Journal of Global Optimization (2006) 34: 409–426
DOI 10.1007/s10898-005-3249-2

A Combined Global & Local Search (CGLS) Approach to Global Optimization*

U.M. GARCIA-PALOMARES[1,2], F.J. GONZALEZ-CASTAÑO[2] and J.C. BURGUILLO-RIAL[2]
[1] *Universidad Simón Bolívar, Dep. Procesos y Sistemas, Apartado 89000, Caracas, Venezuela, (E-mail: garciap@usb.ve.)*
[2] *Universidade de Vigo, Dep. Ingeniería Telemática, ETSI Telecomunicación, Campus, Apartado 36310, Vigo, Spain, (E-mail: ubaldo,javier,jrial@det.uvigo.es)*

(Received 29 July 2004; accepted in revised form 12 September 2005)

Abstract. This paper presents a general approach that combines global search strategies with local search and attempts to find a global minimum of a real valued function of n variables. It assumes that derivative information is unreliable; consequently, it deals with derivative free algorithms, but derivative information can be easily incorporated. This paper presents a nonmonotone derivative free algorithm and shows numerically that it may converge to a better minimum starting from a local nonglobal minimum. This property is then incorporated into a random population to globalize the algorithm. Convergence to a zero order stationary point is established for nonsmooth convex functions, and convergence to a first order stationary point is established for strictly differentiable functions. Preliminary numerical results are encouraging. A Java implementation that can be run directly from the Web allows the interested reader to get a better insight of the performance of the algorithm on several standard functions. The general framework proposed here, allows the user to incorporate variants of well known *global* search strategies.

Key words: Derivative-free, Global optimization, Nonmonotone

1. Introduction

This section proposes a nonmonotone derivative-free algorithm that converges to a point x satisfying the zero order necessary *minimality* conditions **ZONC** for nonsmooth convex functions $f(\cdot)$: $S \subseteq IR^n \to IR$. We assume that derivative information is either nonexistent or unreliable; however, convergence to a first order necessary point x is established when $f(\cdot)$ happens to be strictly differentiable at x, but not necessarily convex; that is

*Research done under the cooperation agreement between Universidade de Vigo and Universidad Simón Bolívar.

$$\lim_{\substack{x_i \to x \\ d_i \to d \\ \eta \to 0}} \frac{f(x_i + \eta d_i) - f(x_i)}{\eta ||d_i||} = \nabla f(x)^T d.$$

Let X_1 be the set of first-order stationary points of $f(\cdot)$, which is characterized by $X_1 = \{x \in S\colon\ \nabla f(x) = 0\}$, and let X_0 be the set of points that satisfy the zero-order necessary condition (ZONC) for functions with directional derivatives $f'(x, d)$ everywhere defined.

ZONC: (Zero-order necessary condition)

Let $D = \{d_1, \ldots, d_m\}$ be a finite set of unit vectors that positively spans $I\!R^n$. We say that x satisfies ZONC if it belongs to the set X_0 defined as:

$$X_0 = \{x \in S : f'(x, d) \geq 0, \text{for all } d \in D\}. \tag{1}$$

Let us remark beforehand that the first order necessary condition, namely, $\nabla f(x) = 0$, holds if $x \in X_0$ and $f(\cdot)$ is strictly differentiable at x (García-Palomares and Rodríguez, 2002, Lemma 4.2).

An ultimate goal is to find all global minima of $f(\cdot)$, i.e., find all $x \in X^*$ defined as

$$X^* = \{z \in X_0 : f(z) \leq f(x) \text{ for all } x \in S\}. \tag{2}$$

Diener (1994) proposed to find X^*, the set of global minimizers, as the solution trajectories of ordinary differential equations of first and second order. He explicitly states that he proved no convergence results. Indeed, most researchers in this field agree that even the simpler problem of finding a single element of X^* without assuming a structured problem, may be intractable, mainly because a practical mathematical characterization of a global optimum is lacking (Groenwold and Snyman, 2002), and because Törn and Žilinskas (1989) proved that a sequence of iterates must be dense to ensure that it converges to the global minimum. On top of this, X^* may not be a countable set, or $|X^*|$, the cardinality of X^*, may be so large as to render impractical any algorithm that tries to determine all of its elements. These impediments may persist on X_1. Another major hindrance is that derivative information may be absent, so an element of either X^* or X_1 must be determined on the basis of function evaluations only. This excludes the use of efficient gradient based methods to carry out a local search; so the paper's objectives are restricted to find *some* element(s) in X_0 or X_1 without computing nor approximating derivatives.

Despite (or due to) the cited shortcomings a flood of articles on global optimization strategies continues to appear and its pace does not seem to slow down. It looks a formidable task to cite all pertinent references that have appeared in the open literature. Good surveys and many references are included in the volumes of handbook of global optimization edited by Horst and Pardalos (1994), and Pardalos and Edwin Romeijn (2002). Gray et al. (1997) wrote a survey of algorithms that can be accessed in the Web. Haim et al. (1999) compare different software packages and Pintér (1995) describes a good range of real world applications.

The decision maker is often interested in finding the best minima in a compact set S. We could assume that this set is given by bounds on the variables or by constraints that can be easily handled with straightforward modifications of the basic ideas presented in this paper. To avoid additional technicalities that could stray the reader's attention from the main ideas, we will assume for the sake of simplicity of the presentation that our algorithms generate a sequence of points that remain in S. This is valid, at least theoretically, if we assume that $f(x) \to \infty$ whenever $||x|| \to \infty$.

This paper is organized as follows: Next section points out the role that local search algorithms play in several global optimization strategies and gives the convergence proof of a nonmonotone variant of the derivative-free local search algorithm for nonsmooth functions recently proposed by García-Palomares and Rodríguez (2002). Section 3 describes an algorithm that combines this nonmonotone search with different global minimization strategies. It is proved that the proposed scheme generates (sub)sequences with limit points in $X_0(X_1)$ if $f(\cdot)$ is locally convex (strictly differentiable). Section 4 discusses implementation issues and reports encouraging numerical results. Section 5 states the conclusion and additional remarks.

