# Application of some recent stochastic global optimization algorithms to practical problems


## M. M. Ali

Research Fellow, Department of Computer Science, Åbo Akademi University,
Turku Centre for Computer Science, Turku, Finland

## C. Storey

Research Professor, Department of Mathematical Sciences,
DeMontfort University, The Gateway, Leicester, U.K.

## A. Törn

Professor, Department of Computer Science, Åbo Akademi University,
Turku, Finland



Turku Centre for Computer Science
TUCS Technical Report No 47
October 1996
ISBN 951–650–842–1
ISSN 1239–1891

**Abstract**

We describe global optimization problems from three different fields representing many body potentials in physical chemistry, optimal control of a chemical reactor, and fitting a statistical model to empirical data. Historical backgrounds for each of the problems as well as the practical significance of the first two are given. The problems are solved by using eight recently developed stochastic global optimization algorithms representing control random search (4 algorithms), simulated annealing (2 algorithms) and clustering (2 algorithms) and the results are discussed. The importance of global optimization in respective field are focussed.

**Keywords:** Global optimization, real life problem, pig-liver, likelihood function, many-body potentials, optimal control.

## 1. Introduction

Increasing demand for global optimization in many areas such as engineering design, applied and social sciences has encouraged many researchers to devote themselves to finding a reliable global optimization algorithm. Despite the research in the last couple of decades, the global optimization problems remain intractable, theoretically at least. Research is continuing on both the probabilistic and the deterministic algorithms. As a result considerable advances have been made in both directions and some algorithms have been developed. A good account of deterministic methods can be found in (Ref.1,2). Among the stochastic methods Multilevel Single Linkage (MSL) (Ref.3), Topographical Multilevel Single Linkage (TMSL) (Ref.4), Simulated Annealing (SA) (Ref.5), Aspiration based Simulated Annealing (ASA) (Ref.6,7), Controlled Random Search (CRS1, CRS2, CRS3) (Ref.8,9,10) and Modified Controlled Random Search Algorithms (CRS4, CRS5) (Ref.11) are recent.

The variety of techniques that have been proposed is impressive, but their relative merits have neither been analysed in a systematic manner nor properly investigated by computational experiments on a wide range of problems. A handful of problems (Ref.12) that have been used so far to test the global optimization algorithms are simple and well defined mathematical functions. They all have a small number of local minima and are categorized as easy-to-solve problems, according to a recent study by Törn, et al (Ref.13). With the advent and the blending of sciences demands for the development of efficient global optimization algorithms to tackle difficult optimization problems rapidly is in rise. Recent advances in global optimization together with the development of massively parallel high performance computers may help solve larger problems bringing more successes to many advanced engineering fields. With this view in mind, we feel that there is an important need for more difficult and relatively large scale global optimization test problems to be considered in order to assess recent global optimization algorithms. It is also important to see which algorithms suit which types of problems. The problems we consider are of practical interest and they may form a set of test functions for global optimization. The first two represent two large classes of problems and their dimensions can easily be increased if desired. We introduce these problems of great scientific importance in different practical fields of application, aim to solve them and discuss the results and the performances of the stochastic algorithms on them. The three problems we consider are from the fields of material science, chemical engineering and applied statistics. They are very hard in the sense that they have many local minima and that they represent high nonlinearities in their mathematical representations. The aim of the paper is therefore twofold : firstly to bring the attention of these problems to the global optimization community and secondly to make aware of the successful use of some global optimization methods to those working in the fields of the problems. An extensive numerical investigation involving these problems using the above mentioned global optimization algorithms can be found in Ali (Ref.6).

The objective of this paper is also to describe the problems in detail so that one can use them as standard problems to test the global optimization algorithms that are yet to be developed in years to come. We also present summarised numerical results for each problem. The organization of the paper is as follows. In section 2 the full description of each problem is given. In section 3 numerical results are discussed. Finally in section 4 some concluding remarks towards solving these practical problems based on the performance of each method are given.

## 2. Practical problems

**2.1 Many-body Potential Function (MP):** Global optimization techniques are central to the solution of macromolecular modelling and simulation problems, since many fundamental problems in these areas are formulated as large scale global optimization problems. The problem of finding the least energy structure for a given molecular system arises in : (i) the study of protein structures in biological chemistry – where it is assumed that the native protein structure corresponds to the global minimum of the potential energy function (Ref.14,15); (ii) the study of large confined ionic systems in plasma physics (Ref.16,17,18). Configurations of large ion structures are of special interest because of an important phase transition and (iii) molecular optimization of clusters – a group of identical atoms or molecules with specific geometrical and chemical properties (Ref.19). Much of the interest is due to the recent discovery of stable carbon fullerene (Ref. 20). Although, small clusters ejected by simple potentials have been studied in (Ref.21,22), no numerical studies have been carried out with more involved and complex potential such as the semi-empirical potential. Our current interest lies with such a potential associated with the molecular optimization of clusters, namely the Tersoff's semi-empirical potentials for silicon (Ref.23,24) and Tersoff-like potentials (parameterizations of Tersoff's potential) for arsenic (Ref.25,26). The problem arose as a result of some work involving molecular dynamics (MD) to simulate the ejection of particles after a solid surface had been bombarded with energetic ions. In any simulation of this process it is necessary that the interaction potentials give roughly the correct energetics and structure of the clusters formed by the ejected particles. Therefore the problem is to calculate the minimum energy of clusters of particles which interact through well-defined many-body interaction potentials. Before we define the underlying function to be optimized we need to describe the following terms explicitly.

The binding energy in the Tersoff formulation (see, Ref.24) is written as a sum over atomic sites in the form

$$E_i = \frac{1}{2} \sum_{j \neq i} f_c(r_{ij}) \big( V_R(r_{ij}) - B_{ij} V_A(r_{ij}) \big) \ , \quad \forall i \tag{2.1.1}$$

where $r_{ij}$ is the distance between atoms $i$ and $j$, $V_R$ is a repulsive term, $V_A$ is an attractive term, $f_c(r_{ij})$ is a switching function and $B_{ij}$ is a many-body term that depends on the positions of atoms $i$ and $j$ and the neighbours of atom $i$. More details of each of these quantities can be found in (Ref.23,25,26,27). The term $B_{ij}$ is given by

$$B_{ij} = (1 + \gamma^{n_1} \xi_{ij}^{n_1})^{-1/2n_1} \tag{2.1.2}$$

where $n_1$ and $\gamma$ are known fitted parameters (Ref.24). The term $\xi_{ij}$ for atoms $i$ and $j$ is given by

$$\xi_{ij} = \sum_{k \neq i,j} f_c(r_{ik}) g(\theta_{ijk}) \tag{2.1.3}$$

where $\theta_{ijk}$ is the bond angle between bonds $ij$ and $ik$ and $g$ is given by

$$g(\theta_{ijk}) = 1 + c^2/d^2 - c^2/[d^2 + (h - \cos\theta_{ijk})^2] \ . \tag{2.1.4}$$

The quantities $c, d$ and $h$ which appear in (2.1.4) are also known fitted parameters. The terms $V_R(r_{ij})$ and $V_A(r_{ij})$ are given by

$$\begin{aligned} V_R(r_{ij}) &= Ae^{-\lambda_1 r_{ij}} \\ V_A(r_{ij}) &= Be^{-\lambda_2 r_{ij}} \end{aligned} \tag{2.1.5}$$

where $A, B, \lambda_1$ and $\lambda_2$ are given fitted parameters. The switching function $f_c(r_{ij})$ is given by

$$f_c(r_{ij}) = \begin{cases} 1 \ , & r_{ij} \leq R - D \\ \frac{1}{2} - \frac{1}{2}\sin[\pi(r_{ij} - R)/(2D)] \ , & R - D \leq r_{ij} \leq R + D \\ 0 \ , & r_{ij} \geq R + D \end{cases} \tag{2.1.6}$$

Three distinct versions of this problem have been considered denoted by $As$, Smith (Ref.16); $Si(B)$ and $Si(C)$, Tersoff (Ref.24,27). Each problem has its own set of parameters that are given in Table 2.1.1.

