Computers & Operations Research 51 (2014) 338–349

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Test problem generator for unconstrained global optimization ☆

Chi-Kong Ng *, Duan Li

Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong S.A.R., PR China

ARTICLE INFO

Available online 28 June 2014

Keywords:
Global optimization
Large-scale optimization
Test problems
Fourth degree polynomials

ABSTRACT

We develop in this paper a high performance test problem generator for generating analytic and highly multimodal test problems for benchmarking unconstrained global optimization algorithms. More specifically, we propose in this research a novel and computationally efficient procedure for generating nonlinear nonconvex not separable unconstrained test problems with (i) analytic test functions, (ii) known local minimizers that are distributed uniformly in the interior of a compact box, among which only one is the global solution, and (iii) controllable difficulty levels. A standard set of test problems with different sizes and different difficulty levels is produced for both MATLAB and GAMS and is available for downloading. Numerical experiments have demonstrated the stability of the generating process and the difficulty of solving the standard test problems.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Optimization, as a powerful search engine in decision-making, finds wide applications in almost all fields of engineering, finance, and management as well as social science. The existence of multiple local minima of a general nonconvex objective function makes global optimization a great challenge. Since the publication of the two-volume books entitled "Towards Global Optimization" in 1975 [9] and 1978 [10], the study of global optimization has grown by leaps and bounds. The last four decades have witnessed rapid development in both theory and numerical techniques for global optimization. Nevertheless, it is difficult to compare the efficiencies of different solution algorithms directly. Thus, empirical computational testing is always necessary.

Construction of test problems for benchmarking global optimization algorithms is a challenging task. There are three main types of test problems in the literature: problems modeling a variety of real-world applications, problem instances with certain designated characteristics, and randomly generated test problems with known solutions. Excellent collections of test problems of the first two types can be found in [12,13,20,29,33], for example. For the third type, some test problem generators have been proposed for constrained global optimization (see, e.g., [5,11,27,31,32]. However, the generation of nontrivial test problems for unconstrained global optimization algorithms seems to be difficult as evidenced by the fact that very few papers have addressed this subject.

Unconstrained global optimization is an important subject within the optimization community. It not only forms the foundation of global optimization, but also has many applications in the real world. Various practical problems can be modeled by or be transformed into unconstrained global optimization problems. Examples include seismic analysis in earthquake and exploration seismologies; protein folding problems in biomedical sciences; solution to polynomial equations that arise frequently in symbolic computation, algebraic geometry and computer algebra; and satisfiability problems that arise frequently in computer vision, VLSI design and computer-aided design. Numerous papers in the literature have been devoted to the development of theories and/or algorithms for unconstrained global optimization, including tunneling algorithms (see, e.g., [3,6,24]), filled function methods (see, e.g., [17,18,39]), multistart algorithms (see, e.g., [2,21,28]), simulated annealing algorithms (see, e.g., [7,19]), DIRECT algorithm (see, e.g., [14,22]), methods using Peano curves (see, e.g., [35,36]), and global descent method (see [30]). Benchmarking unconstrained global optimization algorithms is thus an important and interesting subject both from the theoretical and practical points of view.

In 1993, Schoen [34] proposed, probably, the first test problem generator for benchmarking unconstrained global optimization algorithms. Denote the usual Euclidean norm of a vector by $\|\cdot\|$. Suppose that k distinct points, $z_i \in (0,1)^n$, $i=1,\dots,k$, and their values, $f_i \in \Re$, $i=1,\dots,k$, are given. Assume further that k parameters, $a_i>1$, $i=1,\dots,k$, are also given. Schoen's generator generates the following family of functions:

$$f(x)=\frac{\sum_{i=1}^{k} f_i \prod_{j\neq i}\|x-z_j\|^{a_j}}{\sum_{i=1}^{k}\prod_{j\neq i}\|x-z_j\|^{a_j}} \quad \text{where } x\in[0,1]^n.$$

☆This research was partially supported by the Research Grants Council of Hong Kong, Grant number 414207.

* Corresponding author.
E-mail addresses: ckng@se.cuhk.edu.hk (C.-K. Ng), dli@se.cuhk.edu.hk (D. Li).

http://dx.doi.org/10.1016/j.cor.2014.06.019
0305-0548/© 2014 Elsevier Ltd. All rights reserved.

It can be proved that the k points, z_i, $i=1,\dots,k$, are stationary points, and $\min_i f_i \le f(x) \le \max_i f_i$, $\forall x \in [0,1]^n$. Moreover, if $a_i \ge 2 \in \mathbf{Z}$, $\forall i \in \{1,\dots,k\}$, then $f \in C^\infty$. Note, however, that there is no prior knowledge or ways to control the properties of the stationary points, in general.

In 1998, Gaviano and Lera [16] introduced a family of functions that included a class of C^1 and a class of C^2 test functions with a priori known local minimizers and their regions of attraction (also known as basins) (see, e.g., [8,17] for the definition of the region of attraction of a minimizer). They, with other coworkers, further extended the family to include a class of C^0 test functions in [15] in 2003. The idea of their method is to dig some deep holes in the hillside of a convex quadratic function. The width of a hole is actually the region of attraction of its minimum. For the class of C^0 test functions, they used quadratic polynomials to maintain the continuity of the resulting class of composite functions. For the class of C^1 test functions, they used cubic polynomials to maintain the first-order continuous differentiability. While for the class of C^2 test functions, they used quintic polynomials to maintain both the first- and second-order continuous differentiabilities. Denote the box-constrained admissible region of x by X. Suppose that k distinct points, $z_i \in \text{int}(X)$, $i=1,\dots,k$, and their values, $f_i \in \Re$, $i=1,\dots,k$, are given. Let $S_i = \{x \in \Re^n : \|x-z_i\| \le \rho_i,\ \rho_i > 0\}$, $i=2,\dots,k$, where ρ_i, $i=2,\dots,k$, are chosen such that $S_i \cap S_j = \varnothing$, $\forall i \ne j$. Their generator produces the following family of functions:

$$f(x) = \begin{cases} g_i(x), & x \in S_i \ (i \in \{2,\dots,k\}), \\ \|x-z_1\|^2 + f_1, & x \notin S_2 \cup \cdots \cup S_k, \end{cases}$$

where g_i, $i=2,\dots,k$, are suitable quadratic, cubic or quintic polynomials for C^0, C^1 or C^2-class test functions, respectively. Interested readers can find the details of these polynomials in [15]. As mentioned in their papers, all three classes have many parameters to be coordinated. The correlations of the parameters indeed do not allow simple and fast generation. The parameter coordination problem was finally relaxed by their test problem generator presented in [15]. Since they provided a complete mechanism for tuning the parameters, from our point of view, their generator is so far the most controllable one that is capable of generating numerous test problems. Nevertheless, their generator has two obvious weaknesses. First, the region of attraction of z_1 is not well defined since the hillside of the convex quadratic function, $\|x-z_1\|^2 + f_1$, has been destroyed by the holes $S_2,\dots,S_k$. Moreover, it is well known within the global optimization community that just the value, and the first- and second-order derivatives of a function, in general, do not provide sufficient information for finding a global minimizer of the function. Since the functions in Gaviano et al. are composite functions with at most second-order continuous differentiability, it can be seen from the definition of f that no information about the function g_i, $i=2,\dots,k$, is available at any $x \notin S_2 \cup \cdots \cup S_k$. It can thus never provide sufficient information for locating the minimizers $z_2,\dots,z_k$. Without doubt, this property makes their test functions hard to be minimized globally and thus excellent for competitive testing. However, due to the same reason, only a few papers in the literature have actually adopted it for a comparison purpose of global optimization algorithms. We thus believe that analytic functions are better choices of test problems for benchmarking global optimization algorithms.

In 2007, Addis and Locatelli [1] proposed a test problem generator for generating a class of functions that are analogous with the molecular conformation problems. They first defined two types of one-dimensional component with multiple local minimizers obtained through some oscillation terms based on cosine functions. By addition and linear transformation of several one-dimensional components with different parameters, they obtained a class of not separable n-dimensional component, namely basic component function. Through their combination operations on two instances of the basic component function, they finally obtained a basic test function. In fact, by using basic component functions or the result of previous combination operations as parameters of their combination operations, more test functions can be obtained. As their functions contain numerous components and each component contains several parameters, interested readers should refer to [1] for more details. Essentially, one of the two types of one-dimensional component in [1] is

$$d_{p,K}(x) = \xi_p(x) + O^{K,H}_{c_1,c_2}(x)$$

where p, K, c_1, c_2, and H are parameters, $O^{K,H}_{c_1,c_2}(x)$ is an oscillation term based on a cosine function, and $\xi_p(x)$ is a composite function with continuous first derivative. Therefore, their generator produces at most C^1 test problems. It is evident that some significant difficulties still exist to block the generation of analytic functions for benchmarking unconstrained global optimization algorithms, notwithstanding the above-mentioned promising progress.

The main purpose of this research is to develop a high performance test problem generator for generating analytic and highly multimodal test problems for benchmarking unconstrained global optimization algorithms. Unlike Schoen's functions in [34], the proposed test problems have a priori known minimizers, a maximizer and saddle points as well as their values. Unlike Gaviano et al.'s functions in [15,16] and Addis and Locatelli's function in [1], the proposed test problems are analytic polynomial-type functions; the parameters of the proposed test problems can be generated directly without any coordination problem. This paper indeed describes a novel and computationally efficient procedure for generating nonlinear nonconvex not separable unconstrained test problems with (i) analytic test functions, (ii) known minimizers that are distributed uniformly in the interior of a compact box, and (iii) controllable difficulty levels.

This paper is organized as follows. In Section 2, we first construct n univariate nonlinear nonconvex unconstrained minimization problems with known optimal solutions. Combining the n univariate problems, we produce a separable nonlinear nonconvex unconstrained n-variable problem. In Section 3, we disguise the separability of the problem and introduce randomness by adopting a technique from Calamai et al. [5] to obtain a not separable function by linear transformations of variables of a separable function. Since many algorithms for solving unconstrained global minimization problems require that all local minimizers of the objective function be contained in the interior of a compact box, we pay special attentions to this issue in Section 4. We devote Section 5 to discuss the realization of the generator and the developed software packages, while generation of random numbers, control of the difficulty levels of the test problems, description of the software packages, and summary of the parameters are the topics of this section. In Section 6, we report the results of the stability tests on the standard test problems. We also report the results of some test experiments with a multistart algorithm and with a deterministic global optimization solver, namely GAMS/BARON. Finally, we draw some conclusions in Section 7.

2. Generation of separable nonlinear nonconvex unconstrained test problems

2.1. Construction of separable problems

Consider a general class of separable problems for unconstrained global minimization which takes the following form:

$$(P) \quad \min_{x \in \Re^n} f(x),$$

where

$$f(x)=\sum_{i=1}^{n} a_i f_i(x_i), \tag{1}$$

$x=[x_1,\ldots,x_n]^T\in\Re^n$ is a vector of n decision variables, each $f_i(x_i)$ $(\forall i\in\{1,\ldots,n\})$ is a real-valued function, and each a_i $(\forall i\in\{1,\ldots,n\})$ is a positive constant.

