Journal of Global Optimization (2006) 34: 569–587
DOI 10.1007/s10898-005-4384-5


# Global Optimization using a Dynamical Systems Approach

STEFAN SERTL and MICHAEL DELLNITZ*
*Faculty of Computer Science, Electrical Engineering and Mathematics, University of Paderborn, D-33095 Paderborn, Germany (e-mail: dellnitz@uni-paderborn.de)*

(Received 31 July 2003; accepted in revised form 20 March 2005)

**Abstract.** We develop new algorithms for global optimization by combining well known branch and bound methods with multilevel subdivision techniques for the computation of invariant sets of dynamical systems. The basic idea is to view iteration schemes for local optimization problems – e.g. Newton's method or conjugate gradient methods – as dynamical systems and to compute set coverings of their fixed points. The combination with bounding techniques allow for the computation of coverings of the global optima only. We show convergence of the new algorithms and present a particular implementation.

**Mathematics Subject Classifications (2000).** 37N40, 65K05, 90C30.

**Key words:** Branch and bound optimization, Computation of fixed points, Dynamical systems, Global optimization

## 1. Introduction

The solution of global optimization problems plays an important role in many applications in science and engineering. Classical iteration procedures (e.g. Newton's method or conjugate gradient methods) are often not appropriate for finding a global minimum of a given objective function, since they usually get stuck in local minima. Alternative approaches like *random search methods* (see e.g. Dixon and Szegö, 1978; Zhigljavsky, 1991) or *genetic algorithms* (Goldberg, 1989; Davis, 1996) have the disadvantage that there always is some uncertainty whether or not the global optimum has actually been found at the end of the searching procedure. On the other hand rigorous methods based on interval analysis as proposed for instance in Ratschek and Rokne (1988), Hansen (1992) or Kearfott (1996) are usually only applicable to low-dimensional problems.

In this article we propose a new method for the computation of the global minimum of a given objective function. This method is based on a *set oriented approach* which is similar in spirit as the one in Dellnitz

*Research of the authors is partially supported by the Deutsche Forschungsgemeinschaft within the Sonderforschungsbereich 376.

and Hohmann (1997) where set oriented numerical methods have been developed for the analysis of discrete dynamical systems (see also Dellnitz and Hohmann, 1996; Dellnitz and Junge, 2002). Roughly speaking these methods work as follows: starting with a big compact region in state space one constructs successively refined box coverings of the invariant sets of the dynamical system under consideration. Thus, the respective numerical multilevel scheme is based on two basic ingredients: subdivision and selection. Similar approaches can be found e.g. in Hsu (1987) or Osipenko and Komarchev (1995).

Already in Dellnitz et al. (2002) subdivision techniques were successfully introduced for the computation of all the zeros of a nonlinear function in a compact domain. The underlying idea is to view iteration schemes (such as Newton's method) as dynamical systems. Then the subdivision and selection procedures are adapted to this context such that all the respective fixed points can be detected. In this paper we go even one step further and adapt the algorithms to the context of global optimization. That is, by an additional combination with branch and bound strategies we propose particular subdivision schemes which allow to approximate the global minima of a given function in a reliable way.

A more detailed outline of the article is as follows. In Section 2 we present a modification of the subdivision procedure for the computation of set coverings of the fixed points of a given dynamical system (Proposition 1 and Algorithm 2). This allows us to compute all the local extremal points of arbitrary nonlinear functions within compact domains. Since we want to find the global optima of the given objective function we propose an algorithm which combines the subdivision procedure of Section 2 with branch and bound techniques as described in Horst and Tuy (1996) and discuss its convergence properties (Section 3). In Section 4 we present several numerical examples which illustrate the efficiency and reliability of our approach.

## 2. The Computation of Fixed Points of Discrete Dynamical Systems

We consider discrete dynamical systems of the type

$$x_{j+1} = f(x_j), \quad j = 0, 1, 2, \ldots,$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a continuous mapping and $x_j \in \mathbb{R}^n$, $j = 0, 1, 2, \ldots$, are the state variables.

Our purpose is to develop a set oriented numerical method for the approximation of all the fixed points of $f$ within some given compact subset $Q \subset \mathbb{R}^n$, i.e. the set

$$FP_f(Q) = \{x \in Q : f(x) = x\}.$$

Previously a subdivision technique has been developed in Dellnitz and Hohmann (1997) for the approximation of general *invariant sets* of dynamical systems, that is sets $A \subset \mathbb{R}^n$ with the property that $f(A) = A$. Here we adapt this technique to the present context in order to compute successively finer coverings of $FP_f(Q)$.

### 2.1. COMPUTATION OF INVARIANT SETS

We begin by briefly recalling the subdivision procedure of Dellnitz and Hohmann (1997). Starting with a (big) initial compact subset $Q \subset \mathbb{R}^n$ this algorithm produces successively refined coverings of the invariant sets under consideration for a given dynamical system $f$. More concretely, this procedure works as follows:

ALGORITHM 1. Computation of invariant sets.

**Step 0. Initialization.** Let $\mathcal{B}_0$ be an initial collection of finitely many subsets of the compact set $Q$ such that $\cup_{B \in \mathcal{B}_0} B = Q$.

**Step $k$. Iteration ($k = 1, 2, \ldots$).** At the beginning of Step $k$ a collection $\mathcal{B}_{k-1}$ of finitely many subsets is available.

**$k$.1.** Choose $\theta_k$ with $0 < \theta_{\min} \leqslant \theta_k \leqslant \theta_{\max} < 1$.

**$k$.2. Subdivision:** Construct a new (refined) system $\widehat{\mathcal{B}}_k$ of subsets such that

$$\bigcup_{B \in \widehat{\mathcal{B}}_k} B = \bigcup_{B \in \mathcal{B}_{k-1}} B$$

and

$$\operatorname{diam}(\widehat{\mathcal{B}}_k) = \theta_k \operatorname{diam}(\mathcal{B}_{k-1}).$$

**$k$.3.** Set $\mathcal{B}_k = \emptyset$ to be an empty collection.

**$k$.4. Selection:**

For every $B \in \widehat{\mathcal{B}}_k$ do:

Add $B$ to the collection $\mathcal{B}_k$ if there exists a $\widehat{B} \in \widehat{\mathcal{B}}_k$ such that $f^{-1}(B) \cap \widehat{B} \neq \emptyset$.

end for

**$k$.5.** Continue with Step $k+1$.

Concerning the details and convergence properties of Algorithm 1 the reader is referred to Dellnitz and Hohmann (1997).

