The Normal Score Transformation Applied to a Multi-Univariate Method of Global Optimization

CARY D. PERTTUNEN and BRUCE E. STUCKMAN
Dept. of Electrical Engineering, University of Louisville, Louisville, KY 40292, U.S.A.

(Received: 5 October 1990; accepted: 15 August 1991)

Abstract. Nonparametric global optimization methods have been developed that determine the location of their next guess based on the rank-transformed objective function evaluations rather than the actual function values themselves. Another commonly-used transformation in nonparametric statistics is the normal score transformation. This paper applies the normal score transformation to the multi-univariate method of global optimization. The benefits of the new method are shown by its performance on a standard set of global optimization test problems. The normal score transformation yields a method that gives equivalent searches for any monotonic transformation of the objective function.

Key words. Bayesian global optimization, nonparametric statistics.

Introduction

The most general situation that can be encountered in global optimization is one in which we have no prior knowledge about the form of the objective function, and have access only to evaluations of this function (i.e. no derivative information). This suggests the use of stochastic processes to model the unknown objective function. A general stochastic process has multiple maxima and minima, and permits a wide range of functions as allowable realizations of the process. Kushner [1] proposed a univariate global optimization method based on using a Brownian motion model for the objective function. This method selects the point at each iteration that maximizes the probability of exceeding the current optimum by a predetermined constant. Stuckman [2] heuristically extended this method to higher dimensions using a multi-univariate approach.

One disadvantage in assuming a Brownian motion model for the objective function is that the resulting conditional probability density functions are normal (Gaussian). From a computational view, normality simplifies the calculations involved in finding the next point. However, strict normality cannot be assumed or justified in a typical objective function. Perttunen [3] proposed a nonparametric method of global optimization that did not require the assumption of normality. This method transforms the objective function evaluations into ranks, and performs all calculations using the ranked objective function evaluations rather than the actual objective function evaluations themselves. Perttunen and Stuckman [4] applied the rank transformation to the multi-univariate method. This application significantly decreased the number of evaluations needed for conver-

Journal of Global Optimization **2**: 167–176, 1992.

gence with 1% error for the standard test functions proposed by Dixon and Szegö [5].

Another common transformation used in nonparametric statistics is the normal score transformation. Normal scores are defined as "typical" values of the order statistics in a sample from a normal distribution. By performing standard parametric statistical tests on the normal scores of the data, the validity of the statistical test no longer depends on whether the data are normal [6]. Furthermore, the resulting test has the same asymptotic efficiency as parametric tests for normally-distributed data, and larger efficiency for non-normal data [7].

The application of the normal score transformation to the multi-univariate method is presented in this paper. The performance of the new method is compared to the multi-univariate and rank-transformed multi-univariate methods using standard test functions.

Multi-Univariate Search with Normal Score Transformation

Let our objective function be given by $f(\boldsymbol{x})$, where f: $\mathcal{R}^n \rightarrow \mathcal{R}$ over a closed hypercube S defined by:

$$S = \text{all } \boldsymbol{x}_i \text{ satisfying } b_{j1} \leq x_{ij} \leq b_{j2} \tag{1}$$

where x_{ij} is the jth component of $\boldsymbol{x}_i$, $j = 1, 2, \ldots, n$ and where b_{j1} and b_{j2} are the lower and upper bounds on the jth component of $\boldsymbol{x}_i$. The goal is to find the global minimum $f(\boldsymbol{x}^0)$ such that

$$f(\boldsymbol{x}^0) \leq f(\boldsymbol{x}) \text{ for all } \boldsymbol{x} \in S\,. \tag{2}$$

The multi-univariate method assumes that the unknown function can be modeled as samples of a Brownian motion random process along each possible one-dimensional line segment passing through the search domain. An outline of the n-dimensional search strategy (as given in [2]) is as follows.

(1) Evaluate the function at the 2^n corner points of the n-dimensional hypercube. In other words, evaluate $f(\boldsymbol{x}_i)$ for all $\boldsymbol{x}_i$ such that $x_{ij} = b_{j1}$ or $x_{ij} = b_{j2}$ for all x_{ij}, $j = 1, 2, \ldots, n$.

(2) Form a set of line segments l_{st} which connect each pair of corner points $\boldsymbol{x}_s$ and $\boldsymbol{x}_t$ (for $s \neq t$) where the function has been evaluated.

