial point* $\tilde{x}^k$ *by performing a weighted reflection: let*

$$f_w^k = \sum_{j=1}^n w_j^k f(x_{i_j}^k);$$

then take

$$\tilde{x}^k = \begin{cases} c_w^k - \alpha^k(x_{i_0}^k - c_w^k), & \text{if} \quad f_w^k \le f(x_{i_0}^k); \\ x_{i_0}^k - \alpha^k(c_w^k - x_{i_0}^k), & \text{if} \quad f_w^k > f(x_{i_0}^k); \end{cases}$$

where α^k *is given by (3.10).*

If $\tilde{x}^k \notin D$ *go to Step 2; otherwise compute* $f(\tilde{x}^k)$.

Step 4. *If* $f(\tilde{x}^k) \ge f_{max}^k$ *then take*

$$S^{k+1} = S^k.$$

Set $k = k + 1$ *and go to Step 2.*

Step 5. *If* $f_{min}^k \le f(\tilde{x}^k) < f_{max}^k$ *then take*

$$S^{k+1} = S^k \cup \{\tilde{x}^k\} - \{x_{max}^k\},$$

set $k = k + 1$ *and go to Step 1.*

Step 6. *If* $f(\tilde{x}^k) < f_{min}^k$ *let*

$$\tilde{S} = S^k \cup \{\tilde{x}^k\} - \{x_{max}^k\}$$

and select the subset S_{min} *of* $2n + 1$ *points in* $\tilde{S}$ *corresponding to the smallest values of* f.

Determine the diagonal matrix Q, *the vector* c *and the scalar* d *such that*

$$f(x_i) = \frac{1}{2} x_i' Q x_i + c' x_i + d, \quad x_i \in S_{min}, \quad i = 1, \ldots, 2n + 1.$$

Step 7. *If the diagonal entries of* Q *are not all positive, then take*

$$S^{k+1} = \tilde{S},$$

set $k = k + 1$ *and go to Step 1.*

Step 8. *If Q is positive definite let*

$$\tilde{x}_q^k = -Q_c^{-1}.$$

If $\tilde{x}_q^k \notin D$ or $f(\tilde{x}_q^k) \geq f(\tilde{x}^k)$, then take

$$S^{k+1} = \tilde{S};$$

else take

$$S^{k+1} = \tilde{S} \cup \{\tilde{x}_q^k\} - \{\tilde{x}_{max}^k\},$$

where $\tilde{x}_{max}^k$ is defined by (3.13).

Set $k = k + 1$ and go to Step 1.

We note that, in the improved algorithm, the number of function evaluations required to determine a candidate to be the current estimate of the global optimizer is the same as in the basic algorithm. An additional objective function evaluation is performed only at Step 8 and, hence, only in a subset of the cases in which the current estimate is updated; so that we may expect that the number of additional function evaluations is small with respect to the total number.

4. Remarks on the Convergence of the Algorithm

Price's algorithm is called a *controlled random search method* because it follows a strategy which is a compromise between a pure random search strategy and a clustering strategy. As described in Section 2, an essential element of the Price's algorithm is the set S^k: initially, this set is constituted by m points chosen at random over D and then it collects the best points produced in the procedure. At each iteration a new trial point is produced along a direction which is randomly chosen over a finite number of vectors determined by the points belonging to S^k. The rationale behind the approach of the Price's algorithm is that, as the number of iterations increases, the set S^k should cluster round the global minimum points and the directions used should become more effective than directions chosen at random on R^n.

Since the number of points chosen at random over D is given by m and the directions used by the algorithm are not chosen at random over all R^n, it is not possible to state any convergence property of the Price's algorithm. In practice, the possibility of locating a global minimum point rests on the fact that the number of points randomly chosen at the initial step is not small and on the fact that global minimum points do not have narrow region of attraction. From this point of view, it appears clearly that the Price's algorithm, as described in Section 2, is a heuristic.

The modified version of the Price's algorithm proposed in this paper tries, as we said before, to improve the local search without affecting significantly the global

search, therefore it is still a heuristic. In this respect, we refer to section 7.2 of [5] where it is reported a thorough discussion on "why heuristics are necessary" and on how many are "heuristics elements in global optimization".

