Journal of Global Optimization **28:** 229–238, 2004.
© 2004 *Kluwer Academic Publishers. Printed in the Netherlands.*

A Hybrid Descent Method for Global Optimization

K.F.C. YIU[2], Y. LIU[1] and K.L. TEO[1]
[1]*Department of Applied Mathematics, Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: cedric_yiu@hkucc.hku.hk)*
[2]*Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Pokfulam, Hong Kong*

(Received and accepted in revised form 21 January 2003)

Abstract. In this paper, a hybrid descent method, consisting of a simulated annealing algorithm and a gradient-based method, is proposed. The simulated annealing algorithm is used to locate descent points for previously converged local minima. The combined method has the descent property and the convergence is monotonic. To demonstrate the effectiveness of the proposed hybrid descent method, several multi-dimensional non-convex optimization problems are solved. Numerical examples show that global minimum can be sought via this hybrid descent method.

Key words: Descent method, Global minimum, Simulating annealing

Nomenclature:
$\{a, b\}$, integers ranging from a to b $(a < b)$; $[a, b]$, real numbers between a and b $(a < b)$, T, the current temperature; α, the cooling speed; N_c, the number of cooling steps; N, the number of random perturbations for each temperature.

1. Introduction

Multi-dimensional non-convex continuous optimization problems are important in many practical applications. Many approaches which are supported by relevant convergence analysis are for finding local minima only (Rubinov, 2000). However, many local minima are useless in practice, as their corresponding cost values are much too inferior to the global minimum cost value. There are several stochastic optimization methods proposed to solve for the global minimum. They are usually heuristic in nature and very expensive to apply. Therefore, methods which are hybrid of different type of algorithms are becoming more popular.

One method is to use gradient-type methods coupled with certain auxiliary functions to move successively from one local minimum to another better one. This includes the tunnelling method (Levy an Montalvo, 1985; Yao, 1989; Cetin et al., 1993), the bridging method (Liu and Teo, 1999) and the filled function (Ge and Qin, 1987; Ge, 1987, 1990; Liu, 2001; Xu et al., 2001; Zhang et al., 2001). These methods rely heavily on the successful construction of a tunnelling function, a bridging function or a filled function to by-pass previously converged local minima. Another interesting method is to apply the cutting angle method to by-pass local minima. This involves solving subproblems of the minimax-type (Bagirov and Rubinov, 2002). For low dimensions, some of these methods have been shown to provide very reliable algorithms to locate the global minimum (Barhen et al., 1997). However, most of these methods have been tested successfully on relatively low dimensions, it is therefore unsure whether the numerical behaviour of these methods for large scale non-convex minimization problems can be retained.

In general, a good global optimization technique should have the capability of avoiding local minima, and the speed of convergence to approach stationary points. For the case of continuous decision variables, the stochastic optimization approach provides a good methodology to move away from stationary points, but it is computationally intensive to be applied. One of the reasons for this is because the method is very slow when it tries to approach or descent to stationary points. On the other hand, an analytic approach based on the gradient information is much more efficient in finding a stationary point. Ideally, we hope to retain the robustness of the stochastic optimization method and the speed of the local minimizing algorithm when it comes close to a stationary point. In this paper, a hybrid technique is proposed to combine an analytic approach with a simulated annealing algorithm.

The simulated annealing algorithm (Kirkpatrick et al., 1983; Cerny, 1985) is a stochastic optimization method. It has very nice convergent property (Locatelli, 2000 and has been widely applied for global optimization. By examining the simulated annealing algorithm, we see that its main advantage is in escaping from local minima, rather than in finding the global minimum with pre-defined precision. Therefore, we propose to combine simulated annealing algorithm with a gradient-based algorithm to form a hybrid method. The simulated annealing algorithm is used so as to escape from local minima. Then, a gradient-based algorithm is used to speed up the task of local search. If a solution obtained is not global, the set of possible descent points is still very large and therefore the probability of locating a descent point is strictly positive. Since the simulated annealing algorithm is used to locate a descent point from a previously converged local solution, the probability of finding a descent point is much larger then finding a better local minimum. Thus, it is much more efficient computationally.