### 2.2. COMPUTATION OF FIXED POINTS

For a given dynamical system $f$ the invariant set within $Q$ in general consists not only of fixed points but contains also periodic points or even more complicated sets. Thus, the above algorithm cannot be used directly to compute the set of fixed points $FP_f(Q)$. We therefore present a modified version of the subdivision scheme. The main difference is in the selection step, which is now adapted to the purpose of finding $FP_f(Q)$.

ALGORITHM 2. Computation of $FP_f(Q)$.

**Step 0. Initialization.** Let $\mathcal{B}_0$ be an initial collection of finitely many subsets of the compact set $Q$ such that $\cup_{B\in\mathcal{B}_0} B = Q$.

**Step $k$. Iteration ($k = 1, 2, \ldots$).** At the beginning of Step $k$ a collection $\mathcal{B}_{k-1}$ of finitely many subsets is available.

**$k$.1.** Choose $\theta_k$ with $0 < \theta_{\min} \leqslant \theta_k \leqslant \theta_{\max} < 1$.

**$k$.2. Subdivision:** Construct a new (refined) system $\widehat{\mathcal{B}}_k$ of subsets such that

$$\bigcup_{B\in\widehat{\mathcal{B}}_k} B = \bigcup_{B\in\mathcal{B}_{k-1}} B$$

and

$$\operatorname{diam}(\widehat{\mathcal{B}}_k) = \theta_k \operatorname{diam}(\mathcal{B}_{k-1}).$$

**$k$.3.** Set $\mathcal{B}_k = \emptyset$ to be an empty collection.

**$k$.4. Selection:**
For every $B \in \widehat{\mathcal{B}}_k$ do:
add $B$ to the collection $\mathcal{B}_k$ if $f(B) \cap B \neq \emptyset$.
end for

**$k$.5.** Continue with Step $k+1$.

As a first result we show that this algorithm converges to the set $FP_f(Q)$ for $k \to \infty$.

PROPOSITION 1. *Let $Q_k$ be the union of the subsets in $\mathcal{B}_k$,*

$$Q_k = \bigcup_{B\in\mathcal{B}_k} B.$$

*Then the following holds:*

(*a*) $Q_k$ *is a covering of* $FP_f(Q)$ *for every* $k \geqslant 0$.

(*b*) $Q_\infty = \bigcap_{k=0}^{\infty} Q_k = FP_f(Q)$.

*Proof.*

(a) Let $x \in FP_f(Q)$. By definition we have $x \in Q = Q_0$. Now assume that $x \in Q_{k-1}$ for some $k > 0$. Then by construction there exists a $B(x) \in \widehat{\mathcal{B}}_k$ with $x \in B(x)$. Since $f(x) = x$ it follows that $f(B(x)) \cap B(x)$ is nonempty. Therefore $B(x) \in \mathcal{B}_k$ which implies $x \in Q_k$.

(b) Let $x \in Q_\infty$. Then for every $k \geqslant 0$ there exists a $B_k(x) \in \mathcal{B}_k$ with $x \in B_k(x)$. Assume that $x \notin FP_f(Q)$, i.e. $\varepsilon = ||f(x) - x|| > 0$. Since $f$ is continuous there exists a $\delta > 0$ such that

$$||f(x') - f(x)|| < \varepsilon/3$$

for all $x' \in \mathbb{R}^n$ with $||x' - x|| < \delta$. Using the fact that $\operatorname{diam}(\mathcal{B}_k) \to 0$ for $k \to \infty$ we conclude that there exists a $K > 0$ such that

$$\operatorname{diam}(B_k(x)) < \min(\delta, \varepsilon/3)$$

for all $k > K$. But this implies that $\operatorname{diam}(f(B_k(x))) < 2\varepsilon/3$ for all $k > K$. On the other hand by construction $f(B) \cap B \neq \emptyset$ for all $B \in \mathcal{B}_k$ and all $k > 0$. Thus we have a contradiction to $||f(x) - x|| = \varepsilon$ and therefore $x$ must be in $FP_f(Q)$. Since $x \in Q_\infty$ was arbitrary we conclude that $Q_\infty \subset FP_f(Q)$. In combination with (a) it immediately follows that $Q_\infty = FP_f(Q)$. □

*Remark* 2. The statements of Proposition 1 still hold if we replace $f(B)$ in the selection step of the subdivision procedure by some outer approximation $U_k(f(B))$ of $f(B)$ as long as

$$\max_{B \in \widehat{\mathcal{B}}_k} \operatorname{dist}(U_k(f(B)), f(B)) \to 0$$

for $k \to \infty$.

### 2.3. IMPLEMENTATION OF THE ALGORITHM

The numerical realization of Algorithm 2 is very similar to the classical subdivision algorithm for the approximation of arbitrary invariant sets of dynamical systems as described in Dellnitz and Hohmann (1997). For the sake of completeness we briefly review its important aspects.

For the implementation of the collections $\mathcal{B}_k$ we use generalized rectangles (also called *boxes*) of the form

$$R(c,r)=\{y\in\mathbb{R}^n : |y_i-c_i|\leqslant r_i \text{ for } i=1,\dots,n\},$$

where $c, r\in\mathbb{R}^n$, $r_i>0$ for $i=1,\dots,n$, are the center and radius, respectively. We start the subdivision procedure with a single rectangle $\mathcal{B}_0=\{R\}$. Given a collection $\mathcal{B}_k$ we construct the refined collection $\widehat{\mathcal{B}}_k$ by bisection of the rectangles in $\mathcal{B}_k$ with respect to the $j$th coordinate, where $j$ is varied cyclically. The subdivision of a rectangle $R(c,r)$ leads to two rectangles $R_-(c^-,\hat{r})$ and $R_+(c^+,\hat{r})$, where

$$\hat{r}_i=\begin{cases} r_i & \text{for } i\neq j,\\ r_i/2 & \text{for } i=j,\end{cases}\qquad c_i^{\pm}=\begin{cases} c_i & \text{for } i\neq j,\\ c_i\pm r_i/2 & \text{for } i=j.\end{cases}$$

This allows for a very efficient storage scheme: a collection $\mathcal{B}_k$ is completely determined by the initial box $R(c,r)$ and a binary tree representing the subdivision structure (see Figure 1).

