INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2007; **71**:678–702
Published online 29 December 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.1960

Improved global convergence probability using multiple independent optimizations

J. F. Schutte[1], R. T. Haftka[1,*,†] and B. J. Fregly[2]

[1]*Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611-6250, U.S.A.*
[2]*Department of Biomedical Engineering, University of Florida, Gainesville, FL 32611-6250, U.S.A.*

SUMMARY

For some problems global optimization algorithms may have a significant probability of not converging to the global optimum or require an extremely large number of function evaluations to reach it. For such problems, the probability of finding the global optimum may be improved by performing multiple independent short searches rather than using the entire available budget of function evaluations on a single long search. The main difficulty in adopting such a strategy is to decide how many searches to carry out for a given function evaluation budget. The basic premise of this paper is that different searches may have substantially different outcomes, but they all start with rapid initial improvement of the objective function followed by much slower progress later on. Furthermore, we assume that the number of function evaluations to the end of the initial stage of rapid progress does not change drastically from one search to another for a given problem and algorithmic setting. Therefore we propose that the number of function evaluations required for this rapid-progress stage be estimated with one or two runs, and then the same number of function evaluations be allocated to all subsequent searches. We show that these assumptions work well for the particle swarm optimization algorithm applied to a set of difficult analytical test problems with known global solutions. For these problems we show that the proposed strategy can substantially improve the probability of obtaining the global optimum for a given budget of function evaluations. We also test a Bayesian criterion for estimating the probability of having reached the global optimum at the end of the series of searches and find that it can provide a conservative estimate for most problems. Finally, we demonstrate the approach on a particularly challenging engineering design problem constructed so as to have at least 32 widely separated local optima. Copyright © 2006 John Wiley & Sons, Ltd.

Received 28 November 2005; Revised 2 November 2006; Accepted 3 November 2006

KEY WORDS: global optimization; multiple independent optimizations; convergence probability

*Correspondence to: R. T. Haftka, Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611-6250, U.S.A.
†E-mail: haftka@ufl.edu

1. INTRODUCTION

If we consider the general global unconstrained optimization problem for the real valued function $f(\mathbf{x})$ defined on the set $\mathbf{x} \in D$ in $\mathbb{R}^n$ we cannot state that a global solution has been found unless an exhaustive search of the set $\mathbf{x} \in D$ is performed [1–3]. With a finite number of function evaluations, at best we can only estimate the probability of arriving at or near the global optimum. To solve global optimization problems reliably, the optimizer needs to achieve an efficient balance between sampling the entire design space and sampling promising regions more densely for a more refined search [4]. Many algorithms achieve this balance, such as the deterministic DIRECT optimizer [5] or stochastic algorithms such as genetic algorithms [6], simulated annealing [7, 8], clustering [9], and the particle swarm optimizer [10].

Although these global optimization algorithms are fairly robust, they can be attracted, at least temporarily, towards optima that are not global (see, for example, the Griewank problem in Figure 1). This difficulty can be addressed by allowing longer optimization runs or an increased population size. Both options often result in a decrease in the algorithm efficiency, with no guarantee that the optimizer will reach the global optimum.

It is possible that restarting the algorithm when its progress slows down may be a more efficient approach. This idea is based on the hypothesis that several limited independent optimization runs, each with a small likelihood of finding the global optimum, may be combined to yield a vastly improved global convergence probability. This approach is routinely used for global search using multi-start, or re-start local optimizers [11–13]. Le Riche and Haftka [14], for example, have applied this approach with genetic algorithms for solving complex composite laminate optimization problems.

When using local optimizers, a restart is simple because convergence is usually well defined. With global optimizers on the other hand it is not that clear when to stop individual runs. That is, the main difficulty is deciding how a fixed budget of computational resources should be

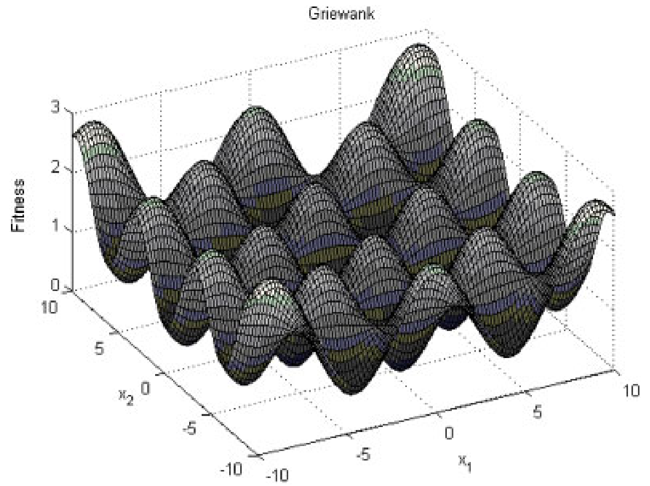

Figure 1. Multiple local minima for Griewank analytical problem surface plot in two dimensions.

divided among optimization runs. The basic premise of this paper is that different runs may have substantially different outcomes, but they all start with initial rapid progress followed by much slower progress later on. Furthermore, we assume that the number of function evaluations to the end of the rapid progress stage does not change drastically from one run to another for a given problem and algorithmic setting. Therefore we propose that the number of function evaluations required for this rapid-progress stage be estimated with one or two runs, and then the same number of function evaluations be allocated to all subsequent searches The objective of this paper is to test these assumptions and to demonstrate the effectiveness of the resulting multi-run strategy for particle swarm optimization (PSO).

The organization of this manuscript is as follows: first, a brief description of the optimization algorithm applied in this study, the PSO algorithm, is given (Section 2.1). Next, a set of analytical test problems with known global solutions are described (Section 2.2), along with details on calculating global convergence probabilities. A practical structural sizing problem is introduced in Section 2.3. After that, the multiple run methodology is outlined (Section 2.4) and a general budget strategy is presented for dividing a fixed number of fitness evaluations among multiple searches on a single processor. The use of this method on a parallel processing machine is also discussed. Numerical results based on the multiple run strategy for both single and multi-processor machines are then reported and discussed (Section 3). Finally, general conclusions about the multi-run methodology are presented (Section 4).