We end this section by pointing out some peculiarities of our notation, which is otherwise standard: $\{\cdot\}_{i=1}^{\infty}$ is an infinite sequence of iterates, and $\{\cdot\}_{i\in I}$ is a subsequence. $I\!R^n$ is the n-dimensional Euclidean space; $I\!R_+$ is the set of nonnegative real scalars. Small Latin letters $i, \ldots, q$ are integers. All other Latin letters are vectors in some Euclidean space. Subscripts denote different entities, and superscripts denote components; e.g., x_i^k is the k-component of vector x_i. Capital Latin letters denote sets and lower case Greek letters denote elements in $I\!R$. The notation $o(\tau)$ denotes a function from $I\!R_+ \to I\!R$ with the property $\lim_{\tau\downarrow 0} o(\tau)/\tau = 0$.

2. Local Search

Gradient-like algorithms using first and second order derivative information are often used in the local search phase of deterministic global optimization algorithms of smooth functions. An obvious approach is to start

a local search on the best point given by a global optimization algorithm (Isenor et al., 2003 and references therein). In general, a local search is combined with heuristics and/or strategies that augment the chances of finding a global minimum. Known strategies that include local search are:

Multistart + local search. A local search is applied on multiple starting points randomly generated. We can see Smith et al. (1990), Pintér (1995), and references therein.

Local search + splitting of the search space. The search space is split in boxes (intervals) and a local search is carried out in smaller boxes (intervals). These *interval-like* techniques have been proposed by Jones et al. (1993), Csendes and Ratz (1997), Huyer and Neumaier (1999), and others.

Local search + bypass. Local searches of $f(\cdot)$ and/or auxiliary functions are carried out to obtain a local minimum. Once a local minimum x^* of $f(\cdot)$ is detected, descent steps are carried out on an auxiliary function $\tilde{f}(\cdot)$ until theoretically a point x is found with a functional value lower than $f(x^*)$. Some examples are the tunneling function (Levy and Gómez, 1985), the fill function (Ge, 1990; Liu and Xu, 2004) and the cutting angle function (Bagirov et al., 2002). A typical tunneling function is:

$$\tilde{f}(x) = (f(x) - f(x^*) + \delta)\left(1 + \frac{1}{||x - x^*||^2}\right). \tag{3}$$

Local search of approximating functions. Locatelli and Schoen (2002) find $\tilde{x}$, a local minimum of a function $\tilde{f}(\cdot)$ that shares some properties with $f(\cdot)$, but at the same time, it is more amenable to work with. Then a local search on $f(\cdot)$ is carried out, starting at $\tilde{x}$.

Combine local search and stochastic optimization. Yiu et al. (2004) proposed to perform alternatively local search and simulated annealing. A switch between both techniques will occur when the function value $f(\cdot)$ strictly decreases.

This brief survey reveals that a local search algorithm; i.e., an algorithm devised to locate a local minimum, is the core of several deterministic approaches in global optimization. Conceivably, the better the local search, the faster the algorithm. A major drawback of these strategies (except Yiu et al. 2004) is that *many* local minima must be found, which is computationally expensive. This paper is closer in spirit to Yiu's, because it combines an *incomplete* local search with a global strategy, but it differs from his because the stochastic optimization is replaced by an evolutionary approach, and the local search does not require strict decrease of function values. It works with a population of individuals at each iteration, an

incomplete nonmonotone local search is carried out from each individual and the search results are used to update the population.

This paper assumes that gradient-like search may become impractical and must be replaced by a derivative-free search, even though the latter is not necessarily endowed with the well established superlinear rate of convergence of quasi Newton methods; nonetheless, it is nowadays accepted that derivative-free methods are essential for solving real problems (Wright, 1996; Kolda et al., 2003). Recent global optimization approaches that do not compute or approximate gradients by difference of function values have been proposed by Jones (2001), who locates the *global* minimum as a sequence of *local* minima of interpolating functions made up by a linear combination of basis functions. In a series of papers, Price (1977, 1983, 1987) introduced the Controlled Random Search (CRS) algorithms that have been later modified and tested numerically with encouraging results (Ali et al., 1997; Ali and Törn, 2004). The main drawback of these algorithms is that no theoretical convergence proof is provided.

We now prove convergence of a nonmonotone version of the derivative-free algorithm proposed by García-Palomares and Rodríguez (2002). It will be the core of the Combined Global and Local Search (CGLS) optimization approach described in the next section. Let us mention in passing that nonmonotone strategies used with gradient information have been introduced in both unconstrained and constrained optimization. Numerical tests show that they are competitive with conventional gradient-like monotone techniques. Furthermore, it has been observed that nonmonotone strategies may avoid convergence to local nonglobal solutions of the problem (Grippo et al., 1986; Toint, 1997; Ulbrich, 2001). Section 4 (Table VI) supports these results.

Tables I and III describe, respectively, the conceptual and the actual implementation of the nonmonotone algorithm. Theorem 2 shows that it generates a (sub)sequence of iterates that converges to $x \in X_0$ if $f(\cdot)$ is nonsmooth and locally convex around x, or converges to $x \in X_1$ if $f(\cdot)$ is strictly differentiable at x. The key idea of the algorithm is the nonmonotone sufficient decrease (NMSD) condition (4). Given a *solution* estimate $x_i \in S, \alpha_i \in [0,\ 1], \nu(\tau_i)$ satisfying **A4** below, and φ_i satisfying **A5** below, any point $y \in S$ is a potential candidate to become a new *solution* estimate if

$$f(y) \leq f(x_i) + \alpha_i(\varphi_i - f(x_i)) - \nu(\tau_i). \tag{4}$$

Every time we pinpoint $y \in S$ that fulfills (4) we say that the iteration has been *successful*. Otherwise, we declare that x_i is *temporarily* blocked and reduce $\nu(\tau_i)$ to try to unblock it. Theorem 2 below shows that the reduction of $\nu(\cdot)$ will eventually unblock x, unless this point satisfies ZONC. In a sense this is reminiscent of *trust region* approaches. The monotone