The selection step is usually discretized via *test points* within each box. The discretized selection criterion thus reads as follows:

add $B$ to the collection $\mathcal{B}_k$ if $f(x)\in B$ for at least on test point $x\in B$.

For low-dimensional problems we typically use a fixed grid of test points within each box, in higher-dimensional problems the points are chosen at random.

*Remark* 3.

(a) Rigorous convergence results for the realization of the subdivision scheme can be obtained when outer approximations of $f(B)$ can be computed which satisfy the condition of Remark 2. This can be done for example using appropriate interval extensions of $f$ (see e.g. Moore, 1966; Alefeld and Herzberger, 1983). If local Lipschitz estimates on $f$

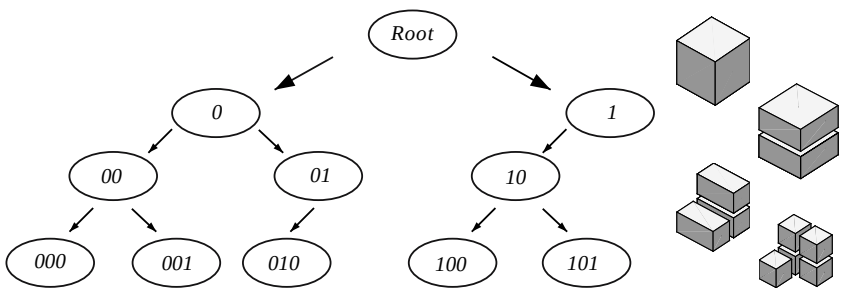



*Figure 1.* Storage scheme for the box collections $\mathcal{B}_k, k=0,1,2,3$.

are available then the methods presented in Junge (1999) can be used, too.

(b) Results on the complexity of general subdivision schemes can be found in Schütze (2004).

We now illustrate the method by the following elementary example.

EXAMPLE 4. We want to compute the fixed points of the Hénon map (Hénon, 1976)

$$f : \mathbb{R}^2 \to \mathbb{R}^2, \quad f(x) = \begin{pmatrix} 1 - ax_1^2 + bx_2 \\ x_1 \end{pmatrix}$$

with a standard set of parameter values $a = 1.2$ and $b = 0.2$. It is known that for these values the Hénon map exhibits complicated dynamics. However, this does not affect our fixed point computations.

Using Algorithm 2 with a grid of $4 \times 4$ test points per box in the selection step we get the coverings shown in Figure 2. After 40 subdivision steps the covering consists of 5 boxes within two clusters (see Table 1).

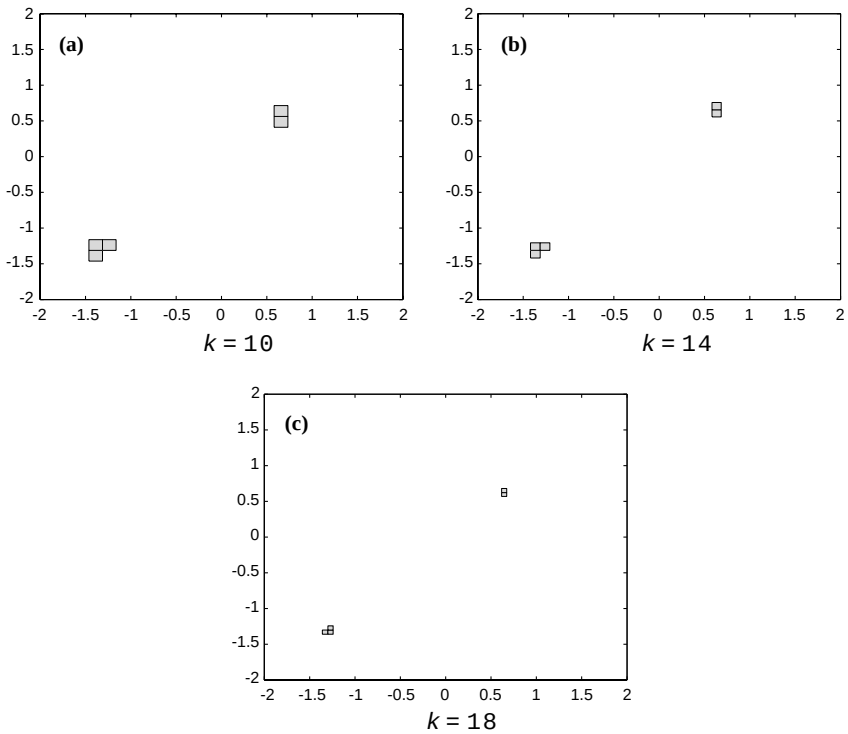



*Figure 2.* Box coverings $\mathcal{B}_{10}$, $\mathcal{B}_{14}$ and $\mathcal{B}_{18}$ of the fixed points of the Hénon map.