Each atom, say atom $i$, has its own potential energy, $E_i$, given by (2.1.1). The sum in (2.1.1) is taken over $j$, all neighbours of $i$. Therefore to determine the potential energy of a single particle one has to calculate (2.1.1) which involves the calculation of (2.1.2)-(2.1.6) for each neighbour of that particle. Notice that the energy of a particle depends upon the distances and angles subtended with respect to the other particles and therefore different particles have different energies. The objective function becomes the total energy for all atoms, i.e.,

$$f = \sum_i E_i, \quad \forall i \ , \tag{2.1.7}$$

of which the global minimum $f^*$ has to be found. It is clear from (2.1.7) that $f$ is a function of atomic coordinates. Therefore we consider the atomic positions in two and three dimensional space as variables. In the next section we describe more details of all the variables involved.

Table 2.1.1 : Parameters for *As*, *Si(B) and Si(C)*

| | As | Si(B) | Si(C) |
|---|---|---|---|
| c | 5.2731318 | 4.8381 | 1.0039E+5 |
| d | 0.75102662 | 2.0417 | 16.216 |
| h | 0.15292354 | 0.0000 | $-0.59826$ |
| $n_1$ | 0.60879133 | 22.956 | 0.78734 |
| $\gamma$ | 0.00748809 | 0.33675 | 1.0999E-6 |
| $\lambda_1$ | 6.739581257 | $3.2394 \times CS_i$ | $2.4799 \times CS_i$ |
| $\lambda_2$ | 4.886847795 | $1.3258 \times CS_i$ | $1.7322 \times CS_i$ |
| A | 10.45561332 | $3.2647E3$ | $1.8308E3$ |
| B | 14.41961332 | $9.5373E1$ | $4.7118E2$ |
| R | $3.50/CA_s$ | $3.0/CS_i$ | $2.85/CS_i$ |
| D | $0.15/CA_s$ | $0.20/CS_i$ | $0.15/CS_i$ |

$CA_s = 5.6537/2$, $CS_i = 5.4307/2$

**Problem formulation**

In order to calculate the minimum potential energy for small numbers of particles, we first fix a particle at the origin and choose our second particle to lie on the positive x-axis. The third particle is chosen to lie in the x-y plane. Therefore the variables involving the third particle are radial distance of the position of the particle from origin and its polar angle. Since the position of the first particle is always fixed and the second particle is restricted to the positive x-axis, this gives a minimisation problem involving three variables for three particles (P3). For four particles (P4), additionally three variables (the cartesian co-ordinates of the 4-th particle) are required to give a minimisation problem in six independent variables. Further particles (P5 and P6) are added to determine the energetics of small clusters. Therefore, for clusters of 4, 5 and 6 particles the number of dimensions become 6, 9 and 12 respectively. The physics involved in the problems imposes the following restrictions on the variables. The first two variables (distances between first-second, first-third particles) are taken to lie in $[0, 1.16]$ for *Si* and $[0, 1.30]$ for *As*. The third variable (polar angle) for all cases is taken to lie in, $[0, \pi]$, and all other variables are restricted to $[-1.5, 1.5]$ for both *Si* and *As*. However, we invoke symmetry arguments to constrain the third coordinates of the fourth and fifth particles to be non-negative and non-positive respectively. We do not constrain the coordinates of the sixth particle, preferring instead to calculate all the symmetric structures (with respect to the 6th particle) which are identical apart from rotational symmetry as a check on the accuracy of the calculation. The complete search regions for each group of partices are given in Table 2.1.2.

Table 2.1.2 : Search Region for each Problem

| | $As$ | $Si(B)$, $Si(C)$ | |
|---|---|---|---|
| $n$ | | | |
| 3 | $0 \le x_1, x_2 \le 1.30$<br>$0 \le x_3 \le \pi$ | $0 \le x_1, x_2 \le 1.16$<br>$0 \le x_3 \le \pi$ | P3 |
| 6 | $-1.5 \le x_4, x_5 \le 1.5$<br>$0 \le x_6 \le 1.5$ | $-1.5 \le x_4, x_5 \le 1.5$<br>$0 \le x_6 \le 1.5$ | P4 |
| 9 | $-1.5 \le x_7, x_8 \le 1.5$<br>$-1.5 \le x_9 \le 0$ | $-1.5 \le x_7, x_8 \le 1.5$<br>$-1.5 \le x_9 \le 0$ | P5 |
| 12 | $-1.5 \le x_i \le 1.5$<br>$i$=10, 11, 12 | $-1.5 \le x_i \le 1.5$<br>$i$=10, 11, 12 | P6 |

To demonstrate the complex nature of the objective function for this problem we now define this function for four atoms $i, j, k$ and $l$. Therefore, for P4, let $x_1$ be the $x$-coordinate of the 2$^{\text{nd}}$ particle, $x_2$ be the radial distance from the origin to the 3$^{\text{rd}}$ particle, $x_3$ the angle the 3$^{\text{rd}}$ particle subtends with the $x$-axis, $x_4, x_5, x_6$ the coordinates of the 4$^{\text{th}}$ particle. Then

$$f = E_i + E_j + E_k + E_l = f(x_1, \ldots, x_6)$$

where, for example the $E_i$ is given by

$$\begin{aligned}
E_i = 0.5\Big[&f_c(r_{ij})\,[V_R(r_{ij}) - B_{ij}V_A(r_{ij})] + f_c(r_{ik})\,[V_R(r_{ik}) - B_{ik}V_A(r_{ik})] + \\
&f_c(r_{il})\,[V_R(r_{il}) - B_{il}V_A(r_{il})]\Big] \\
= 0.5\Big[&f_c(r_{ij})\Big[Ae^{-\lambda_1 r_{ij}} - Be^{-\lambda_2 r_{ij}}\,\{1 + \gamma^{n_1}\,(f_c(r_{ik})g(\theta_{ijk}) + f_c(r_{il})g(\theta_{ijl}))^{n_1}\}^{-\frac{1}{2n_1}}\Big] \\
&+f_c(r_{ik})\Big[Ae^{-\lambda_1 r_{ik}} - Be^{-\lambda_2 r_{ik}}\,\{1 + \gamma^{n_1}\,(f_c(r_{ij})g(\theta_{ikj}) + f_c(r_{il})g(\theta_{ikl}))^{n_1}\}^{-\frac{1}{2n_1}}\Big] \\
&+f_c(r_{il})\Big[Ae^{-\lambda_1 r_{il}} - Be^{-\lambda_2 r_{il}}\,\{1 + \gamma^{n_1}\,(f_c(r_{ij})g(\theta_{ilj}) + f_c(r_{ik})g(\theta_{ilk}))^{n_1}\}^{-\frac{1}{2n_1}}\Big]\Big]
\end{aligned}$$

Notice that expressions for the other particles, $E_j$, $E_k$ and $E_l$, are similar which directly follows from (2.1.1). For brevity the functional expressions in terms of the independent variables for these particles are omitted from this paper but can be found in full in (Ref.6).