Table I. Local nonmonotone algorithm

Input: $0<\epsilon, 0<\mu_m\le\mu_M<1\le\gamma$	Constants
$i=0$	iteration counter
Pick $x_1\in S, 0<\tau_1\in I\!R$	initial values
DO $i=i+1$	iteration
Update $D_i, \varphi_i, \alpha_i\in[0,\ 1]$	satisfying **A3,A5**
$\phi_i=f(x_i)+\alpha_i(\varphi_i-f(x_i))$	
IF $\exists(d\in D_i): f(x_i+\tau_i d)\le\phi_i-\nu(\tau_i)$	
Find $\eta\ge\tau_i: f(x_i+\eta d)\le\phi_i-\nu(\tau_i)$	local search
$x_{i+1}=x_i+\eta d$	move
$\nu(\tau_i)\le\nu(\tau_{i+1})\le\gamma\nu(\tau_i)$	may expand ν
ELSE	x_i is blocked
$x_{i+1}=x_i$	no move
$\mu_m\nu(\tau_i)\le\nu(\tau_{i+1})\le\mu_M\nu(\tau_i)$	reduce ν
END IF	
WHILE $(\tau_i>\epsilon)$	no convergence

algorithm is derived if $p=1, \varphi_i=f(x_i)$ in (6). Fortunately, theorem 2 needs no specialized material to prove convergence. It closely follows the arguments given by García-Palomares and Rodríguez (2002) and the reader is advised to look at that reference for further details.

In the sequel we will assume:

A1. $f(\cdot): I\!R^n\to I\!R$ is bounded below, and $\{x_i\}_1^\infty$ remains in the compact set S.

A2. $f(\cdot)$ has directional derivatives $f'(x,d)$ everywhere defined, and

$$\eta>0\Rightarrow\begin{cases} f'(x,\eta d) &= \eta f'(x,d),\\ f(x+\eta d) &= f(x)+\eta f'(x,d)+o(\eta).\end{cases}\tag{5}$$

A3. The finite set $D=\{d_1,\ldots,d_m\}$ of unit search directions positively spans $I\!R^n$, and $f(\cdot)$ is strictly differentiable or locally convex at all limit points of the sequence $\{x_i\}_1^\infty$ generated by the algorithm.

A4. The function $\nu(\cdot): I\!R_+\to I\!R_+$ is strictly increasing and $\lim_{\tau\downarrow 0}\nu(\tau)/\tau=0$.

A5. The sequence of reference values $\{\varphi_i\}_1^\infty$ is a nonincreasing sequence of upper bounds of the functional values $\{f(x_i)\}_1^\infty$ and decreases sufficiently every bounded number of iterations; specifically
(a) $\varphi_{i+1}\ \le\varphi_i$ for all i,
(b) $f(x_i)\le\varphi_i$, for all i, and
(c) If the number of successful iterations is infinite then there exist an infinite subset of iteration indices K and a finite $p>0$ such that

$$\forall(k \in K)\ \exists(i \in K) \Rightarrow \begin{cases} \varphi_{k+1} \le \varphi_k - \nu(\tau_j) \text{for some} j : i < j \le k \\ \text{No more than } p \text{ successful iterations} \\ \text{between } i \text{ and } k. \end{cases} \tag{6}$$

Assumptions **A1–A5** are rather mild. In particular **A2** holds for nonsmooth functions with directional derivatives everywhere defined, like convex functions, norm of differentiable functions, the maximum and/or minimum of a group of differentiable functions, and continuous differentiable functions. We are now ready to prove convergence of the algorithm framework shown in Table I.

LEMMA 1. $\{\tau_i\}_1^\infty \to 0$.

Proof. If the number of successful iterations is finite, then $\nu(\tau_{i+1}) \le \mu_M \nu(\tau_i)$ for all i large enough, and by **A4** the lemma is true. Let us now assume that the number of successful iterations is infinite, and let $k, i \in K, k < i$ be two consecutive indices in K as defined by (6); hence $\varphi_i \le \varphi_{k+1} \le \varphi_k - \nu(\tau_j)$ for some $j \le k$. We assert that $\{\nu(\tau_j)\}_{j \in J} \to 0$, where J is the index set of ν values satisfying (6); for otherwise $\{\varphi_k\}_{k \in K}$ would decrease without bounds, and **A5b** implies that $\{f(x_k)\}_{k \in K}$ is unbounded, contradicting **A1**.

Given any two consecutive elements $j, k \in J$ there are at most $2p$ successful iterations between them; hence $\nu(\tau_i) \le \gamma^{2p} \nu(\tau_j)$, for $j \le i < k$ and we thus assert that $\{\nu(\tau_i)\} \to 0$. The validity of the lemma follows from **A4**. □

COROLLARY. *There is an infinite number of blocked points.*

The algorithm is well defined, in the sense that if x_i is permanently blocked, then $f'(x_i, d) \ge 0$ for all $d \in D_i$. Note that $f(x_i + \tau d) > \phi_i - \nu(\tau) > f(x_i) - \nu(\tau)$ for all $d \in D, \tau > 0$. The result now follows from García-Palomares and Rodríguez (2002, lemma 3.1).