The other main desirable property of the proposed hybrid descent method is that the convergence is monotonic. Since the simulating annealing is mainly used for seeking descent points, the decrease in the objective function after executing each simulated annealing search might be very small. But it is sufficient to bypass previously converged local solutions and resume local neighbourhood search. To demonstrate the effectiveness of the proposed hybrid method, several multi-dimensional non-convex optimization problems are solved. For each example, the proposed hybrid descent method locates its global solution.

2. Algorithms

Let $f(\boldsymbol{x})$ be a twice continuously differentiable non-convex function on the set $\Omega = \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{a} \leqslant \boldsymbol{x} \leqslant \boldsymbol{b}\}$, with $\boldsymbol{a}$ and $\boldsymbol{b} \in \mathbb{R}^n$. We assume that all the minima of $f(x)$ are isolated minima and that there is a finite number of them. We consider the problem of finding the global minimum of $f(\boldsymbol{x})$.

Since $f(\boldsymbol{x})$ is not convex, we cannot expect to locate its global minimum using a gradient-based approach. Theoretically, the global minimum of $f(\boldsymbol{x})$ could be sought by using the simulated annealing algorithm alone. However, its convergence rate is usually very slow. On the other hand, a gradient-based algorithm is much more efficient in converging to a stationary point. Thus, by combining the simulated annealing algorithm with a gradient-based local minimization algorithm, we obtain an efficient hybrid descent method, which is formally stated in the following:

A Hybrid Descent Algorithm

(1) Generate $\boldsymbol{x}^{(0)}$ randomly and evaluate $f(\boldsymbol{x}^{(0)})$. Set $k = 0$.

(2) Solve for the local minimum of $f(\boldsymbol{x})$ via a gradient-based minimization method with $\boldsymbol{x}^{(k)}$ as the initial guess to give $\boldsymbol{x}^{(k),*}$ such that $f(\boldsymbol{x}^{(k),*}) - f(\boldsymbol{x}^{(k)}) \leqslant -\epsilon_k$, where ϵ_k is a positive parameter.

(3) Starting from $\boldsymbol{x}^{(k),*}$, execute $\mathcal{N}$ simulating annealing iterations until a point $\boldsymbol{x}^{(k+1)}$ is obtained, such that $f(\boldsymbol{x}^{(k+1)}) - f(\boldsymbol{x}^{(k),*}) \leqslant -\delta_k$ for some positive parameter δ_k.

(4) Set $k := k + 1$. Return to Step 2 until convergence.

In Step 3 of the algorithm, the simulated annealing iterations composes of three key steps, namely

(1) the generation of the next trial point in the solution space via random perturbations,

(2) a choice of a probability distribution to govern the acceptance of uphill steps,

(3) an annealing schedule.

In this paper, following Kirkpatrick et al. (1983) and Cerny (1985), the Boltzmann probability distribution is used. The annealing schedule is determined by the parameters α, the cooling speed; N_c, the number of cooling steps; N, the number of random perturbations for each temperature; and the initial temperature, T. Typical choices of these parameters can be found in (Press et al., 1992). The simulated algorithm algorithm can be implemented as follows:

Initiation. Select α, N_c, N, and initial T. Evaluate $f(\boldsymbol{x}^{(k),*})$.

Cooling.

(a) Let j be the cooling step. Set $j = 1$.

(b) If $j \in (1, \cdots, N_c)$

   (i) $i = \text{random}\{1, 2, 3\}$.

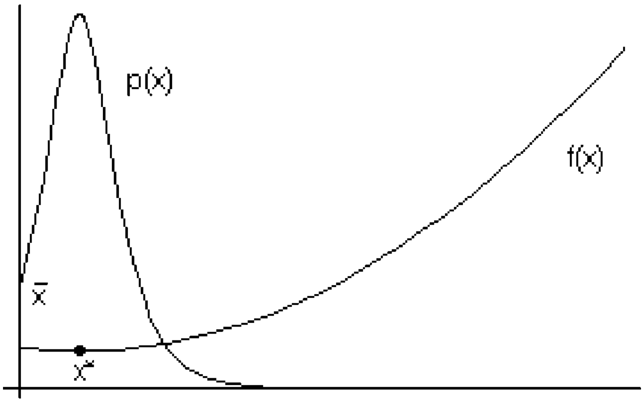

Figure 1. A typical auxiliary function.

