A Wide Class of Test Functions for Global Optimization

FABIO SCHOEN
Dept. Computer Sciences, University of Milano, via Comelico 39/41, I-20135 Milano, Italy.

(Received: 9 July 1991; accepted: 11 May 1992)

Key words. Global optimization, test functions, metric interpolation.

In this short note a class of test functions is proposed for use as a common benchmark for (essentially unconstrained) global optimization problems. This class of functions was first introduced in Gordom and Wixom (1978) and used in a global optimization context in Betrò (1984) and in Betrò and Schoen (1991).

The main advantages of functions within this class are:

- they are easily built for any dimension
- their global minimum and maximum are known *a priori*
- their smoothness is controllable by means of a set of parameters
- the number and location of stationary points is controllable by the user.

The above advantages make this class, in the author's opinion, an ideal testbed for global optimization both by building special instances and by randomly generating test functions with similar structure.

The definition of the test functions is the following

$$f(x) = \frac{\Sigma_{i=1}^{k} f_i \Pi_{j \neq i} \|x - z_j\|^{\alpha_j}}{\Sigma_{i=1}^{k} \Pi_{j \neq i} \|x - z_j\|^{\alpha_j}} ,$$

where:

- $x \in [0, 1)^N, \quad N \in \mathbb{N}$
- $k \in \mathbb{N}$
- $z_j \in [0, 1]^N \quad \forall j = 1, \ldots, k$
- $f_i \in \mathbb{R} \quad \forall i = 1, \ldots, k$
- $\alpha_i \in \mathbb{R}^+ \quad \forall i = 1, \ldots, k$

and the norm used is the euclidean norm (although different norms might be used as well).

Journal of Global Optimization **3**: 133–137, 1993.

The main properties of these functions are the following:

1. $f(z_i) = f_i \quad \forall i = 1, \ldots, k$
2. $\min_{i=1,k} f_i \leq f(x) \leq \max_{i=1,k} f_i \quad \forall x \in [0, 1]^N$
3. If $\alpha_i > 1$ then $\lim_{x \mapsto z_i} \nabla f(x) = 0$
4. If $\alpha_i > 1$ then $f(x) - f(z_i) = O(\|x - z_i\|^{\alpha_i})$.

It is possible to generate functions with $\alpha_i \leq 1$, but, in general, they will not be differentiable at z_i.

Of course in the literature many other test functions have been proposed for unconstrained or simply bounded global optimization, starting with those first introduced in Dixon and Szëgo (1978) and including good collections like, e.g., those in Pardalos and Floudas (1990); the author however feels that the proposed class deserves special attention in that

- it is easy to generate any number of test functions in any dimension with as many stationary points as needed;
- families of randomly generated test functions possessing similar structure can be easily built;
- the accuracy of a global optimization algorithm can be tested *a posteriori* thanks to the knowledge of the global optimum.

The following is a C program for computing function values as well as gradients. Please notice that the code was written without using specific C constructs (like, e.g., arrays of pointers) only to let it easy to port the code to different languages.[1]

```
/**
        Test functions for global optimization
**/

#include <math.h>

/* definition of a few constants: this values may be modified */

#define SMALL 1.0E-7 /* Treshold under which the distance between 2
points is considered to be 0 */

#define MAXPTS 100  /* max number of stationary points */
#define MAXDIM 50   /* max dimension */

/**
        Function prototypes
**/

/* prototypes for ANSI - C
```

```
double function(double* x);
void gradient(double *x, double *grad)
*/

double function();
void gradient();

/**
      Global variables (to be randomly generated or read from a file)
**/

extern double  z_shep[MAXPTS][MAXDIM];  /* Stationary points */
extern double  fz_shep[MAXPTS];         /* function values   */
extern double  alpha_shep[MAXPTS];      /* smoothness parameter */
extern int     dim_shep;                /* n. of variables */
extern int     k_shep;                  /* n. of stationary points */
double
function(x)
double x[ ];
{

   int    i,j,jmin;
   double den,romin,ff;
   double roalfa[MAXPTS];

   den=0;
   roalfa[0]=0;
   for(i=0; i<dim_shep;i++)
      roalfa[0] +=(x[i] - z_shep[0][i])*(x[i] - z_shep[0][i]);
   roalfa[0]=pow(roalfa[0],(0.5*alpha_shep[0]));
   romin=roalfa[0];
   jmin=0;
   for(j=1; j<k_shep; j++) {
      roalfa[j]=0;
      for (i=0; i<dim_shep; i++)
         roalfa[j] +=(x[i] - z_shep[j][i])*(x[i] - z_shep[j][i]);
      roalfa[j]=pow(roalfa[j],(0.5*alpha_shep[j]));
      if(romin > roalfa[j]){
         romin = roalfa[j];
         jmin = j;
      }
   }
   if (romin <SMALL) {
      ff = fz_shep[jmin];
      return(ff);
   }
   ff=0;
   for(j=0; j<k_shep; j++) {
      den +=1.0/roalfa[j];
```

```
        ff += fz_shep[j]/roalfa[j];
    }
    ff /= den;
    return(ff);
}


void
gradient(x, grad)
double grad[ ],x[ ];
{
    double sumro, sumfro, sumaxro, sumafxro;
    double tmp;
    double distsq[MAXPTS];
    double distalfa[MAXPTS];
    int    i,j,l;

    for(j=0; j<k_shep; j++) {
        distsq[j]=0.0;
        for (i=0; i<dim_shep; i++) {
            distsq[j] += (x[i]-z_shep[j][i])*(x[i]-z_shep[j][i]);
        }
        if (fabs(distsq[j]) < SMALL) {
            for(i=0; i<dim_shep; i++)
                grad[i]=0.0;
            return;
        }
        distalfa[j] =pow(distsq[j],(0.5*alpha_shep[j]));
    }
    for (l=0; l<dim_shep; l++) {
        sumro=0.0;
        sumfro=0.0;
        sumaxro=0.0;
        sumafxro=0.0;
        for (j=0; j<k_shep; j++) {
            sumro += 1.0/distalfa[j];
            sumfro += fz_shep[j]/distalfa[j];
            tmp = alpha_shep[j] * (x[l] - z_shep[j][l]) /
                        (distalfa[j] * distsq[j]);
            sumaxro += tmp;
            sumafxro += tmp * fz_shep[j];
        }
        grad[l] = (sumfro * sumaxro - sumro * sumafxro) / (sumro * sumro);
    }
    return;
}
```

Note

[1]The source is available for anonymous FTP at `ghost.dsi.unimi.it`, directory `ftp/pub/schoen`. Users without FTP access may request a copy through my electronic mail address: `schoen@ghost.dsi.unimi.it`.

References

1. B. Betrò (1984), Bayesian Testing of Nonparametric Hypotheses and Its Applications to Global Optimization, *J.O.T.A.* **42**, 31–50.
2. B. Betrò and F. Schoen (1991), A Stochastic Technique for Global Optimization, *Computers Math. Applic.* **21**, 127–133.
3. L. C. W. Dixon and G. P. Szëgo (eds.) (1978), *Towards Global Optimization 2*, North-Holland, Amsterdam.
4. W. J. Gordom and J. A. Wixom (1978), Shepard's Method of "Metric Interpolation" to Bivariate and Multivariate Interpolation, *Mathematics of Computation* **32**, 253–264.
5. P. M. Pardalos and C. A. Floudas (1990), *A Collection of Test Problems for Constrained Global Optimization Algorithms*, Lecture Notes in Computer Sciences, 455, Springer-Verlag, Berlin.