*Table 1.* Boxes obtained after 40 steps of Algorithm 2 applied to the Hénon map

| Center | | Radius | |
|---|---|---|---|
| $x_1$ | $x_2$ | $x_1$ | $x_2$ |
| −1.30516148 | −1.30516148 | $2.86102295 \times 10^{-6}$ | $2.86102295 \times 10^{-6}$ |
| −1.30515575 | −1.30516148 | $2.86102295 \times 10^{-6}$ | $2.86102295 \times 10^{-6}$ |
| −1.30515575 | −1.30515575 | $2.86102295 \times 10^{-6}$ | $2.86102295 \times 10^{-6}$ |
| 0.63849163 | 0.63849163 | $2.86102295 \times 10^{-6}$ | $2.86102295 \times 10^{-6}$ |
| 0.63849163 | 0.63849735 | $2.86102295 \times 10^{-6}$ | $2.86102295 \times 10^{-6}$ |

Since the number of clusters stays constant after the first few subdivision steps we conclude that there are two fixed points within the rectangles

$$[-1.3051645, -1.3051528] \times [-1.3051645, -1.3051528]$$

and

$$[0.6384887, 0.6384945] \times [0.6384887, 0.6385003],$$

respectively. The exact results can either be found by additional subdivision steps until a prescribed accuracy is achieved or by switching to some local search method which uses the obtained covering as input.

In addition to the above values $a = 1.2$ and $b = 0.2$ we also performed computational tests on a wide range of different values for these parameters. Algorithm 2 was always able to approximate the corresponding fixed points.

## 3. Application to Optimization Problems

In this section we show how to use the methods presented in the previous section to solve unconstrained optimization problems.

### 3.1. COMPUTATION OF EXTREMAL POINTS

The main idea is to view iteration schemes for local optimization (e.g. Newton's method or conjugate gradient methods) as discrete dynamical systems and to compute their fixed points using the subdivision procedure of Section 2. Since these fixed points correspond to the (local) extremal solutions of the objective function under consideration this method allows for the computation of coverings of *all* the minima (or maxima) within a given compact subset of the phase space.

EXAMPLE 5. We want to compute the extremal points of the following objective function (Himmelblau, 1972):

$$g(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

Within the compact set $Q = [-5, 5] \times [-5, 5]$ this function possesses 9 critical points as listed in Table 2. Using our subdivision algorithm with Newton's method as the dynamical system and a grid of $3 \times 3$ test points per box we obtain the covering shown in Figure 3. It consists of 9 boxes with each of them containing one of the critical points of $g$.

Having computed such a box covering one can obtain the exact solutions of the optimization problem using standard iterative optimization algorithms with a small number of initial points within each of the boxes.

*Table 2.* Location and types of the critical points of $g$ within the region $Q = [-5, 5] \times [-5, 5]$

| $x_1$ | $x_2$ | Type |
|---|---|---|
| −3.7793 | −3.2832 | minimum |
| −2.8051 | 3.1313 | minimum |
| 3.0000 | 2.0000 | minimum |
| 3.5844 | −1.8481 | minimum |
| −3.0730 | −0.0814 | saddle point |
| −0.1280 | −1.9537 | saddle point |
| 0.0867 | 2.8843 | saddle point |
| 3.3852 | 0.0739 | saddle point |
| −0.2708 | −0.9230 | maximum |

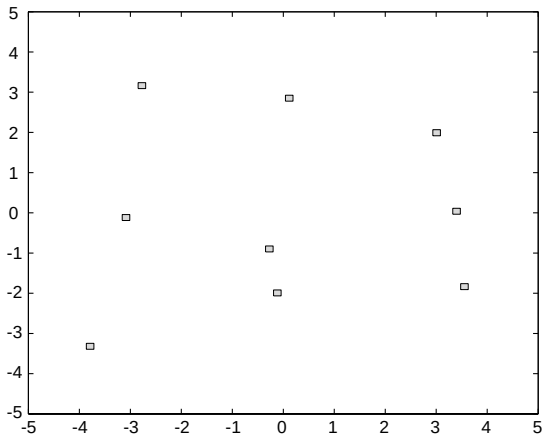



*Figure 3.* Box covering of all the critical points of the function $g$ of Example 5 obtained after 14 subdivision steps.

### 3.2. COMBINATION WITH BRANCH AND BOUND TECHNIQUES

Using the method described above we are in principle able to approximate all the local extremal solutions of the given optimization problem. Selection of the global minima (or maxima) can then be done by comparing the corresponding function values.

However, if one is only interested in the global minima our method can be combined with well known branch and bound techniques (see e.g. Horst and Pardalos, 1995; Horst and Tuy, 1996; Pardalos and Romeijn, 2002) yielding an even more efficient approach. We therefore introduce the following modified subdivision scheme for the computation of coverings of the set of global minimizers of a given continuous function $g: \mathbb{R}^n \to \mathbb{R}$.

ALGORITHM 3. Computation of global minimizers.

**Step 0. Initialization.** Choose an iteration scheme $f: \mathbb{R}^n \to \mathbb{R}^n$ such that the local minimizers of $g$ are fixed points of $f$.
Set $\alpha_0 = +\infty$ and let $\mathcal{B}_0$ be an initial collection of finitely many subsets of a compact set $Q$ such that $\cup_{B\in\mathcal{B}_0} B = Q$.

**Step $k$. Iteration ($k = 1, 2, \ldots$).** At the beginning of Step $k$ a collection $\mathcal{B}_{k-1}$ and an upper bound $\alpha_{k-1}$ on the global minimum of $g$ is available.

**$k$.1.** Choose $\theta_k$ with $0 < \theta_{\min} \leqslant \theta_k \leqslant \theta_{\max} < 1$.

**$k$.2. Subdivision:** Construct a new (refined) system $\widehat{\mathcal{B}}_k$ of subsets such that

$$\bigcup_{B\in\widehat{\mathcal{B}}_k} B = \bigcup_{B\in\mathcal{B}_{k-1}} B$$

and

$$\operatorname{diam}(\widehat{\mathcal{B}}_k) = \theta_k \operatorname{diam}(\mathcal{B}_{k-1}).$$

**$k$.3.** Set $\alpha_k = \min(\alpha_{k-1}, \min g(S_k))$, where $S_k$ is a finite set of sample points within the union of the boxes in $\widehat{\mathcal{B}}_k$.