(3) Restrict the search to points along the line segments. Consider a given line segment l_{st}. Assuming the unknown function can be represented by a Brownian motion process along each one-dimensional trajectory, the expected value conditioned on the known evaluations of the function $f(\boldsymbol{x}_s)$ and $f(\boldsymbol{x}_t)$ can be found by [1]:

$$E[f(\boldsymbol{x})] = f(\boldsymbol{x}_s) + \lambda(f(\boldsymbol{x}_t) - f(\boldsymbol{x}_s))/|l_{st}| \tag{3}$$

where $\boldsymbol{x}$ is a point on the line segment l_{st}, some distance λ from the point $\boldsymbol{x}_s$,

$0 \leq \lambda \leq |l_{st}|$. The conditional variance is given by

$$\mathrm{Var}[f(x)] \propto \lambda(|l_{st}| - \lambda)/|l_{st}| . \tag{4}$$

One result obtained by assuming the Brownian motion model is that the probability density function of the objective function is normally-distributed at each point [1]. Therefore, for each point in this line segment, we can find the probability that the unknown objective function will be less than $f_m(x) - K_m$, where $f_m(x)$ is the smallest value of the function found after m guesses and K_m is some positive constant. This probability is equal to the cumulative normal distribution function evaluated at $(f_m(x) - K_m) - E[f(x)]/\sqrt{\mathrm{Var}[f(x)]}$. The location of the point that maximizes this probability over the line segment will be included in the set of possible choices for the next guess. The expression for λ, the point of maximum probability in the line segment, is given by [1], [2]:

$$\lambda = \frac{[f_m(x) - K_m - f(x_s)]|l_{st}|}{[2f_m(x) - 2K_m - f(x_s) - f(xt)]} . \tag{5}$$

The probability of reducing the current minimum by K_m at this point λ is a monotonic function of the argument of the cumulative normal distribution function. The square of this argument, denoted by $A_{\min}$, is given by [1], [2]:

$$A_{\min} = \frac{4(f_m(x) - K_m - f(x_s))(f_m(x) - K_m - f(x_t))}{|l_{st}|} . \tag{6}$$

Now, the normal score transformation will be applied to the functional evaluations. In other words, for all of the above calculations the kth smallest functional evaluation will be replaced by a "typical" value of the kth smallest number from a normal sample with mean zero and standard deviation 1. The equations for λ and $A_{\min}$ become

$$\lambda = \frac{[\psi\{f_m(x), m\} - K_m - \psi\{f(x_s), m\}]|l_{st}|}{[2\psi\{f_m(x), m\} - K_m - \psi\{f(x_s), m\} - \psi\{f(x_t), m\}]} \tag{7}$$

$$A_{\min} = \frac{4(\psi\{f_m(x), m\} - K_m - \psi\{f(x_s), m\})(\psi\{f_m(x), m\} - K_m - \psi\{f(x_t), m\})}{|l_{st}|} \tag{8}$$

where $\psi\{f(x), m\}$ is the normal score of the function evaluation at x after m iterations.

Assuming there are no ties in the data, the normal score is defined to be the $j/(m+1)$ quartile of the normal distribution. This can be expressed as [6]:

$$\psi\{j, m\} = \Phi^{-1}\left\{\frac{j}{m+1}\right\} \tag{9}$$

where $\Phi^{-1}\{\cdot\}$ is the inverse of the standard normal distribution function. Since no closed-form expression exists for $\Phi^{-1}\{\cdot\}$, it must be calculated numerically. The

approximation used by the MINITAB statistical software package is the following [8]:

$$\psi\{j, m\} \approx 4.91\left(\left(\frac{j-3/8}{m+1/4}\right)^{0.14} - \left(1-\frac{j-3/8}{m+1/4}\right)^{0.14}\right). \tag{10}$$

This approximation is used in the computer implementation of the method.

After performing the calculations in equation (8) over each line segment, the values of $A_{\min}$ are compared to find the line segment which contains the point with the lowest $A_{\min}$ (since $A_{\min}$ is a monotonic function of the probability of reducing the current minimum by K_m). In the line segment with the minimum $A_{\min}$, we can use equation (7) to give the location of the point which has the highest probability of being less than the smallest value found by at least K_m over the set of line segments. This point becomes the next guess $\boldsymbol{x}_{m+1}$.