THEOREM 2 (Convergence). *Let $\{D_i\}_1^\infty \to D$, and let x be any limit point of blocked points of the sequence $\{x_i\}_1^\infty$. If $f(\cdot)$ is locally convex around x then $f'(x, d) \ge 0$ for all $d \in D$. Moreover, if $f(\cdot)$ is strictly differentiable at x, then $\nabla f(x) = 0$.*

Proof. Let x_i be a blocked point and $d \in D_i$. By construction we have that

$$f(x_i + \tau_i d) > \phi_i - \nu(\tau_i) \ge f(x_i) - \nu(\tau_i).$$

The rest of the convergence proof follows from García-Palomares and Rodríguez (2002, theorems 3.4, 3.5, 4.1, lemma 4.2 and the comment below theorem 4.1). □

It is worth mentioning that this convergence result only ensures convergence to stationary points, unless $f(\cdot)$ enjoys some particular properties, like smoothness and convexity. To the author's knowledge there is no general derivative-free algorithm that ensures convergence to a local minimum for nonsmooth convex functions, a quest that has been so far elusive.

3. Global Optimization Algorithm

The previous section was concerned with the problem of locating a stationary point. Numerical experiments reported in the next section show that the nonmonotone feature of the local search algorithm may *occasionally* make it jump over small hills; but this effect could be offset with a careless implementation. A major drawback *of any local algorithm* is that it does not search the whole domain, which is evidently necessary in global optimization (Stephens and Baritompa, 1998).

An early attempt to consider global information is multi start, and the NonMonotone (NM) algorithm could be used on a set of random starting points. Locatelli and Schoen report several difficulties with the use of this approach (2002); so we decided to work on an initial population $\mathcal{P}$ of p individuals randomly chosen in S and combine global and local search as follows. We identify x_b as the best point in $\mathcal{P}$ if $f(x_b) \leq f(x), x \in \mathcal{P}$. We also identify x_w as the worst point in $\mathcal{P}$ if $f(x_w) \geq f(x), x \in \mathcal{P}$. Ties are broken arbitrarily. The algorithm attempts to find some $y \in S$ that fulfills the (NMSD) condition (4). Genetic-like, simulated annealing or deterministic techniques that have been proposed in the open literature can be used, as long as they provide a point $y \in S$ satisfying (4). If this is the case, it replaces x_w by y; otherwise it picks $x \in \mathcal{P}$ and carries out a *local* NM search as described in Tables II and VII.

As $\varphi_i = \max(f(x)), x \in \mathcal{P}_i$, and the worst point is always replaced by a better point at all successful iterations, **A5ab** trivially holds. Convergence would follow if we can identify a set K satisfying (6). Note that after p successful iterations the population $\mathcal{P}$ has *evolved* to a completely different set of *individuals*. Let $i, k, i < k$ be two consecutive indices in the set K if i is the biggest iteration index for which $\mathcal{P}_i \neq \mathcal{P}_k$, but $\mathcal{P}_j \cap \mathcal{P}_k \neq \emptyset$ for all $i < j < k$. Hence, by construction

$$\varphi_{k+1} \leq \varphi_k - \nu(\tau_j), \text{ for some } j : i < j \leq k. \tag{7}$$

Convergence follows almost verbatim from lemma 1 and theorem 2 if in the local search phase we reduce $\nu(\cdot)$ only if $f(x + \tau d) > \varphi_i - \nu(\tau_i)$ for all $d \in D$. The following propositions are therefore valid and are merely given to collect the convergence results. We recall that assumptions **A1 – A5** hold.

Table II. i-th Iteration of the CGLS algorithm

Input: $0 < \epsilon, 0 < \mu_m \leq \mu_M < 1 \leq \gamma_i, \tau_i, D_i, \mathcal{P}_i$

Identify x_w, and let $\varphi_i = f(x_w)$
Select $\mathcal{Q}_i \subseteq \mathcal{P}_i, \alpha_i \in [0,\ 1]$
$success = false, \phi_i = f(x) + \alpha_i(\varphi_i - f(x))$
FOR ALL $x \in \mathcal{Q}_i$
  Generate $x_{i+1} \in S$ with a **global** strategy
  IF $f(x_{i+1}) \leq \phi_i - \nu(\tau_i)$
    $success =$ TRUE
  END IF
END FOR
IF NO *success*
  Choose $x \in \mathcal{P}_i$ to start the **local** search
  IF $x_{i+1} = x_i + \sum_{k=1}^{m} \eta_k d_k, \eta_k \geq \tau_i$ and $f(x_{i+1}) \leq \phi_i - \nu(\tau_i)$
    $success =$ TRUE
  END IF
END IF
IF *success*
  $\mathcal{P}_{i+1} = \mathcal{P}_i + \{x_{i+1}\} - \{x_w\}$, recompute x_w
  $\nu(\tau_i) \leq \nu(\tau_{i+1}) \leq \gamma_i \nu(\tau_i)$
ELSE
  $\mu_m \nu(\tau_i) \leq \nu(\tau_{i+1}) \leq \mu_M \nu(\tau_i)$
END IF

PROPOSITION 3. $\{\tau_i\}_1^\infty \to 0$.

PROPOSITION 4. *Let* $D_i = \{d_{i1}, \ldots, d_{im}\} \to D = \{d_1, \ldots, d_m\}$. *If the algorithm gets stuck at* x_i *and* $f(\cdot)$ *is locally convex around* x_i, *then* $f'(x_i, d_k) \geq 0$, *for* $d_k \in D$. *Moreover,* $\nabla f(x_i) = 0$ *if* $f(\cdot)$ *is strictly differentiable at* x_i.

PROPOSITION 5. *Let* $D_i = \{d_{i1}, \ldots, d_{im}\} \to D = \{d_1, \ldots, d_m\}$, *and let* x *be a limit point of a (sub)sequence of blocked points. If* $f(\cdot)$ *is locally convex around* x, *then* $f'(x, d_k) \geq 0, d_k \in D$. *Besides, if* $f(\cdot)$ *is strictly differentiable at* x, *then* $\nabla f(x) = 0$.