2. METHODOLOGY

2.1. Particle swarm optimization algorithm

The optimizer applied in this study is the particle swarm optimizer (PSO). It has strong global search capabilities and, being a stochastic population-based optimizer, is an ideal candidate for parallelization [15]. The PSO, introduced by Kennedy and Eberhart [10], is modelled on the social behaviour of swarming animal groups such as bees, birds or fish. Communications among individuals in the swarm serve to direct the search effort towards regions in the design space with high feasibility.

Consider a swarm of p particles, with each particle's position $\mathbf{x}_k^i$ representing a possible solution point in the design space $\mathbf{D}$. For each particle i, Kennedy and Eberhart [10] proposed that the position $\mathbf{x}_{k+1}^i$ be updated in the following manner:

$$\mathbf{x}_{k+1}^i = \mathbf{x}_k^i + \mathbf{v}_{k+1}^i \tag{1}$$

with a pseudo-velocity $\mathbf{v}_{k+1}^i$ calculated as follows:

$$\mathbf{v}_{k+1}^i = w_k \mathbf{v}_k^i + c_1 r_1 (\mathbf{p}_k^i - \mathbf{x}_k^i) + c_2 r_2 (\mathbf{p}_k^g - \mathbf{x}_k^i) \tag{2}$$

Here, subscript k indicates a (unit) pseudo-time increment, $\mathbf{p}_k^i$ is the best-found fitness location by particle i at timestep k, which represents the cognitive contribution to the search vector $\mathbf{v}_{k+1}^i$. $\mathbf{p}_k^g$ is the global best-found position among all particles in the swarm at time k and forms the social contribution to the velocity vector. Fitness values assigned to $\mathbf{p}_k^i$ and $\mathbf{p}_k^g$ are denoted f_{best}^i and f_{best}^g, respectively. Random numbers r_1 and r_2 are uniformly generated in the interval [0, 1]. Shi and Eberhart [16] proposed that the cognitive and social scaling parameters c_1 and c_2 be selected

Table I. Particle swarm algorithm parameters.

Parameter	Description	Value
p	Population size (number of particles)	20
c_1	Cognitive trust parameter	2.0
c_2	Social trust parameter	2.0
w_0	Initial inertia	1
w_d	Inertia reduction parameter	0.01
κ	Bound on velocity fraction	0.5
v_d	Velocity reduction parameter	0.01
d	Dynamic inertia/velocity reduction delay (function evaluations)	200

such that $c_1 = c_2 = 2$ to allow the product $c_1 r_1$ or $c_2 r_2$ to have a mean of 1. The result of using these proposed values is that the particles overshoot the target half the time, thereby maintaining separation in the group and allowing for a greater area to be searched. The variable w_k, set to 1 at initialization, is a modification to the original PSO algorithm [16]. By reducing its value dynamically, Fourie and Groenwold [17] suggested that a more refined search can be enforced by observing the fitness improvement rate. At the initialization time step $k=0$ particle velocities $\mathbf{v}_0^i$ are initialized to random values within the limits $0 \leqslant \mathbf{v}_0^i \leqslant \mathbf{v}_0^{\max}$. The particle velocity upper limit $\mathbf{v}_0^{\max}$ is calculated as a fraction of the distance between the upper and lower bound on variables in the design space $\mathbf{v}_0^{\max} = \kappa(\mathbf{x}_{\mathrm{UB}} - \mathbf{x}_{\mathrm{LB}})$ with $\kappa = 0.5$ as suggested in [17].

The impact of population size on the performance of the PSO algorithm was previously investigated by Carlisle and Dozier [18] and Shi and Eberhart [16], among others. Carlisle and Dozier showed that, for some of the test problems evaluated in their paper, a certain minimum threshold on population size is required for reliable convergence. They also showed that increasing the swarm size excessively beyond this value results in decreased algorithm efficiency. We evaluate the efficiency of the multi-run strategy by comparing this approach with the two alternatives of increased population size or allowing an increased number of algorithm iterations. To this end we solved each test problem with different population sizes for a very high number of fitness evaluations. No effort was made to fine tune the algorithm to a particular problem in this study and standard parameters were utilized as reported in Table I.

2.2. *Analytical test set*

The convergence behaviour of the PSO algorithm was analysed with the Griewank [19], Shekel [20] and Hartman [20] analytical problems (see Appendix A for problem definitions), each of which possess multiple local minima. Analytical test problems were used because global solutions f^* are known *a priori*. The known solution value allows us to ascertain if an optimization has converged to the global minimum. To estimate the probability of converging to the global optimum, we performed 1000 optimization runs for each problem, with each run limited to 500 000 fitness evaluations. These optimization runs are performed with identical parameter settings, with the exception of a different random number seed for each run to start the population at different initial points in the design space. To evaluate the global convergence probability of the PSO algorithm as a function of population size, we solved each problem using a swarm of 10, 20, 50 and 100 particles. We assumed that convergence to the global optimum was achieved when the fitness value

Table II. Problem convergence tolerances.

Problem	Convergence tolerance ε_t
Griewank	0.1
Shekel	0.001
Hartman	0.001

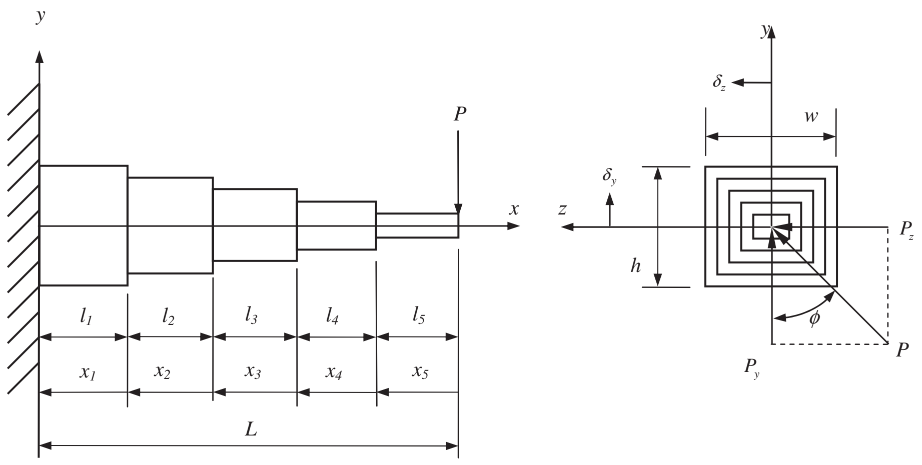

Figure 2. Stepped cantilever beam.

was within a predetermined fixed tolerance ε_t (see Table II) of the known global optimum f^*

$$f \leqslant f^* + \varepsilon_t \tag{3}$$

For the Shekel and Hartman problems, the tolerance ensures that the minimum corresponding to the global optimum for the problem has been found. That is, because $\varepsilon \neq 0$ the exact optimum is not obtained, but if a local optimizer is started from the PSO solution found with the given tolerance it will converge to the global optimum. For the Griewank problem, however, starting a local optimizer at the PSO solution will not guarantee convergence to the global optimum, since this noisy, shallow convex problem has several local minima grouped around the global optimum that will defeat a local optimizer.