Hence in terms of the variables $x_i,\ i = 1, 2, \ldots, 6$,

$$f = 0.5\Bigg[f_c(x_1)\Big[Ae^{-\lambda_1 x_1} - Be^{-\lambda_2 x_1}\Big\{1 + \gamma^{n_1}\big(f_c(x_2)(1 + c^2/d^2 - c^2/[d^2 + (h - \cos x_3)^2]) + f_c(\sqrt{(x_4^2 + x_5^2 + x_6^2)})(1 + c^2/d^2 - c^2/[d^2 + (h- \frac{x_4}{\sqrt{(x_4^2 + x_5^2 + x_6^2)}})^2])\big)^{n_1}\Big\}^{-\frac{1}{2n_1}}\Big] + f_c(x_2)\Big[Ae^{-\lambda_1 x_2} - Be^{-\lambda_2 x_2}\Big\{1 + \gamma^{n_1}(f_c(x_1)(1+ c^2/d^2 - c^2/[d^2 + (h - \cos x_3)^2]) + f_c(\sqrt{(x_4^2 + x_5^2 + x_6^2)})(1 + c^2/d^2 - c^2/[d^2 + (h - \frac{x_4 \cos x_3 + x_5 \sin x_3}{\sqrt{(x_4^2 + x_5^2 + x_6^2)}})^2])\big)^{n_1}\Big\}^{-\frac{1}{2n_1}}\Big] + f_c(\sqrt{(x_4^2 + x_5^2 + x_6^2)})\Big[Ae^{-\lambda_1\sqrt{(x_4^2+x_5^2+x_6^2)}} - Be^{-\lambda_2\sqrt{(x_4^2+x_5^2+x_6^2)}}\Big\{1 + \gamma^{n_1}\big(f_c(x_1)(1 + c^2/d^2 - c^2/[d^2 + (h - \frac{x_4}{\sqrt{(x_4^2 + x_5^2 + x_6^2)}})^2]) + f_c(x_2)(1 + c^2/d^2 - c^2/[d^2 + (h - \frac{x_4 \cos x_3}{\sqrt{(x_4^2 + x_5^2 + x_6^2)}} - \frac{x_5 \sin x_3}{\sqrt{(x_4^2 + x_5^2 + x_6^2)}})^2])\big)^{n_1}\Big\}^{-\frac{1}{2n_1}}\Big]\Bigg] + \text{expressions for } E_j + E_k + E_l\,.$$

Skillful programming may reduce the lengthy and complex presentation of the objective function. For user convenience a FORTRAN program for calculating the function value of *Si(B)P6* is provided in the Appendix.

The molecular optimization problem that we have formalised is difficult because the presence of a large number of local minimizers, even for a system with a small number of atoms, creates numerous regions of attraction for local searches. Some characteristics of the problems were extracted by performing local search from 10000 random points within the search region. This gave the probabilities that a local search started from a point sampled at random in the search region would reach the best known minimum, see Table 2.1.3. One typical such experiment for *AsP6* gave the following data: 5173 points escaped the sampling region, 10 points reached the single global minimizer giving $f^* = -19.58$, 144 minima were reached multiple times and 192 minima were represented by a single point. Several of the minima could be recognized as extended 3, 4 and 5 body minima. For $P4, P5, P6$ experiments gave an indication that there might be an infinite number of local minimizers. Based on the estimated probabilities we conclude that $P3$,$P4$ and $Si(B)P5$ are rather easy to solve. The others are more difficult with $Si(C)P5, P6$ already respectably difficult. The problems $Pn,\ n = 7, 8...$ are increasingly difficult to solve.

Table 2.1.3 : Best known minima and their probabilities in %

| | As | | Si(B) | | Si(C) | |
|---|---|---|---|---|---|---|
| $P3$ | $-6.86$ | (14.) | $-7.87$ | (34.) | $-5.33$ | (26.) |
| $P4$ | $-10.65$ | (7.5) | $-15.71$ | (11.) | $-8.00$ | (6.7) |
| $P5$ | $-14.78$ | (1.3) | $-20.40$ | (4.7) | $-12.08$ | $(< .01)$ |
| $P6$ | $-19.58$ | (0.1) | $-26.52$ | (0.3) | $-15.88$ | $(< .01)$ |

**2.2 Tank Reactor Problem (TR):** In the determination of optimal control of nonlinear systems, numerical methods must be used since the problem cannot be solved analytically. The standard control vector iteration procedure based on Pontryagin's maximum principle (Ref.28) cannot solve a control problem with multiple minima, even if numerous starting values are used. The dynamic programming (DP) method (Ref.29,30) was used to overcome this but DP involves many parameters and it can only handle problems of small size. In (Ref.31) a specially designed multistart method with line search is proposed. However, solving nonlinear optimal control problems with multiple minima by global optimization in general has never been addressed either by control engineers or by the global optimization community. For the purpose of our aforementioned objective we apply global optimization algorithms to the control problem after discretizing it into a form suitable for nonlinear programming. The approach is independent of the size of the problem, i.e., the number of equations involved. This we hope would pave the way towards the solution of nonlinear optimal control problems for the control engineers which they may be unaware of so far. The problem we consider here is a model of a nonlinear continuous stirred tank reactor which involves two different local minima. The problem was studied in (Ref.30). The equations describing the chemical reactor are:

$$\begin{aligned}
\dot{x}_1 &= -(2+u)(x_1+0.25) + (x_2+0.5)\exp\left(\frac{25x_1}{x_1+2}\right) \ , \\
\dot{x}_2 &= 0.5 - x_2 - (x_2+0.5)\exp\left(\frac{25x_1}{x_1+2}\right) \ , \\
\dot{x}_3 &= x_1^2 + x_2^2 + 0.1\ u^2.
\end{aligned} \tag{2.2.1}$$

The control $u$ is unconstrained and the performance index is given by

$$f = x_3(0.78)\ . \tag{2.2.2}$$

The initial condition is $x(0) = (0.09, 0.09, 0.0)$ and the interval of integration is $0 \leq t \leq 0.78$. It is required to compute the control variable $u(t)$ $(0 \leq u(t) \leq 5)$ which will minimize the performance index $x_3(0.78)$. The problem has a global minimum $x_3(0.78) = 0.13309$ and a local minimum $x_3(0.78) = 0.24442$. These two values of the performance index are directly associated with two different control trajectories (Ref.30).

For this problem we discretize the time interval so that the number of time steps $t_i = 0.06(i-1), i = 1, \ldots, 13$ becomes the number of variables, $n = 13$, and the constant controls used for each time step become the variables. Notice that the number of variables can easily be doubled by halving each time step for a greater accuracy and this has been discussed in (Ref.6). We used a variable step and variable order Runge-Kutta routine, D02CAF, from the NAG library for integration. The routine therefore uses constant control, $u(i-1)$, to integrate the system from time step $t_{i-1}$ to step $t_i$, $i = 1, 2, \ldots, n$. Clearly in order to calculate a function value the system (2.2.1) has to be integrated from the left to the right with the given initial condition. If $x_i$, $i$=1, …, $n$ be the constant control to integrate the system of equations (2.2.1) then $x_i \in [0, 5]^n$.

**2.3 Pig-Liver Likelihood Function (PL):** This example arises from a statistical analysis of the elimination rates of flowing substrates in pig-liver. The problem is to estimate the parameters of a model of steady-state elimination by the standard statistical procedure of maximum likelihood estimation. The full details of the mathematical model are given in (Ref.32). Experimental measurements of elimination rates, $V_{ij}$ (for $j$-th experiment on the $i$-th pig-liver), on 5 pig livers, each measured under four or five different conditions are given in Table 2.3.1 and these are given data for PL. Notice that in this table there is only four experimental values for the third pig-liver (i.e., $n_3 = 4$ in equations (2.3.1) and (2.3.2) below). The statistical model fitted to this data has 12 parameters, only two of which are of interest, the other 10 are nuisance parameters. This model was first investigated by Robinson, *et al.*,(Ref.32), who considered a Bayesian approach to the problem, using uninformative priors, and obtained the marginal posterior densities of the two main parameters of interest by quadrature methods. The two parameters of interest are the Michaelis constant for the enzyme-substrate interaction ($k_m$) and the coefficient of variation for the properties of the capillaries assumed to make up the liver ($\epsilon^2$). The 10 nuisance parameters are, for each pig-liver $i$, the standard deviation $\sigma_i$ of $\ln V_{ij}$ and the maximum elimination rate, $V_{\max_i}$, of the whole liver.