Note that the last proposition only states that there must be at least one limit blocked point $x \in \mathcal{P}$ that satisfies the zero or first order necessary condition. Had we regularly chosen x_b in the local phase, it would have been obvious to prove that a subsequence of the best points converges to x. However, the numerical experiments suggest that this is not the best strategy, because the algorithm can terminate prematurely at a local minimum, which is a limit point of the best ones. CGLS should therefore end with a local search starting at x_b. The next section deals with these practical implementation issues and reports preliminary numerical results

on a moderately difficult and on a difficult test problem. We emphasize that CGLS subsumes any strategy that admits (NMSD) in function values (Equation (4)).

4. Implementation and Numerical Results

A lot of variants can be derived from the algorithmic descriptions given in Tables I and II, and deep insight will be an asset to secure an efficient implementation. In general, sic: *a number of heuristic enhancements are needed to obtain a high quality model* (Huyer and Neumaier 1999). In addition, parameter values seem to be problem dependent in all derivative free methods.

The actual implementation of the (NM) algorithm (Table III) proceeds along the guidelines suggested by García-Palomares and Rodríguez (2002). We keep in mind that a proper updating of D satisfying **A3** is of utmost importance to improve the performance of the algorithm; so $D = \{\pm u, \pm d_1, \ldots, \pm d_n\}$,where $u \in I\!R^n$ is a unit direction generator vector, and $d_1, \ldots, d_n$ are the n orthogonal directions taken from the columns of the Householder orthogonal matrix $(I - 2uu^T)$. In this implementation (Table III) the direction generator u is a descent direction that is updated after the algorithm cyclically searches all directions. This is reminiscent of Hooke and Jeeves (1961) and has worked well when coupled with the interpolation procedure explained below. The generator u is randomly generated at all *temporarily* blocked points to give the algorithm extra directions to search on. This latter feature is particularly convenient for nonsmooth convex functions, since $f'(x, d) \geq 0$ may hold on a larger number of search directions. Table III shows the list of parameters that can be set by the user. We used $\epsilon_\tau = \epsilon_f = 1e-6, \gamma = 1.3, \mu = 0.3, \tau = 0.1,$ and $\varphi = 200$ for the Rosenbrock function. We set $\varphi = f(x)$ if a successful move appears after a local search is carried out on all directions. We expand τ when (NMSD) holds on more than $n/2$ directions of search and reduce τ when (4) is not fulfilled on any $d \in D_i$. We also chose $\nu(\tau) = 0.1\tau^2$ and terminate the algorithm under two normal *convergence* conditions: $\tau < \epsilon_\tau$, or $f(x_w) \leq f(x_b) + \epsilon_f(|f(x_b)| + 1)$. As a safeguard (not explicitly given in Table III) we also terminate the algorithm when the number of function evaluations exceeds 10^6. The common discrete *local* search procedure, often used in search methods, was replaced by a quadratic interpolation procedure (Table IV) *to try to* find a better f value on the direction d, and it is roughly as follows: Given $x \in S, d \in I\!R^n, \tau > 0$, the procedure computes $z_{\text{right}} = x + \tau d, z_{\text{left}} = x - \tau d$ and defines $y = \arg\min(f(z_{\text{right}}), f(z_{\text{left}}))$. If $f(y) > f(x)$ the procedure considers that x is blocked along d and returns $f(y)$ and the corresponding y point; otherwise, it enters into the proper interpolation procedure, which first detects a region of positive curvature,

Table III. Local nonmonotone implementation

Parameters: $\epsilon_\tau, \epsilon_f, \gamma, \mu, \tau, \varphi$	
Get x, $v^k \in [x^k - \tau,\ x^k + \tau], k = 1, \ldots, n$	random v
DO $u = x - v, v = x, success = 0$	
IF $\|\|u\\| < \tau$, let $u^k \in [-\tau,\ \tau], k = 1, \ldots, n$	random u
Set $d_0 = u, d_j = e_j - 2u^j u, j = 1, \ldots, n$	
FOR $j = 0$ TO n	
$y =$ INTERPOLATE (x, d_j, τ)	table IV
$\alpha = \min(\tau, \alpha), \phi = f(x) + \alpha(\varphi - f(x))$	
IF $(f(y) \le \phi - 0.1\tau^2)$	
$success = success + 1$	
$x = y$	move
ENDIF	
ENDFOR	
IF $(success = 0)$	
$\tau = \mu\tau$	
ELSE $\varphi = f(x)$	
ENDIF	
IF $(success > n/2)$	
$\tau = \gamma\tau$	
ENDIF	
WHILE $(\tau > \epsilon_\tau)$ and $(f(v) > f(x) + \epsilon_f(\|f(x)\| + 1))$	

i.e., it searches for 3 points where it makes sure that $f(\cdot)$ will be interpolated by a convex quadratic, it then finds the minimum of the latter and returns the lowest $f(\cdot)$ value observed. Note that the interpolation is only carried out when $f(x + \eta d) \le f(x)$, when $\eta = \tau$ or $\eta = -\tau$ (See Table IV).

We should advise the reader that the choice of α may influence the performance of the algorithm significantly; $\alpha = 1$ and a large initial φ value gives more diversity, and helps the algorithm escape from local minima; but this implies more function evaluations. On the contrary, a value of $\alpha = 0$ emphasizes points that improve $f(x)$. It tends to find the local minimum closest to the initial estimate. Table VI reports the number of times the global minimum was found and the number of function evaluations required by the algorithm for $\alpha = 0, \alpha = \min(\tau, \alpha)$ and $\alpha = 1$, on the 10-dimensional Rosenbrock banana function (8), which is considered as a moderately difficult problem by Törn et al. (1999). The Rosenbrock function is defined by:

$$f(x) = \sum_{k=1}^{n-1} \left[100 \left(x^{k+1} - (x^k)^2 \right)^2 + (1 - x^k)^2 \right]. \tag{8}$$

Table IV. $y =$ INTERPOLATE(x, d, τ)