2.3. Structural hollow beam optimization

As an example of a practical engineering problem we present the maximization of tip displacement of a stepped hollow beam (Figure 2). This problem proves difficult to solve even when using global optimization algorithms due to the presence of multiple local minima.

The stepped cantilever beam consists of five sections, each section defined by four design variables, width (w), height (h), top and bottom wall thickness (t_h) and left and right wall thickness (t_w) (see Figure 3). A load P with y-direction component P_y and z-direction component

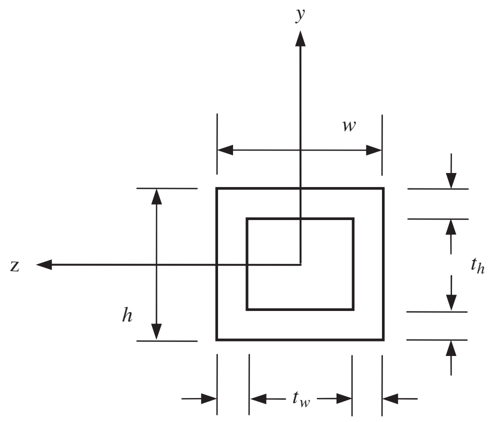

Figure 3. Beam cross-section parameters.

P_z is applied to the tip of the beam. The tip displacement of this prismatic cantilever beam is maximized subject to a material stress constraint in addition to aspect ratio constraints, detailed below. It is the aspect ratio limit (it must be between 0.2 and 5) that creates the local optima. Each section is most flexible when the aspect ratio is at its limit but it is possible for the height to be five times the width or *vice versa*. The feasible region bound by aspect ratio constraints for section 5 of the beam is illustrated in Figure 4, with the two possible solutions indicated. The global optimum is achieved when all five sections have the same aspect ratio, and there are several local optima corresponding to different combinations of extreme height to width ratios for the five sections that are not aligned.

Equations describing tip displacement contributions δ_{yi} and δ_{zi} for each beam section i can be obtained using Castigliano's method

$$\begin{aligned}\delta_y &= \int_0^L \frac{M}{EI}\frac{\partial M}{\partial P}\,\mathrm{d}x \\ &= \int_0^{l_5} \frac{P_y x_5^2}{EI_{z5}}\,\mathrm{d}x + \int_0^{l_4} \frac{P_y(x_4+l_5)^2}{EI_{z4}}\,\mathrm{d}x + \int_0^{l_3} \frac{P_y(x_3+l_5+l_4)^2}{EI_{z3}}\,\mathrm{d}x \\ &\quad + \int_0^{l_2} \frac{P_y(x_2+l_5+l_4+l_3)^2}{EI_{z2}}\,\mathrm{d}x + \int_0^{l_1} \frac{P_y(x_1+l_5+l_4+l_3+l_2)^2}{EI_{z1}}\,\mathrm{d}x \qquad (4)\end{aligned}$$

where E is the Young's modulus and I_{z1} through I_{z5} are the moment of inertia's about the neutral z-axis of the particular beam section under consideration.

If we choose $l_1=l_2=l_3=l_4=l_5=1$, Equation (4) becomes

$$\begin{aligned}\delta_y &= \frac{P_y}{3EI_{z5}} + \frac{7P_y}{3EI_{z4}} + \frac{19P_y}{3EI_{z3}} + \frac{37P_y}{3EI_{z2}} + \frac{61P_y}{3EI_{z1}} \\ &= \delta_{y5} + \delta_{y4} + \delta_{y3} + \delta_{y2} + \delta_{y1} \qquad (5)\end{aligned}$$

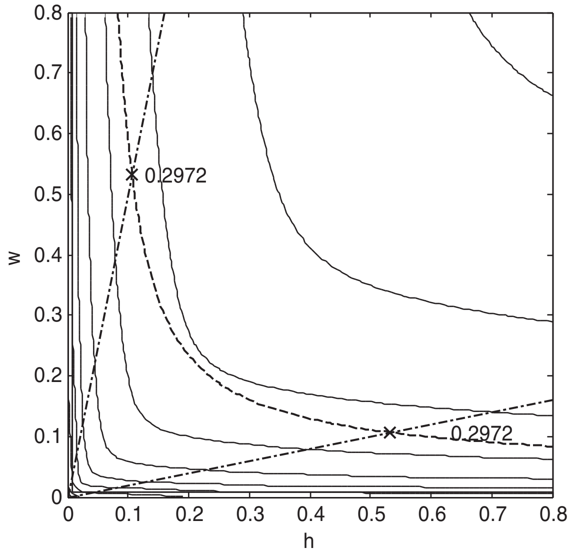

Figure 4. Tip deflection contour plot as a function of beam section 5 with height *h*, and width *w* with yield stress and aspect ratio constraints indicated by dashed and dash dotted lines, respectively. Crosses indicate the two solutions (one dominated by bending in *y*-axis, and the other in the *z*-axis) for maximum tip displacement.