**$k$.4.** Set $\mathcal{B}_k = \emptyset$ to be an empty collection.

**$k$.5. Selection:**
For every $B \in \widehat{\mathcal{B}}_k$ do
Determine a lower bound $\beta_k(B) \leqslant \inf g(B)$.
Add $B$ to the collection $\mathcal{B}_k$ if $f(B) \cap B \neq \emptyset$ and $\beta_k(B) \leqslant \alpha_k$.
end for

**$k$.6.** Continue with Step $k+1$.

From Proposition 1 we know that in the limit the resulting set $Q_\infty = \bigcap_{k=0}^{\infty} Q_k$ with $Q_k = \cup_{B\in\mathcal{B}_k} B$ contains only fixed points of $f$. We now show under which conditions Algorithm 3 converges to the set

$$GM_g(Q) = \{x^\star \in Q : g(x^\star) \leqslant g(x)\ \forall x \in Q\}$$

of all global minimizers of $g$ with respect to the set $Q$.

THEOREM 6. *The set* $Q_\infty = \bigcap_{k=0}^{\infty} Q_k$ *with* $Q_k = \cup_{B \in \mathcal{B}_k} B$ *generated by Algorithm 3 is equal to the set* $GM_g(Q) = \{x^\star \in Q : g(x^\star) \leqslant g(x)\ \forall x \in Q\}$ *of global minimizers of* $g$ *with respect to* $Q$ *if the following conditions on the bounds* $\alpha_k$ *and* $\beta_k(B)$ *are satisfied:*

(i) $(\alpha_k - \min g(Q)) \to 0$ *for* $k \to \infty$.
(ii) $\max_{B \in \mathcal{B}_k} (\inf g(B) - \beta_k(B)) \to 0$ *for* $k \to \infty$.

*Proof.* We first show that no global minimizer is removed in the selection step of the algorithm: Let $x \in Q$ be a global minimizer of $g$ with respect to $Q$. Assume that $x \in Q_{k-1} = \bigcup_{B \in \mathcal{B}_{k-1}} B$ for some $k > 0$. By construction there exists a $B(x) \in \widehat{\mathcal{B}}_k$ with $x \in B(x)$. Since $x$ is a fixed point of $f$ we have $f(B(x)) \cap B(x) \neq \emptyset$. Furthermore $\alpha_k \geqslant g(x)$ since $x$ is a global minimizer and $g(x) \geqslant \beta_k(B(x))$ by construction. It follows that $B(x) \in \mathcal{B}_k$ and therefore $x \in Q_k$. Since $x \in Q = Q_0$ we conclude that $x \in Q_k$ for all $k \geqslant 0$ and hence $x \in Q_\infty$.

Since $Q_\infty$ is a subset of $FP_f(Q)$ by Proposition 1 it remains to show that each $x \in FP_f(Q)$ which is not a global minimizer of $g$ is not contained in $Q_\infty$: Let $x \in FP_f(Q)$ with $\varepsilon = \big(g(x) - \min\ g(Q)\big) > 0$. For contradiction assume that $x \in Q_\infty$. This implies that for every $k \geqslant 0$ there exists a $B_k(x) \in \mathcal{B}_k$ with $x \in B_k(x)$. Since $g$ is continuous and $\operatorname{diam}(\mathcal{B}_k) \to 0$ for $k \to \infty$ there exists a $k_1$ with

$$(g(x) - \inf g(B_k(x))) < \varepsilon/3$$

for all $k > k_1$. The condition on $\beta_k$ implies that there is a $k_2$ with

$$(\inf g(B_k(x)) - \beta_k(B_k(x))) < \varepsilon/3$$

for all $k > k_2$. Furthermore there exist a $k_3$ such that

$$(\alpha_k - \min\ g(Q)) < \varepsilon/3$$

for all $k > k_3$ due to the convergence property of $\alpha_k$. It follows that for all $k > K = \max(k_1, k_2, k_3)$

$$\begin{aligned}\alpha_k - \beta_k(B_k(x)) &= \alpha_k - \min\ g(Q) + \min\ g(Q) - g(x) \\ &\quad + g(x) - \inf\ g(B_k(x)) + \inf g(B_k(x)) - \beta_k(B_k(x)) \\ &< \varepsilon/3 - \varepsilon + \varepsilon/3 + \varepsilon/3 = 0.\end{aligned}$$

Therefore $\alpha_k < \beta_k(B_k(x))$ for all $k > K$ which contradicts the fact that $B_k(x) \in \mathcal{B}_k$ for all $k \geqslant 0$. □

*Remark* 7.

(a) Observe that Algorithm 3 can be viewed as a particular realization of the general branch and bound algorithm by Horst and Tuy (1996). In fact, the additional selection criterion $f(B) \cap B \neq \emptyset$ turns out to be very useful for the purpose of global optimization.
(b) Using the definitions of Horst and Tuy a bounding operation which satisfies conditions (i) and (ii) of Theorem 6 is called *consistent*.

In numerical realizations of Algorithm 3 the following two questions arise:

(i) How to choose the sets $S_k$ of sample points to satisfy the condition on the $\alpha_k$?
(ii) How to determine suitable lower bounds $\beta_k(B)$?

One possible answer to the first question is given by the following result:

PROPOSITION 8. *Let the sets of sample points $S_k$ for the computation of $\alpha_k$ in Algorithm 3 be defined in such a way that $S_k \cap B \neq \emptyset$ for all $k > 0$ and for all $B \in \mathcal{B}_k$. Furthermore assume that condition (ii) of Theorem 6 is satisfied. Then the coverings $\mathcal{B}_k$ converge to the set of global minimizers of $g$, i.e.*

$$\operatorname{dist}(Q_k, GM_g(Q)) \to 0, \quad \text{for } k \to \infty.$$

*Proof.* We have to show that $\alpha_k \to \min g(Q)$ for $k \to \infty$. Let $x^\star \in Q$ be a global minimizer of $g$, i.e. $g(x^\star) = \min g(Q)$, and choose an arbitrary $\varepsilon > 0$. For each $k > 0$ there exists a $B_k(x^\star) \in \mathcal{B}_k$ with $x^\star \in B_k(x^\star)$. Since $\operatorname{diam}(\mathcal{B}_k) \to 0$ for $k \to \infty$ and by continuity of $g$ we conclude that there exists a $\bar{k} > 0$ such that $g(y) - g(x^\star) < \varepsilon$ for all $y \in B_k(x^\star)$ and all $k > \bar{k}$. By assumption we can choose $y_k \in S_k$ with $y_k \in B_k(x^\star)$ for all $k > 0$. It follows that $g(y_k) - g(x^\star) < \varepsilon$ for all $k > \bar{k}$. By definition of $\alpha_k$ we therefore have $\alpha_k - g(x^\star) < \varepsilon$ for all $k > \bar{k}$. □