It is evident that $x^*=[x_1^*,\ldots,x_n^*]^T$ is a stationary point of $f(x)$ if and only if x_i^* is a stationary point of $f_i(x_i)$ for all $i\in\{1,\ldots,n\}$. Furthermore, x^* satisfies the sufficient conditions to be a strict local minimizer (maximizer) of $f(x)$ if and only if x_i^* satisfies the sufficient conditions to be a strict local minimizer (maximizer) of $f_i(x_i)$ for all $i\in\{1,\ldots,n\}$.

As problem (P) can be decomposed into the following n univariate subproblems:

$$(P_i)\quad \min_{x_i\in\Re} f_i(x_i),\quad i=1,\ldots,n,$$

we now consider the generation of the ith subproblem (P_i). It is believed that good candidates for $f_i(x_i)$ should have the following four properties:

1. $f_i(x_i)$ is of a simple structure for easy analysis.
2. $f_i(x_i)$ is a twice continuously differentiable function and preferably an analytic function.
3. $f_i(x_i)$ is a coercive function, i.e., $f_i(x_i)\to\infty$ as $|x_i|\to\infty$. (Note that the continuity and coercivity properties guarantee that all global minimizers of $f_i(x_i)$ are finite.)
4. $f_i(x_i)$ possesses more than one local minimizer.

It is well known that a polynomial with at least third degree is one of the simplest forms for univariate nonlinear nonconvex functions, whereas a cubic polynomial is, in general, not coercive, a quartic polynomial with a positive coefficient at its fourth degree term is a coercive function. Moreover, a classical result in 1970 (see [4]) proved that "An (unconstrained) global optimization method for quartic polynomials can solve (system of) polynomial equations of any degree." Thus, it is worth paying special attention to quartic-polynomial-type objective functions in unconstrained global optimization. With the above understanding, we now confine $f_i(x_i)$ to a quartic polynomial:

$$f_i(x_i)=x_i^4+4p_ix_i^3+6q_ix_i^2+s_ix_i, \tag{2}$$

where p_i, q_i and s_i are real constants. Note that the first-order necessary condition for achieving the optimality of f_i is

$$f_i'(x_i)=4x_i^3+12p_ix_i^2+12q_ix_i+s_i=0. \tag{3}$$

Note also that, if α_i is a real root of $f_i'(x_i)$, then (3) gives

$$s_i\equiv s_i(\alpha_i)=-4\alpha_i(\alpha_i^2+3p_i\alpha_i+3q_i).$$

Theorem 1. *Let α_i be a strict local minimizer of $f_i(x_i)$. Then, $f_i(x_i)$ has two distinct strict local minimizers if and only if the following conditions hold:*

$$p_i^2>q_i;$$

$$\alpha_i\in(-p_i-2r_i,-p_i-r_i)\cup(-p_i+r_i,-p_i+2r_i), \tag{4}$$

where $r_i=\sqrt{p_i^2-q_i}>0$. Let

$$\beta_i\equiv\beta_i(\alpha_i)=\frac{-(3p_i+\alpha_i)-\sqrt{\Delta_i(\alpha_i)}}{2}, \tag{5}$$

$$\gamma_i\equiv\gamma_i(\alpha_i)=\frac{-(3p_i+\alpha_i)+\sqrt{\Delta_i(\alpha_i)}}{2}, \tag{6}$$

where

$$\Delta_i(\alpha_i)=3(2r_i+p_i+\alpha_i)(2r_i-p_i-\alpha_i)>0. \tag{7}$$

1. *If $\alpha_i\in(-p_i-2r_i,-p_i-r_i)$, then*
$$-p_i-2r_i<\alpha_i<-p_i-r_i<\beta_i<-p_i+r_i<\gamma_i<-p_i+2r_i, \tag{8}$$
where α_i and γ_i are strict local minimizers of $f_i(x_i)$ and β_i is the strict local maximizer of $f_i(x_i)$; and
$$F_i(\gamma_i(\alpha_i),\alpha_i)\equiv f_i(\gamma_i(\alpha_i))-f_i(\alpha_i)$$
is a strictly decreasing function of α_i. Moreover, if $\alpha_i\in(-p_i-2r_i,-p_i-r_i\sqrt{3})$, then $f_i(\alpha_i)<f_i(\gamma_i)$.
2. *If $\alpha_i\in(-p_i+r_i,-p_i+2r_i)$, then*
$$-p_i-2r_i<\beta_i<-p_i-r_i<\gamma_i<-p_i+r_i<\alpha_i<-p_i+2r_i, \tag{9}$$
where β_i and α_i are strict local minimizers of $f_i(x_i)$ and γ_i is the strict local maximizer of $f_i(x_i)$; and
$$F_i(\beta_i(\alpha_i),\alpha_i)\equiv f_i(\beta_i(\alpha_i))-f_i(\alpha_i)$$
is a strictly increasing function of α_i. Moreover, if $\alpha_i\in(-p_i+r_i\sqrt{3},-p_i+2r_i)$, then $f_i(\alpha_i)<f_i(\beta_i)$.

Proof. First note that the cubic polynomial $f_i'(x_i)$ in (3) has exactly three roots: either three real roots or one real and one complex conjugate pair of roots. It is evident that any point of inflection of $f_i(x_i)$ is a double root of $f_i'(x_i)$. By the continuity and the coercivity of $f_i(x_i)$, we conclude that exactly one of the following situations must hold:

1. Let $x_i^L<\hat{x}_i<x_i^R$. Then, $f_i'(x_i)$ has three distinct real roots x_i^L, $\hat{x}_i$ and x_i^R if and only if x_i^L and x_i^R are strict local minimizers of $f_i(x_i)$ and $\hat{x}_i$ is the strict local maximizer of $f_i(x_i)$.
2. $f_i'(x_i)$ has a double root if and only if $f_i(x_i)$ has a single local minimizer and a point of inflection.
3. $f_i'(x_i)$ has a triple root or a complex conjugate pair of roots if and only if $f_i(x_i)$ has one local minimizer only.

Now, to prove the condition that if $f_i(x_i)$ has two distinct strict local minimizers then $p_i^2>q_i$, we can prove its equivalent condition that if $p_i^2\le q_i$ then $f_i(x_i)$ has only one local minimizer. Note that if either $p_i^2<q_i$ or ($p_i^2=q_i$ and $x_i\ne -p_i$), then $f_i''(x_i)>0$, $\forall x_i\in\Re$. Thus, $f_i(x_i)$ has no local maximizer. If $p_i^2=q_i$ and $x_i=-p_i$, then $-p_i$ is the triple root of $f_i'(x_i)$. From the above possible situations the results are sustained.

Now, let $p_i^2>q_i$ and $r_i=\sqrt{p_i^2-q_i}>0$. Suppose that α_i, $\hat{x}_i$ and x_i^* are the three distinct real roots of $f_i'(x_i)$. Let

$$f_i'(x_i)=4(x_i-\alpha_i)(x_i-\hat{x}_i)(x_i-x_i^*).$$

By comparing the coefficients of x_i^2 and that of x_i in the above equation and in (3), and eliminating x_i^* in the results of the above comparisons by substitution, we have

$$\hat{x}_i^2+(3p_i+\alpha_i)\hat{x}_i+(\alpha_i^2+3p_i\alpha_i+3q_i)=0.$$

Its discriminant is $\Delta_i(\alpha_i)$ in (7). β_i and γ_i in (5) and (6) are the solutions to the above quadratic equation, respectively, and are the distinct real roots of $f_i'(x_i)$ if and only if $\Delta_i(\alpha_i)>0$, that is,

$$|\alpha_i+p_i|<2r_i.$$

Moreover, from the second-order necessary conditions, if α_i is a strict local minimizer of $f_i(x_i)$, then $|\alpha_i+p_i|\ge r_i$. However, if $\alpha_i=-p_i\pm r_i$, then α_i is a double root of $f_i'(x_i)$ and thus it is a point of inflection of $f_i(x_i)$. Combining these results with the second-order sufficient conditions, we conclude that α_i is a strict local minimizer of $f_i(x_i)$ if and only if

$$|\alpha_i+p_i|>r_i.$$

Now, combining these two conditions leads to (4).

We now consider case 1: $\alpha_i\in(-p_i-2r_i,\ -p_i-r_i)$. By elementary calculus, it is easy to show that $\beta_i(\alpha_i)$ in (5) is a strictly convex function of α_i with minimizer at $-p_i+r_i$. Hence

$$-p_i-r_i=\beta_i(-p_i-r_i)<\beta_i(\alpha_i)<\beta_i(-p_i-2r_i)=-p_i+r_i.$$

With the same technique, we can show that $\gamma_i(\alpha_i)$ in (6) is a strictly concave function of α_i with maximizer at $-p_i-r_i$. Hence

$$-p_i+r_i=\gamma_i(-p_i-2r_i)<\gamma_i(\alpha_i)<\gamma_i(-p_i-r_i)=-p_i+2r_i.$$

Combining these two results and the condition of case 1 yields (8). From the above situation 1, we conclude that $x_i^L\equiv\alpha_i$ and $x_i^R\equiv\gamma_i$ are strict local minimizers of $f_i(x_i)$ and $\hat{x}_i\equiv\beta_i$ is the strict local maximizer of $f_i(x_i)$. Next, we define

$$F_i(\gamma_i(\alpha_i),\alpha_i)\equiv f_i(\gamma_i(\alpha_i))-f_i(\alpha_i).$$

Since $\alpha_i\in(-p_i-2r_i,\ -p_i-r_i)$ in case 1, we have

$$\begin{aligned}\frac{dF_i(\gamma_i(\alpha_i),\alpha_i)}{d\alpha_i}\\ &=18(\alpha_i+p_i+r_i)(\alpha_i+p_i-r_i)\left[\alpha_i+p_i+\frac{(\alpha_i+p_i+2r_i)(\alpha_i+p_i-2r_i)}{\sqrt{\Delta_i(\alpha_i)}}\right]\\ &<0.\end{aligned}$$

Thus F_i is a strictly decreasing function of α_i in case 1. Moreover, the unique root of $F_i(\gamma_i(\alpha_i),\alpha_i)$ in the range specified in case 1 is $-p_i-r_i\sqrt{3}$. Therefore, if $\alpha_i\in(-p_i-2r_i,\ -p_i-r_i\sqrt{3})$, then it is the strict global minimizer of $f_i(x_i)$, while γ_i is the strict non-global minimizer of $f_i(x_i)$.

To prove case 2, we simply rename α_i in (8) as β_i, β_i in (8) as γ_i, and γ_i in (8) as α_i. We then have (9). Therefore, the remaining of case 2 also holds. □

Theorem 2. *Suppose that the assumptions in Theorem 1 hold. If*

$$\alpha_i\in(-p_i-2r_i,\ -p_i-r_i\sqrt{3})\cup(-p_i+r_i\sqrt{3},\ -p_i+2r_i),\tag{10}$$

for all $i\in\{1,\dots,n\}$, then $x^{**}=\alpha=[\alpha_1,\dots,\alpha_n]^T$ is the strict global minimizer of $f(x)$. Besides the strict global minimizer, $f(x)$ has 2^n-1 strict non-global minimizers, one strict local maximizer, and 3^n-2^n-1 saddle points.