Similarly the displacement in the *z*-direction is

$$\begin{aligned}\delta_z &= \frac{P_z}{3EI_{y5}} + \frac{7P_z}{3EI_{y4}} + \frac{19P_z}{3EI_{y3}} + \frac{37P_z}{3EI_{y2}} + \frac{61P_z}{3EI_{y1}} \\ &= \delta_{z5} + \delta_{z4} + \delta_{z3} + \delta_{z2} + \delta_{z1}\end{aligned} \tag{6}$$

yielding a total tip deflection of

$$\delta = \sqrt{\delta_y^2 + \delta_z^2} \tag{7}$$

For the tip deflection formulation stated (7) there are 2 possible optima yielding the maximum displacement (all sections at aspect ratio 0.2, or all at 5). For the purpose of clearly identifying the global minimum in this work we desire the maximization of the tip deflection to favour only one of the possible solutions, and therefore reformulate the maximization to be in a fixed direction θ

$$\max_{w_1,h_1,t_{w1},t_{h1},w_2,h_2,\ldots,t_{w5},t_{h5}} \quad \delta = \delta_y \cos\theta + \delta_z \sin\theta \tag{8}$$

We choose $\theta = \pi/3$, which biases the tip deflection in the *y* direction. With the bias in the *y* direction, a section with $h_i \gg w_i$ is preferred to a section with $w_i \gg h_i$. However, both are more effective than a section with $w_i = h_i$, which creates two local optima for each section.

The tip deflection maximization is subject to the stress constraints for each beam section *i* (with the allowable stress reduced by safety factor SF). The stresses are calculated at the root of each beam segment, at a maximum distance from the bending axis (i.e. the four corners of each root cross-section) using the Euler–Bernoulli beam equations

$$\begin{aligned} \sigma_i - \frac{\sigma_{\text{allow}}}{\text{SF}} &\leqslant 0 \\ \left(\frac{M_y h}{2I_z}\right)_i + \left(\frac{M_z w}{2I_y}\right)_i - \frac{\sigma_{\text{allow}}}{\text{SF}} &\leqslant 0 \end{aligned} \tag{9}$$

where M_y and M_z are the bending moment introduced at the base of section i by P_y and P_z respectively. In the above equation

$$\begin{aligned} I_{yi} &= 2\left[\frac{t_{hi}(w_i - 2t_{wi})^3}{12} + \frac{h_i t_{wi}^3}{12} + t_{wi} h_i \left(\frac{w_i - t_{wi}}{2}\right)^2\right] \\ I_{zi} &= 2\left[\frac{(w_i - 2t_{wi}) t_{hi}^3}{12} + \frac{t_{wi} h_i^3}{12} + t_{hi}(w_i - 2t_{wi})\left(\frac{h_i - t_{hi}}{2}\right)^2\right] \end{aligned} \tag{10}$$

In addition to the stress constraints the aspect ratio of each beam segment is limited to

$$0.2 \leqslant \frac{w_i}{h_i} \leqslant 5 \tag{11}$$

The hollow cavity along the axial direction and lower bounds on wall thicknesses are accommodated through the following geometric constraints:

$$\begin{aligned} 0.01 &\leqslant t_{hi} \leqslant \frac{h_i}{4} \\ 0.01 &\leqslant t_{wi} \leqslant \frac{w_i}{4} \end{aligned} \tag{12}$$

The above constraints are then normalized

$$\begin{aligned} g_1 &= \left[\sigma_i \Big/ \left(\frac{\sigma_{\text{allow}}}{\text{SF}}\right)\right] - 1 \leqslant 0 \\ g_2 &= \frac{4t_{hi}}{h_i} - 1 \leqslant 0 \\ g_3 &= \frac{4t_{wi}}{w_i} - 1 \leqslant 0 \\ g_4 &= 1 - \frac{t_{hi}}{0.01} \leqslant 0 \\ g_5 &= 1 - \frac{t_{wi}}{0.01} \leqslant 0 \\ g_6 &= \left[\left(\frac{w_i}{h_i}\right) \Big/ 5\right] - 1 \leqslant 0 \\ g_7 &= \left[\left(\frac{h_i}{w_i}\right) \Big/ 5\right] - 1 \leqslant 0 \end{aligned} \tag{13}$$

This yields seven constraints per section, for a total of 35 constraints. The beam material properties and applied load is given in Table III.

Table III. Beam material properties and end load configuration.

Material property	Value
Young's modulus E	72e9 Pa
Safety factor SF	3
Allowable stress σ_{allow}	505e6 Pa
l_1, l_2, l_3, l_4, l_5	1 m
P_y	3535.5 N
P_z	3535.5 N

Due to the PSO's poor local search ability [21] a hybrid approach was used where the final solution of a swarm optimization was used as a starting point of a local search (Sequential Quadratic Programming).

2.4. Multiple-run methodology

The use of multiple optimizations for genetic algorithms was proposed by Le Riche and Haftka [14]. However, no criterion was given on the division of computational resources between the multiple optimizations. The method entails running multiple optimizations with a reduced number of fitness evaluations, either by limiting the number of algorithm iterations or reducing the population while keeping the number of iterations constant. Individually, the convergence probability of each optimization run may only be a fraction of a single traditional optimization run. On the other hand, the cumulative convergence probability obtained by combining the limited runs can be significantly higher than that of the single traditional run. Previously, similar studies have been undertaken to investigate the efficiency of repeated optimizations using simple search algorithms such as pure random search, grid search and random walk [22, 23]. The use of multiple local optimizations or clustering [24] is a common practice but for some algorithms the efficiency of this approach decreases rapidly when problems with a high number of local minima are encountered [22].

For estimating the efficiency of the proposed strategy and for comparison with optimizations with increased populations/allowed iterations, we are required to calculate the probability of convergence to the global optimum for an individual optimization run, P_i. This convergence probability cannot be easily calculated for practical engineering problems with unknown solutions. For the set of analytical problems, however, the solutions are known and a large number of optimizations of these problems can be performed at little computational cost. With some reasonable assumptions these two facts allow us to estimate the probability of convergence to the global optimum for individual optimization runs. The efficiency and exploration run considerations derived from the theoretical analytical results are equally applicable to practical engineering problems where solutions are not known *a priori*. The first step in calculating P_i is using the convergence ratio, C_r, which is calculated as follows:

$$C_{\text{r}} = \frac{N_{\text{c}}}{N} \tag{14}$$

where N_{c} is the number of globally converged optimizations and N is the number of optimizations, in this case 1000. For a very large number of optimizations, the probability P_i that any individual run converges to the global optimum approaches C_{r}. For a finite number of runs, however, the standard error s_e in P_i can be quantified using

$$s_e = \sqrt{\frac{P_i(1-P_i)}{N}} \tag{15}$$

which is an estimate of the standard deviation of C_r. For example, if we obtain a convergence probability of $P_i = 0.5$ with $N = 1000$ optimizations, the standard error would be $s_e = 0.016$.