In any case, the basic and the improved algorithms could be easily modified in order to produce a sequence of points globally convergent in probability towards a global minimum point. In fact, it is sufficient to continue, once in a while, to evaluate points chosen at random over D (see, e.g., [3]). In this way it is possible to guarantee the global convergence of the algorithm even if the values of the objective function are affected by the presence of noise. The only requirement is that the noise is distributed with zero mean ([3]).

An example of such a modification is to replace Step 3 of the basic algorithm or Step 4 of the improved algorithm with the following new step:

Step N *If* $f(\tilde{x}^k) \geq f^k_{max}$ *then choose a point* $\hat{x}$ *at random over* D*:*

if $f(\hat{x}) < f^k_{max}$ *then take*

$$S^{k+1} = S^k \cup \{\hat{x}\} - \{x^k_{max}\},$$

set $k = k + 1$ *and go to Step 1;*

otherwise (if $f(\hat{x}) \geq f^k_{max}$*) take*

$$S^{k+1} = S^k,$$

set $k = k + 1$ *and go to Step 2.*

This new step, needed to guarantee convergence properties, requires an additional function evaluation at a point determined without using any information on the objective function. This task could be too expensive when the basic algorithm is used for solving a global minimization problem with feature (i). Since the algorithm proposed in this section should save function evaluations in the local search, it appears to be more viable to use Step N in the improved algorithm.

However, in this work we have implemented the modified version of the Price's algorithm without using Step N. This is due to the fact that our original aim was to tackle the global minimization problem of estimating the parameters in the Wilson–Devinney model. Since the previous numerical experiences (see, e.g., [17], [18]), performed on such problem, show that the Price's algorithm is able to cluster round the global minimum points and that the convergence is achieved even if after a very large number of function evaluations, in practice, the introduction of Step N appeared to be not necessary to enforce the global convergence in probability of the Price's algorithm and of its modifications, at least for the particular global optimization problem of main concern here.

Table I.

Functions	n	Basic Algorithm			Improved Algorithm			$\Delta_{nf}(\%)$
		n_f	n_p	f_{min}	n_f	n_p	f_{min}	
1	2	5554	3881	$.19E-3$	3837	1428	$.65E-8$	30.90
2	3	5159	3611	$.19E-2$	1648	634	$.19E-2$	68.05
3	3	5266	3239	$.66E-7$	3150	1632	$.22E-7$	40.18
4	4	5649	4118	$.17E-8$	3500	2318	$.47E-9$	38.04
5	4	16477	13004	$.41E-7$	5089	2417	$.32E-7$	69.11
6	2	1573	839	$-.10E+1$	722	229	$-.10E+1$	54.10
7	2	2181	1244	$.21E-7$	903	261	$.58E-11$	58.59
”	4	8088	5066	$.11E-6$	2374	688	$.77E-10$	70.64
”	6	16017	10100	$.14E-6$	3921	1092	$.38E-9$	75.51
”	8	26718	16992	$.26E-6$	5427	1440	$.42E-9$	79.68
”	10	36838	23004	$.36E-6$	7081	1815	$.18E-9$	80.77
8	2	1890	1079	$.23E-7$	800	263	$.22E-8$	57.67
”	4	7172	4461	$.59E-7$	2195	642	$.22E-8$	69.39
”	6	14375	8939	$.23E-6$	3790	1082	$.22E-9$	71.56
”	8	23583	14535	$.26E-6$	5191	1331	$.17E-8$	77.98
”	10	32402	19649	$.35E-6$	7037	1826	$.97E-10$	78.54
9($m=5$)	4	7567	4187	-10.05	5403	2841	-10.05	28.59
9($m=7$)	4	7492	4114	-10.06	5386	2837	-10.06	28.10
9($m=10$)	4	7526	4157	-10.07	5862	3235	-10.07	22.11
10	3	2936	1533	-3.86	1014	250	-3.86	62.05
”	6	14071	8946	-3.32	4154	1432	-3.32	70.47
11	2	2148	1202	3.00	936	279	3.00	56.42
12	2	1239	659	-1.00	586	162	-1.00	52.70
”	4	4675	2580	-1.00	1655	754	-1.00	64.59
13	2	1783	1036	$-.20$	723	225	$-.20$	59.45
”	4	6720	4164	$-.40$	2327	826	$-.40$	65.37
14	2	1745	915	-95.28	710	164	-95.28	59.31
15($m=4$)	1	530	299	15.28	236	89	15.28	55.47
15($m=10$)	1	523	285	44.95	203	61	44.95	61.18
15($m=25$)	1	790	420	261.81	332	113	261.78	57.97

5. Numerical Results

The improved algorithm has been experimented on a large set of test functions, taken from the literature, with n ranging from 2 to 10. In all cases D is a hypercube. We report in the Appendix the description of all test functions and the corresponding hypercubes.