Proof. Under the conditions of the theorem, each $f_i(x_i)$ ($\forall i\in\{1,\dots,n\}$) has three stationary points including two strict local minimizers and one strict local maximizer. Hence, the results are sustained.

2.2. Interconnections and specifications of parameters

On the generation of p: From Theorem 1, we see that all stationary points of $f_i(x_i)$ are contained in $[-p_i-2r_i,-p_i+2r_i]$. Thus $x_i=-p_i$ is the center of the foregoing interval. To ensure that the test problem generator to be unbiased to positive or negative values, we generate p_i uniformly in $[\underline{p},\overline{p}]\equiv[-\overline{p},\overline{p}]$ with $\overline{p}>0$. In the set of standard test problems, we set $\overline{p}=1$.

On the generation of q: The extent of the interval $[-p_i-2r_i,-p_i+2r_i]$ is $4r_i$, where $r_i=\sqrt{p_i^2-q_i}$. Since we generate $p_i\in[-\overline{p},\overline{p}]$ and its value is uncertain before generation, it is possible that $p_i\to0$ for some i in some instance of test problems. To ensure that the extent of the foregoing interval not to be too small, we generate q_i uniformly in $[\underline{q},\overline{q}]$ with $\underline{q}<\overline{q}\le-1$ for all i. In the set of standard test problems, we set $\underline{q}=-2$ and $\overline{q}=-1$.

On the generation of α in open sets: To ensure that each subproblem (P_i) has two local minimizers with α_i being its global minimizer, Theorem 1 shows that α_i must be generated such that

$$\alpha_i\in(-p_i-2r_i,\ -p_i-r_i\sqrt{3})\cup(-p_i+r_i\sqrt{3},\ -p_i+2r_i).$$

However, if α_i is generated "too close" to the boundary points of the above open intervals, the computer round-off errors may affect the stability of the generation. In view of this consideration, we generate α_i uniformly in a fraction α' of the middle parts of the open intervals, where $\alpha'\in(0,1)$ is a user-defined real number. In other words, we generate α_i uniformly in

$$\begin{aligned}&[-p_i-(2-l)r_i,\ -p_i-(\sqrt{3}+l)r_i]\\ &\quad\cup[-p_i+(\sqrt{3}+l)r_i,\ -p_i+(2-l)r_i],\end{aligned}\tag{11}$$

where

$$l=(1-\alpha')(2-\sqrt{3})/2.\tag{12}$$

In the set of standard test problems, we set $\alpha'=0.95$.

On the minimum eigenvalue and the condition number of $\nabla^2 f(x^{**})$, and the generation of a: Since $f(x)$ is a separable function, its Hessian matrix $\nabla^2 f(\alpha)$ is thus a diagonal matrix with its ith ($i=1,\dots,n$) element equal to

$$\lambda_i\equiv\lambda_i(\alpha_i)=a_if_i''(\alpha_i)=12a_i(\alpha_i^2+2p_i\alpha_i+q_i),\tag{13}$$

with $a_i>0$. Thus, for any given $i\in\{1,\dots,n\}$, $\lambda_i(\alpha_i)$ is a quadratic function with minimizer at $\alpha_i=-p_i$. Since we generate α_i uniformly in the region stated in (11), a lower bound and an upper bound for λ_i are, respectively, given by

$$\begin{aligned}\underline{\lambda}_i&=\lambda_i(-p_i-(\sqrt{3}+l)r_i)=\lambda_i(-p_i+(\sqrt{3}+l)r_i)\\ &=12[(\sqrt{3}+l)^2-1]a_ir_i^2,\\ \overline{\lambda}_i&=\lambda_i(-p_i-(2-l)r_i)=\lambda_i(-p_i+(2-l)r_i)\\ &=12[(2-l)^2-1]a_ir_i^2.\end{aligned}$$

To ensure that $\nabla^2 f(\alpha)$ has a sufficiently large minimum eigenvalue and a sufficiently small condition number, we generate a_i uniformly in $[\underline{a},\overline{a}]$ such that $1\le\underline{a}<\overline{a}$ and $\overline{a}/\underline{a}\le10$. Since we generate $p_i\in[-\overline{p},\overline{p}]$ with $\overline{p}>0$ and $q_i\in[\underline{q},\overline{q}]$ with $\underline{q}<\overline{q}\le-1$, the minimum eigenvalue of $\nabla^2 f(\alpha)$ is thus

$$\lambda^*=\min_i\lambda_i\ge\min_i\underline{\lambda}_i\ge12[(\sqrt{3}+l)^2-1]\underline{a}(-\overline{q})$$

and the 2-norm condition number of $\nabla^2 f(\alpha)$ is

$$\kappa_f=\frac{\max_i\lambda_i}{\min_j\lambda_j}\le\frac{\max_i\overline{\lambda}_i}{\min_j\underline{\lambda}_j}\le\frac{(2-l)^2-1}{(\sqrt{3}+l)^2-1}\cdot\frac{\overline{a}}{\underline{a}}\cdot\frac{\overline{p}^2-\underline{q}}{-\overline{q}}.\tag{14}$$

In the set of standard test problems, we set $\underline{a}=1$ and $\overline{a}=2$. Recall that we define l in (12), and we set $\overline{p}=1$, $\underline{q}=-2$ and $\overline{q}=-1$, thus

$$\lambda^*>24\quad\text{and}\quad\kappa_f<9.$$

2.3. Algorithm

Based on the theoretical results in Section 2.1 and the practical consideration in Section 2.2, an algorithm for generating a general class of separable problems for unconstrained global minimization which takes the form (P) is now described as follows.

Algorithm 1.

1. Input
   - n: the dimension of the problem;
   - $\underline{a}$ and $\overline{a}$ ($1\le\underline{a}<\overline{a}$ and $\overline{a}/\underline{a}\le10$): the lower and upper bounds for each component of the random n-vector a;
   - $\overline{p}$ (>0): the upper bound (i.e., the negative lower bound) for each component of the random n-vector p;

- $\underline{q}$ and $\overline{q}$ ($\underline{q} < \overline{q} \le -1$):the lower and upper bounds for each component of the random n-vector q;
- α' ($0 < \alpha' < 1$): a fraction in the middle parts of the open admissible regions of each α_i.

2. For each $i \in \{1, \ldots, n\}$, generate, uniformly, $a_i \in [\underline{a}, \overline{a}]$, $p_i \in [-\overline{p}, \overline{p}]$ and $q_i \in [\underline{q}, \overline{q}]$, and calculate $r_i = \sqrt{p_i^2 - q_i}$.
3. For each $i \in \{1, \ldots, n\}$, generate α_i uniformly in (11) and calculate

$$\hat{x}_i = \begin{cases} \beta_i & \text{if } -p_i - (2-l)r_i \le \alpha_i \le -p_i - (\sqrt{3}+l)r_i, \\ \gamma_i & \text{if } -p_i + (\sqrt{3}+l)r_i \le \alpha_i \le -p_i + (2-l)r_i, \end{cases}$$

where β_i, γ_i and l are defined in (5), (6) and (12), respectively.
4. Now, for any given $x \in \Re^n$, the objective function for problem (P) is $f(x)$ in (1), where $f_i(x_i)$ is defined in (2). Its gradient vector and Hessian matrix are, respectively, given by

$$\nabla f(x) = \begin{bmatrix} a_1 f_i'(x_1) \\ \vdots \\ a_n f_i'(x_n) \end{bmatrix} \quad \text{and} \quad \nabla^2 f(x) = \begin{bmatrix} a_1 f_1''(x_1) & & 0 \\ & \ddots & \\ 0 & & a_n f_n''(x_n) \end{bmatrix}$$

where

$$f_i'(x_i) = 4x_i^3 + 12p_i x_i^2 + 12q_i x_i + s_i,$$
$$f_i''(x_i) = 12(x_i^2 + 2p_i x_i + q_i).$$

Its strict global minimizer and global minimum value are, respectively, given by

$$x^{**} = \alpha = [\alpha_1, \ldots, \alpha_n]^T \quad \text{and} \quad f(\alpha).$$

Its strict local maximizer and local maximum value are, respectively, given by

$$\hat{x} = [\hat{x}_1, \ldots, \hat{x}_n]^T \quad \text{and} \quad f(\hat{x}).$$

The eigenvalues of $\nabla^2 f(x^{**})$ are λ_i in (13), for all i. The 2-norm condition number of $\nabla^2 f(x^{**})$ is κ_f in (14).

2.4. Example

Fig. 1 shows the contours of $f(x)$ for the standard test problem ngli001 (which is explained in detail in Section 5), where "$*$", "$\times$", "▲" and "●" indicate global minimizer, non-global minimizer, local maximizer and saddle point, respectively. For this example, its strict global minimizer and global minimum value are $x^{**} = \alpha \approx [-2.29, -2.34]^T$ and $f(\alpha) \approx -286.56$, while the minimum eigenvalue and the 2-norm condition number of $\nabla^2 f(x^{**})$ are, respectively, $\lambda^* \approx 112.07$ and $\kappa_f \approx 1.21$.

3. Generation of not separable nonlinear nonconvex unconstrained test problems

3.1. Construction of not separable problems

We now consider the generation of a general class of not separable problems for unconstrained global minimization which takes the following form:

$$(\overline{P}) \quad \min_{y \in \Re^n} g(y),$$

where $y = [y_1, \ldots, y_n]^T \in \Re^n$ is a vector of n decision variables and $g(y)$ is a real-valued function. To achieve this goal, we disguise the separability of the problem (P) and introduce randomness via some simple linear transformations of variables.