In applications we usually discretize the selection criterion $f(B) \cap B \neq \emptyset$ of Algorithm 3 by mapping test points from the box $B$. We therefore just have to compute the values of the objective function for these points to satisfy the above condition. In practice we additionally compute the objective values for the image points $f(x)$ of those test points for which $f(x) \in B$ for $x \in B$. The reason is that in general these values are lower than those for the test points themselves. This leads to sequences $\{\alpha_k\}_k$ which converge faster to the global minimum and therefore boxes with 'big' lower bounds are eliminated earlier in the subdivision process.

Obviously the computation of lower bounds $\beta_k(B)$ on the function values of $g$ within a given box $B$ is in general a very difficult task. If the objective function is simple enough methods of interval arithmetic can be used. Otherwise estimates on lower bounds have to be computed based on numerical approximations of the given function and/or other available information on its local behavior.

## 4. Numerical Examples

We now illustrate the efficiency of our approach by several numerical examples.

EXAMPLE 9. As a first example we apply our global optimization algorithm to the function

$$g(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2,$$

which has previously been considered in Example 5. The four local minimizers of $g$ (cf. Table 2) have the same function value 0 and therefore all of them are also global minimizers. As shown in Tables 3 and 4 and Figure 4 our branch and bound procedure locates all these points efficiently in a reliable way. In this simple case interval arithmetic has been used for the computation of the lower bounds. Table 4 also shows how the combination with branch and bound techniques decreases the computational effort compared to Algorithm 2.

EXAMPLE 10. Our second example is taken from Moré et al. (1981):

$$g_2 : \mathbb{R}^3 \to \mathbb{R}, \qquad g_2(x) = \sum_{i=1}^{10} h_i(x)^2,$$

*Table 3.* Number of boxes obtained during an application of the subdivision process to the objective function $g$ of Example 9

| Subdivision steps | Number of boxes |
|---|---|
| 2 | 4 |
| 4 | 9 |
| 6 | 5 |
| 8 | 5 |
| 10 | 5 |
| 12 | 4 |
| 14 | 4 |

*Table 4.* Performance of the subdivision procedure for Example 9

| Method | Function evaluations | | |
|---|---|---|---|
| | $g$ | $\nabla g$ | $\nabla^2 g$ |
| Algorithm 2 | 9 | 1644 | 1644 |
| Algorithm 3 | 332 | 212 | 212 |

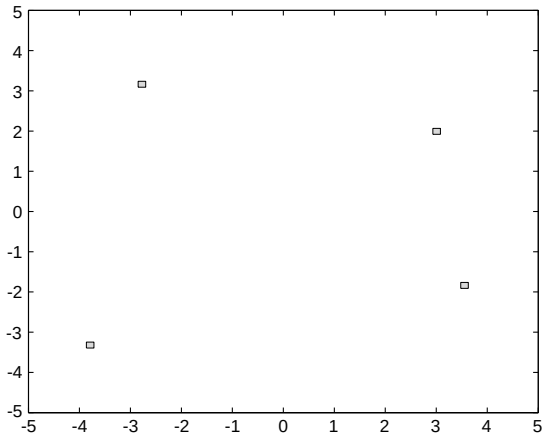


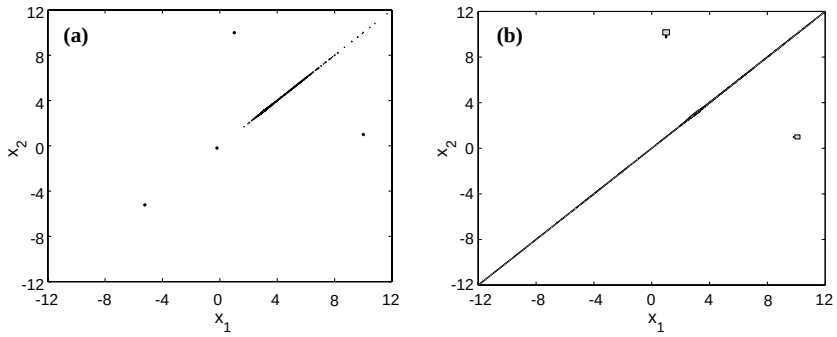

*Figure 4.* Box covering of the global minimizers of Example 9 obtained after 14 subdivision steps.

where

$$h_i(x) = \exp(-0.1ix_1) - \exp(-0.1ix_2) - x_3(\exp(-0.1i) + \exp(-i)).$$

The points $(1, 10, 1)^t$ and $(10, 1, -1)^t$ are global minimizers. The line $x_1 = x_2$, $x_3 = 0$ consists of global minimizers, too.

Using Newton's method with randomly chosen initial points the two global minima at $(1, 10, 1)^t$ and $(10, 1, -1)^t$ can easily be found. On the other hand such a direct approach makes it difficult to detect that the line $x_1 = x_2$, $x_3 = 0$ completely consists of global minimizers as well (see Figure 5(a)). In contrast to this we are able to compute coverings of all the global minimizers within a given box $B \in \mathbb{R}^3$ using Algorithm 3 (cf. Figure 5(b)). As shown in Table 5 a comparable computational effort is required to obtain this result. Observe that for this example Algorithm 2 shows a similar behavior since $g_2$ has only a few local minimizers besides the global ones. In that case the number of boxes during the course of the algorithm increases and therefore also slightly the number of test points for which the iteration scheme has to be evaluated. On the other hand no




Newton iteration with 25000 randomly chosen initial points

Box covering after 30 steps of Algorithm 3

*Figure 5.* Global minimizers found for Example 10 (projection onto the $(x_1, x_2)$- plane).