To obtain the combined cumulative probability of finding the global optimum by multiple independent optimizations we apply the statistical law for calculating the probability of success with repeated independent events. We denote the combined or cumulative probability of N multiple independent optimization runs converging to the solution as P_c. Then using the fact that the convergence events are uncorrelated, P_c can be found from

$$P_c = 1 - \prod_1^N (1-P_i) \tag{16}$$

where P_i is the probability of the ith individual optimization run converging to the global optimum. If we assume that individual optimization runs with similar parameter settings, as in the case of the following study, have equal probability of convergence, we can simplify Equation (16) to

$$P_c = 1 - (1-P_i)^N \tag{17}$$

The increase in cumulative probability P_c for fixed values of P_i as a function of increasing number of optimization runs N is illustrated in Figure 5. It must be stressed that the above relations are only valid for uncorrelated optimizations, which may not be the case when a poor quality random number generator is used to generate initial positions in the design space. Certain generators can exhibit a tendency to favour regions of design space, biasing the search and probability of convergence to a minimum in these regions.

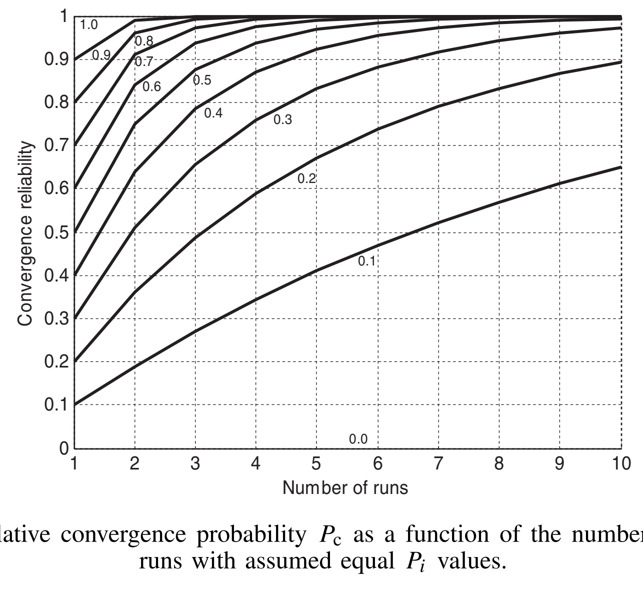

Figure 5. Cumulative convergence probability P_c as a function of the number of optimization runs with assumed equal P_i values.

2.5. Exploratory run and budgeting scheme

Using the multiple run methodology requires a budget strategy by which to divide a fixed number of fitness evaluations among the independent optimization runs. The budget of fitness evaluations n_{b} is usually dictated by how much time the user is willing to allocate on a machine to solve a problem divided by how long a single fitness evaluation takes to execute. The proposed budget strategy is based on two assumptions. The first is that each run should be limited to a number of function evaluations corresponding to the stage in the algorithms when progress is relatively rapid. The second assumption is that for a given problem and algorithmic settings the number of function evaluations in this stage does not vary drastically from one run to the next, in other words some runs stall far from the global optimum while others terminate near it. An exploratory optimization utilizing a fraction of n_{b}, denoted n_{e} is required to determine the length of the rapid-progress stage. The fitness history of this optimization is used to obtain an estimate of the number of fitness evaluations to be allocated to each run n_i. For the problems used in this study, Figure 5 shows that the second assumption holds. The figure also shows a strong correlation between the point where the fitness history levels off (end of rapid-progress stage) and the point where the convergence history levels off (Figure 6). Consequently, it appears reasonable to assume that the algorithm will either converge quickly to the optimum or stall at a similar number of fitness evaluations for optimizations started at different initial positions (Figure 7), which motivates the first assumption above.

This exploratory run is stopped using a criterion that monitors the rate of change of the objective fitness value as a function of the number of fitness evaluations. As soon as this rate of improvement drops below a predetermined value (i.e. the fitness value plot levels off), the exploratory optimization is stopped and the number of fitness evaluations noted as n_{e}. The selected stopping criterion was a change of less than 0.01 in fitness value for at least 500 function evaluations. With n_{b} and n_{e} known, it is easy to calculate the number of independent optimizations to be performed, N, as well as the number of fitness evaluations per run n_i. After the exploratory optimization of n_{e} the

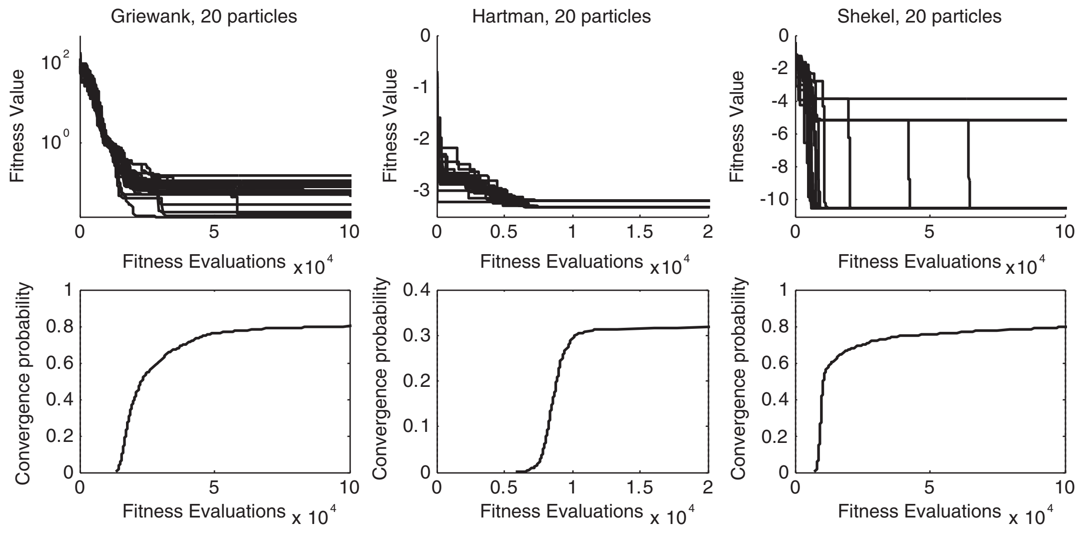

Figure 6. Fitness history and convergence probability P_{c} plots for Griewank, Hartman and Shekel problems.