Note that

$$f(x) = \sum_{i=1}^{n} a_i x_i^4 + 4 \sum_{i=1}^{n} a_i p_i x_i^3 + 6x^T AQx + s^T Ax,$$

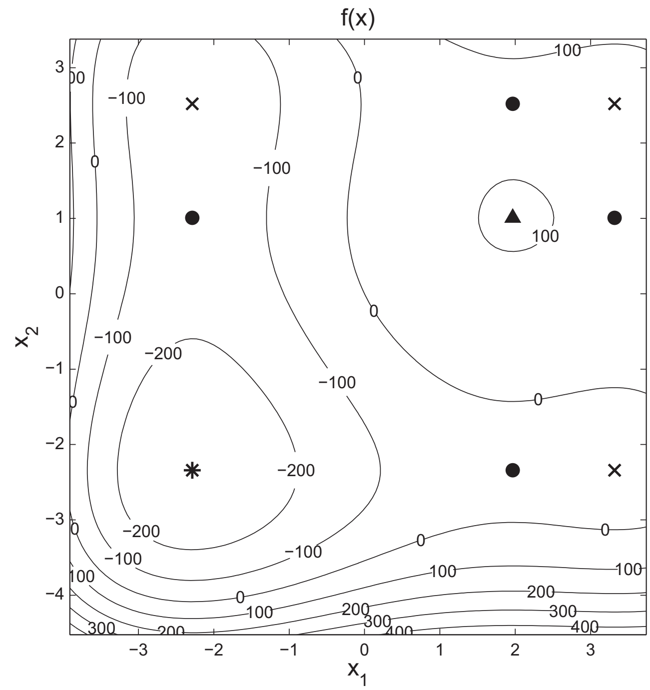

Fig. 1. Contours of $f(x)$ for the level 0 problem ngli001 ($*$=global minimizer, $\times$ =non-global minimizer, ▲=local maximizer, ●=saddle point).

where $A = \text{diag}(a_1, \ldots, a_n)$, $Q = \text{diag}(q_1, \ldots, q_n)$ and $s = [s_1, \ldots, s_n]^T$. Now, we let

$$x = DHy, \tag{15}$$

where $D = \text{diag}(d_1, \ldots, d_n)$ is a positive definite diagonal matrix with $d_i > 0$ for all i, and

$$H = I - 2vv^T$$

is a random Householder matrix with I being the n-dimensional identity matrix and $v^T v = 1$, where $v = [v_1, \ldots, v_n]^T \in \Re^n_+$ is a random non-negative vector. Then

$$x_i = d_i H_i y,$$

where H_i is the ith row of H. Since H is symmetric, we have

$$\begin{aligned} g(y) &\equiv f(DHy) \\ &= \sum_{i=1}^{n} a_i d_i^4 (H_i y)^4 + 4 \sum_{i=1}^{n} a_i p_i d_i^3 (H_i y)^3 + 6y^T (HDAQDH) y + s^T ADHy, \end{aligned}$$

which is a not separable function of y.

Note that if α_i satisfies (10) for all $i \in \{1, \ldots, n\}$, then $x^{**} = \alpha$ is the strict global minimizer of $f(x)$. Since H is a Householder matrix which satisfies

$$H^{-1} = H,$$

the transformation in (15) thus gives

$$y^{**} = H^{-1} D^{-1} x^{**} = HD^{-1}\alpha.$$

Therefore, $y^{**} = HD^{-1}\alpha$ is the strict global minimizer of $g(y)$ and $g(y^{**}) = f(x^{**}) = f(\alpha)$. Similarly, if x^*, $\hat{x}$ and $\tilde{x}$ are strict local minimizer, strict local maximizer and saddle point of $f(x)$, respectively, then $y^* = HD^{-1}x^*$, $\hat{y} = HD^{-1}\hat{x}$ and $\tilde{y} = HD^{-1}\tilde{x}$ are corresponding strict local minimizer, strict local maximizer and saddle point of $g(y)$, respectively, and $g(y^*) = f(x^*)$, $g(\hat{y}) = f(\hat{x})$ and $g(\tilde{y}) = f(\tilde{x})$.

Note also that

$$(DH)^T = HD.$$

Thus

$$\nabla g(y) \equiv \nabla_y f(DHy) = HD\ \nabla_x f(DHy),$$
$$\nabla^2 g(y) \equiv \nabla^2_{yy} f(DHy) = HD\ \nabla^2_{xx} f(DHy)\ DH.$$

3.2. Interconnections and specifications of parameters

On the minimum eigenvalue and the condition number of $\nabla^2 g(y^{**})$, and the generation of D:

Theorem 3. *The eigenvalues and the corresponding eigenvectors of $\nabla^2 f(x)$ are $\lambda_i(x_i) = 12a_i(x_i^2 + 2p_i x_i + q_i)$ and e_i, where e_i is the ith unit vector, for all $i \in \{1, \dots, n\}$. While the eigenvalues and the corresponding eigenvectors of $\nabla^2 g(y)$ are $\lambda_i(x_i) d_i^2$ and He_i, for all $i \in \{1, \dots, n\}$.*

Proof. Since $\nabla^2 f(x) = \mathrm{diag}(\lambda_1(x_1), \dots, \lambda_n(x_n))$, its eigenvalues and the corresponding eigenvectors are, respectively, $\lambda_i(x_i) = 12a_i (x_i^2 + 2p_i x_i + q_i)$ and e_i, for all $i \in \{1, \dots, n\}$. Now consider

$$\begin{aligned}\nabla^2 g(y) He_i &= HD\ \nabla^2 f(x)\ DHHe_i \\ &= HD\ \nabla^2 f(x)\ De_i \\ &= H\ \mathrm{diag}(\lambda_1(x_1)d_1^2, \dots, \lambda_n(x_n)d_n^2)\ e_i \\ &= \lambda_i(x_i) d_i^2 He_i.\end{aligned}$$

This completes the proof. □

Theorem 3 shows that the eigenvalues of $\nabla^2 g(y^{**})$ are $\mu_i = \lambda_i d_i^2$ $(\forall i \in \{1, \dots, n\})$ where $\lambda_i \equiv \lambda_i(\alpha_i) = 12a_i(\alpha_i^2 + 2p_i\alpha_i + q_i)$. To ensure that $\nabla^2 g(y^{**})$ has a sufficiently large minimum eigenvalue and a sufficiently small condition number, we generate d_i uniformly in $[\underline{d}, \overline{d}]$ such that $0.1 \le \underline{d} < \overline{d}$ and $\overline{d}/\underline{d} \le 10$. As discussed in Section 2.2, we generate $p_i \in [-\overline{p}, \overline{p}]$ with $\overline{p} > 0$, $q_i \in [\underline{q}, \overline{q}]$ with $\underline{q} < \overline{q} \le -1$, and $a_i \in [\underline{a}, \overline{a}]$ such that $1 \le \underline{a} < \overline{a}$ and $\overline{a}/\underline{a} \le 10$, thus the minimum eigenvalue of $\nabla^2 g(y^{**})$ is

$$\mu^* = \min_i \mu_i \ge \left(\min_i \underline{\lambda}_i\right)\left(\min_i d_i\right)^2 \ge 12[(\sqrt{3}+l)^2 - 1]\underline{a}(-\overline{q})\underline{d}^2$$

and the 2-norm condition number of $\nabla^2 g(y^{**})$ is

$$\begin{aligned}\kappa_g &= \frac{\max_i \mu_i}{\min_j \mu_j} \\ &\le \frac{\max_i \overline{\lambda}_i}{\min_j \underline{\lambda}_j}\left(\frac{\max_i d_i}{\min_j d_j}\right)^2 \\ &\le \frac{(2-l)^2 - 1}{(\sqrt{3}+l)^2 - 1} \cdot \frac{\overline{a}}{\underline{a}} \cdot \frac{\overline{p}^2 - \underline{q}}{-\overline{q}} \cdot \frac{\overline{d}^2}{\underline{d}^2}.\end{aligned} \tag{16}$$

In the set of standard test problems, we set $\underline{d} = 0.25$ and $\overline{d} = 0.50$. Recall that we define l in (12), and we set $\underline{a} = 1$, $\overline{a} = 2$, $\overline{p} = 1$, $\underline{q} = -2$ and $\overline{q} = -1$, thus

$$\mu^* > 1.5 \quad \text{and} \quad \kappa_g \le 36.$$

Error analysis: Let the n-vector ε be the error in the n-vector x. From (15), we have $y = HD^{-1}x$. Thus, the error in y is $HD^{-1}\varepsilon$. Since H is a Householder matrix which satisfies $H^{-1} = H^T = H$, therefore

$$\|HD^{-1}\varepsilon\| = \|D^{-1}\varepsilon\| = \sqrt{\sum_{i=1}^n \left(\frac{\varepsilon_i}{d_i}\right)^2} \le \frac{\|\varepsilon\|}{\underline{d}}.$$

Thus, $\underline{d}$ should not be too small.

3.3. Algorithm

Based on the transformation discussed in Section 3.1 and the practical consideration in Section 3.2, an algorithm for generating a general class of not separable problems for unconstrained global minimization which takes the form $(\overline{P})$ is now described as follows.

Algorithm 2.

1. Perform Algorithm 1.
2. Input $\underline{d}$ and $\overline{d}$ $(0.1 \le \underline{d} < \overline{d}$ and $\overline{d}/\underline{d} \le 10)$.
3. For each $i \in \{1, \dots, n\}$, generate $d_i \in [\underline{d}, \overline{d}]$. Let $D = \mathrm{diag}(d_1, \dots, d_n)$ be an n-dimensional positive definite diagonal matrix.
4. Generate a random n-vector $v \in [0, 1]^n$. Set $v_i := v_i/\sqrt{v^T v}$ for $i \in \{1, \dots, n\}$. Compute the Householder matrix $H = I - 2vv^T$.
5. Now, for any given $y \in \mathfrak{R}^n$, the objective function for problem $(\overline{P})$ is defined by

   $$g(y) \equiv f(DHy).$$

   Its gradient vector and Hessian matrix are, respectively, given by

   $$\nabla g(y) = HD\ \nabla f(x),$$
   $$\nabla^2 g(y) = HD\ \nabla^2 f(x)\ DH,$$

   where $x = DHy$. Its strict global minimizer and global minimum value are, respectively, given by

   $$y^{**} = HD^{-1}\alpha \quad \text{and} \quad g(y^{**}) = f(\alpha).$$

   Its strict local maximizer and local maximum value are, respectively, given by

   $$\hat{y} = HD^{-1}\hat{x} \quad \text{and} \quad g(\hat{y}) = f(\hat{x}).$$

   The eigenvalues of $\nabla^2 g(y^{**})$ are

   $$\mu_i = \lambda_i d_i^2, \quad \forall i.$$

   The 2-norm condition number of $\nabla^2 g(y^{**})$ is κ_g in (16).

3.4. Example

Fig. 2 shows the contours of $g(y)$ of the standard test problem `ngli001` (which is explained in detail in Section 5), where "∗", "×", "▲" and "●" indicate global minimizer, non-global minimizer, local maximizer and saddle point, respectively. For this example, its strict global minimizer and global minimum value are $y^{**} \approx [2.44, 8.60]^T$ and $g(y^{**}) = f(\alpha) \approx -286.56$, while the minimum eigenvalue and the 2-norm condition number of $\nabla^2 g(y^{**})$ are, respectively, $\mu^* \approx 16.48$ and $\kappa_g \approx 1.01$.

4. Test problems over compact boxes

Before the derivations of compact boxes, we need to emphasize that these boxes are not involved in the construction of the test problems.