(4) The function is evaluated at the point $\boldsymbol{x}_{m+1}$. The line segment which contains the new point is then subdivided into two segments about the new point. Line segments are then added which connect the new point to the nearest 3 old points, and the value of $A_{\min}$ is found for each of the line segments including the new ones. This process of employing equations (7) and (8) to find the next guess, evaluating the function at the next guess, and forming new segments is repeated iteratively.

Specifying the Search Parameter K_M

Before this algorithm can be implemented, the selection of the search parameter K_m must be considered. In the multi-univariate method [2], K_m is selected at each iteration as a function of the difference between the maximum and minimum function evaluations found. In the rank-transformed multi-univariate method, K_m is found as follows [4]. If n is the predetermined total number of functional evaluations that we want to use, then ideally we want to minimize the expected minimum found after N evaluations. At each step, the point that maximizes the pseudo-probability of being the minimum for all N evaluations is selected. Thus, on the mth iteration the rank we are trying to attain is $m - N$ (which is obviously impossible by definition of ranks, but acts as a relaxation-type scheme for the search method). This can be done by letting K_m be $N - m + 1$ on the mth iteration. This gives a more global search in the early iterations and a progressively more local search as the number of iterations approach N.

For the normal-score transformed method, one way of forming K_m is in terms of standard deviations of a standard normal distribution. We can say that our goal is to find normal-scored function evaluations that are a least some constant number of standard deviations below the mean. If we select 3 standard deviations as our constant (which corresponds to a probability of approximately 0.001) then the expression for K_m is:

$$K_m = \psi\{f_m(\boldsymbol{x}), m\} + 3\,. \tag{11}$$

In this case, we must keep the total number of evaluations less than the reciprocal of 0.001, in other words, less than 1000. After substituting K_m into equations (7) and (8) and simplifying, the equations for λ and $A_{\min}$ become:

$$\lambda = \frac{[3 + \psi\{f(\boldsymbol{x}_s), m\}]|l_{st}|}{[6 + \psi\{f(\boldsymbol{x}_s), m\} + \psi\{f(\boldsymbol{x}_t), m\}]} \tag{12}$$

$$A_{\min} = \frac{4(3 + \psi\{f(\boldsymbol{x}_s), m\})(3 + \psi\{f(\boldsymbol{x}_t), m\})}{|l_{st}|} . \tag{13}$$

Using this technique for choosing K_m has the advantage that, unlike the rank-transformed multi-univariate method, the rate of convergence of the optimization method will not be dependent upon the *a priori* selection of the number of evaluations. Thus, a search which evaluates the unknown function 200 times reaches the same results for the first 100 evaluations as does a search of only 100 evaluations. This makes the method more straightforward for the user and makes the comparison to global optimization less ambiguous if based upon the number of evaluations necessary for convergence.

Results on Standard Test Functions

The performance of the new method is examined using the standard set of test functions proposed by Dixon and Szegö [5]. The observed $f_{\min}$ is given at each iteration in which a new minimum function evaluation is found. The results are shown for up to 500 evaluations.

The first test functions used are the Shekel family:

$$f(x) = -\sum_{i=1}^{m} [(\boldsymbol{x} - \boldsymbol{a}_i)^T(\boldsymbol{x} - \boldsymbol{a}_i) + c_i]^{-1} \tag{14}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$, $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)^T$, $0 \leq x_j \leq 10$, $j = 1, 2, \ldots, n$, and the values for $\boldsymbol{a}_i$ and c_i are shown in Table I.

The observed $f_{\min}$ vs. the number of evaluations for $m = 5$, $m = 7$, and $m = 10$ for Shekel's (SQRIN) function are given in Tables II, III, and IV respectively.

Table I. Data for Shekel function

i	$\boldsymbol{a}_i$				c_i
1	4.0	4.0	4.0	4.0	0.1
2	1.0	1.0	1.0	1.0	0.2
3	8.0	8.0	8.0	8.0	0.2
4	6.0	6.0	6.0	6.0	0.4
5	3.0	7.0	3.0	7.0	0.4
6	2.0	9.0	2.0	9.0	0.6
7	5.0	5.0	3.0	3.0	0.3
8	8.0	1.0	8.0	1.0	0.7
9	6.0	2.0	6.0	2.0	0.5
10	7.0	3.6	7.0	3.6	0.5