For all test functions a comparison has been made with the basic Price's algorithm. Both algorithms have been stopped whenever

$$f^k_{max} - f^k_{min} \leq 10^{-6}.$$

In table 1 we report the results of the comparison between the basic and the improved Price's algorithm. In the table:

- n_f is the number of function evaluations;
- n_p is the number of function evaluations at the trial points $\tilde{x}^k$ that have not given a value $f(\tilde{x}^k)$ smaller than f^k_{max}, and hence that have been discarded;
- f_{min} is the value of f^k_{min} when the stop occurs;
- Δ_{nf} is the ratio between the difference of the values of n_f in the two algorithms and the largest value of n_f.

From Table I we point out that the new algorithm performs better with respect to all test problems. In particular both n_f and n_p are smaller; in 14 cases f_{min} is smaller and in the other cases f_{min} is equal to the one obtained by Price's algorithm. Therefore we have that the new algorithm requires a significantly smaller number of function evaluations and, in many cases, yields a smaller value of the objective function.

From the analysis of the behaviour of the new algorithm, we remark that the use of a weighted centroid and of a weighted reflection is most effective in reducing the number of function evaluations, while the use of a quadratic model of the objective function is effective mainly in achieving a better accuracy of the estimated minimum value f_{min}.

6. Application to the Parameter Extraction of Close Eclipsing Binary Stars

As we already said, the new algorithm has been motivated by the problem of determining the principal geometrical and physical parameters characterizing a binary star system, from photometric observations at different wavelengths of their eclipses. The determination of these parameters is of main interest in astrophysics, since the behaviour of binary systems is analysed in order to validate the theories on stellar evolution.

The physical model of the light curve generation in a binary system has been developed by Wilson and Devinney in 1971 [19], and it is represented by a rather complicated function of 18 variables that characterize the system. Among these, 3 variables are data of the model, and 7 are dependent on data and on other variables; the remaining 8 variables are independent, and have to be determined by comparing the light curves generated by the model with the observed light curves, at different wavelengths.

Let us denote by l^0_{ij} the $i-th$ observation on the light curve at the wavelength j, and by $l^c_{ij}(x)$ the light intensity predicted by the model, where x denotes the vector of independent variables. Then the parameter extraction problem consists in determining a global minimizer of the weighted square error function:

$$S(x) = \sum_{j=1}^{r} \frac{1}{s_j} \sum_{i=1}^{s_j} p_{ij}(l^0_{ij} - l^c_{ij}(x))^2$$

Table II.

Binary system	n	Basic Alg.		Improved Alg.	
		n_f	f_{min}	n_f	f_{min}
AT CAM $(q > 1)$	8	37944	.0001359	10897	.0001358
AT CAM $(q < 1)$	8	26100	.0001351	12432	.0001350
V677	8	36900	.0027367	15544	.0027365
A0 CAM	8	32400	.002962	10980	.002962

where r is the number of light curves, s_j is the number of observations on the $j-th$ light curve, and p_{ij} is a weight associated with the error.

The features of $S(x)$ are that:

(i) its evaluation is very expensive;

(ii) the values of the objective function are affected by measurement noise;

(iii) its derivatives are not available.

In particular, as regards the first point, we notice that the evaluation of the function $S(x)$ requires about $30''$ of CPU time of an IBM RISC System 6000/520. Thus the reduction in the number of function evaluations is a crucial step to solve efficiently the problem.

In Table II we report the numerical results obtained by the two algorithms on the data of four binary star systems.

The results reported in Table II show that the improved version of Price's algorithm allows us to decrease drastically the number of function evaluations and, in this way allows us reduce the computation time.