(ii) Depending on the outcome of i, within the set Ω, re-generate randomly one of the following: one element of $\boldsymbol{x}$, or $m \in \text{random}\{1, \cdots, n\}$ elements of $\boldsymbol{x}$, or the whole vector of $\boldsymbol{x}$. This gives $\tilde{\boldsymbol{x}}$.

(iii) Calculate $D{=}f(\tilde{\boldsymbol{x}}){-}f(\boldsymbol{x})$. If $D< -\delta_k$ or random$[0,1] < T\exp(-D/T)$, then $\boldsymbol{x} = \tilde{\boldsymbol{x}}$.

(iv) Set $j := j+1$ and return to (i) until N perturbations are executed.

(c) Set $T := \alpha T$ and $j := j+1$. Return to Step (b) until N_c cooling steps are executed.

2.1. A COMPARISON WITH OTHER METHODS

Many deterministic global optimization methods rely on the successful construction of an auxiliary function to move successively from one local minimum to another better one. Most auxiliary functions constructed attempt to penalize a local minimum by weighting heavily at that point. A typical example of an auxiliary function is

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

where $\boldsymbol{x}^*$ is a local minimum. Theoretically, the minimum of this function should be different from $\boldsymbol{x}^*$. When $p(\bar{\boldsymbol{x}}) \leqslant 0$ is achieved by some $\bar{\boldsymbol{x}}$, this is sufficient to by-pass $\boldsymbol{x}^*$. It is therefore attempted to minimize this function or to enforce it as an inequality constraint. However, although $\bar{\boldsymbol{x}}$ is clearly exist, it is not a trivial task to look for the right trajectory. Starting from a fixed perturbation to $\boldsymbol{x}^*$ is not sufficient to local $\bar{\boldsymbol{x}}$. This is illustrated in Figure 1 where the variables are assumed to be greater than zeroes. It $p(\boldsymbol{x})$ is minimized with any starting points $\boldsymbol{x}^* - \epsilon$, $\bar{\boldsymbol{x}}$ sought will still be in the vicinity of $\boldsymbol{x}^*$ and will therefore be trapped by the same

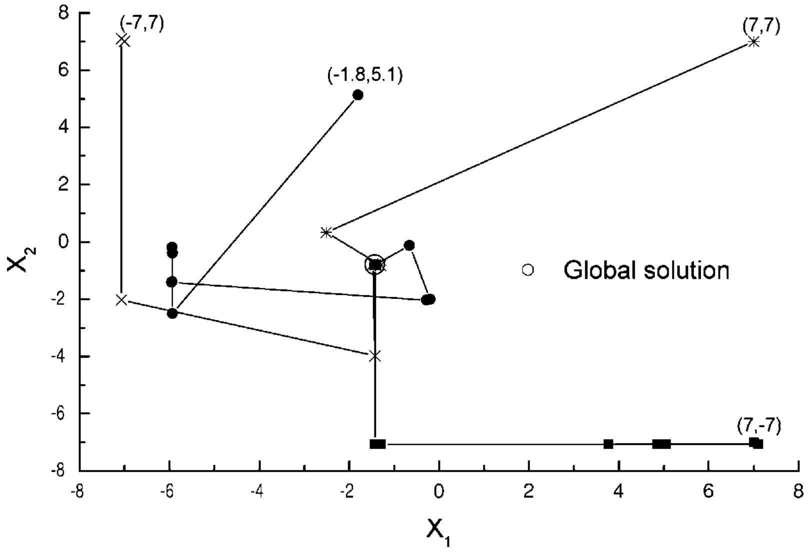

Figure 2. The trajectory of convergence for the algorithm in the (x_1, x_2)-plane.

local minimum. We believe the same situation can happen even without the nonnegative constraint. Some deterministic strategies have been attempted to perturb $\boldsymbol{x}^*$ successively (Ng and Li, 2001). This will inevitably involve the solution of a large number of nonlinear programming subproblems.

Another deterministic global optimization method is to combine the cutting angle method with a local search. The method requires the solution of a number of subproblems which is of the minimax-type and is combinatorial in nature. Although it is demonstrated to work well for relatively small problems, its efficiency towards larger problems is yet to be established.