4.1. Construction of compact boxes

Many algorithms for solving unconstrained global minimization problems require that all local minimizers of $g(y)$ be contained in the interior of a compact box:

$$Y = \{y : \underline{y}_i \le y_i \le \overline{y}_i,\ i = 1, \dots, n\}. \tag{17}$$

To derive a compact box to confine the vector of decision variables, y, in $(\overline{P})$, we notice from Theorem 1 that if $\alpha_i \in (-p_i - 2r_i, -p_i - r_i\sqrt{3})$, then $x_i^L \equiv \alpha_i$, $\hat{x}_i \equiv \beta_i$ and $x_i^R \equiv \gamma_i$ are the global minimizer, local maximizer and non-global minimizer of $f_i(x_i)$, respectively, with $x_i^L < \hat{x}_i < x_i^R$, and thus

$$L_i \equiv \hat{x}_i - x_i^L = \beta_i - \alpha_i = (-3p_i - 3\alpha_i - \sqrt{\Delta_i})/2 > 0,$$
$$R_i \equiv x_i^R - \hat{x}_i = \gamma_i - \beta_i = \sqrt{\Delta_i} > 0;$$

whereas if $\alpha_i \in (-p_i + r_i\sqrt{3}, -p_i + 2r_i)$, then $x_i^L \equiv \beta_i$, $\hat{x}_i \equiv \gamma_i$ and $x_i^R \equiv \alpha_i$ are the non-global minimizer, local maximizer and global minimizer of $f_i(x_i)$, respectively, with $x_i^L < \hat{x}_i < x_i^R$, and thus

$$L_i \equiv \hat{x}_i - x_i^L = \gamma_i - \beta_i = \sqrt{\Delta_i} > 0,$$
$$R_i \equiv x_i^R - \hat{x}_i = \alpha_i - \gamma_i = (3p_i + 3\alpha_i - \sqrt{\Delta_i})/2 > 0.$$

To avoid any local minimizer of $f(x)$ to be too close to the boundaries of the required box, we set

$$x_i^L - \delta_i^L L_i = \underline{x}_i \le x_i \le \overline{x}_i = x_i^R + \delta_i^R R_i, \tag{18}$$

where δ_i^L and δ_i^R are positive random numbers.

The transformation in (15) gives

$$y_i = \frac{1-2v_i^2}{d_i} x_i - 2v_i \sum_{j \ne i} \frac{v_j}{d_j} x_j, \quad \forall i.$$

Since $d_j > 0$ and $v_j \ge 0$ for all $j \in \{1, \dots, n\}$, thus, if $v_i^2 \ge 1/2$, then we set

$$\frac{1-2v_i^2}{d_i}\overline{x}_i - 2v_i \sum_{j \ne i} \frac{v_j}{d_j}\overline{x}_j = \underline{y}_i \le y_i \le \overline{y}_i = \frac{1-2v_i^2}{d_i}\underline{x}_i - 2v_i \sum_{j \ne i} \frac{v_j}{d_j}\underline{x}_j; \tag{19}$$

whereas if $v_i^2 < 1/2$, then we set

$$\frac{1-2v_i^2}{d_i}\underline{x}_i - 2v_i \sum_{j \ne i} \frac{v_j}{d_j}\overline{x}_j = \underline{y}_i \le y_i \le \overline{y}_i = \frac{1-2v_i^2}{d_i}\overline{x}_i - 2v_i \sum_{j \ne i} \frac{v_j}{d_j}\underline{x}_j. \tag{20}$$

4.2. Generation of an upper bound of the objective value

Besides the compact box that contains all local minimizers of the objective function, some algorithms for solving unconstrained global minimization problems require an upper bound for the objective value.

For problem (P), its objective function $f(x)$ is separable and its decision variables are contained in the box:

$$\overline{X} = \{x : \underline{x}_i \le x_i \le \overline{x}_i, \ i = 1, \dots, n\},$$

where $\underline{x}_i$ and $\overline{x}_i$ are defined in (18). Since $f_i(x_i)$ is a coercive function with one local maximizer, thus

$$f_i(x_i) \le \overline{f}_i = \max\{f_i(\underline{x}_i), f_i(\hat{x}_i), f_i(\overline{x}_i)\}, \tag{21}$$

where $\hat{x}_i$ is the local maximizer of $f_i(x_i)$ as given in Theorem 1. Hence,

$$f(x) = \sum_{i=1}^n a_i f_i(x_i) \le \overline{f} = \sum_{i=1}^n a_i \overline{f}_i.$$

For problem ($\overline{P}$), however, its objective function $g(y)$ is not separable, it is, in general, difficult to find an upper bound for it directly. In the following, we introduce a simple method for finding an upper bound for $g(y)$ over the compact box Y. We first transform the not separable objective function $g(y)$ back to the separable objective function $f(x)$ with a compact box

$$\overline{\overline{X}} = \{x : \underline{\underline{x}}_i \le x_i \le \overline{\overline{x}}_i, \ i = 1, \dots, n\}$$

that contains the transformed box Y. We then find an upper bound for each univariate function $f_i(x_i)$ and finally obtain upper bounds for both $f(x)$ and $g(y)$.

The transformation in (15) gives

$$x_i = (1-2v_i^2) d_i y_i - 2d_i v_i \sum_{j \ne i} v_j y_j, \quad \forall i.$$

Since $d_j > 0$ and $v_j \ge 0$ for all $j \in \{1, \dots, n\}$, thus, if $v_i^2 \ge 1/2$, then we have

$$(1-2v_i^2) d_i \overline{y}_i - 2d_i v_i \sum_{j \ne i} v_j \overline{y}_j = \underline{\underline{x}}_i \le x_i \le \overline{\overline{x}}_i = (1-2v_i^2) d_i \underline{y}_i - 2d_i v_i \sum_{j \ne i} v_j \underline{y}_j; \tag{22}$$

whereas if $v_i^2 < 1/2$, then we have

$$(1-2v_i^2) d_i \underline{y}_i - 2d_i v_i \sum_{j \ne i} v_j \overline{y}_j = \underline{\underline{x}}_i \le x_i \le \overline{\overline{x}}_i = (1-2v_i^2) d_i \overline{y}_i - 2d_i v_i \sum_{j \ne i} v_j \underline{y}_j. \tag{23}$$

Again, since $f_i(x_i)$ is a coercive function with one local maximizer, thus,

$$f_i(x_i) \le \overline{\overline{f}}_i = \max\{f_i(\underline{\underline{x}}_i), f_i(\hat{x}_i), f_i(\overline{\overline{x}}_i)\}. \tag{24}$$

Hence,

$$f(x) = \sum_{i=1}^n a_i f_i(x_i) \le \overline{\overline{f}} = \sum_{i=1}^n a_i \overline{\overline{f}}_i.$$

Since Y is contained in $\overline{\overline{X}}$ after the transformation, therefore $g(y) \le \overline{g} \equiv \overline{\overline{f}}$.

Note, however, that

$$\overline{f} < \max_{y \in Y} g(y) \ll \overline{\overline{f}}$$

when n is large.

4.3. Interconnections and specifications of parameters

On the relative sizes of the region-of-attraction of minimizers:

Theorem 4. *Suppose that the assumptions in Theorem 1 hold. Assume that δ_i^L and δ_i^R in (18) are independent positive random numbers that follow identical uniform distribution with lower bound $\underline{\delta}$ and upper bound $\overline{\delta}$. Denote the expected ratio of the size of the region of attraction of the non-global minimizer to that of the global minimizer α_i of $f_i(x_i)$ by $\mathrm{Ratio}_i(\alpha_i)$. If $\alpha_i \in (-p_i - 2r_i, -p_i - r_i\sqrt{3})$, then $\mathrm{Ratio}_i(\alpha_i)$ is a strictly increasing function; whereas if $\alpha_i \in (-p_i + r_i\sqrt{3}, -p_i + 2r_i)$, then $\mathrm{Ratio}_i(\alpha_i)$ is a strictly decreasing function.*

Proof. Denote the local maximizer of $f_i(x_i)$ by $\hat{x}_i$. Now, if $\alpha_i \in (-p_i - 2r_i, -p_i - r_i\sqrt{3})$, then the expected ratio is

$$\begin{aligned}
\mathrm{Ratio}_i(\alpha_i) &= E\left(\frac{\overline{x}_i - \hat{x}_i}{\hat{x}_i - \underline{x}_i}\right) \\
&= E\left\{\frac{(1+\delta_i^R)[\gamma_i(\alpha_i) - \beta_i(\alpha_i)]}{(1+\delta_i^L)[\beta_i(\alpha_i) - \alpha_i]}\right\} \\
&= \frac{\gamma_i(\alpha_i) - \beta_i(\alpha_i)}{\beta_i(\alpha_i) - \alpha_i} \cdot E\left(\frac{1+\delta_i^R}{1+\delta_i^L}\right) \\
&= \frac{2k\sqrt{\Delta_i}}{-3(p_i+\alpha_i) - \sqrt{\Delta_i}},
\end{aligned}$$

where

$$\begin{aligned}
k &= E\left(\frac{1+\delta_i^R}{1+\delta_i^L}\right) \\
&= \int_{\underline{\delta}}^{\overline{\delta}} \int_{\underline{\delta}}^{\overline{\delta}} \left(\frac{1+\delta_i^R}{1+\delta_i^L}\right) \frac{1}{(\overline{\delta} - \underline{\delta})^2} \, d\delta_i^R \, d\delta_i^L \\
&= \frac{(1+\overline{\delta})^2 - (1+\underline{\delta})^2}{2(\overline{\delta} - \underline{\delta})^2} \cdot \ln\left(\frac{1+\overline{\delta}}{1+\underline{\delta}}\right)
\end{aligned}$$

is a positive constant. Differentiating the expected ratio with respect to α_i yields

$$\frac{d\,\mathrm{Ratio}_i(\alpha_i)}{d\alpha_i} = \frac{72kr_i^2}{[3(p_i+\alpha_i) + \sqrt{\Delta_i}]^2 \sqrt{\Delta_i}} > 0.$$

Similarly, if $\alpha_i \in (-p_i + r_i\sqrt{3}, -p_i + 2r_i)$, then

$$\mathrm{Ratio}_i(\alpha_i) = E\left(\frac{\hat{x}_i - \underline{x}_i}{\overline{x}_i - \hat{x}_i}\right)$$

$$= \frac{\gamma_i(\alpha_i)-\beta_i(\alpha_i)}{\alpha_i-\gamma_i(\alpha_i)} \cdot E\left(\frac{1+\delta_i^L}{1+\delta_i^R}\right) = \frac{2k\sqrt{\Delta_i}}{3(p_i+\alpha_i)-\sqrt{\Delta_i}},$$

where

$$k = E\left(\frac{1+\delta_i^L}{1+\delta_i^R}\right) = \frac{(1+\overline{\delta})^2-(1+\underline{\delta})^2}{2(\overline{\delta}-\underline{\delta})^2} \cdot \ln\left(\frac{1+\overline{\delta}}{1+\underline{\delta}}\right)$$

is a positive constant. Differentiating the expected ratio with respect to α_i yields

$$\frac{d\ \mathrm{Ratio}_i(\alpha_i)}{d\alpha_i} = -\frac{72kr_i^2}{[3(p_i+\alpha_i)+\sqrt{\Delta_i}]^2\sqrt{\Delta_i}} < 0.$$

This completes the proof. □

In order to have the desired properties as stated in Theorem 4, δ_i^L and δ_i^R must be generated from identical uniform distribution. Moreover, the introduction of δ^L and δ^R is to avoid the minimizers of $f_i(x_i)$ to get too close to the boundaries of the box in (18). So, δ_i^L and δ_i^R cannot be too small. On the other hand, δ_i^L and δ_i^R cannot be too large, otherwise, both $\overline{f}_i$ in (21) and $\overline{\overline{f}}_i$ in (24) will become extremely large. With the above consideration, we generate both δ_i^L and δ_i^R uniformly in $[\underline{\delta}, \overline{\delta}]$ such that $0.1 \le \underline{\delta} < \overline{\delta} \le 1$. In the set of standard test problems, we set $\underline{\delta} = 0.3$ and $\overline{\delta} = 0.7$.