As an example, in the case of the binary star system AT CAM $(q > 1)$, the overall CPU time of an IBM RISC System 6000/520 is reduced from about 312 hours to about 91 hours, that is from more than 13 days to less than 4 days. Finally, in Figure 1 we show, in the case of the binary stars AT CAM $(q > 1)$, how f^k_{min} decreases with the number of function evaluations, when this number is larger than 5000. This figure confirms that the improved algorithm performs like the basic algorithm at the initial iterations, corresponding to the global search phase, and is much faster in the final iterations, corresponding to the local search phase. Similar behaviours are observed for all binary stars of Table II.

Appendix: Test Functions

1. *Extended Rosenbrock* [20]

$$f(x) = \sum_{i=1}^{n-1} \{(x_i - 1)^2 + 100(x_i^2 - x_{i+1})^2\},$$

$$x^* = [1, 1, \ldots, 1]^T \quad f(x^*) = 0.$$

The region of interest is $-1000 \leq x_i \leq 1000 \quad i = 1, \ldots, n.$

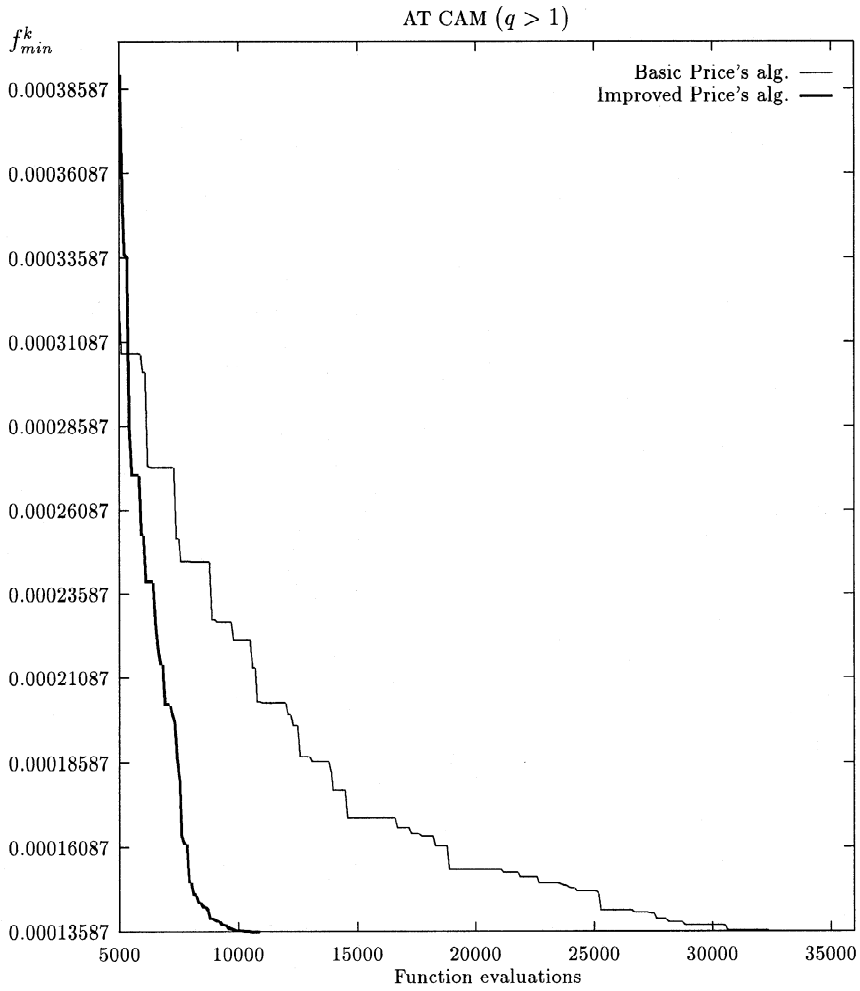

Figure 1. Comparison between the basic and the improved Price's algorithm on AT CAM $(q > 1)$ data

2. *Meyer and Roth* [20]

$$f(x_1, x_2, x_3) = \sum_{i=1}^{m} \{Y_i(t; v; x) - y_i\}^2,$$

in which

$$Y_i(t; v; x) = \frac{x_1 x_3 t_i}{(1 + x_1 t_i + x_2 v_i)},$$

and the t_i, v_i and the y_i are given in the following table.

i	t_i	v_i	y_i
1	1.0	1.0	0.126
2	2.0	1.0	0.219
3	1.0	2.0	0.076
4	2.0	2.0	0.126
5	0.1	0.0	0.186

For this problem $m = 5$ and $n = 3$

$x^* = [3.13, 15.16, 0.78]^T \quad f(x^*) = 0.4 \times 10^{-4}$.