The present method is stochastic in nature and it relies on the simulated annealing algorithm to by-pass local minimum. This resembles successive random perturbations to $\boldsymbol{x}^*$ and is able to escape from any trap with a positive probability as shown in Figure 1. Comparatively, the simulated annealing algorithm requires only re-evaluation of cost functions instead of solving computationally expensive subproblems.

3. Numerical Results

In the following, the Matlab 6 subroutine *fmincon* is used to solve for the local minima of $f(\boldsymbol{x})$. Moreover, we choose $\mathcal{N} = 1$, $\alpha = 0.9$, initial $T = 0.7$ in the simulated annealing algorithm. N_c and N are chosen to be large to prevent premature exit from the simulated annealing algorithm. All the results are executed on a Pentium IV 1.6G PC running a Linux operating system.

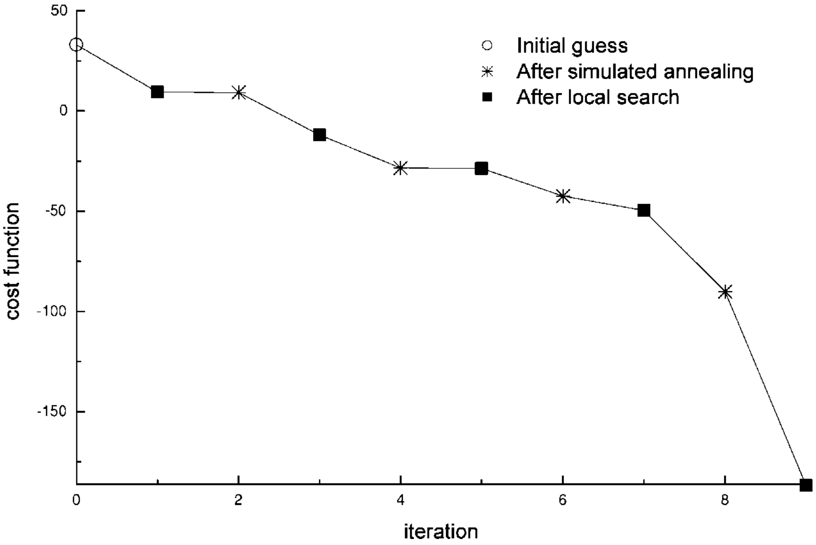

Figure 3. A typical convergence history for the algorithm in test 1.

3.1. TEST 1

The two-dimensional Shubert function (Shubert, 1972)

$$f(x_1,x_2) = \left(\sum_{i=1}^{5} i\cos[(i+1)x_1+i]\right)\left(\sum_{i=1}^{5} i\cos[(i+1)x_2+i]\right)$$

$$+\frac{1}{2}((x_1+1.42513)^2+(x_2+0.80032)^2), \quad -10 \leqslant x_i \leqslant 10, \quad i=1,2$$

is used, which has one global minimum located at

$$x_1 = -.142513, \qquad x_2 = -0.80032$$

with a function value $f(\boldsymbol{x}) = -186.73091$. The starting points are chosen to be $(7,7), (7,-7), (-7,7), (0,0)$, and (-1.8,5.1). The convergence trajectory in the x_1-x_2-plane is shown in Figure 2. A typical convergence history is depicted in Figure 3 which exhibits a perfect monotonic convergence. From different starting points, the number of function evaluations required in the simulated annealing are summarized as follows:

Starting point	Function evaluation in SA	No. of local search
(7,7)	2	2
(-7,7)	12	3
(0,0)	16	3
(7,-7)	572	5
(-1.8,5.1)	63	6

This is to compare with approximately 100 000 function evaluations if the simulated annealing algorithm is applied alone in finding the global minimum within a reasonable precision. Local search is usually very efficient in terms of number of function evaluations in locating local minima. Therefore, a significant saving has been achieved in using the hybrid descent method.