Input: $x, d \in IR^n, \tau > 0$	Comments
$d = \tau d, z_{\text{right}} = x + d, z_{\text{left}} = x - d$	
$y = \arg\min(f(z_{\text{right}}), f(z_{\text{left}}))$	
IF $(f(y) > f(x))$	no interpolation
RETURN $f(y)$ and its argument y	
ELSEIF $f(y) = f(z_{left})$	change sense
$d = -d$	
ENDIF	
$\gamma = 2, z = x + d, z_{left} = x, z_{right} = z + 2d$	
WHILE $2f(z_{\text{left}}) + f(z_{\text{right}}) - 3f(z) \leq 0$	no convexity
$\gamma = 2\gamma$	
$z_{\text{left}} = z, z = z_{\text{right}}, z_{\text{right}} = z + \gamma d$	
ENDWHILE	
$z_{\min} = z + \dfrac{4f(z_{\text{left}}) - f(z_{\text{right}}) - 3f(z)}{4f(z_{\text{left}}) + 2f(z_{\text{right}}) - 6f(z)}\, d$	
$y = \arg\min(f(z_{\min}), f(z), f(z_{\text{right}}))$	
RETURN $f(y)$ and its argument y	

The test was carried out on 3 different initial points:

$$x_s = (-1.2, 1, -1.2, 1, \ldots, -1.2, 1),$$
$$x_g = (1, 1, 1, \ldots, 1, 1, 1),$$
$$x_l = (-0.9933, 0.9966, 0.9982, 0.9990, 0.9992, 0.9991, 0.9985, 0.9971, 0.9942, 0.9884),$$

which are respectively the standard initial point, the global minimum and a local minimum with 4 digits of precision. Each case was run 100 times. The results shown on table VI reveal that the nonmonotone feature may become quite useful for global minimization. It converged with a suitable number of function evaluations to the global minimum in 46% of the cases when it started at the local minimum x_l. Likewise, the algorithm escaped from the global minimum over 20% of the cases. Safeguards should be included to recover the best minimum found.

A similar test was repeated for the 5-dimensional Shekel's Foxhole function (Equation (9), Table V), which is considered as a difficult problem by Törn et al. (1999).

$$f(x) = -\sum_{j=1}^{30} \frac{1}{c_j + \sum_{i=1}^{5} (x_i - A_{ji})^2}. \tag{9}$$

Table V. Data for Shekel's function (9)

c	A				
0.806	9.681	0.667	4.783	9.095	3.517
0.517	9.400	2.041	3.788	7.931	2.882
0.100	8.025	9.152	5.114	7.621	4.564
0.908	2.196	0.415	5.649	6.979	9.510
0.965	8.074	8.777	3.467	1.863	6.708
0.669	7.650	5.658	0.720	2.764	3.278
0.524	1.256	3.605	8.623	6.905	4.584
0.902	8.314	2.261	4.224	1.781	4.124
0.531	0.226	8.858	1.420	0.945	1.622
0.876	7.305	2.228	1.242	5.928	9.133
0.462	0.652	7.027	0.508	4.876	8.807
0.491	2.699	3.516	5.874	4.119	4.461
0.463	8.327	3.897	2.017	9.570	9.825
0.714	2.132	7.006	7.136	2.641	1.882
0.352	4.707	5.579	4.080	0.581	9.698
0.869	8.304	7.559	8.567	0.322	7.128
0.813	8.632	4.409	4.832	5.768	7.050
0.811	4.887	9.112	0.170	8.967	9.693
0.828	2.440	6.686	4.299	1.007	7.008
0.964	6.306	8.583	6.084	1.138	4.350
0.789	0.652	2.343	1.370	0.821	1.310
0.360	5.558	1.272	5.756	9.857	2.279
0.369	3.352	7.549	9.817	9.437	8.687
0.992	8.798	0.880	2.370	0.168	1.701
0.332	1.460	8.057	1.336	7.217	7.914
0.817	0.432	8.645	8.774	0.249	8.081
0.632	0.679	2.800	5.523	3.049	2.968
0.883	4.263	1.074	7.286	5.599	8.291
0.608	9.496	4.830	3.150	8.270	5.079
0.326	4.138	2.562	2.532	9.661	5.611

Shekel's function exhibits multiple local minima and its global minimum is

$$x = (8.0249, 9.1517, 5.1139, 7.6209, 4.5641), \quad f(x) = -10.406.$$

The global minimum for the Shekel function was rarely found (1–3% in several runs) and convergence to more than 12 different minima was observed regardless of the α value used. However, when the method started on a local minimum and $\alpha = 0$ it could not escape from that minimum.

We now turn our attention to the CGLS described in Table II and implemented as shown in Table VII. We chose $Q = (\mathcal{P} - \{x_b, x_w\})$. For each $x \in Q$ the algorithm fuses a **global** strategy with the local NM search. By the use of a finite number of iterations of NM it exploits the property that NM has of escaping from a local nonglobal minimum. To also take into account a **global** strategy it uses the downhill direction $u = x - x_w$ as direction generator of the search set $D = \{\pm u, \pm d_1, \ldots, \pm d_n\}$ and attempts to find some $y \in \{z \in S : z = x + \eta d, \eta \geq \tau, d \in D\}$ satisfying (4). This set D tends