The region of interest is $-10 \leq x_i \leq 10 \quad i = 1, 2, 3$.

3. *Fletcher and Powell* [20]

$$f(x_1, x_2, x_3) = 100\{(x_3 - 10\theta)^2 + (r - 1)^2\} + x_3^2,$$

where

$$r = |(x_1^2 + x_2^2)^{1/2}|$$

and

$$\theta = \begin{cases} \dfrac{1}{2\pi} \tan^{-1} \dfrac{x_2}{x_1}, & (x_1 > 0) \\ \dfrac{1}{2\pi} \tan^{-1} \dfrac{x_2}{x_1} + \dfrac{1}{2}, & (x_1 < 0) \end{cases}$$

$x^* = [1, 0, 0]^T \quad f(x^*) = 0$.

The region of interest is $-10 \leq x_i \leq 10 \quad i = 1, 2, 3$.

4. *Miele and Cantrell* [20]

$$f(x_1, x_2) = (\exp(x_1) - x_2)^4 + 100(x_2 - x_3)^6 + \{\tan(x_3 - x_4)\}^4 + x_1^8,$$

$x^* = [0, 1, 1, 1]^T \quad f(x^*) = 0$.

The region of interest is $-10 \leq x_i \leq 10 \quad i = 1, 2, 3, 4$.

5. *Wood's function, quoted by Colville* [20]

$$\begin{aligned} f(x) \;=\; & 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 \\ & + 10.1\{(x_2 - 1)^2 + (x_4 - 1)^2\} + 19.8(x_2 - 1)(x_4 - 1), \end{aligned}$$

$x^* = [1, 1, 1, 1]^T \quad f(x^*) = 0$.

The region of interest is $-10 \leq x_i \leq 10 \quad i = 1, 2, 3, 4$.

6. *Six hump camel back function* [2]

$$f(x_1, x_2) = (4 - 2.1x_1^2 + x_1^4/3)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2,$$

$f^* \simeq -1.0316285$.

The region of interest is $-2.5 \leq x_1 \leq 2.5 \quad -1.5 \leq x_2 \leq 1.5$.

This function exhibits six local minimizers, two of which are also global.

7. 10^n *local minima* [21]

$$\begin{aligned} f(x) \;=\; & (\pi/n)\{10\sin^2(\pi x_1) + \sum_{i=1}^{n-1}[(x_i - 1)^2(1 + 10\sin^2(\pi x_{i+1}))] \\ & + (x_n - 1)^2\}, \end{aligned}$$

$f^* = 0$.

The region of interest is $-10 \leq x_i \leq 10 \quad i = 1, \ldots, n$.

This function has roughly 10^n local minimizers and a unique global minimizer located at $x_i^* = 1, \quad i = 1, \ldots, n$.

8. 15^n *local minima* [21]

$$f(x) = (1/10)\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}[(x_i - 1)^2(1 + 10\sin^2(3\pi x_{i+1}))]\}$$
$$+(1/10)(x_n - 1)^2[1 + \sin^2(2\pi x_n)],$$
$$f^* = 0.$$

The region of interest is $-10 \leq x_i \leq 10 \quad i = 1, \ldots, n$.

This function has roughly 15^n local minimizers and a unique global minimizer located at $x_i^* = 1, \quad i = 1, \ldots, n$.

9. *Shekel's family* [2]

$$f(x) = -\sum_{i=1}^{m} \frac{1}{(x - a_i)^T(x - a_i) + c_i}.$$

We studied this function with $m = 5, m = 7, m = 10$ and $n = 4$.