3.2. TEST 2

Another problem is taken from (Pardalos et al., 1991) which is a constrained optimization problem and can be summarized as follows:

$$f(\boldsymbol{x}) = \sum_{i=1}^{k-1}\left(x_i - \frac{r_i}{r_{i+1}}x_{i+1}\right)^2 + \sum_{i=k+1}^{n} x_i^2 - \left(\sum_{i=1}^{k} x_i\right)^2,$$

$$\sum_{i=1}^{n} x_i = 1, \quad x_i \geqslant 0, \quad i = 1, \cdots, n, \quad k = [n/2] + 1,$$

$$r_i = \frac{r'_i}{\sum_{i=1}^{k} r'_i}, \qquad r'_i = 5|\sin(i)| + 0.1, \quad i = 1, \cdots, k.$$

The problem has a global minimum at $\boldsymbol{x}^{*T} = (r_1, \cdots, r_k, 0, \cdots, 0)^T$ with $f(\boldsymbol{x}^*) = -1$. The dimension $n = 100$ is used. The starting point is chosen to be $\boldsymbol{x}^{0T} = (0, \cdots, 0, 1)^T$ which has a function value $f(\boldsymbol{x}^0) = 0$. It took a few minutes to reduce the cost function to the global value with two local searches taken.

3.3. TEST 3

Another problem is taken from (Griewank, 1981; Bagirov and Rubinov, 2002) and can be written as:

$$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1.$$

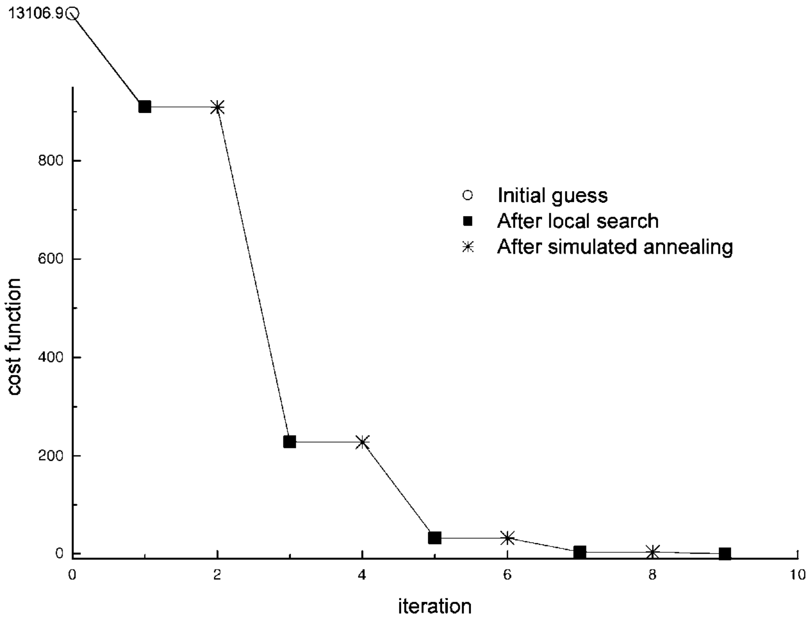

Figure 4. A typical convergence history for the algorithm in test 4.

The problem has a global minimum at $\boldsymbol{x}^{*T} = (0, \cdots, 0)^T$ with $f(\boldsymbol{x}^*) = 0$. The starting point is chosen to be $\boldsymbol{x}^{0T} = (-n, \cdots, -n)^T$ which is very far from the global solution. The dimension $n = 1000$ is used and it took about 30 min to reduce the cost function from 2.5×10^5 to the global value with two local searches taken.

3.4. TEST 4

We use the n dimensional Levy function (Levy and Montalvo, 1985)

$$f(\boldsymbol{x}) = \frac{\pi}{n}\left(k\sin^2(\pi y_1) + \sum_{i=1}^{n-1}((y_i - a)^2(1 + k\sin^2(\pi y_{i+1})) + (y_n - a)^2\right),$$

$$y_i = 1 + 0.25(x_i - 1), \qquad -10 \leqslant x_i \leqslant 10, \quad i = 1, 2, \cdots, n$$

where the constants k and a are fixed at 10 and 1, respectively. This function has many local minima, where only one of these is the global minimum. The location of this global minimum is at

$$x_i = 1, \quad i = 1, 2, \cdots, n$$

and the function value at this point is equal to 0 irrespective of the dimension of the problem. Different dimension are chosen and different starting points are generated randomly. The comparison is summarized in the following:

Dimension	cpu time	no. of local search
10	1.57–2.25 s	2–3
100	183.4–739.1 s	4–6
1000	4.5–8.8 h	4–6

A typical convergence history is shown in Figure 4. The global minimum has been found for all starting points. From Figure 4, we see that since the simulating annealing is mainly used for seeking a descent point, the decrease in the objective function after *SA* can be very small. But it is sufficient for algorithm to by-pass previous converged solutions and resume local neighbourhood search.