*Table 5.* Comparison of the subdivision techniques with other optimization methods for the function $g_2$

| Method | Function evaluations | | | CPU | Global minimizers found | | |
|---|---|---|---|---|---|---|---|
| | $g_2$ | $\nabla g_2$ | $\nabla^2 g_2$ | time | $(1, 10, 1)^t$ | $(10, 1, -1)^t$ | $x_1 = x_2$ |
| Algorithm 2 | 3873 | $7.1 \times 10^5$ | $7.1 \times 10^5$ | 23 s | yes | yes | all[1] |
| Algorithm 3 | $6.6 \times 10^5$ | $9.8 \times 10^5$ | $3.6 \times 10^5$ | 20 s | yes | yes | all[1] |
| Newton | 125 | $6.7 \times 10^4$ | $6.6 \times 10^4$ | 2 s | yes | yes | 121 |
| | 1424 | $6.7 \times 10^5$ | $6.6 \times 10^5$ | 19 s | yes | yes | 1420 |
| | 3488 | $1.7 \times 10^6$ | $1.7 \times 10^6$ | 55 s | yes | yes | 3486 |

[1] The line $x_1 = x_2$, $x_3 = 0$ is completely contained in the covering.

lower bounds have to be computed, which in this example, were obtained from numerical estimates based on evaluations of $g_2$ and $\nabla g_2$.

EXAMPLE 11. The next example shows that the subdivision algorithm also reliably finds the global minimizers of objective functions with a larger number of variables. For this we introduce the following function

$$g_3 : \mathbb{R}^n \to \mathbb{R}, \quad g_3(x) = \sum_{i=1}^{n} 1 + x_i^2 (x_i - 0.2)^2 (x_i + 0.2)^2 - \cos(10\pi\, x_i),$$

which has $3^n$ global minimizers in $[-2, 2]^n$.

Using the Algorithm 3 we always find all the global minimizers of $g_3$. For $n = 5$ we also used Algorithm 2 to compute a covering of all the local extremal points of $g_3$ and selected the global minimizers afterwards. As shown in Table 6 much more function evaluations are needed compared to the branch and bound variant of the subdivision method. This is due to the fact that $g_3$ possesses a large

number of local minimizers in addition to the global ones. The ability to delete boxes containing only local minimizers based on lower bounds therefore leads to a significant improvement of the performance in this case.

For $n = 5$ it is also possible to find all global minimizers running the quasi-Newton method implemented in the NAG C-library (`www.nag.com`) with randomly chosen initial points. However, the number of function and gradient evaluations is much higher than for Algorithm 3 (cf. Table 6). For $n = 10$ it seems to be impossible to compute all the minimizers using this method – even for a very large number of initial points – while the subdivision algorithm finds all of them successfully.

In addition to the above examples we carried out computational experiments with several test problems from the literature. The following list specifies these functions, their global minimizers and the search space we used in our computations:

(i) Levy test problem no. 13 (Levy and Gomez, 1985):

$$g(x) = \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1})) + (x_n - 1)^2 (1 + \sin^2(2\pi x_n)), \quad -5 \leqslant x_i \leqslant 5, \quad i = 1, \ldots, n,$$
$$x^* = (1, 1, \ldots, 1)^t, \quad g(x^*) = 0.$$

(ii) The Griewank test function (Griewank, 1981):

$$g(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad -500 \leqslant x_i \leqslant 700,$$
$$i = 1, \ldots, n, \quad x^* = (0, 0, \ldots, 0)^t, \quad g(x^*) = 0.$$

*Table 6.* Comparison of the results for the subdivision algorithms and an optimization method implemented in the NAG C-library applied to the function $g_3$

| $n$ | Method | Function evaluations $g_3$ | Function evaluations $\nabla g_3$ | CPU time | Minimizers found |
|---|---|---|---|---|---|
| 5 | Algorithm 2 | $3.6 \times 10^{11}$ | $1.2 \times 10^{11}$ | 4960 m | 243 |
| | Algorithm 3 | $1.0 \times 10^{7}$ | $4.3 \times 10^{6}$ | 9 s | 243 |
| | NAG | $7.6 \times 10^{5}$ | $7.6 \times 10^{5}$ | 2 s | 38 |
| | | $3.8 \times 10^{6}$ | $3.8 \times 10^{6}$ | 9 s | 125 |
| | | $3.8 \times 10^{7}$ | $3.8 \times 10^{7}$ | 98 s | 243 |
| 10 | Algorithm 3 | $1.6 \times 10^{10}$ | $6.7 \times 10^{9}$ | 460 m | 59049 |
| | NAG | $7.0 \times 10^{8}$ | $7.0 \times 10^{8}$ | 46 m | 31 |
| | | $7.0 \times 10^{9}$ | $7.0 \times 10^{9}$ | 462 m | 328 |
| | | $3.5 \times 10^{10}$ | $3.5 \times 10^{10}$ | 2329 m | 1468 |

(iii) A modified version of the Griewank function presented in Locatelli (2003):

$$g(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \sum_{i=1}^{n} \log\left[2 + \cos\left(\frac{x_i}{\sqrt{i}}\right)\right] + n \log(3),$$
$$-200 \leqslant x_i \leqslant 400, \quad i = 1, \ldots, n, \quad x^* = (0, 0, \ldots, 0)^t, \quad g(x^*) = 0.$$

For all this test problems our new global optimization algorithm reliably detects the global minimizers in a reasonable amount of time (cf. Table 7). Observe that in some cases the NAG iterative solver with randomly chosen initial points allows to find the global minimizer much faster. But in practice one never knows when to stop the search since – in contrast to these experiments – one usually does not know if the best solution found so far actually is the global minimizer. For example in case of the modified Griewank function with $n > 5$ after a few hours of computation the smallest function value encountered so far did not change anymore although – even after more than a week – the global minimizer still had not been found.

We finish this section with some notes about complexity issues of the proposed algorithm.