The values of $a_i = (a_{i1}, \ldots, a_{in})^T$ and $c_i > 0$ (for $i = 1, \ldots, m$) are given in the following table:

i	a_{i1}	a_{i2}	a_{i3}	a_{i4}	c_i
1	4.	4.	4.	4.	.1
2	1.	1.	1.	1.	.2
3	8.	8.	8.	8.	.2
4	6.	6.	6.	6.	.4
5	3.	7.	3.	7.	.4
6	2.	9.	2.	9.	.6
7	5.	5.	3.	3.	.3
8	8.	1.	8.	1.	.7
9	6.	2.	6.	2.	.5
10	7.	3.6	7.	3.6	.5

The region of interest is $0 \leq x_j \leq 10 \quad j = 1, \ldots, n$.

This function has m minima in positions a_i with levels c_i.

10. *Hartman's family* [2]

$$f(x) = -\sum_{i=1}^{m} c_i \exp\left(-\sum_{j=1}^{n} a_{ij}(x_j - p_{ij})^2\right).$$

We studied this function with $m = 4, n = 3$ and $n = 6$.

The values of $a_i = (a_{i1}, \ldots, a_{in})^T, p_i = (p_{i1}, \ldots, p_{in})^T$ and $c_i > 0$ (for $i = 1, \ldots, n$) are given in the following tables:

i	a_{i1}	a_{i2}	a_{i3}	c_i	p_{i1}	p_{i2}	p_{i3}
1	3.	10.	30.	1.	.3689	.1170	.2673
2	.1	10.	35.	1.2	.4699	.4387	.7470
3	3.	10.	30.	3.	.1091	.8732	.5547
4	.1	10.	35.	3.2	.03815	.5743	.8828

i	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	a_{i6}	c_i
1	10.	3.	17.	3.5	1.7	8.	1.
2	.05	10.	17.	.1	8.	14.	1.2
3	3.	3.5	1.7	10.	17.	8.	3.
4	17.	8.	.05	10.	.1	14.	3.2

i	p_{i1}	p_{i2}	p_{i3}	p_{i4}	p_{i5}	p_{i6}
1	.1312	.1696	.5569	.0124	.8283	.5886
2	.2329	.4135	.8307	.3736	.1004	.9991
3	.2348	.1451	.3522	.2883	.3047	.6650
4	.4047	.8828	.8732	.5743	.1091	.0381

The region of interest is $0 \le x_j \le 1 \quad j = 1, \ldots, n$.

This function has m minima in positions p_i with levels c_i.

11. Goldstein and Price [2]

$$f(x) = [1+(x_1+x_2+1)^2(19 - 14x_1+3x_1^2 - 14x_2+6x_1x_2+3x_2^2)] \\ [30+(2x_1 - 3x_2)^2(18 - 32x_1+12x_1^2+48x_2 - 36x_1x_2+27x_2^2)],$$

$$x^* = [0, -1]^T \quad f(x^*) = 3.$$

The region of interest is $-2 \le x_j \le 2 \quad j = 1, 2$.

12. *Exponential* [22]

$$f(x) = \exp\left(-0.5\sum_{i=1}^{m} x_i^2\right),$$

$$f^* = -1$$

The region of interest is $-1 \le x_i \le 1 \quad i = 1, \ldots, m$.

13. *Cosine mixture* [22]

$$f(x) = 0.1\sum_{i=1}^{m} \cos(5\pi x_i) - \sum_{i=1}^{m} x_i^2,$$

$f^* = -2$ if $m = 2$ and $f^* = -4$ if $m = 4$.

The region of interest is $-1 \le x_i \le 1 \quad i = 1, \ldots, m$.

14. *Poissonian pulse-train likelihood* [22]

$$f(x) = \sum_{i=1}^{p}(-\lambda_i(x) + n_i \log(\lambda_i(x)),$$

where

$$\lambda_i(x) = 2\left[1 + 2.5\exp\left\{-0.5\left(\frac{i - x_1}{x_2}\right)^2\right\}\right] + 3$$

and $p = 21$ and the values of n_i, $i = 1, \ldots, 21$ are: 5, 2, 4, 2, 7, 2, 4, 5, 4, 4, 15, 10, 8, 15, 5, 6, 3, 4, 5, 2, 6.

$f^* \simeq -95.28$.

The region of interest is $1 \leq x_1 \leq 21, \quad 1 \leq x_2 \leq 8$.

15. *Cauchy likelihood* [22]

$$f(x) = -\sum_{i=1}^{n}[\log(\pi) + \log(1 + (y_i - x)^2)].$$

We studied this function with $n = 4, n = 10$ and $n = 25$.