4.4. Algorithm

Based on the theoretical results in Sections 4.1 and 4.2, and the practical consideration in Section 4.3, an algorithm for generating a compact box Y in (17) that contains all local minimizers of $g(y)$ in its interior, and for computing an upper bound for $f(x)$ and for $g(y)$ is now described as follows.

Algorithm 3.

1. Perform Algorithm 2.
2. Input $\underline{\delta}$ and $\overline{\delta}$ ($0.1 \le \underline{\delta} < \overline{\delta} \le 1$).
3. For each $i \in \{1,\dots,n\}$, generate both δ_i^L and δ_i^R uniformly in $[\underline{\delta}, \overline{\delta}]$.
4. For each $i \in \{1,\dots,n\}$, define the relaxed lower and upper bounds for x_i as follows:
   - if $\alpha_i \in (-p_i-2r_i, -p_i-r_i\sqrt{3})$, define
     $$\underline{x}_i = \alpha_i + \delta_i^L \cdot \frac{3p_i+3\alpha_i+\sqrt{\Delta_i}}{2} \quad \text{and} \quad \overline{x}_i = \gamma_i + \delta_i^R\sqrt{\Delta_i};$$
   - if $\alpha_i \in (-p_i+r_i\sqrt{3}, -p_i+2r_i)$, define
     $$\underline{x}_i = \beta_i - \delta_i^L\sqrt{\Delta_i} \quad \text{and} \quad \overline{x}_i = \alpha_i + \delta_i^R \cdot \frac{3p_i+3\alpha_i-\sqrt{\Delta_i}}{2};$$

   and calculate $\overline{f}_i$ in (21).
5. For each $i \in \{1,\dots,n\}$, define the lower and upper bounds for y_i as follows:
   - if $v_i^2 \ge 1/2$, then define $\underline{y}_i$ and $\overline{y}_i$ as in (19);
   - if $v_i^2 < 1/2$, then define $\underline{y}_i$ and $\overline{y}_i$ as in (20).
6. For each $i \in \{1,\dots,n\}$, define the lower and upper bounds for x_i of the compact box $\overline{X}$ as follows:
   - if $v_i^2 \ge 1/2$, then define $\underline{\underline{x}}_i$ and $\overline{\overline{x}}_i$ as in (22);
   - if $v_i^2 < 1/2$, define $\underline{\underline{x}}_i$ and $\overline{\overline{x}}_i$ as in (23);

   and calculate $\overline{\overline{f}}_i$ in (24).
7. Compute the upper bounds for $f(x)$ and $g(y)$
   $$\overline{f} = \sum_{i=1}^n a_i\overline{f}_i \quad \text{and} \quad \overline{g} = \overline{\overline{f}} = \sum_{i=1}^n a_i\overline{\overline{f}}_i.$$

4.5. Example

For the standard test problem `ngli001` (which is explained in detail in Section 5), we have

$$\underline{y} \approx \begin{bmatrix} -12.92 \\ -13.53 \end{bmatrix} \quad \text{and} \quad \overline{y} \approx \begin{bmatrix} 15.34 \\ 15.33 \end{bmatrix}$$

$$\overline{f} \approx 482.47, \quad \max_{y \in Y} g(y) \approx 6463.80 \quad \text{and} \quad \overline{g} = \overline{\overline{f}} \approx 10184.39.$$

5. Realization and software packages

5.1. Generation of random vectors

To perform Algorithms 1–3, we need a total of eight uniformly distributed random n-vectors: a, p, q, α, d, v, δ^L and δ^R. We use the Mersenne Twister (the default random number generator in MATLAB 7.4 Release 2007a and later) to generate these random vectors in $(0,1)$. As mentioned in the online documents of MATLAB, a full description of the Mersenne Twister algorithm can be found in http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html.

For each test problem in the standard set, we use the corresponding problem-number as the seed of the random number generator. To ensure the same set of random numbers to be used to generate the standard set of test problems in other versions of MATLAB or in different programming languages, we first generate the required random numbers and store them in text files. We then read the random numbers back from the file to generate the standard test problem. More specifically, we generate a by the pre-generated uniformly distributed random numbers 1 to n; p by $n+1$ to $2n$; q by $2n+1$ to $3n$; α by $3n+1$ to $4n$; d by $4n+1$ to $5n$; v by $5n+1$ to $6n$; δ^L by $6n+1$ to $7n$; and δ^R by $7n+1$ to $8n$.

5.2. Control of the difficulty levels of test problems

The difficulty of a global optimization problem depends on many factors (see, e.g., [9,10,23,26,37,38]). Three of these factors that make a global optimization problem difficult are:

1. large number of local minima,
2. small difference between the global and non-global minimum values, and
3. small size of the basin at the global minimizer.

From Theorem 2, our test problem has 2^n local minima. Thus, by changing the dimension n, we can change the number of local minima of the problem. (Actually, the number of stationary points is fixed for a given n.) Although the effect of the number of local minima to the difficulty of a problem is rather disputable (see, e.g, [25]), we will demonstrate the validity of this factor for our set of test problems in the next section.

We know from Theorem 1 that the difference between the non-global and the global minimum values of $f_i(x_i)$ is a decreasing function of α_i if $\alpha_i \in (-p_i-2r_i, -p_i-r_i\sqrt{3})$ and it is an increasing function if $\alpha_i \in (-p_i+r_i\sqrt{3}, -p_i+2r_i)$. Hence, the difficulty of solving subproblem (P_i) is an increasing function of α_i if $\alpha_i \in (-p_i-2r_i, -p_i-r_i\sqrt{3})$ and it is a decreasing function if

$\alpha_i \in (-p_i + r_i\sqrt{3}, -p_i + 2r_i)$. Thus, we can increase or decrease the difficulty level of a subproblem via selecting the location of α_i.

Theorem 4 shows that the difficulty of solving a subproblem is an increasing function of α_i if $\alpha_i \in (-p_i - 2r_i, -p_i - r_i\sqrt{3})$, and it is a decreasing function if $\alpha_i \in (-p_i + r_i\sqrt{3}, -p_i + 2r_i)$. These results conform with the second factor.

Using these results and (11), we can generate a "*relatively easy*" subproblem (P_i) by generating an α_i in

$$\left[-p_i - (2-l)r_i, -p_i - \frac{2+\sqrt{3}}{2}r_i\right] \cup \left[-p_i + \frac{2+\sqrt{3}}{2}r_i, -p_i + (2-l)r_i\right].$$

Similarly, we can generate a *"relatively difficult"* subproblem (P_i) by generating an α_i in

$$\left[-p_i - \frac{2+\sqrt{3}}{2}r_i, -p_i - (\sqrt{3}+l)r_i\right]$$
$$\cup \left[-p_i + (\sqrt{3}+l)r_i, -p_i + \frac{2+\sqrt{3}}{2}r_i\right].$$

We now give the following definition for the difficulty level of problem (P).

Definition 1. Problem (P) is said to be a *level* 0 *problem* if all its n subproblems are relatively easy. It is a *level* 1 *problem* if $\lceil n/2 \rceil$ subproblems are relatively difficult and $\lfloor n/2 \rfloor$ subproblems are relatively easy. It is a *level* 2 *problem* if all its n subproblems are relatively difficult.

It should be emphasized that higher-level problems are only statistically more difficult to be solved globally than lower-level problems. It does not imply that any higher-level problem is more difficult than every lower-level problem.

In the generating process, we first generate all the relatively difficult subproblems and then the relatively easy subproblems. Since problem (P) will be transformed to problem $(\overline{P})$ in Algorithm 2, the above procedure will not affect the generality of the test problems. We will demonstrate the validity of the difficulty level of problem $(\overline{P})$ in the next section.

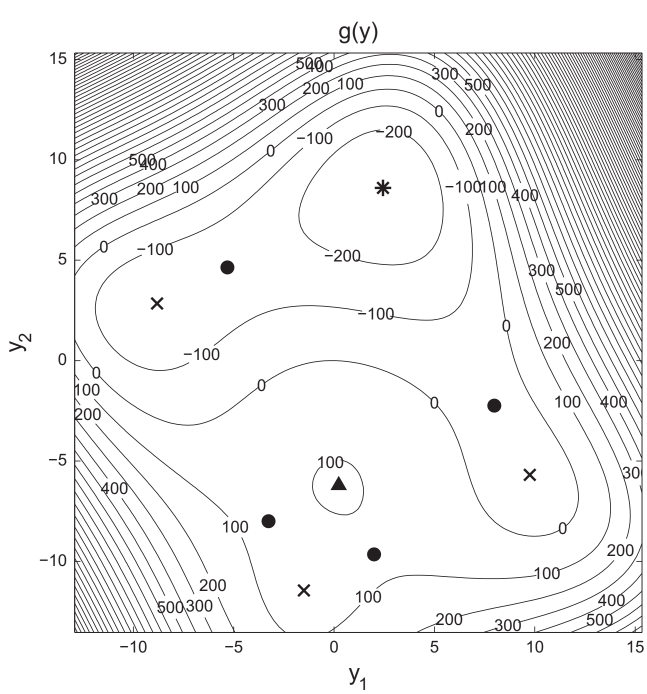

Fig. 2. Contours of the level 0 problem `ngli001` ($*$=global minimizer, $\times$=non-global minimizer, ▲=local maximizer, ●=saddle point).

Figs. 2, 3 and 4 show the contours of the standard test problems `ngli001` (level 0), `ngli011` (level 1), and `ngli021` (level 2), where "$*$", "$\times$", "▲" and "●" indicate global minimizer, non-global minimizer, local maximizer and saddle point, respectively.

Remark: Note that instead of increasing the dimension n of a test problem, it is easy to increase the number of local minima of the test problem by introducing an oscillation term to the

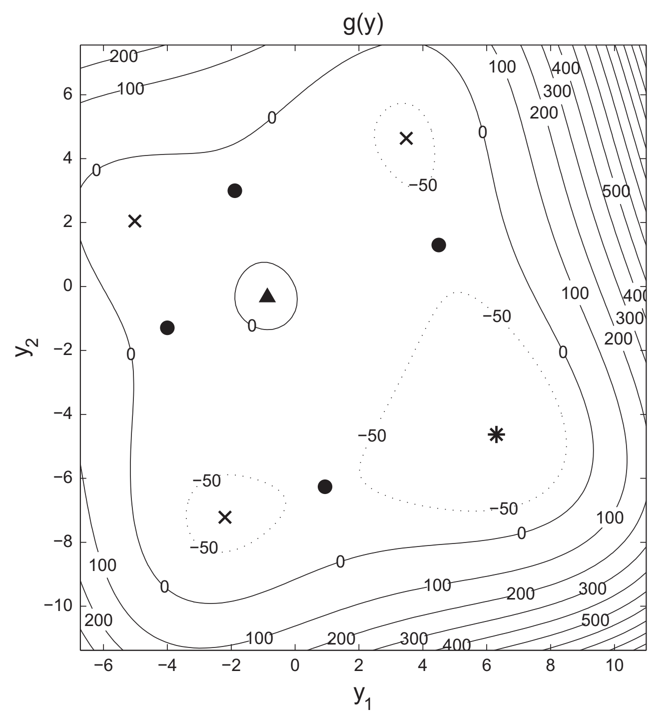

Fig. 3. Contours of the level 1 problem `ngli011` ($*$=global minimizer, $\times$=non-global minimizer, ▲=local maximizer, ●=saddle point).