Table II. Results for Shekel 5 function

Number of Evaluations	Observed f_{min}	% Error
1	−0.27311	97.310
17	−4.86009	52.132
63	−10.03485	1.166
203	−10.15290	0.003
393	−10.15307	0.001
495	−10.15309	0.001
499	−10.15310	0.001

Table III. Results for Shekel 7 function

Number of Evaluations	Observed f_{min}	% Error
1	−0.293618	97.178
17	−3.980522	61.736
52	−7.261699	30.195
79	−8.299829	20.216
86	−10.30500	0.941
203	−10.39877	0.040
342	−10.40026	0.025
381	−10.40254	0.003
400	−10.40274	0.002
442	−10.40289	0.000
498	−10.40289	0.000

Table IV. Results for Shekel 10 function

Number of Evaluations	Observed f_{min}	% Error
1	−0.3217291	96.946
17	−1.522084	85.554
25	−1.957289	81.423
43	−2.889846	72.573
60	−4.613623	56.212
61	−5.646820	46.406
94	−8.808858	16.395
105	−8.866760	15.846
184	−9.080604	13.816
250	−9.102548	13.608
260	−10.35516	1.719
262	−10.36986	1.580
264	−10.50632	0.285
323	−10.52791	0.080
346	−10.53527	0.010
364	−10.53611	0.002
410	−10.53631	0.000
497	−10.53631	0.000
500	−10.53632	0.000

The true global minimum of each of these functions is -10.1532 for Shekel 5, -10.4029 for Shekel 7, and -10.5363 for Shekel 10.

Next, the method will be used on the 3-dimensional Hartman function.

$$f(x_1, x_2, x_3) = -\sum_{i=1}^{4} \left[c_i \exp\left(-\sum_{j=1}^{3} [a_{ij}(x_j - p_{ij})^2] \right) \right] \quad 0 \leq x_i \leq 1, \quad i = 1, 2, 3. \tag{15}$$

where:

$$a_{11} = 3.0,\ a_{12} = 10,\ a_{13} = 30,\ p_{11} = 0.3689,\ p_{12} = 0.1170,\ p_{13} = 0.2673,$$

$$a_{21} = 0.1,\ a_{22} = 10,\ a_{23} = 35,\ p_{21} = 0.4699,\ p_{22} = 0.4387,\ p_{23} = 0.7470,$$

$$a_{31} = 3.0,\ a_{32} = 10,\ a_{33} = 30,\ p_{31} = 0.1091,\ p_{32} = 0.8732,\ p_{33} = 0.5547,$$

$$a_{41} = 0.1,\ a_{42} = 10,\ a_{43} = 35,\ p_{41} = 0.03815,\ p_{42} = 0.5743,\ p_{43} = 0.8828,$$

$$c_1 = 1.0,\ c_2 = 1.2,\ c_3 = 3.0,\ c_4 = 3.2.$$

The global minimum of this function is $f_{\min} = -3.8628$. The results for the Hartman 3 function are given in Table V.

Table V. Results for Hartman 3 function

Number of Evaluations	Observed $f_{\min}$	% Error
1	−0.0679741	98.240
5	−0.0913323	97.636
7	−0.334829	37.392
9	−2.41803	14.187
13	−3.314803	14.187
15	−3.408731	11.755
23	−3.620394	6.275
40	−3.735579	3.293
42	−3.770615	2.386
58	−3.780158	2.139
73	−3.789788	1.890
94	−3.796772	1.709
106	−3.837480	0.655
108	−3.838003	0.642
110	−3.838687	0.624
168	−3.839350	0.607
277	−3.839473	0.604
279	−3.839741	0.597
281	−3.839815	0.595
488	−3.839875	0.593
489	−3.839913	0.592
491	−3.839971	0.591
492	−3.839994	0.590
494	−3.840031	0.589
497	−3.840072	0.588

The first of the two-dimensional test functions used is Branin's RCOS function.

$$f(x_1, x_2) = a(x_2 - bx_1^2 + cx_1 - d)^2 + h(1-f)\cos(x_1) + h - 5 \leq x_1 \leq 10\,,$$
$$0 \leq x_2 \leq 15\,, \tag{16}$$

where $a = 1$, $b = 5.1/4\pi^2$, $c = 5/\pi$, $d = 6$, $h = 10$, $f = 1/8\pi$. This function has three global minima at $(x_1, x_2) = (-3.14159, 12.275)$, $(3.14159, 2.275)$, $(9.42478, 2.475)$ with $f_{\min} = 0.397887$. The observed $f_{\min}$ vs. number of evaluations is shown in Table VI.