Table 2.3.1 : Experimental values of $V_{ij}$

| $i \downarrow$ | $j \rightarrow$ | | $V_{ij}$ | | |
|---|---|---|---|---|---|
| Liver 1 | 0.09 | 0.23 | 0.23 | 0.33 | 0.38 |
| Liver 2 | 0.05 | 0.11 | 0.17 | 0.24 | 0.35 |
| Liver 3 | 0.26 | 0.36 | 0.55 | 0.57 | |
| Liver 4 | 0.15 | 0.21 | 0.36 | 0.41 | 0.41 |
| Liver 5 | 0.16 | 0.33 | 0.67 | 0.70 | 0.74 |

The log-likelihood function to be maximized can be written as

$$f\left(k_m, \epsilon^2, \sigma_1, \ldots, \sigma_5, V_{\max_1}, \ldots, V_{\max_5}\right) = -\sum_{i=1}^{5}(n_i \ln \sigma_i + R_i^2/(2\sigma_i^2)) \tag{2.3.1}$$

where $n_i$ is the number of experiments on the $i$-th pig-liver (see Ref.32,33). The data is analyzed using Bayesian statistical techniques with the error distribution of $\ln V_{ij}$ assumed to be normal (Ref.32). Therefore,

$$R_i^2(V_{\max_i}, k_m, \epsilon^2) = \sum_{j=1}^{n_i} (\ln V_{ij} - \ln \hat{V}_{ij})^2 \quad , \tag{2.3.2}$$

where $\hat{V}_{ij}$=modelled value for $j$-th experiment on the $i$-th pig-liver, based on $k_m, \epsilon^2, V_{\max_i}$, etc and again $n_i$ is the number of experiments on the $i$-th pig-liver. The model values of $\hat{V}_{ij}$ are found by solving the following non-linear equation

$$\frac{V_{\max_i}}{\hat{V}_{ij}} \left(1 - \left(\epsilon^2 \frac{V_{\max_i}}{2F_i k_m}\right) \Big/ \left[1 + \frac{\hat{V}_{ij}/F_i k_m}{\exp\{(V_{\max_i} - \hat{V}_{ij})/F_i k_m\} - 1}\right]^2\right) = \frac{k_m}{\hat{c}_{ij}} + 1 \tag{2.3.3}$$

where, $\hat{c}_{ij}$, the logarithmic average of $c_{ij}$ and $\bar{c}_{ij}$, is given by

$$\hat{c}_{ij} = \frac{c_{ij} - \bar{c}_{ij}}{\ln(c_{ij}/\bar{c}_{ij})} \quad . \tag{2.3.4}$$

$F_i$, $c_{ij}$ and $\bar{c}_{ij}$ are known constants associated with the $j$-th experiment of the $i$-th pig-liver whose values are given in Table 2.3.2. It is clear from (2.3.3) that $\hat{V}_{ij}$ is not expressible explicitly and must be determined by numerical solution of (2.3.3) with nonzero $\epsilon^2$. A PASCAL program for the calculation of the function value for PL can be found in (Ref.6). The number of local maxima for this problem is not known but the global maximum value 59.84 was obtained by us. We have also found two local maxima of values 54.67 and -23.025 (see Ref.6). The search region for this problem was $[0.03, 1]^{12}$.

Table 2.3.2 : Parameter Values for Each Experiment

| $i \downarrow$ | $F_i \downarrow$ | $j \rightarrow$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|
| 1 | 1.03 | $c_{ij}$ | 0.14 | 0.43 | 0.46 | 1.69 | 3.69 |
| | | $\bar{c}_{ij}$ | 0.04 | 0.20 | 0.23 | 1.30 | 3.20 |
| 2 | 0.94 | $c_{ij}$ | 0.08 | 0.15 | 0.28 | 0.43 | 1.65 |
| | | $\bar{c}_{ij}$ | 0.02 | 0.04 | 0.09 | 0.17 | 0.95 |
| 3 | 0.96 | $c_{ij}$ | 0.36 | 0.63 | 2.63 | 26.5 | |
| | | $\bar{c}_{ij}$ | 0.09 | 0.26 | 1.92 | 26.0 | |
| 4 | 1.14 | $c_{ij}$ | 0.23 | 0.36 | 0.80 | 1.47 | 4.12 |
| | | $\bar{c}_{ij}$ | 0.08 | 0.13 | 0.48 | 1.14 | 3.78 |
| 5 | 1.22 | $c_{ij}$ | 0.16 | 0.35 | 2.0 | 18.4 | 24.4 |
| | | $\bar{c}_{ij}$ | 0.04 | 0.13 | 1.53 | 18.0 | 23.6 |

## 3. Numerical results and comparisons

In this section we discuss the numerical results obtained and the performance of all algorithms used. Based on their performance further comments are made. More details of problems and extensive numerical investigations of the stochastic methods on the problems are given in (Ref.6).

In order to be able to directly compare the performance of algorithms one must either keep the effort constant and compare the quality of the results or keep the quality of the result constant (i.e., 'the global minimum must be found found') and compare the efforts needed to achieve the required quality. Because it is difficult to callibrate the methods to use the same effort for a solution and because qualities are very difficult to compare we prefer to use the latter approach, when possible, requiring that the same estimate of the global minimum must be reached by all methods.

Comparisons of the algorithms are thus made using the number of function evaluations and the cpu time on problems for which the global minima are obtained. This is complemented by using a quality measure on problems where only local minima were found by several algorithms. Note that it has not been possible in this comparison to keep the effort constant over the algorithms. This means that the result is partly due to the effect 'more effort better result'. Details are further clarified below.

The computing was carried out on the HP9000/870 computer at Loughborough University of Technology using the Pascal programming language. The random numbers were generated using the NAG routine.

**3.1 Many-body Potential Function (MP)** : We now consider all methods on problem MP. All algorithms are compared in Table 3.1.1-3.1.5 with Table 3.1.1-3.1.4 giving the number of function evaluations, FE and cpu, the cpu times in seconds, needed to achieve the global minima for problems $P3$ and $P4$. We have compared $P3$ and $P4$ separately because almost all algorithms have found the global minimum for these problems. In Table 3.1.5 the results for the problems $P5$ and $P6$ are given. A comprehensive presentation of the numerical results for this problem can be found in (Ref.6). In Table 3.1.1-3.1.4 the columns under different algorithms represent the normalized and non-normalized FE and cpu. The normalization is carried out by division with the smallest FE and cpu needed by the best algorithm (that located the global solution) for each function. Thus, the number 5.3 in the first row of Table 3.1.1 means that CRS2 needed 5.3 times more functions evaluations to reach $f^* = -7.87$ than the best algorithm, MSL, marked with the number 1.0 (i.e., $5.3 \times 210 \simeq 1123$). The normalization makes all the six problems equally important in the evaluation.

Table 3.1.1 $As$ ($P3: f^* = -6.85, P4: f^* = -10.65$)

| | | CRS2 | | CRS3 | | CRS4 | | CRS5 | | ASA | | SA | | TMSL | | MSL | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P3 | FE | 952 | 6.3 | 1281 | 8.5 | 951 | 6.3 | 820 | 5.4 | 20017 | 132.6 | 14938 | 98.9 | 228 | 1.5 | 151 | 1.0 |
| P4 | | 4075 | 2.4 | 5111 | 3.0 | 1694 | 1.0 | 2113 | 1.3 | 60994 | 36.3 | 52499 | 31.2 | 1682 | 1.0 | 7228 | 4.3 |
| AN | | | 4.4 | | 5.8 | | 3.7 | | 3.4 | | 86.6 | | 65.0 | | 1.3 | | 2.7 |
| P3 | cpu | 0.42 | 5.3 | 1.16 | 14.5 | 0.44 | 5.5 | 0.41 | 5.1 | 8.96 | 112.0 | 6.56 | 82.0 | 0.52 | 6.5 | 0.08 | 1.0 |
| P4 | | 3.92 | 2.4 | 7.70 | 4.8 | 1.62 | 1.0 | 2.06 | 1.3 | 50.97 | 31.5 | 58.51 | 36.1 | 5.63 | 3.5 | 5.48 | 3.4 |
| AN | | | 3.9 | | 9.7 | | 3.3 | | 3.2 | | 71.8 | | 59.1 | | 5.0 | | 2.2 |