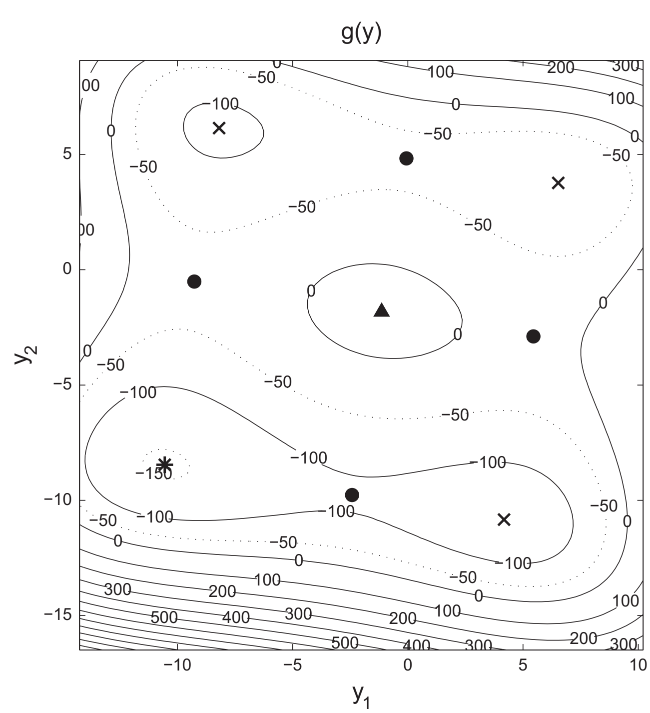

Fig. 4. Contours of the level 2 problem `ngli021` ($*$=global minimizer, $\times$=non-global minimizer, ▲=local maximizer, ●=saddle point).

objective function of each univariate problem (see, e.g., [1]). Define

$$(\hat{P}) \quad \min_{x \in \Re^n} \hat{f}(x) = \sum_{i=1}^{n} a_i \hat{f}_i(x_i),$$

where

$$\hat{f}_i(x_i) = f_i(x_i) + c_i(x_i),$$

$$c_i(x_i) = \begin{cases} -b_i \cos\left[2\pi k_i \left(\dfrac{x_i - \alpha_i}{\gamma_i - \alpha_i}\right)\right] & \text{if } \alpha_i \in (-p_i - 2r_i, -p_i - r_i\sqrt{3}), \\ -b_i \cos\left[2\pi k_i \left(\dfrac{x_i - \beta_i}{\alpha_i - \beta_i}\right)\right] & \text{if } \alpha_i \in (-p_i + r_i\sqrt{3}, -p_i + 2r_i), \end{cases}$$

$$b_i > 0 \quad \text{and} \quad k_i \ge 1 \text{ integer},$$

a_i and $f_i(x_i)$ are defined in (2), and α_i is the global minimizer of $f_i(x_i)$. Note that the univariate function $f_i(x_i)$ has two minimizers under the conditions in Theorem 1. It is easy to show that the global minimizer of $f_i(x_i)$ is the global minimizer of $\hat{f}_i(x_i)$, and the non-global minimizer of $f_i(x_i)$ is a non-global minimizer of $\hat{f}_i(x_i)$. Moreover, if b_i is sufficiently large, then $\hat{f}_i(x_i)$ has $k_i - 1$ additional non-global minimizers between the two minimizers of $f_i(x_i)$. The techniques described in Sections 3 and 4 can also be used to disguise the separability and to find a compact box for the variables. Note that, however, the exact locations and the function values of the additional $k_i - 1$ local minima of $\hat{f}_i(x_i)$ are hard to find. The exact number of local minima of the transformed problem resulting from the transformation described in Sections 3 and 4 are not known, either. Therefore, we did not incorporate any oscillation term to the objective function of any univariate problem in our test problem generator.

5.3. Software packages

The developed test problem generator is implemented in MATLAB; and a set of 300 standard test problems is generated by the generator for test campaigns. The generator package works well in MATLAB 7.4 Release 2007a and later, while the standard set of test problems works well in MATLAB 7.0 Release 14 (2004) and later, and GAMS Distribution 23.3 and later. The packages are available for downloading from the authors' webpage, http://www.se.cuhk.edu.hk/~ckng/generator/.

The details of file names and the ways to generate the test functions from a practical point of view can be found in the Appendix.

5.4. A summary of the user-defined and random parameters

User-defined parameters:

1. n: the number of decision variables.
2. $L \in \{0, 1, 2\}$: the difficulty level.
3. $\underline{a}$ and $\overline{a}$ ($1 \le \underline{a} < \overline{a}$ and $\overline{a}/\underline{a} \le 10$): the lower and upper bounds for each component of the random n-vector a (where $\underline{a} = 1$ and $\overline{a} = 2$ in the set of standard test problems).
4. $\overline{p}$ ($\overline{p} > 0$): the upper bound (i.e., the negative lower bound) for each component of the random n-vector p (where $\overline{p} = 1$ in the set of standard test problems).
5. $\underline{q}$ and $\overline{q}$ ($\underline{q} < \overline{q} \le -1$):the lower and upper bounds for each component of the random n-vector q (where $\underline{q} = -2$ and $\overline{q} = -1$ in the set of standard test problems).
6. $\alpha' \in (0, 1)$: a fraction in the middle parts of the open admissible regions of each α_i (where $\alpha' = 0.95$ in the set of standard test problems).
7. $\underline{d}$ and $\overline{d}$ ($0.1 \le \underline{d} < \overline{d}$ and $\overline{d}/\underline{d} \le 10$): the lower and upper bounds for each component of the random n-vector d (where $\underline{d} = 0.25$ and $\overline{d} = 0.50$ in the set of standard test problems).
8. $\underline{\delta}$ and $\overline{\delta}$ ($0.1 \le \underline{\delta} < \overline{\delta} \le 1$):the lower and upper bounds for each component of the random n-vectors δ^L and δ^R (where $\underline{\delta} = 0.3$ and $\overline{\delta} = 0.7$ in the set of standard test problems).

Random parameters:

1. $a \in [\underline{a}, \overline{a}]^n$: the weight of the polynomial $f_i(x_i)$ in $f(x)$.
2. $p \in [-\overline{p}, \overline{p}]^n$: where $4p_i$ is the coefficient of the third-degree term of $f_i(x_i)$.
3. $q \in [\underline{q}, \overline{q}]^n$: where $6q_i$ is the coefficient of the second-degree term of $f_i(x_i)$.
4. α: the global minimizer of $f(x)$.
5. $d \in [\underline{d}, \overline{d}]^n$: the diagonal elements of the diagonal matrix D.
6. $v \in [0, 1]^n$ with $v^T v = 1$: the vector for defining the Householder matrix H.
7. $\delta^L \in [\underline{\delta}, \overline{\delta}]^n$ and $\delta^R \in [\underline{\delta}, \overline{\delta}]^n$: the factors of enlargement for defining the box Y.

6. Computational experiment

6.1. Efficiency tests and stability tests

To test the efficiency of the generator, we generated each class of test problems 50 times by a personal computer with Intel Core 2 Duo E6700 CPU, and recorded the average CPU times per problem for generation in Table 1. Since the time is very short for generating every standard test problem, we conclude that the generation process is efficient.

To test the stability of the generating process, we performed four phases of stability test. First, we compared $f_i(\alpha_i)$, $f_i(\beta_i)$ and $f_i(\gamma_i)$ for each $i \in \{1, \ldots, n\}$ during the generating process of the standard test problems. We concluded that $f_i(\alpha_i) < f_i(\beta_i)$ and $f_i(\alpha_i) < f_i(\gamma_i)$ for all i of every standard test problem.

Next, we computed the following during the generating process of the standard test problems:

- $\|\nabla g(y^{**})\|$: the Euclidean norm of $\nabla g(y^{**})$,
- μ^*: the minimum eigenvalue of $\nabla^2 g(y^{**})$,
- κ_g: the condition number of $\nabla^2 g(y^{**})$, and
- $(\overline{f} - f(\alpha))/n$: the average range of values of $f_i(x_i)$.

A summary of results can be found in Table 1. From the results shown in the table, we conclude that $\|\nabla g(y^{**})\|$ is sufficiently close to 0, $\nabla^2 g(y^{**})$ is positive definite with a sufficiently small condition number, and the range of objective values is sufficiently small for every standard test problem.

Table 1
Summary of the efficiency and stability test results.

n	IDs	Avg. CPU/prob. (Seconds)	Avg. $\|\nabla g(y^{**})\|$	Avg. μ^*	Avg. κ_g	Avg. $\dfrac{\overline{f} - f(\alpha)}{n}$
2	ngli001–ngli030	0.016	5.07×10^{-14}	8.96	1.99	206.88
5	ngli031–ngli060	0.018	5.86×10^{-14}	5.78	3.36	176.21
10	ngli061–ngli090	0.021	9.16×10^{-14}	4.78	5.79	193.01
20	ngli091–ngli120	0.026	1.64×10^{-13}	4.18	6.48	212.18
50	ngli121–ngli150	0.064	2.63×10^{-13}	3.40	8.88	191.34
100	ngli151–ngli180	0.105	4.70×10^{-13}	3.13	10.02	201.31
200	ngli181–ngli210	0.188	8.75×10^{-13}	2.94	11.74	192.71
500	ngli211–ngli240	0.401	2.13×10^{-12}	2.56	13.96	192.91
1000	ngli241–ngli270	0.918	4.15×10^{-12}	2.42	15.74	194.13
2000	ngli271–ngli300	3.359	8.25×10^{-12}	2.34	17.22	194.02

Third, for each test problem we run 10,000 local searches from randomly generated starting points within a multistart algorithm. The results of the test experiments show that y^{**} is the global minimizer of the corresponding test problem. The details of the multistart test experiments can be found in the next subsection.

Fourth, we solved some standard test problems by a deterministic global optimization solver. The results of the test experiments also indicate that y^{**} is the global minimizer of the corresponding test problem. The details of the test experiments can be found in Section 6.3.

6.2. Test experiments with a multistart algorithm

The whole set of standard test problems, `ngli001–ngli300`, was solved by the unconstrained local minimization algorithm FMINUNC in MATLAB 10,000 times with different initial points which were generated uniformly in the box Y. A summary of the computational results is shown in Table 2.

It is easy to see from Table 2 that the average number of successful trials is decreasing from the top to the bottom of the table. Therefore, we conclude that the relative size of the attraction region of the global minimizer to the box Y decreases as n and/or L increase. Moreover, the impact of n is more significant than that of L.