Table VI. Global minimum and function evaluations

Report on the 10-dimensional Rosenbrock function (8) ($\epsilon = 10^{-8}$, $\tau = 1$, $\varphi = 200$, 100 runs each case)			
starting point	$\alpha = 0$	$\alpha_{i+1} = \min(\alpha_i, \tau_i)$	$\alpha = 1$
standard	70	80	87
global	100	75	87
local	0	46	46
Global minimum found (%)			
starting point	$\alpha = 0$	$\alpha_{i+1} = \min(\alpha_i, \tau_i)$	$\alpha = 1$
standard	7180	7378	87710
global	353	4690	66329
local	374	4780	74165
Function evaluations (Average)			

to spread the population on points away from the worst point and successfully introduced the necessary global character to the algorithm. We also tried the downhill direction $u = x_b - x$, but results not reported here were rather discouraging. It seems that this generator has the undesirable effect of attracting the population towards x_b. Numerically it was also worthwhile to exclude x_b from the set $\mathcal{Q}$. As x_b and its offspring may all eventually belong to the population, they may cause a cluster of points akin to x_b, which is undesirable when x_b is close to a local nonglobal minimum. This seems to occur quite often for Shekel and other functions with multiple local minima. The worst point x_w was left out for mere convenience. This simply avoids to set $u = x - x_w = 0$ when $x = x_w$. As a safeguard though we generate u randomly when $||x - x_w|| < \tau$. The rest of the algorithm is sketched in Table VII. To end the algorithm we perform a complete local NM search starting at x_b. The convergence theorem justifies this final step. We ran some numerical tests for the Rosenbrock and the Shekel functions with an initial population randomly generated. As previously commented, $\alpha = 0$ works better for Shekel and $\alpha_{i+1} = \min(\alpha_i, \tau_i)$ is better for Rosenbrock. Table VIII shows the number of function evaluations and the percentage of times that the global minimum is attained for both functions. Each case was run 100 times. The results are highly competitive when compared with numerical tests that have been recently reported in the open literature (Ali et al., 1997; Ali and Törn, 2004). Research is ongoing to test the CGLS approach on a wider range of academic and practical problems and different **global** strategies.

Table VII. CGLS actual implementation: Local NM to $x \in \mathcal{Q}$ with **global** D

Input: $\epsilon = 10^{-6}, \mu = 0.3, \gamma = 1.3, \tau = 0.1, 0 \leq \alpha \leq 1$, random $\mathcal{P}$

```
FOR x ∈ (P − {x_b, x_w})
  u = x − x_w. If ||u|| < τ, generate u randomly
  d_0 = u, d_j = e_j − (2u^j)u, j = 1, ..., n
  success = 0
  FOR j = 0 TO n
    y = INTERPOLATE(x, d_j, τ), α = min(α, τ)
    IF f(y) ≤ f(x) + α(f(x_w) − f(x)) − 0.1τ^2
      x = y, success = success + 1
    ENDIF
  ENDFOR
  IF success = 0
    τ = μτ
  ELSE
    P_{i+1} = P_i + {x} − {x_w}
    Recompute x_w, x_b
    IF success > n/2
      τ = γτ
    ENDIF
  ENDIF
ENDFOR
REPEAT UNTIL (τ < ε) OR (f(x_w) < f(x_b) + ε(|f(x_b)| + 1))
Perform a final local NM search starting at x_b
```

Table VIII. Average CGLS function evaluations(% Global minimum found)

function	Population size 20	50	100	500
Rosenbrock	25358(99%)	55310(100%)	111340(100%)	
Shekel	2065(21%)	4124(46%)	7109(74%)	24718(100%)

This paper includes a companion Java implementation (Burguillo et al., 2004) that can be accessed and run directly from the Web

http://www.det.uvigo.es/~jrial/Proyectos/Global/Global.html

and the interested scholar may analyze the influence of the parameter values on the performance of our implementation on various test functions, including an *easy* 2-dimensional quadratic function, the moderately difficult Rosenbrock banana function (8) with an adjustable number of variables, and the difficult 5-dimensional Shekel's Foxholes function (Equation (9) and Table V) taken from Ali and Törn (2004), which has multiple local minima.

5. Final Remarks

The key factor in our convergence analysis was the nonmonotone sufficient decrease (NMSD) condition, which does not force a monotone decrease in function values. It was observed numerically that the nonmonotone (NM) algorithm may converge to a global minimum, even when the starting point is a local *nonglobal* minimum. Based on this property we propose to use (NM) on individuals in a population $\mathcal{P}$ randomly generated in the domain of $f(\cdot)$. Preliminary numerical results obtained on a modrately difficult function (Rosenbrock) and on a difficult function (Shekel) are remarkable when the number of function evaluations and the number of times the global minimum is attained are taken as the performance indices. We suppose that nonmonotone algorithms for differentiable functions should be explored in this context.

As it has been stated, there is in general no theoretical way to ensure convergence to a global minimum; however, the numerical experiments carried out so far may imply that our approach satisfies key features (Khompatraporn et al., 2004). It is

general, because the global minimum may be attained under mild conditions,
reliable, because the global minimum may be located with better performance indices than those found in the literature,
easy to use, because novice users can readily implement the algorithm, and
efficient, because the nonmonotone derivative-free algorithm for unconstrained minimization, which is the core of our study, seems to be highly competitive among those that do not admit gradient evaluation nor approximation.

We end this section with a remark on constrained problems. It is straightforward to prove that convergence theorem 2 holds when S is the bound constraint convex set $S = \{x \in I\!R^n : s \leq x \leq t\}$, D is the set of unit vectors along the coordinate axes and at blocked *feasible* points we require that

$$d \in D \Rightarrow f(P_S(x_i + \tau_i d)) > \phi_i - \nu(\tau_i) \geq f(x_i) - \nu(\tau_i), \tag{10}$$

where $P_S(x)$ is the projection of x onto S. The next step in this line of research is to analyze if (10) is sufficient for solving convex systems. Penalty functions, that transform a constrained model into an unconstrained problem might also be useful for solving more difficult constrained optimization problems. Another approach that will deserve our attention in future research is to evaluate $f(\cdot)$ at feasible random points close to one individual x. If no success is obtained after a predetermined number of

trials we search on another individual. If no individual generates a point not too much worse than himself we decrease $\nu(\cdot)$. The population is updated as described for the unconstrained case.