AN : Average of normalized numbers

Table 3.1.2 $Si(B)$ ($P3: f^* = -7.87, P4: f^* = -15.70$)

| | | CRS2 | | CRS3 | | CRS4 | | CRS5 | | ASA | | SA | | TMSL | | MSL | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P3 | FE | 1123 | 5.3 | 1094 | 5.2 | 755 | 3.6 | 808 | 3.8 | 13992 | 66.6 | 13978 | 66.6 | 212 | 1.0 | 210 | 1.0 |
| P4 | | 4304 | 1.8 | 3724 | 1.5 | 2418 | 1.0 | 2967 | 1.2 | 27988 | 11.6 | 46458 | 19.3 | 2520 | 1.0 | 5312 | 2.2 |
| AN | | | 3.6 | | 3.4 | | 2.3 | | 2.5 | | 39.1 | | 43.0 | | 1.0 | | 1.6 |
| P3 | cpu | 0.48 | 3.4 | 0.76 | 5.4 | 0.34 | 2.4 | 0.39 | 2.8 | 5.23 | 37.4 | 5.91 | 42.2 | 0.14 | 1.0 | 0.17 | 1.2 |
| P4 | | 5.33 | 8.3 | 6.26 | 9.8 | 2.25 | 3.5 | 0.64 | 1.0 | 16.03 | 25.0 | 49.30 | 77.0 | 4.40 | 6.9 | 3.80 | 5.9 |
| AN | | | 5.9 | | 7.6 | | 3.0 | | 1.9 | | 31.2 | | 59.6 | | 4.0 | | 3.6 |

Table 3.1.3 $Si(C)$ ($P3: f^* = -5.33, P4: f^* = -7.99$)

| | | CRS2 | | CRS3 | | CRS4 | | CRS5 | | ASA | | SA | | TMSL | | MSL | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P3 | FE | 998 | 6.3 | 1092 | 6.9 | 705 | 4.4 | 612 | 3.8 | 6073 | 38.2 | 12768$^\dagger$ | - | 254 | 1.6 | 159 | 1.0 |
| P4 | | 9829 | 6.1 | 11446 | 7.1 | 7369 | 4.6 | 8521 | 5.3 | 53485 | 33.3 | 56365$^\dagger$ | - | 1606 | 1.0 | 3820 | 2.4 |
| AN | | | 6.2 | | 7.0 | | 4.5 | | 4.6 | | 35.8 | | - | | 1.3 | | 1.7 |
| P3 | cpu | 0.26 | 1.9 | 0.58 | 4.1 | 0.19 | 1.4 | 0.22 | 1.6 | 2.69 | 19.2 | 4.27$^\dagger$ | - | 0.15 | 1.1 | 0.14 | 1.0 |
| P4 | | 4.96 | 2.1 | 16.48 | 6.9 | 3.24 | 1.4 | 3.50 | 1.5 | 20.49 | 8.6 | 25.82$^\dagger$ | - | 2.82 | 1.2 | 2.39 | 1.0 |
| AN | | | 2.0 | | 5.5 | | 1.4 | | 1.6 | | 13.9 | | - | | 1.2 | | 1.0 |

$^\dagger$ : Local minimum found

Table 3.1.4 Averages of AN over $As$, $Si(B)$ and $Si(C)$

| | CRS2 | CRS3 | CRS4 | CRS5 | ASA | SA | TMSL | MSL |
|---|---|---|---|---|---|---|---|---|
| FE | 4.7 | 5.4 | 3.5 | 3.5 | 53.8 | 54.0 | 1.2 | 2.0 |
| cpu | 3.9 | 7.6 | 2.6 | 2.2 | 39.0 | 59.4 | 3.4 | 2.3 |

Table 3.1.5 : Best minima found by the methods for MP: $P5$ and $P6$

| $i$ | MP | | CRS2 | CRS3 | CRS4 | CRS5 | ASA | SA | TMSL | MSL |
|---|---|---|---|---|---|---|---|---|---|---|
| | *As* | | | | | | | | | |
| 1 | P5 | $f$ | $[-14.78]$ | $[-14.78]$ | $[-14.78]$ | $[-14.78]$ | $[-14.78]$ | -14.57 | -14.35 | -14.57 |
| | | FE | 8877 | 43173 | 6130 | 9124 | 34285 | 108406 | 1831 | 20170 |
| 2 | P6 | $f$ | $[-19.57]$ | $[-19.57]$ | -18.49 | -16.51 | -18.49 | -18.49 | -15.62 | -18.35 |
| | | FE | 38381 | 67241 | 33192 | 30784 | 10462 | 205151 | 3878 | 35147 |
| | *Si(B)* | | | | | | | | | |
| 3 | P5 | $f$ | $[-20.39]$ | -20.31 | $[-20.39]$ | -15.70 | -20.31 | -20.31 | -18.97 | $[-20.39]$ |
| | | FE | 11111 | 18348 | 3971 | 4321 | 36417 | 112951 | 3594 | 12954 |
| 4 | P6 | $f$ | -24.51 | -24.51 | $[-26.51]$ | -24.51 | -25.98 | -23.12 | -24.44 | -24.51 |
| | | FE | 37677 | 64067 | 44980 | 42134 | 151109 | 206166 | 4139 | 18666 |
| | *Si(C)* | | | | | | | | | |
| 5 | P5 | $f$ | -10.66 | -11.30 | -10.57 | -10.57 | -10.14 | -11.30 | $[-12.08]$ | -10.66 |
| | | FE | 58063 | 45173 | 12479 | 10362 | 113077 | 114438 | 5542 | 8202 |
| 6 | P6 | $f$ | -15.10 | -15.19 | $[-15.88]$ | -15.10 | -13.03 | -10.57 | -13.32 | -13.24 |
| | | FE | 182720 | 329089 | 156916 | 175233 | 66274 | 272927 | 8944 | 14027 |

Overall best minimum $f^*$ for each problem is denoted in brackets

In Table 3.1.5 it can be seen that the algorithms have failed to produce the same estimate to the global minimum for the problems $P5$ and $P6$. The best minima obtained are given in brackets. From this table we extract the reliabilities of all algorithms in finding the best solution using a quality measure. For each algorithm we therefore consider the six problems and calculate

$$\frac{1}{6}\sum_{i=1}^{6} f_i/f_i^* \tag{3.1.1}$$

in order to rank order the algorithms (see colunm three in Table 3.1.6). In the above expression for each problem $i$, $f_i^*$ is the best minimum from Table 3.1.5 and $f_i$ is the minimum obtained by the considered algorithm. For the algorithm which produces the best minimum for problem $i$, $f_i^* = f_i$ . A similar ranking is obtained just by counting the number of $f^*$ for each method giving the order 1: CRS4; 2: CRS2; 3: CRS3; 4: ASA, CRS5, MSL, TMSL; 8: SA.

A comparison based on the results in Table 3.1.4 (global minimum achieved) shows that in terms of FE and cpu the algorithms can be listed in the order of merit given by the columns for FE and cpu in Table 3.1.6. These rankings correlate as can be expected. The third column represents the reliability rank computed from Table 3.1.5 using formula (3.1.1). This column is roughly negatively correlated with the first two which also can be expected. The exception is SA and ASA which perform badly for all measures.

Table 3.1.6

| | FE | cpu | $f^*$ |
|---|---|---|---|
| 1 | TMSL | CRS5 | CRS4 |
| 2 | MSL | MSL | CRS3 |
| 3 | CRS4 | CRS4 | CRS2 |
| 4 | CRS5 | TMSL | ASA |
| 5 | CRS2 | CRS2 | MSL |
| 6 | CRS3 | CRS3 | TMSL |
| 7 | ASA | ASA | SA |
| 8 | SA | SA | CRS5 |

FE, cpu from Table 3.1.4; $f^*$ from Table 3.1.5

The cpu varies slightly over runs of the same algorithm depending on the load of the computer. We tried as best as we could to minimize this effect. We also provide unit times for all problems on the machine used. This permits cpu time comparisons over different machines. For this reason we consider one unit cpu time as the time taken to calculate 1000 FE. This unit time for each problem is given in Table 3.1.7. Notice that the unit time for *As* is comparatively higher than for *Si*. This is because of some extra numerical work is involved in the calculation of the parameters (see Table 2.1.1) for *As*. The unit times also permit to calculate the overhead of the algorithms, i.e., the time spent in addition to function evaluations. For instance in Table 3.2.1 below CRS4 used 8997 FE for the TR problem. This gives $67.8 - 8.997 \times 8.3 = 6.9$ sec, i.e., about 10% of the cpu is overhead.