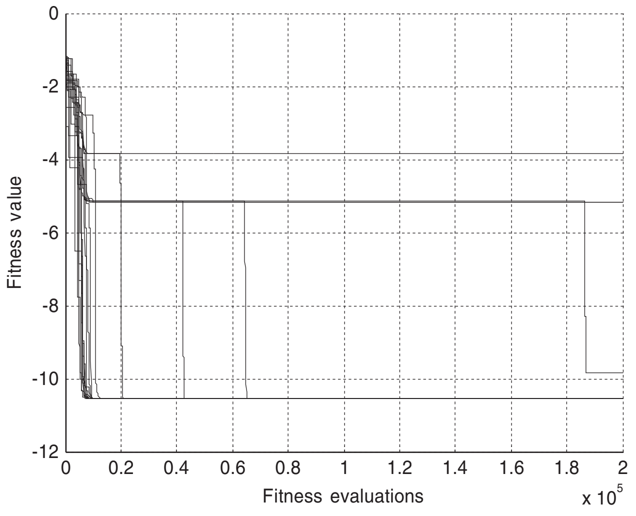

Figure 7. Typical Shekel fitness history plots of 20 optimizations (sampled out of 1000).

remainder of the budgeted fitness evaluations is distributed among a number of N independent optimizations, which may be calculated as follows:

$$N = \left\lfloor \frac{n_\mathrm{b} - n_\mathrm{e}}{n_\mathrm{e}} \right\rfloor \tag{18}$$

with an allowed number of fitness evaluations per run calculated by

$$n_i = \left\lfloor \frac{n_\mathrm{b} - n_\mathrm{e}}{N} \right\rfloor \tag{19}$$

If a multi-processor machine is available, very high P_c values may be reached using a multiple run strategy. If we take the simple case where each independent optimization run is assigned to a separate node, the multiple run approach will be constrained somewhat differently than the previous single-processor case. Rather than the number of multiple optimizations being limited by a fixed budget of fitness evaluations (which is divided equally among the set of multiple independent optimizations using Equation (18)), the number of optimization runs will be defined by the number of computational nodes and the wall clock time available to the user. A similar method to that followed for a single processor machine for determining algorithm/problem behaviour must still be followed to determine the optimal number of fitness evaluations for a single optimization run. This exploratory run can, however, be done using a parallel implementation of the population-based algorithm under consideration, in which concurrent processing is achieved through functional decomposition [15].

2.6. Bayesian convergence probability estimation

To estimate the probability that the global optimum has been reached at the end of the process, a Bayesian convergence probability estimation method may be used, as proposed by Groenwold and Snyman [25] and Groenwold and Hindley [26]. The probability that the best solution found among all optimizations $\tilde{f}$ will be the global solution f^* is given in [26] as

$$\Pr[\tilde{f} = f^*] \geqslant 1 - \frac{(N+\overline{a})!(2N+\overline{b})!}{(2N+\overline{a})!(N+\overline{b})!} \tag{20}$$

where N is the total number of optimizations performed, and $\overline{a} = a + b - 1$, $\overline{b} = b - N_c - 1$ with a, b suitable parameters of a Beta distribution $\beta(a, b)$. Values of parameter a and b were chosen as 1 and 5, respectively, as recommended by Groenwold and Hindley [26].

3. NUMERICAL RESULTS

3.1. Multi-run approach for predetermined number of optimizations

For the three problems under consideration only a limited improvement in global convergence probability is achieved by applying the traditional approaches of increasing the number of fitness evaluations or the population size (Figure 8). For the Shekel problem, using larger swarm sizes and/or allowing an increased number of fitness evaluations yielded higher convergence probabilities only up to a point. Similar results were obtained for the Griewank and Hartman problem cases. On the other hand, optimizations with a small number of particles reached moderate global convergence probabilities at significantly fewer fitness evaluations than did optimizations with large swarms. This behaviour was observed for all the problems in the test set (Figure 6). To exploit this behaviour we replace a single optimization with several PSO runs, each with a limited population and number of iterations. These individual optimizations utilize the same amount of resources allocated to the original single optimization (in this case the number of fitness evaluations).

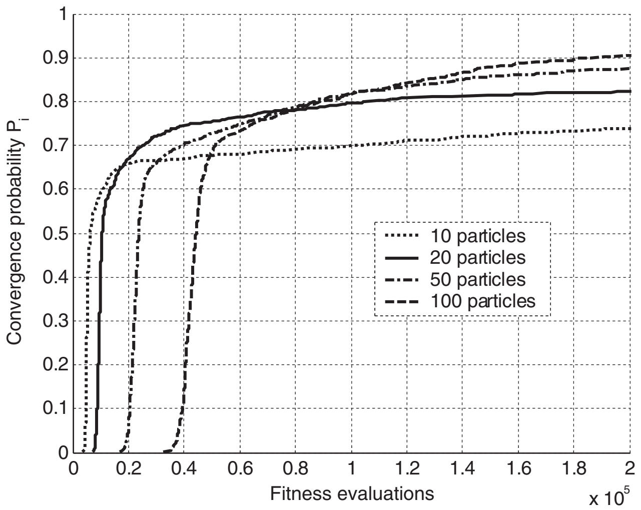

Figure 8. Shekel convergence probability for an individual optimization as a function of fitness evaluations and population size.

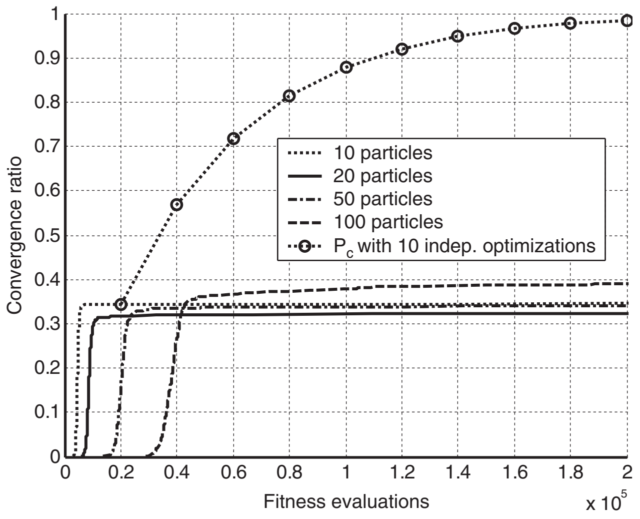

Figure 9. Theoretical cumulative convergence probability P_c as a function of the number of optimization runs with constant P_i for the Hartman problem. Multiple independent runs are with 10 particles.