References

1. Ali, M.M., Storey, C. and Törn, A. (1997), A numerical comparison of some modified controlled random search algorithms, *Journal of Global Optimization* 11, 377–385.
2. Ali, M.M. and Törn, A. (2004), Population set-based global optimization algorithms: some modifications and numerical results, *Computers and Operations Research* 31, 1703–1725.
3. Bagirov, Adil M., Rubinov, Alexander M. and Yearwood, John, (2002), A global optimization approach to classification, *Optimization and Engineering* 3, 129–155.
4. Burguillo, Juan C., García-Palomares, Ubaldo M. and González-Castaño, Francisco J. (2004), Java implementation of the CGLS algorithm, http://www.det.uvigo.es/~jrial/Proyectos/Global/Global.html
5. Csendes, T. and Ratz, D. (1997), Subdivision direction selection in interval methods for global optimization, *SIAM Journal on Numerical Analysis* 34, 922–938.
6. García-Palomares, Ubaldo M. and Rodríguez, José F. (2002), New sequential and parallel derivative-free algorithms for unconstrained optimization, *SIAM Journal on optimization* 13, 79–96.
7. Ge, R. (2000), A filled function method for finding a global minimizer of a function of several variables, *Mathematical Programming* 46, 191–204.
8. Gray, P., Hart, W., Painton, L., Phillips, C., Trahan, M. and Wagner, J. (1997), A Survey of Global Optimization Methods, Sandia National Laboratories, http://www.cs.sandia.gov/opt/survey/main.html
9. Grippo, L. Lampariello, F. and Lucidi, S. (1986), A nonmonotone line search technique for Newton's method, *SIAM Journal on Numerical Analysis* 23, 707–716.
10. Groenwold, Albert A. and Snyman, J.A. (2002), Global Optimization using Dynamic Search Trajectories, *Journal of Global Optimization* 24, 51–60.
11. Haim, D. Giunta, A.A., Holzwarth, M.M., Mason, W.H., Watson, L.T. and Haftka, R.T. (1999), Comparison of optimization software packages for an aircraft multidisciplinary design optimization problem, *Design optimization* 1, 9–23.
12. Horst, R. and Pardalos, Panos M. (1994), *Handbook of Global Optimization* Vol. 1, Kluwer Academic Publishers, ISBN 0-7923-3120-6.
13. Huyer, W. and Neumaier, A. (1999), Global optimization by multilevel coordinate search, *Journal of Global Optimization* 14, 331–355.
14. Isenor, G. Pintér, János D. and Cada, M. (2003), A global optimization approach to laser design, *Optimization and Engineering* 4, 177–196.
15. Jones, Donald R. (2001), A taxonomy of global optimization methods on response surfaces, *Journal of Global Optimization* 21, 345–383.
16. Jones, Donald R., Pertunen, C.D. and Stuckman B.E. (1993), Lipschitzian optimization without the Lipschitz constant, *Journal of Optimization Theory and Applications* 79, 157–181.
17. Khompatraporn, C. Zabinsky, Z.B. and Pintér J.D. (2004), Comparative assessment of algorithms and software for global optimization, *Jurnal of Global Optimization*, to appear
18. Hooke, R. and Jeeves, T.A. (1961), Direct search solution of numerical and statistical problems, *Journal ACM* 8, 212–229.

19. Kolda, Tamara G., Lewis, R.M. and Torczon V. (2003), Optimization by direct search: new perspectives on some classical and modern methods, *SIAM Review* 45–3, 385–482.
20. Levy, A.V. and Gómez S. (1985), The tunneling method applied for the global minimization of functions. In: Boggs, Paul T. (ed.), *Numerical Optimization*, SIAM, 213–244.
21. Liu, X. and Xu, W. (2004), A new filled function applied to global optimization, *Computers and Operations Research* 31, 61–80.
22. Locatelli, M. and Schoen, F. (2002), Fast global optimization of difficult Lennard-Jones clusters, *Computational Optimization and Applications* 21, 55–70.
23. Pardalos, Panos M. and Edwin Romeijn, H. (2002), *Handbook of Global Optimization* Vol. 2, Kluwer Academic Publishers, ISBN 1-4020-0632-2.
24. Pintér J.D. (1995), *Global Optimization in Action (Continuous and Lipschitz optimization: Algorithms, Implementations and Applications)*, Kluwer Academic Publishers, ISBN 0-7923-3757-3.
25. Price, W.L. (1977), A controlled random search procedure for global optimization, *The Computer Journal* 20, 367–370.
26. Price, W.L. (1983), Global optimization by controlled random search, *Journal of Optimization Theory and Applications* 40, 333–348.
27. Price, W.L. (1987), Global optimization algorithm for a CAD workstation, *Journal of Optimization Theory and Applications* 55, 133–146.
28. Smith, Sharon L., Eskow, E. and Schnabel, Robert B. (1990), Adaptive, asynchronous stochastic global optimization algorithms. In: Coleman, Thomas F. and Li, Y. (eds.), *Large-scale Numerical Optimization*, SIAM, ISBN 0-89871-268-8, pp. 207–227.
29. Stephens, C.P. and Baritompa W. (1998), Global optimization requires global information, *Journal of Optimization Theory and Applications* 96–3, 575–588.
30. Toint, P.L. (1997), Non monotone trust region algorithms for non linear optimization subject to convex constraints, *Mathematical Programming* 77, 69–94.
31. Törn, A. and Žilinskas, A. (1989), *Global Optimization*, Springer Verlag, Berlin, Germany.
32. Törn, A., Ali, M. and Viitanen S. (1999), Stochastic global optimization problem classes and solution techniques, *Journal of Global Optimization* 14, 437–447.
33. Ulbrichm, M. (2001), Nonmonotone trust-region methods for bound-constrained semismooth equations with applications to nonlinear mixed complementarity problems, *SIAM Journal on Optimization* 11–4, 889–917.
34. Wright, M.H. (1996), Direct search methods: Once scorned, now respectable. In: Griffiths, D.F. and Watson G. (eds.) *Numerical analysis 1995*, Vol.344, Pittman research notes, pp. 191–208.
35. Yiu, K.F.C., Liu, Y. and Teo, K.L. (2004), A hybrid descent method for global optimization, *Journal of Global Optimization* 28, 229–238.