The other two-dimensional test function used is the Goldstein–Price function.

$$f(x_1, x_2) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$$
$$\times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \tag{17}$$

Table VI. Results for Branin (RCOS) function

Number of Evaluations	Observed $f_{\min}$	% Error
1	3557.588000	–
2	110.156100	–
3	70.138240	–
9	23.655400	–
15	1.883487	373.372
36	1.689419	324.598
40	1.141987	187.013
44	0.746591	87.639
57	0.404955	1.776
114	0.402524	1.165
178	0.401011	0.785
226	0.399969	0.523
240	0.398964	0.271
275	0.398921	0.260
384	0.398892	0.253
431	0.398886	0.251
452	0.398819	0.234

Table VII. Results for Goldstein & Price function

Number of Evaluations	Observed $f_{\min}$	% Error
1	24376.0	–
5	147.5960	–
8	50.94152	–
13	3.627731	20.924
40	3.227506	7.584
77	3.006858	0.229
235	3.004451	0.148
280	3.004364	0.145
290	3.004169	0.139
340	3.004142	0.138

Table VIII. Number of evaluations to be within 1% error

Function	Normal Scores	Ranks	Multi-Univ.
Shekel 5	203	50	500
Shekel 7	86	50	500
Shekel 10	264	100	831
Hartman 3	106	70	93
Branin	178	80	494
Goldstein–Price	77	90	121

This function has 5 local minima, with the global minimum of these being at $(x_1, x_2) = (0.0, -1.0)$ with $f_{\min} = 3$. The observed $f_{\min}$ vs. number of iterations is given in Table VII.

Conclusions

Table VIII shows the number of evaluations needed to be within 1% error of the true minimum for each of the test functions. This is a very conservative termination criterion for a global optimization method (after this point, a local optimization method would be employed to expedite the convergence toward the minimum). These results are compared to the results obtained using the multi-univariate method and the rank-transformed multi-univariate method.

The use of normal scores significantly decreases the number of evaluations needed for convergence within 1% compared to the multi-univariate method. However, the use of the rank transformation still seems to provide faster convergence than the normal scores method. One advantage of using the normal scores method compared to the ranks method is that the total number of evaluations do not need to be prespecified (as long as the number of evaluations is less than the upper bound specified by K_m).

References

1. Kushner, H. J. (1963), A New Method of Locating the Maximum of an Arbitrary Multipeak Curve in the Presence of Noise, *Proceedings of the Joint Automatic Control Conference*, pp. 69–79.
2. Stuckman, B. E. (1988), A Global Search Method for Optimizing Nonlinear Systems, *IEEE Trans. Systems Man & Cybernetics* **18** (6), 965–977.
3. Perttunen, C. D. (1989), A Nonparametric Global Optimization Method Using the Rank Transformation, *Proceedings of the 28th IEEE Conference on Decision and Control*, Tampa, FL, pp. 888–893.
4. Perttunen, C. D. and B. E. Stuckman (1989) The Rank Transformation Applied to a Multi-Univariate Method of Global Optimization, *Proceedings of the IEEE International Conference on Systems Engineering*, Dayton, OH, pp. 217–220.
5. Dixon, L. C. W. and G. P. Szegö (1978), The Optimisation Problem: An Introduction, in L. C. W. Dixon and G. P. Szegö (eds.), *Towards Global Optimisation 2*, North-Holland.
6. van der Waerden, B. L. (1953), Order Tests for the Two-Sample Problem and Their Power, *Indagationes Mathematicae* **15**, 303–316.
7. Conover, W. J. (1980), *Practical Nonparametric Statistics*, Wiley, New York.

8. *MINITAB Reference Manual, Release 6*, Minitab Inc., State College, PA, 1988.
9. Hartman, J. K. (1972), Rep. NP55HH72051A, Naval Postgraduate School, Monterey.
10. Branin, F. K. (1972), A Widely Convergent Method for Finding Multiple Solutions of Simultaneous Non-Linear Equations, *IBM Journal of Research & Development*, pp. 504–522.
11. Goldstein, A. A. and I. F. Price (1971), On Descent from Local Minima, *Mathematics of Computation* **25**, No. 115.