*Remark* 12. Our experiments indicate that the performance of Algorithm 3 is partially affected by the following properties of the optimization problem or rather their combination:

– the number of local and global minimizers of the objective function,
– the problem dimension,
– the quality of the lower bounds.

*Table 7.* Comparison of Algorithm 3 and the NAG solver for several test examples from the literature

| Function | $n$ | CPU time | |
|---|---|---|---|
| | | Algorithm 3 | NAG |
| Levy | 10 | 2 s | 24 ms |
| | 20 | 10 s | 140 ms |
| | 25 | 105 s | 235 ms |
| | 50 | 875 s | 3.5 s |
| Griewank | 10 | 35 ms | 4 ms |
| | 20 | 54 ms | 4 ms |
| | 50 | 243 ms | 5 ms |
| Modified Griewank | 5 | 1 s | 1240 m |
| | 10 | 16 s | failed |
| | 15 | 26 m | failed |
| | 20 | 2673 m | failed |

Let us discuss the first two points in more detail. Boxes containing only local minimizers will only be deleted due to their lower bounds. In the early stage of the subdivision process they often will be kept and require additional computational work not needed to detect the global minimizers. In case of a huge number of local minimizers this dominates the beginning of the computation and therefore leads to a slower performance compared to a function with only a view local minimizers. The number of global optimizers determine the minimal number of boxes needed in the final covering. Hence from some point on in the subdivision process the amount of work per subdivision step depends only on the number of global optimizers.

The dimension of the problem influences mainly the early stage of the subdivision process since more subdivision steps are needed until all coordinate directions are subdivided. The smaller the boxes are in each coordinate direction the more likely it seems to be that they can be deleted if they do not contain parts of the solution set. On the other hand this also crucially depends on the overall behavior of the objective function.

## 5. Conclusion

In this paper we have presented a new global optimization method which combines recently developed set oriented multilevel subdivision techniques for the analysis of dynamical systems with branch and bound techniques. We have analyzed the convergence properties of the algorithm and discussed possible implementations. Numerical examples show that our new method reliably finds the global minima of arbitrary nonlinear functions. Due to its set oriented approach an application of this method is particularly advantageous if the problem possesses a large number of extremal points.

## References

Alefeld, G. and Herzberger, J. (1983), *Introduction to Interval Computations*, Academic Press, New York.

Davis, L. (ed.) (1996), *Handbook of Genetic Algorithms*, International Thomson Computer Press, London.

Dellnitz, M. and Hohmann, A. (1996), The computation of unstable manifolds using subdivision and continuation. In Broer H.W., van Gils S.A., Hoveijn I. and Takens F. (eds.), *Nonlinear Dynamical Systems and Chaos*, vol. 19 of *PNLDE*, Birkhäuser, pp. 449–459.

Dellnitz, M. and Hohmann, A. (1997), A subdivision algorithm for the computation of unstable manifolds and global attractors, *Numerische Mathematik* 75(3), 293–317.

Dellnitz, M. and Junge, O. (2002), Set oriented numerical methods for dynamical systems, In: Fiedler, B., Kopel, N. and Iooss, G. (eds.), *Handbook of Dynamical Systems II: Towards Applications*, World Scientific, Singapore, pp. 221–264.

Dellnitz, M., Schütze, O. and Sertl, S. (2002), Finding zeros by multilevel subdivision techniques, *IMA Journal of Numerical Analysis* 22(2), 167–185.

Dixon, L.C.W. and Szegö, G.P. (eds.) (1978), *Towards Global Optimization 2*. Amsterdam, North-Holland.

Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, New York.

Griewank, A.O. (1981), Generalized descent for global optimization, *Journal of Optimization Theory and Applications* 34(1), 11–39.

Hansen, E. (1992), *Global Optimization Using Interval Analysis*, Marcel Dekker, New York.

Hénon, M. (1976), A two dimensional mapping with a strange attractor, *Communications in Mathematical Physics* 50, 69–77.

Himmelblau, D.M. (1972), *Applied Nonlinear Programming*, McGraw-Hill Book Company, New York.

Horst, R. and Pardalos, P. (eds.) (1995), *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht.

Horst, R. and Tuy, H. (1996), *Global Optimization: Deterministic Approaches*, 3rd edition, Springer, Berlin.

Hsu, C.S. (1987), *Cell-to-Cell Mapping*, Springer, New York.

Junge, O. (1999), Mengenorientierte Methoden zur numerischen Analyse dynamischer Systeme, Ph.D. thesis, University of Paderborn.

Kearfott, B. (1996), *Rigorous Global Search: Continuous Problems*, Kluwer Academic Publishers, Dordrecht.

Levy, A.V. and Gomez, S. (1985), The tunneling method applied to global optimization, In Boggs, P.T., Byrd, R.H. and Schnabel, R.B. (eds.), *Numerical Optimization 1984*. SIAM, Philadelphia, pp. 213–244.

Locatelli, M. (2003), A note on the griewank test function, *Jornal of Global Optimization* 25, 169–174.

Moore, R.E. (1966), *Interval analysis*, Englewood Cliffs, Prentice-Hall, NJ.

Moré, J., Garbow, B. and Hillstrom, K. (1981), Testing unconstrained optimization software, *ACM Transactions on Mathematical Software* 7(1), 17–41.

Osipenko, G. and Komarchev, I. (1995), Applied symbolic dynamics: construction of periodic trajectories, In: Agarwal R.P. (ed.), *Dynamical Systems and Applications*, Vol. 4 of *Worlds Scientific Series in Applicable Analysis*, World Scientific, Singapore, pp. 573–587.

Pardalos, P. and Romeijn, E. (2002), *Handbook of Global Optimization – Volume 2: Heuristic Approaches*, Kluwer Academic Publishers, Dordrecht.

Ratschek, H. and Rokne, J. (1988), *New Computer Methods for Global Optimizations*, Ellis Horwood Limited, Chichester.

Schütze, O. (2004), Set oriented methods for global optimization, Ph.D. thesis, University of Paderborn.

Zhigljavsky, A.A. (1991), *Theory of Global Random Search*, Kluwer Academic Publishers, Dordrecht.