Table 3.1.7: Unit time (1000 FE)

| Seconds | P3 | P4 | P5 | P6 |
|---|---|---|---|---|
| $As$ | 0.28 | 0.42 | 0.56 | 0.77 |
| $Si(B)$ | 0.21 | 0.28 | 0.42 | 0.57 |
| $Si(C)$ | 0.21 | 0.28 | 0.42 | 0.57 |
| TR | 8.3 | | | |
| PL | 4.1 | | | |

**3.2 Tank Reactor Problem (TR)** : For the implementation of all methods, on TR, we have used 13 time steps in order to get a reasonably good accuracy of integration. Table 3.2.1 compares the best result found where DP stands for Dynamic Programming (see Ref.29). Full set of results and comparisons can be found in (Ref.6).

Table 3.2.1

| | CRS2 | CRS3 | CRS4 | CRS5 | SA | ASA | MSL | TMSL | DP |
|---|---|---|---|---|---|---|---|---|---|
| FE | 12136 | 14905 | 8997 | 1342 | 157931 | 101357 | 6900 | 832 | - |
| cpu | 99.9 | 156.6 | 67.8 | 33.7 | 2382.1 | 1380.9 | 147.3 | 16.3 | 44.7 |
| $f^*$ | 0.135 | 0.136 | 0.136 | 0.245 | 0.141 | 0.144 | 0.135 | 0.148 | 0.134 |

From these results it is clear that the minimum number of function evaluations and the cpu time were obtained by TMSL but in terms of the accuracy of the solution it is 10% off the best value found. The CRS5 method failed to locate the global minimum and CRS4 was clearly the best of the 'successful' CRS methods. The overall performance of ASA is better than SA but in terms of the accuracy the SA-type methods are worse than MSL and the CRS methods. The results obtained for DP are quite good, especially in terms of accuracy but these results are deceptive since to obtain them a lot of preliminary work is needed for the determination of the appropriate values of the parameters involved (Ref.34).

**3.3 Pig-liver Likelihood Function (PL)** : Finally Table 3.3.1 summarizes the results of all methods for PL. The problem PL is concerned with finding the global maximum. This Table shows that within the CRS algorithms CRS5 is the best and in the overall comparison TMSL and SA are respectively the best and the worse algorithms. Full set of results can be found in (Ref.6).

Table 3.3.1

| | CRS2 | CRS3 | CRS4 | CRS5 | SA | ASA | MSL | TMSL |
|---|---|---|---|---|---|---|---|---|
| FE | 42796 | 272790 | 11291 | 8367 | 268568 | 89265 | 4443 | 2011 |
| cpu | 212.6 | 1350.3 | 63.8 | 57.0 | 1396.5 | 572.3 | 22.8 | 21.8 |
| $f^*$ | 54.67 | 59.84 | 59.84 | 59.84 | 59.80 | 59.84 | 59.84 | 59.84 |

Robinson, et al.,(Ref.32) solve PL by integrating out the nuisance parameters and found marginal posterior modes at $k_m = 0.225$ and $\epsilon^2 = 0.165$. Schagen (Ref.6,33) considered an alternative maximum likelihood approach, finding a maximum likelihood value of 23.98 by optimizing the full 12 parameter likelihood function. Schagen (Ref.6,33) found that the maximum likelihood estimates of these parameters of interest are somewhat different from the marginal posterior modes and means obtained by Robinson, et al.,(Ref.32). The values of the two parameters of interest obtained by Schagen were $k_m = 0.186$ and $\epsilon^2 = 0.239$. Hence it would seem that nuisance parameters have a considerable effect on the maximum likelihood estimates of the parameters of interest. We used our methods to check this conclusion by recalculating the maximum likelihood estimates. In fact we found that this 12-parameter-optimization problem has a global maximum value 59.84 giving $k_m$ and $\epsilon^2$ values close to those obtained by Robinson, *et al.*, see Table 3.3.3. The model values of $\hat{V}_{ij}$ together with maximum likelihood estimates are given in Table 3.3.2 and 3.3.3 respectively. Comparing Tables 2.3.1 and 3.3.2 it can be seen that good agreements between the experimental and model values of $V_{ij}$ have been achieved.

Table 3.3.2 : Model values ($\hat{V}_{ij}$)

| $i \downarrow$ | $j \rightarrow$ | | $\hat{V}_{ij}$ | | |
|---|---|---|---|---|---|
| Liver 1 | 0.093 | 0.217 | 0.228 | 0.345 | 0.374 |
| Liver 2 | 0.058 | 0.1003 | 0.165 | 0.222 | 0.3362 |
| Liver 3 | 0.2379 | 0.3655 | 0.548 | 0.598 | |
| Liver 4 | 0.1612 | 0.2147 | 0.3313 | 0.3884 | 0.4328 |
| Liver 5 | 0.1657 | 0.3193 | 0.6516 | 0.729 | 0.7312 |

Table 3.3.3

$k_m = 0.224,\ \epsilon^2 = 0.168\ f^* = 59.84$

| $i$ | $\sigma_i$ | $V_{\max_i}$ |
|---|---|---|
| 1 | 0.035 | 0.398 |
| 2 | 0.091 | 0.428 |
| 3 | 0.051 | 0.604 |
| 4 | 0.060 | 0.457 |
| 5 | 0.031 | 0.738 |

## 4. Conclusion

We have evaluated the performance of a number of recent stochastic global optimization algorithms on some complex practical global optimization problems. The conclusion is that using optimization algorithms designed for global optimization for these classes of problems is rewarding. However, in our opinion it is not possible to make a general conclusion about the rank ordering of the algorithms.

Our preliminary investigations and experiences over various implementations of algorithms (Ref. 6) show that for problems with small to a moderate number of local minima TMSL is the best algorithm followed by MSL. For problems with a large number of local minimizers, for example MP : $P5$ and $P6$, CRS-type algorithms are better than MSL-type, especially in finding the better solutions. From our numerical experiments we realised that the adaptive nature of this type of algorithms enable them to escape local minima even when the $N$ points of CRS-type algorithms are close to each other. This has been noticed specially for the MP problem where we believe many local minima are clustered. For this problem the performances of the CRS4 algorithm, was highly satisfactory. Moreover, CRS4 achieved an overall superiority over the rest of the CRS algorithms. Clearly the CRS algorithms are preferable to TMSL and MSL for problems with many minima.

Additional difficulties with the MSL and TMSL algorithms are that they have not only to perform multiple local searches but also that they have to store all distinct local minima and minimizers obtained. This becomes expensive both in terms of cpu time and storage for problems with a large number of local minima. This issue has been thoroughly discussed in (Ref.6). The CRS algorithms can partially overcome this drawback as CRS4 does not perform local searches at all and CRS5 only needs to perform a complete local search to refine the final solution. By nature, however, CRS stores a lot of points anyway. Under these circumstances the ASA algorithm (Ref.6,7), which is superior to SA, may be preferable because the amount of data that has to be stored is negligible. However, on the problems we have considered both SA and ASA proved to be very inefficient in all respects. This is partly because of the inherent inefficiency of simulated annealing as consisting basically of random sampling and partly because of the numerical differentiation involved in the generation mechanism (Ref.7). A direct search simulated annealing algorithm for continuous variables (Ref.35) may overcome these difficulties.

## Acknowledgement

Computational work by S. Viitanen, Åbo Akademi, for exploring characteristics of the many-body potential problems is acknowledged.