6.3. Test experiments with GAMS/BARON

Problems `ngli001–ngli090` were solved by BARON version 9.0.2 under GAMS Distribution 23.3 that runs on a personal computer with Intel Core 2 Duo E6700 CPU. A summary of the computational results is shown in Table 3. In the option file for GAMS/BARON, we increased "`MaxTime`" (the maximum CPU time allowed) from the default value of 1000 s to 3600 s for all problems.

First, we found that the global minimizer and its objective value obtained by GAMS/BARON for each test problem conformed with the desired values. As mentioned in the online documents of GAMS, BARON implements deterministic global optimization algorithms of the branch-and-bound type that guarantees to provide global optima under fairly general assumptions. The computational results demonstrated that the generating process was stable.

Table 2
Summary of computational results of a multistart algorithm.

n	L	IDs	Average no. of successful trials out of 10,000 trials
2	0	ngli001–ngli010	4733.0
	1	ngli011–ngli020	3714.7
	2	ngli021–ngli030	3189.0
5	0	ngli031–ngli040	1368.1
	1	ngli041–ngli050	696.2
	2	ngli051–ngli060	567.6
10	0	ngli061–ngli070	124.2
	1	ngli071–ngli080	51.6
	2	ngli081–ngli090	11.6
20	0	ngli091–ngli100	0.9
	1	ngli101–ngli110	0.4
	2	ngli111–ngli120	0.0
50	0	ngli121–ngli130	0.0
⋮	⋮	⋮	⋮
2000	2	ngli291–ngli300	0.0

Table 3
Summary of computational results of GAMS/BARON.

n	L	IDs	Average No. of Iterations	Average CPU Times (Seconds)
2	0	ngli001–ngli010	9.6	0.40
	1	ngli011–ngli020	15.8	0.43
	2	ngli021–ngli030	20.6	0.46
5	0	ngli031–ngli040	162.0	0.83
	1	ngli041–ngli050	417.6	1.46
	2	ngli051–ngli060	825.7	2.43
10	0	ngli061–ngli070	13 243.8	60.55
	1	ngli071–ngli080	59 867.8	263.63
	2	ngli081–ngli090	214 987.8	892.78

Second, it is easy to see from Table 3 that both the average number of iterations and average CPU times are increasing from the top to the bottom of the table. These results support the assumption that the computational difficulty of solving the problems globally grows exponentially as n increases. Moreover, the effect of n is more significant than that of L.

In fact, we have also tried to solve one problem of each level with $n=20$, namely `ngli100`, `ngli110` and `ngli120`. GAMS/BARON run out of memory after 1,815,000, 1,780,000 and 1,740,000 iterations, and 6.32 h, 5.58 h and 5.05 h, respectively.

Note also that GAMS/BARON not only requires the bounds for the decision variables y_i $(\forall i \in \{1, \ldots, n\})$, but also requires the bounds of the temporary variables x_i $(\forall i \in \{1, \ldots, n\})$ to guarantee fast convergence. That is why we include these bounds in the set of standard test problems for GAMS.

7. Conclusions

This paper has presented a computationally efficient approach for generating a general class of test problems for benchmarking unconstrained global optimization algorithms. This class of test problems is analytic in $\Re^n$, and has 2^n a priori known local minimizers among which only one is the global minimum solution. A set of 300 standard test problems has been produced for both MATLAB and GAMS. Four phases of stability test have shown that the generating process is stable. Preliminary test experiments with a multistart algorithm and with GAMS/BARON have further demonstrated that the relative size of the region of attraction and the difficulty level of test problems can be controlled easily by the values of n and L.

Appendix

The generator package consists of 306 files. Their names and functions are listed as follows:

- `ngli.m` displays the basic information of the generator package.
- `nglistd.m` uses the random numbers stored in the text files `ngli001.rn–ngli300.rn` to generate the 300 standard test problems.
- `ngligen.m` uses the parameters in the text file "`ngli`*nnnn*`.par`" to generate a general (user-defined) test problem in both MATLAB and GAMS, where *nnnn* is an integer between 1000 and 9999.
- `ngliobj.m` produces the objective function value, first and second derivatives of any given test problem at any given point. It is used together with the standard test problems

`ngli001.m`–`ngli300.m` or the user-defined test problems `ngli1000.m`–`ngli9999.m`.

- `nglimain.m` generates any test problem. It is the main procedure of the generator and is used internally.
- `ngli9999.par` stores the parameters for generating the example test problem number 9999 with $n=30$, $L=1$ and the default values for generating any standard test problem. To view the structure of the parameter file, invoke `help ngligen` in MATLAB.
- `ngli001.rn`–`ngli300.rn` store the uniformly distributed random numbers in $(0,1)$ for generating the 300 standard test problems. The random numbers stored in each file are generated using the default random number generator (Mersenne Twister) in MATLAB 7.4 Release 2007a and later with the corresponding problem-number as the initial seed.

The standard test problems package consists of 902 files. Their names and functions are listed as follows:

- `ngli.txt` stores the general information of the set of standard test problems.
- `ngli001.txt`–`ngli300.txt` store the information of the global minimizers and values.
- `ngli001.m`–`ngli300.m` store the parameters of the 300 standard test problems for use with the function `ngliobj.m`.
- `ngli001.gms`–`ngli300.gms` store the 300 standard test problems in GAMS.
- `ngliobj.m` produces the objective function value, first and second derivatives of any given test problem at any given point. It is used together with `ngli001.m`–`ngli300.m`.

References

[1] Addis B, Locatelli M. A new class of test functions for global optimization. J Global Optim 2007;38(3):479–501.
[2] Anderssen RS, Bloomfield P. Properties of the random search in global optimization. J Optim Theory Appl 1975;16:383–98.
[3] Barhen J, Protopopescu V, Reister D. TRUST: a deterministic algorithm for global optimization. Science 1997;276(5315):1094–7.
[4] Bremermann H. A method of unconstrained global optimization. Math Biosci 1970;9:1–15.
[5] Calamai PH, Vicente LN, Júdice JJ. A new technique for generating quadratic programming test problems. Math Progr 1993;61(2):215–31.
[6] Cetin BC, Barhen J, Burdick JW. Terminal repeller unconstrained subenergy tunneling (TRUST) for fast global optimization. J Optim Theory Appl 1993;77(1):97–126.
[7] Dekkers A, Aarts E. Global optimization and simulated annealing. Math Progr 1991;50:367–93.
[8] Dixon LCW, Gomulka J, Herson SE. Reflection on global optimization problems. In: Dixon LCW, editor. Optimization in action: proceedings of the conference on optimization in action held at the University of Bristol in January 1975. New York: Institute of Mathematics and its Applications. Academic Press; 1976. p. 398–435.
[9] Dixon LCW, Szegö GP, editors. Towards global optimization, vol. 1. Amsterdam: North-Holland; 1975.
[10] Dixon LCW, Szegö GP, editors. Towards global optimization, vol. 2. Amsterdam: North-Holland; 1978.
[11] Durazzi C, Ruggiero V, Zanni L. A random generator for large-scale linearly constrained quadratic programming test problems. Monograph 6, Annali Università di Ferrara; 2000.
[12] Floudas CA, Pardalos PM. A collection of test problems for constrained global optimization algorithms. Berlin: Springer-Verlag; 1990.
[13] Floudas CA, Pardalos PM, Adjiman CS, Esposito WR, Gümüs ZH, Harding ST, et al. Handbook of test problems in local and global optimization. Dordrecht: Kluwer Academic Publishers; 1999.
[14] Gablonsky MJ, Kelley CT. A locally-biased form of the DIRECT algorithm. J Global Optim 2001;21(1):27–37.
[15] Gaviano M, Kvasov DE, Lera D, Sergeyev YD. Algorithm 829: software for generation of classes of test functions with known local and global minima for global optimization. ACM Trans Math Softw 2003;29(4):469–80.
[16] Gaviano M, Lera D. Test functions with variable attraction regions for global optimization problems. J Global Optim 1998;13(2):207–23.
[17] Ge R-P. A filled function method for finding a global minimizer of a function of several variables. Math Progr 1990;46(2):191–204.
[18] Ge R-P, Qin Y-F. A class of filled functions for finding global minimizers of a function of several variables. J Optim Theory Appl 1987;54(2):241–52.
[19] Hedar A-R, Fukushima M. Hybrid simulated annealing and direct search method for nonlinear unconstrained global optimization. Optim Methods Softw 2002;17(5):891–912.
[20] Hock W, Schittkowski K. Test examples for nonlinear programming codes. New York: Springer-Verlag; 1981.
[21] Hyuer W, Neumaier A. Global optimization by multilevel coordinate search. J Global Optim 1999;14(4):331–55.
[22] Jones DR, Perttunen CD, Stuckman BE. Lipschitzian optimization without the Lipschitz constant. J Optim Theory Appl 1993;79(1):157–81.
[23] Leary RH. Global optimization on funneling landscapes. J Global Optim 2000;18(4):367–83.
[24] Levy AV, Montalvo A. The tunneling algorithm for the global minimization of functions. SIAM J Sci Stat Comput 1985;6(1):15–29.
[25] Locatelli M. A note on the Griewank test function. J Global Optim 2003;25(2):169–74.
[26] Locatelli M. On the multilevel structure of global optimization problems. Comput Optim Appl 2005;30(1):5–22.
[27] Mathar R, Žilinskas A. A class of test functions for global optimization. J Global Optim 1994;5(2):195–9.
[28] Mockus J. Bayesian approach to global optimization: theory and applications. Dordrecht: Kluwer Academic Publishers; 1989.
[29] Moré JJ, Garbow BS, Hillstrom KE. Testing unconstrained optimization software. ACM Trans Math Softw 1981;7(1):17–41.
[30] Ng C-K, Li D, Zhang L-Z. Global descent method for global optimization. SIAM J Optim 2010;20(6):3161–84.
[31] Pardalos PM. Generation of large-scale quadratic programs for use as global optimization test problems. ACM Trans Math Softw 1987;13(2):133–7.
[32] Rosen JB, Suzuki S. Construction of nonlinear programming test problems. Commun ACM 1965;8(2):113.
[33] Schittkowski K. More test examples for nonlinear programming codes. Berlin: Springer-Verlag; 1987.
[34] Schoen F. A wide class of test functions for global optimization. J Global Optim 1993;3(2):133–7.
[35] Sergeyev YD, Strongin RG, Lera D. Introduction to global optimization exploiting space-filling curves. New York: Springer-Verlag; 2013.
[36] Strongin RG. Algorithms for multi-extremal mathematical programming problems employing the set of joint space-filling curves. J Global Optim 1992;2(4):357–78.
[37] Sutton MA, Whitley D, Lunacek A, Howe M. PSO and multi-funnel landscapes: how cooperation might limit exploration. In: GECCO'06 proceedings of the eighth annual conference on genetic and evolutionary computation. New York: Association for Computing Machinery (ACM); 2006. p. 75–82.
[38] Wales DJ, Doye JPK. Global optimization by basin-hopping and the lowest energy structures of Lennard–Jones clusters containing up to 110 atoms. J Phys Chem A 1997;101(28):5111–6.
[39] Zhang L-S, Ng C-K, Li D, Tian W-W. A new filled function method for global optimization. J Global Optim 2004;28(1):17–43.