The values of y_i are given in the following table:

$n = 4$	3	7	12	17						
$n = 10$	2	5	7	8	11	15	17	21	23	26
$n = 25$	4.1	7.7	17.5	31.4	32.7	92.4	115.3	118.3	119.0	129.6
	198.6	200.7	242.5	255.0	274.7	303.8	334.1	430.0	489.1	
	703.4	978.0	1656.0	1697.8	2745.6					

The region of interest is $y_1 \leq x \leq y_n$.

References

1. L.C.W. Dixon and G.P. Szegö (1975), *Towards Global Optimization 1*, North-Holland, Amsterdam.
2. L.C.W. Dixon and G.P. Szegö (1978), *Towards Global Optimization 2*, North-Holland, Amsterdam.
3. R.Y. Rubinstein (1981), *Simulation and the Monte Carlo Method*, John Wiley & Son, New York.
4. F. Archetti and F. Shoen (1984), A Survey on the Global Optimization Problem: General Theory and Computational Approaches, *Annals of Operations Research*, **1**, 87–110.
5. A. Törn and A. Žilinskas (1989), *Global Optimization*, Lecture Notes in Computer Science 350, Springer-Verlag, Berlin.
6. A.H.G. Rinnooy Kan and G.H. Timmer (1989), Global Optimization, in G.L. Nemhauser, A.H.G. Rinnooy Kan and M.J. Todd (eds), *Optimization*, North-Holland, Amsterdam.
7. R. Horst and H. Tuy (1990), *Global Optimization*, Springer Verlag, Berlin.
8. F. Schoen (1991), Stochastic Techniques for Global Optimization: A Survey of Recent Advances, *Journal of Global Optimization*, **1**, 207–228.
9. C.A. Floudas and P.M. Pardalos (1992), *Recent Advances in Global Optimization*, Princeton University Press, Oxford.
10. R.H. Horst and P.M. Pardalos (1995), *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht.
11. F. Barone, L. Milano and G. Russo (1990), An Optimization Method for Solutions of Close Eclipsing Binaries, in *Active Close Binaries*, ed. C. Ibanoglu, Kluwer Academic Publishers, Dordrecht, 161–188.
12. J. Nocedal (1992), Theory of Algorithms for Unconstrained Optimization, in: *Acta Numerical 1992*, ed. A. Iserles, Cambridge University Press, Cambridge, 199–242.

13. M.J.D. Powell (1992), *A Direct Search Optimization Method that Models the Objective and Constraint Functions by Linear Interpolation*, Numerical Analysis Reports, DAMTP 1992/NA5, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England.
14. J.A. Nelder and R. Mead (1965), A Simplex Method for Function Minimization, Comput. J. **7**, 308–313.
15. W.L. Price (1978), A controlled random search procedure for global optimization, in *Towards Global Optimization 2*, L.C.W. Dixon and G.P. Szegö (eds.), North-Holland, Amsterdam.
16. F. Barone, L. Milano and G. Russo (1990), *The Wilson–Price Code Handbook*, Università di Napoli, Dipartimento di Scienze Fisiche.
17. F. Barone, C. Maceroni, L. Milano and G. Russo (1988), The Optimization of the Wilson–Devinney Method: An Application to CW Cas, *Astron. Astrophys*. **197**, 347–353.
18. F. Barone, L. Fiore, L. Milano and G. Russo (1993), Analysis of Contact Binary Systems: AA Ursae Majoris, V752 Centauri, AO Camelopardalis, and V677 Centauri, *The Astrophysical Journal* **407** 237–251.
19. R.E. Wilson (1979), Eccentric orbit generalization and simultaneous solution of binary star light and velocity curves, *The Astrophysical Journal* **234**, 1054–1066.
20. M.A. Wolfe (1978), *Numerical Methods for Unconstrained Optimization*, Van Nostrand Reinhold Company, New York.
21. S. Lucidi and M. Piccioni, Random Tunneling by Means of Acceptance-Rejection Sampling for Global Optimization, *J. Optim. Theory and Appl.* **62**, 255–278.
22. L. Breiman and A. Cutler (1993), A deterministic algorithm for global optimization, *Mathematical Programming* **58**, 179–199.