## References

1. Horst, R., and Tuy, H., *Global Optimization (Deterministic Approaches)*, Springer-Verlag, Berlin, 1990.

2. Ratschek, H., and Rokne, J., *New Computer Methods for Global Optimization*, Ellis Horwood, Chichester, 1988.

3. Rinnooy Kan, A.H.G. and Timmer, G.T., *Stochastic Global Optimization Methods; Part II : Multilevel Methods*, Mathematical Programming, vol.39, pp. 57-78, 1987.

4. Ali, M.M. and Storey, C., *Topographical Multilevel Single Linkage*, Journal of Global Optimization, vol.5, pp. 349-458, 1994.

5. Dekkers, A. and Aarts, E., *Global Optimization and Simulated Annealing*, Mathematical Programming, vol.50, pp.367-393, 1991.

6. Ali, M.M., *Some Modified Stochastic Global Optimization Algorithms with Applications*, Loughborough University of Technology, Ph.D Thesis, 1994.

7. Ali, M.M. and Storey, C., *Aspiration Based Simulated Annealing Algorithm*, to appear in the Journal of Global Optimization, 1996.

8. Price, W.L., *A Controlled Random Search Procedure for Global Optimization*, Towards Global Optimization 2, Edited by L.C.W. Dixon and G.P. Szegö, North-Holland, Amsterdam, Holland, pp.71-84, 1978.

9. Price, W.L., *Global Optimisation by Controlled Random Search*, Journal of Optimization Theory and Applications, vol. 40, pp.333-348, 1983.

10. Price, W.L., *Global Optimisation Algorithm for CAD Workstation*, Journal of Optimization Theory and Applications, vol. 55, pp.133-146, 1987.

11. Ali, M.M. and Storey, C., *Modified Controlled Random Search Algorithms*, International Journal of Computer Mathematics, Vol.54, No. 3/4, pp.229-235, 1995.

12. Dixon, L.C.W. and Szegö, G.P., *Towards Global Optimization 2*, North-Holland, Amsterdam, 1978.

13. Törn, A. Ali, M.M. and Viitanen, S., *Stochastic Global Optimization: Problem Classes and Solution Techniques*, TUCS Technical Report No. 37, 1996, Åbo Akademi University, Finland.

14. Scheraga, H.A., *Predicting Three-dimensional Structures of Oligopeptides*, in Reviews in Computational Chemistry, K. B. Lipkowitz and D. B. Boyd, eds., vol.3, VCH Publishers, pp.73-142, 1992.

15. Shalloway, D., *Packet annealing : A Deterministic Method for Global Minimization, Application to Molecular Conformation*, in Recent Advances in Global Optimization, C. Floudas and P. Pardaloa, eds., Princeton University Press, pp. 433-477, 1992.

16. Rafac, R., Schiffer, J.P., Hangst, J.S., Dubin, D.H.E. and Wales, D.J, *Stable Configurations of Confined Cold Ionic Systems*, Proc. Natl. Acad. Sci. USA., Vol.88, pp.483-486, 1991.

17. Hasse, R.W. and Schiffer, J.P., *The Structure of the Cylindrically Confined Coulomb Lattice*, Ann. Physics, Vol.203, pp.419-448, 1994.

18. Schiffer, J.P., *Phase Transitions in Anisotropically Confined Ionic Crystals*, Physics Review Letters, Vol.70, pp.818-821, 1993.

19 . Haberland, H., ed., *Clusters of Atoms and Molecules*, Springer Series in Chemical Physics, vol. 52, Springer-Verlag, 1994.

20. Kroto, H.W., Heath, J.R., ÓBrien, S.C., Curl, R.F. and Smalley, R.E., *$C60$ : Buckminsterfullerene*, Nature, Vol.318, pp.162-167, 1985.

21. Hoare, M.R, *Structure and Dynamics of Simple Microclusters*, Advances in Chemical Physics, Vol.40, pp.49-135, 1979.

22. Northby, J.A., *Structure and Binding of Lennard-Jones Clusters:* $13 \leq n \leq 14$, Journal of Chemical Physics, Vol.87, pp.6166-6177, 1987.

23. Ali, M.M. and Smith, R., *The Structures of Small Cluster Ejected by Ion Bombardment of Solids*, Vacuum, Vol.44, No 3/4, pp. 377-379, 1993.

24. Tersoff, J., *Empirical Interatomic Potential for Silicon with Improved Elastic Properties*, Physics Review B, Vol.38, No.14, pp 9902-9905, 1988.

25. Brenner, D. W., *Empirical Potential for Hydrocarbons for use in Simulating the Chemical Vapor Deposition of Diamond Films*, Physics Review B., Vol.42, No.15, pp 9458-9471, 1990.

26. Smith, R., *A Semi-empirical Many-body Interatomic Potential for Modelling Dynamical Processes in Gallium Arsenide*, Nuclear Instruments and Methods in Physics Research B, Vol.67, pp 335-339, 1992.

27. Tersoff, J., *New Empirical Approach for the Structure and Energy of Covalent Systems*, Physics Review B, Vol.37, No.12, pp 6991-7000, 1988.

28. Lapidus, L. and Luus, R., *Optimal Control of Engineering Process*, Blaisdell, Waltham, Mass., pp.243-273, 1967.

29. Luus, R., *Optimal Control by Dynamic Programming using Accessible Grid Points and Region Reduction*, Hungarian Journal of Industrial Chemistry, Vol.17, pp 523-543, 1989.

30. Luus, R. and Galli, M., *Multiplicity of Solutions in using Dynamic Programming for Optimal Control*, Workshop on Chemical Engineering Mathematics and Computation, Göttingen, West Germany; July, pp 15-20, 1990.

31. Rosen, O.and Luus, R., *Global Optimization Approach to Nonlinear Optimal Control*, Journal of Optimization, Theory and Applications, Vol.73, No.3, pp.547-562, 1992.

32. Robinson, P.J., Pettitt, A.N., Zornig, J. and Bass, L., *A Bayesian Analysis of Capillary Heterogeneity in the Intact Pig Liver*, Biometrics, Vol.39, pp 61-69, 1983.

33. Schagen, I.P., *Internal Modelling of Objective Functions for Global Optimization*, Journal of Optimization theory and Applications, Vol.51, Number 2, pp 345-353, 1986.

34. Luus, R., Dittrich, J. and Keil, F.J., *Multiplicity of Solutions in the Optimization of a Bifunctional Catalyst Blend in a Tubular Reactor*, International Workshop on Chemical Engineering Mathematics, Göttingen, Germany, July 7-11, 1991.

35. Ali, M.M., Törn, A. and Viitanen, S., *A direct Search Simulated Annealing Algorithm for Optimization Involving Continuous Variables*, TUCS Technical Report No. 38, 1996, Åbo Akademi University, Finland.