Table IV. Theoretical convergence probability results for Hartman problem.

Number of runs n	Cumulative convergence probability P_c	Cumulative fitness evaluations
1	0.344	10 000
2	0.570	20 000
3	0.718	30 000
4	0.815	40 000
5	0.879	50 000
6	0.920	60 000
7	0.948	70 000
8	0.966	80 000
9	0.978	90 000
10	0.985	100 000

To illustrate the merit of such an approach we optimize the Hartman analytical problem with and without multiple limited optimizations. We observe that for a single optimization the probability of convergence is not significantly improved by allowing more fitness evaluations, or by increasing the population size (Figure 9). We also observe that an optimization with 10 particles quickly attains a probability of convergence of $P_i = 0.344$ after only 10 000 fitness evaluations. Using a multiple run strategy with 10 independent optimizations of 10 000 fitness evaluations yields the theoretical P_c values reported in Table IV (calculated using Equation (17) with $P_i = 0.344$ and $n = 1, \ldots, 10$). These values are indicated as circled data points at a sum equivalent number of fitness evaluations in Figure 9 for comparison with a single extended optimization. The cumulative convergence probability P_c using multiple optimizations is far superior to that of using a single optimization run of up to 100 particles.

3.2. Multi-run efficiency

To investigate the most efficient manner in which to divide a budget of fitness evaluations among multiple independent optimizations, we compare the efficiency of the 2, 5, 10 and 12 independent optimizations of the Griewank problem (Figure 10). A budget of 200 000 fitness evaluations is allowed for optimizing this problem. This results in each optimization in the set being stopped at $n_i = 100\,000$, 40 000, 20 000 and 16 500 fitness evaluations. It can be seen that using a combination of independent runs with high P_i values (with a high number of the associated of fitness evaluations) or multiple runs with low P_i values will not yield the same efficiency (as defined by P_c per fitness evaluation). If the predicted P_c values are plotted on the same graph (Figure 10) it is observed that the two combinations of 5 and 10 optimizations yield the highest P_c values for a given number of fitness evaluations. In both cases the independent runs are stopped at a number of fitness evaluations close to the point where P_i levels off.

The dependence of efficiency on the choice of n_e can be explained as follows: if the independent optimization is stopped prematurely it will result in very low values of P_i. Although a greater number of independent optimizations may then be performed within the single processor budget (Equation (18)) it may still result in very poor efficiency, such as the 12 independent run case in Figure 10. If on the other hand an excessive amount of fitness evaluations are allowed for each

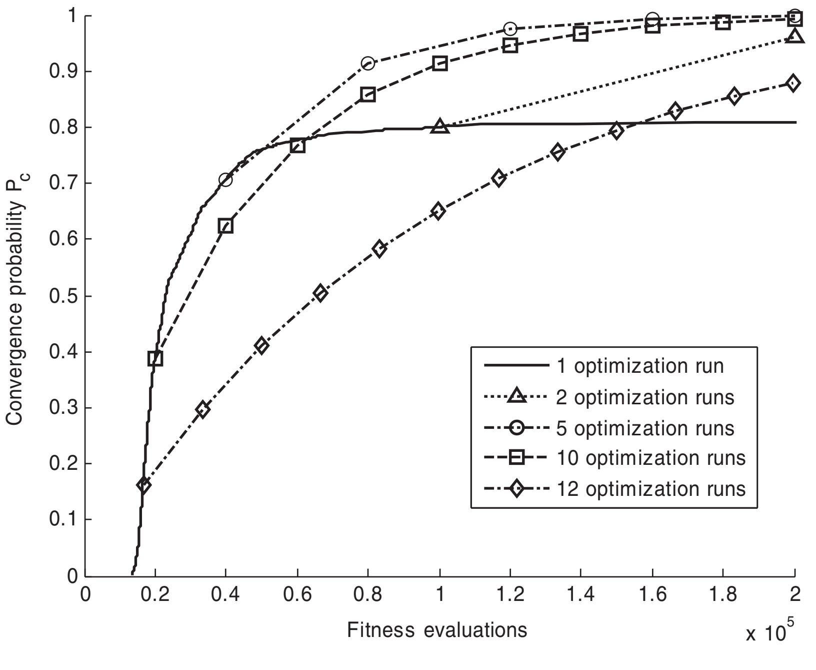

Figure 10. Predicted convergence probability P_c with sets of multiple runs for the Griewank problem.

Table V. Minimum, maximum and median fitness evaluations when applying ratio of change stopping criteria on pool of 1000 optimizations for Griewank, Hartman and Shekel problems.

Problem	Minimum n_{e}	Median n_{e}	Maximum n_{e}
Griewank	23 480	31 810	45 780
Hartman	14 180	16 440	23 360
Shekel	15 460	18 660	35 860

independent run the strategy also suffer since only a reduced number of optimizations can then be performed (see the 2 independent run case in Figure 10). To maximize the efficiency of the multi-run strategy it is therefore desirable to allow only a number of fitness evaluations corresponding to the point where P_i starts levelling off.

From the above we can conclude that to obtain maximum efficiency, the individual runs should be terminated after reaching a number of fitness evaluations near the point where P_i starts levelling off. The exact P_i convergence probability data, however, is not available unless a great number of optimizations are performed, and must be estimated. This estimation is done by observing where the fitness history of a single optimization starts levelling off. The impact and robustness of using a single exploratory optimization fitness history to determine n_i for each optimization was investigated for all three analytical problems. The exploratory optimization was stopped using a rate of change stopping criteria, and N and n_i was calculated using (18) and (19). To verify that a single optimization will be sufficient to give robust results the rate of change stopping criteria was applied to the pool of 1000 optimizations for each of the three analytical test problems, and yielded minimum, maximum and median values of n_{e} reported in Table V. The fitness evaluation history plots for minimum and maximum n_{e} values and corresponding convergence probability plots are given in Figure 11. It can be seen that applying the rate of change stopping criteria on a single run to estimate n_{e} gives fairly robust results, with only a small variation in P_{c} efficiency for all analytical problems.