4. Conclusions

In this paper, a new hybrid descent method has been proposed for solving multi-dimensional non-convex continuous optimization problems. The method employs a gradient-based technique for local neighhourhood improvement and the simulated annealing technique to by-pass local solutions. Numerical results have shown that global minimum can be sought using this hybrid descent method with very nice monotonic convergence history.

5. Acknowledgements

The authors wish to thank the Research Grants Council of Hong Kong to support this research and the referees for their comments.

References

1. Bagirov, A. and Rubinov, A. (2003), Cutting angle method and a local search. *Journal of Global Optimization* 27, 193–213.
2. Barhen, J., Protopopescu, V. and Reister, D. (1997), TRUST: A deterministic algorithm for global optimization, *Science*, 276, 1094–1097.
3. Cerny, V. (1985), Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm, *Journal of Optimization Theory and Applications* 45, 41–51.
4. Cetin, B., Barhen, J. and Burdick, J. (1993), Terminal repeller unconstarined subenergy tunneling (TRUST) for fast global optimization, *Journal of Optimization and Applications* 77, 97–126.
5. Ge, R. (1987), The theory of filled function method for finding global minimizers of nonlinearly constrained minimization problems, *Journal of Computational Mathematics* 5(1), 1–9.
6. Ge, R. (1990), A filled function method for finding a global minimizer of a function of several variables, *Mathematical Programming* 46, 191–204.
7. Ge, R. and Qin, Y. (1987), A class of filled functions for finding global minimizers of a function of several variables, *Journal of Optimization Theory and Applications* 52(2), 240–252.

8. Griewank, A. (1981), Generalized descent for global optimization, *Journal of Optimization Theory and Applications* 34, 11–39.
9. Kirkpatrick, S. Jr., C. D. G. and Vecchi, M. P. (1983), Optimization by simulated annealing, *Science*, 220: 671–680.
10. Levy, A. and Montalvo, A. (1985), The tunneling algorithm for the global minimization of functions, *SIAM Journal of Scientific and Statistical Computing* 6, 15–29.
11. Liu, X. (2001), Finding global minima with a computable filled function, *Journal of Global Optimization*, 19, 151–161.
12. Liu, Y. and Teo, K. (1999), A bridging method for global optimization, *Journal of Australian Mathematical Society Series B* 41, 41–57.
13. Locatelli, M. (2000), Simulated annealing algorithms for continuous global optimization: convergence conditions, *Journal of Optimization Theory and Applications*, 104(1), 121–133.
14. Ng, C. and Li, D. (2001), A sequential constrained minimization method for constrained global optimization. In: *Proceedings of the 5th International Conference on Optimization: Techniques and Applications*, Hong Kong, pp. 990–997.
15. Pardalos, P., Ye, Y. and Han, C. (1991), Algorithms for the solution of quadratic knapsack problems, *Linear Algebra and its Applications* 152, 69–91.
16. Press, W., Teukolsky, S., Vetterling, W. and Flannery, B. (1992), *Numerical Recipes in Fortran: the art of scientific computing (2nd Ed.).* Cambridge University Press, Cambridge.
17. Rubinov, A. (2000), *Abstract Convexity and Global Optimization*. Kluwer Academic Publishers, Dordrecht.
18. Shubert, B. (1972), A sequential method seeking the global maximum of a function, *SIAM Journal of Numerical Analysis*, 9, 379–388.
19. Xu, Z., Huang, H., Pardalos, P. and Xu, C. (2001), Filled functions for unconstrained global optimization, *Journal of Global Optimization*, 20, 49–65.
20. Yao, Y. (1989), Dynamic tunneling algorithm for global optimization, *IEEE Transactions on Systems, Man, and Cybernetics* 19, 1222–1230.
21. Zhang, L., Ng, C., Li, D. and Tian, W. (2001), A new filled function method for global optimization, *Journal of Global Optimization*, to appear.