# Appendix A :

Fortran program for the objective function of problem *Si(B)P6* (Section 2.1).

```
      DOUBLE PRECISION FUNCTION TERCAL(XX)
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (LTMX=1073)
      PARAMETER (IEL=3)
      DOUBLE PRECISION CVL,ALAT
      DOUBLE PRECISION DELTA,A0
      INTEGER N
      DIMENSION CONN(IEL),CONAN(IEL),CONHH(IEL),CONPF(IEL),
     1 CONNM(IEL),CONC2(IEL),CONCD(IEL),COND2(IEL),
     2 CONC(IEL),COND(IEL),CONPE(IEL)
      DIMENSION CONEA(IEL),CONER(IEL),CONFA(IEL),CONFR(IEL),
     1 CONFCA(IEL),CONAZ(IEL)
      DIMENSION CONR(IEL),CONA(IEL),CONRR(IEL),CONDR(IEL),
     1 CONLA(IEL),CONLR(IEL),CONRL(IEL),
     2 CONRL2(IEL),CONRH(IEL),CONRH2(IEL)
      DIMENSION RX(LTMX),RY(LTMX),RZ(LTMX)
      DIMENSION RXI(LTMX),RYI(LTMX),RZI(LTMX),LATYP(LTMX)
      DIMENSION FX(LTMX),FY(LTMX),FZ(LTMX),PPE(LTMX)
      DIMENSION RL(LTMX),RLR(LTMX),LLNUM(LTMX)
      DIMENSION DXC(LTMX),DYC(LTMX),DZC(LTMX)
      DIMENSION XX(12)
        N=6
        LT=6
        L=0
        A0=5.4307
```

```
      ALAT=A0
      DO 21 I=1,N
      LATYP(I)=0
21    CONTINUE
      PI=DACOS(-1.0D0)
      DELTA=0.0
      HPI=0.5*PI
      STP=0.015
      JEL=1
      RX(2)=XX(1)
      RXI(2)=XX(1)
      RX(3)=XX(2)*COS(XX(3))
      RXI(3)=XX(2)*COS(XX(3))
      RY(3)=XX(2)*SIN(XX(3))
      RYI(3)=XX(2)*SIN(XX(3))
      DO 5555 I=4,LT
      RX(I)=XX(3*I-8)
      RY(I)=XX(3*I-7)
      RZ(I)=XX(3*I-6)
      RXI(I)=RX(I)
      RYI(I)=RY(I)
      RZI(I)=RZ(I)
      5555 CONTINUE
        FLAT=ALAT*(1.0+DELTA)
C       LATTICE CONSTANT TO LU UNIT: (THE INTERNAL LENGTH UNIT)
        CVL=0.5*FLAT
C       LATTICE UNITS TO METERS
C       2 for Si(C) 3 for Si(B)
        CONC(2)=1.0039E5
        CONC(3)=4.8381
```

```
COND(2)=16.218
COND(3)= 2.0417
CONHH(2)=-.59826
CONHH(3)= 0
CONN(2)=0.78734
CONN(3)=22.956
CONAZ(2)=1.09999E-6
CONAZ(3)=0.33675
CONLA(2)=1.7322*CVL
CONLA(3)=1.3258*CVL
CONLR(2)=2.4799*CVL
CONLR(3)=3.2394*CVL
CONR(2)=1.8308E3
CONR(3)=3.2647E3
CONA(2)=4.7118E2
CONA(3)=9.5373E1
DO 3987 JEL=1,3
  CONC2(JEL)=CONC(JEL)**2
  COND2(JEL)=COND(JEL)**2
  CONCD(JEL)=1.0+CONC2(JEL)/COND2(JEL)
  CONPE(JEL)=-1.0/(2.0*CONN(JEL))
  CONEA(JEL)=0.5*CONA(JEL)
  CONER(JEL)=0.5*CONR(JEL)
  CONFA(JEL)=CONLA(JEL)*CONA(JEL)
  CONFR(JEL)=CONLR(JEL)*CONR(JEL)
  CONRR(JEL)= 3.0/CVL
  CONDR(JEL)= 0.2/CVL
  CONRL(JEL)=CONRR(JEL)-CONDR(JEL)
  CONRL2(JEL)=CONRL(JEL)**2
  CONRH(JEL)=CONRR(JEL)+CONDR(JEL)
```

```
      CONRH2(JEL)=CONRH(JEL)*CONRH(JEL)
      CONFCA(JEL)=0.50*PI/CONDR(JEL)
 3987 CONTINUE
    DO 3000 I=1,N
      FX(I)=0.0
      FY(I)=0.0
      FZ(I)=0.0
      PPE(I)=0.0
 3000 CONTINUE
    ZIJ=0.0
    COSTH=0.0
    DO 6000 I=1,N
      L=0
      DO 4000 J=1,N
        IF(I.EQ.J) GO TO 4000
        IF(LATYP(I).EQ.0.AND.LATYP(J).EQ.0) JEL=3
        IF(LATYP(I).EQ.1.AND.LATYP(J).EQ.1) JEL=2
        DXL=RX(I)-RX(J)
        DYL=RY(I)-RY(J)
        DZL=RZ(I)-RZ(J)
        RLSQ=DXL**2+DYL**2+DZL**2
        if(rlsq.lt.1.0E-10)rlsq=1.0E-10
        DISIJ=SQRT(RLSQ)
        IF(RLSQ.GT.(CONRH2(JEL))) GO TO 4000
C       GEOMETRICAL PARAMETERS FOR ATOM J INTO L
        L=L+1
        LLNUM(L)=J
        RL(L)=SQRT(RLSQ)
        RLR(L)=1.0/RL(L)
        DXC(L)=DXL*RLR(L)
```

```
      DYC(L)=DYL*RLR(L)
      DZC(L)=DZL*RLR(L)
      IF(L.GT.1) THEN
          COSTH=(DXC(LL)*DXC(L)+DYC(LL)*DYC(L)+DZC(LL)*DZC(L))
      ENDIF
    4000 CONTINUE
    LMAX=L
C   COMPUTE FOR ALL LEGITIMATE NEIGHBORS OF I
    DO 5000 L=1,LMAX
      J=LLNUM(L)
      IF(LATYP(I).EQ.0.AND.LATYP(J).EQ.0) JEL=3
      IF(LATYP(I).EQ.1.AND.LATYP(J).EQ.1) JEL=2
  C   COMPUTE 3-BODY INTERACTIONS
      ZIJ=0.0
      XJ=RL(L)
      YA=EXP(-CONLA(JEL)*XJ)
      FAIJ=-CONEA(JEL)*YA
      YR=EXP(-CONLR(JEL)*XJ)
      FRIJ=CONER(JEL)*YR
      IF(XJ.LT.CONRL(JEL)) THEN
        FCIJ=1.0
      ELSEIF(XJ.GE.CONRL(JEL) . AND. XJ.LE.CONRH(JEL)) THEN
        ARG=CONFCA(JEL)*(XJ-CONRR(JEL))
        FCIJ=0.5*(1.0-SIN(ARG))
      ELSEIF(XJ.GT.CONRH(JEL)) THEN
        FCIJ=0.0
      ENDIF
      DO 4500 LL=1,LMAX
        IF(L.EQ.LL) GO TO 4500
    C   ENERGY PARAMETERS
```

```
      COSTH=(DXC(LL)*DXC(L)+DYC(LL)*DYC(L)+DZC(LL)*DZC(L))
      RJK=RL(L)-RL(LL)
      X=RL(LL)
      IF(X.LT.CONRL(JEL)) THEN
        FCIK=1.0
      ELSEIF(X.GE.CONRL(JEL) . AND. X.LE.CONRH(JEL)) THEN
        ARG=CONFCA(JEL)*(X-CONRR(JEL))
        FCIK=0.5*(1.0-SIN(ARG))
      ELSEIF(X.GT.CONRH(JEL)) THEN
        FCIK=0.0
      ENDIF
      CARG=COND2(JEL)+(CONHH(JEL)-COSTH)**2
      GFACAN=CONCD(JEL)-CONC2(JEL)/CARG
      ZIJ=ZIJ+FCIK*GFACAN
 4500 CONTINUE
      IF(ABS(ZIJ).LT.1.0E-10) THEN
        BIJ=1.0
      ENDIF
      IF((ZIJ.GT.0.00D0)) THEN
        IF((CONAZ(JEL)*ZIJ).GT.30) THEN
          ARG=5.184721D21
        ELSE
          ARG=1.0+( CONAZ(JEL)*ZIJ)**CONN(JEL)
        ENDIF
          BIJ=ARG**CONPE(JEL)
      ENDIF
        ZIJ=0.0
C       POTENTIAL ENERGY FOR I
        X=RL(L)
        VFAC=FRIJ+BIJ*FAIJ
```

```
        VIJ=0.0
        VIJ=FCIJ*VFAC
        PPE(I)=PPE(I)+VIJ
  5000 CONTINUE
6000 CONTINUE
  TERCAL = PPE(1)+PPE(2)+PPE(3)+PPE(4)+PPE(5)+PPE(6)
  RETURN
  END
```

Turku Centre for Computer Science
Lemminkäisenkatu 14
FIN–20520 Turku
Finland
http://www.tucs.abo.fi

University of Turku

- Department of Mathematical Sciences

Åbo Akademi University

- Department of Computer Science
- Institute for Advanced Management Systems Research

Turku School of Economics and Business Administration

- Institute of Information Systems Science