3.3. Stepped cantilever beam

The optimization of the stepped cantilever beam was first performed with the Matlab fmincon function, an implementation of the sequential quadratic programming algorithm. A total of 100 optimizations were performed, each started in a random location in the 20-dimensional search space. The resultant tip deflection solutions were sorted in an ascending fashion and plotted in Figure 12. Only one out of the 100 optimizations reached the optimal tip displacement value of 0.7545, with a median tip displacement result of 0.2707. Several plateaus can be seen in this figure, each indicating several optimizations which converged to the same local maximum. The PSO algorithm was then applied to this problem, with only 100 optimizations performed due to the computational cost associated with this global optimization algorithm. A total budget of 4 000 000 fitness evaluations, which translated to 40 000 fitness evaluations per optimization were allowed. Tip deflection solutions are again sorted in an ascending fashion and given in Figure 13. This yielded an improved median tip displacement of 0.6134. In an attempt to improve over the results obtained using only PSO or fmincon, a hybrid of the two is applied to the problem.

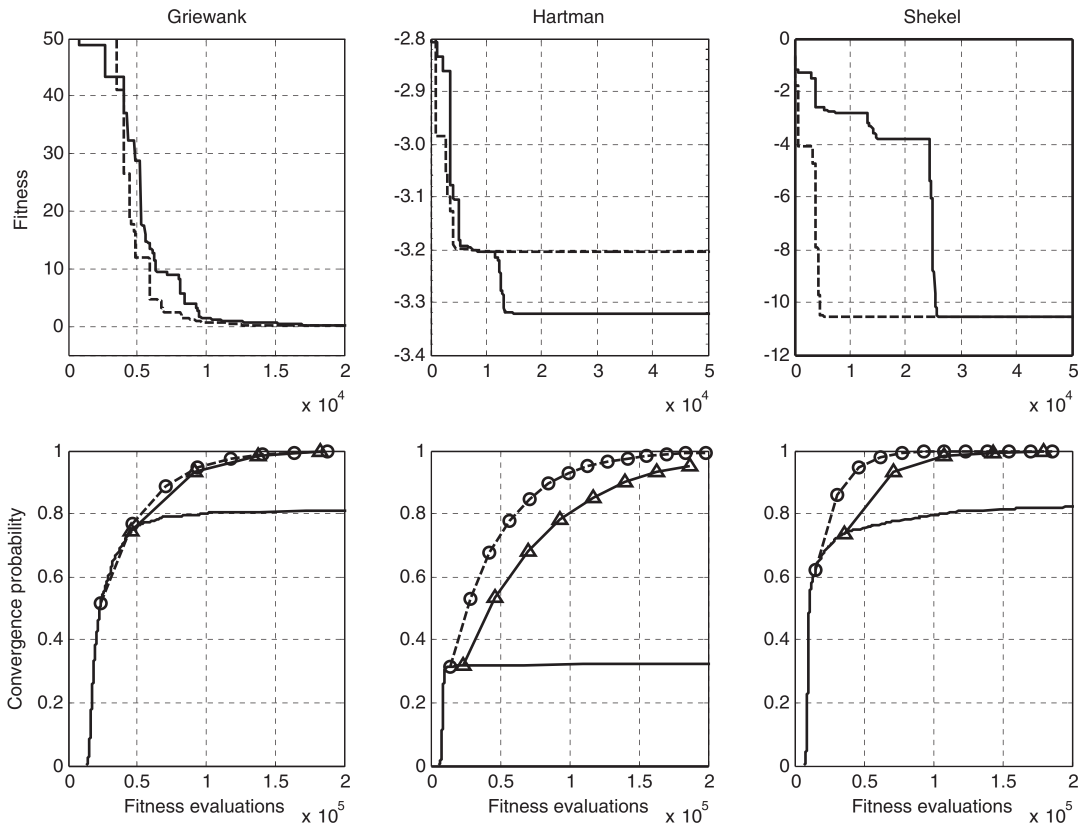

Figure 11. Predicted convergence probability P_c using information from exploratory optimizations which are stopped using a rate of change stopping condition for the Griewank, Hartman and Shekel problems. A solid line with circled datapoints denotes using the longest exploratory run in the pool of 1000 optimizations, and a dashed line with triangular datapoints denote using the shortest. The solid line indicate convergence probability calculated using Equation (14) for the 1000 optimizations as a function of the number of fitness evaluations.

This approach used the PSO as an initial search method to roughly pinpoint the approximate location of the global optimum, and then transitions to the fmincon search function to perform a localized search. Again, due to the required computational effort to solve the problem only 100 optimizations are performed, with 40 000 evaluations per optimization. A marked improvement in the solution quality is observed when using the hybrid approach, as compared to fmincon (Figure 12) or PSO only (Figure 13) with the median tip displacement improved to 0.6442. Three optimizations out of the 100 obtained the optimum displacement of 0.7545 (upper plateau in Figure 13) (Table VI).

To investigate if the allowed budget of fitness evaluations can be more efficiently applied the multiple-run methodology is applied to the stepped cantilever beam problem (Figure 14). In order to investigate the sensitivity of using a single exploratory run, the previous results for 100 hybrid

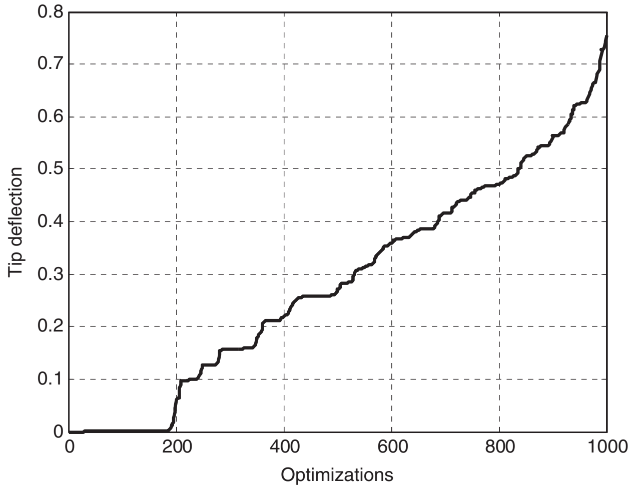

Figure 12. Results for 1000 fmincon optimizations. Tip deflection values sorted ascending and plotted as a function of optimizations. Flat areas in the graph represent optimizations which converged to the same local minimum.

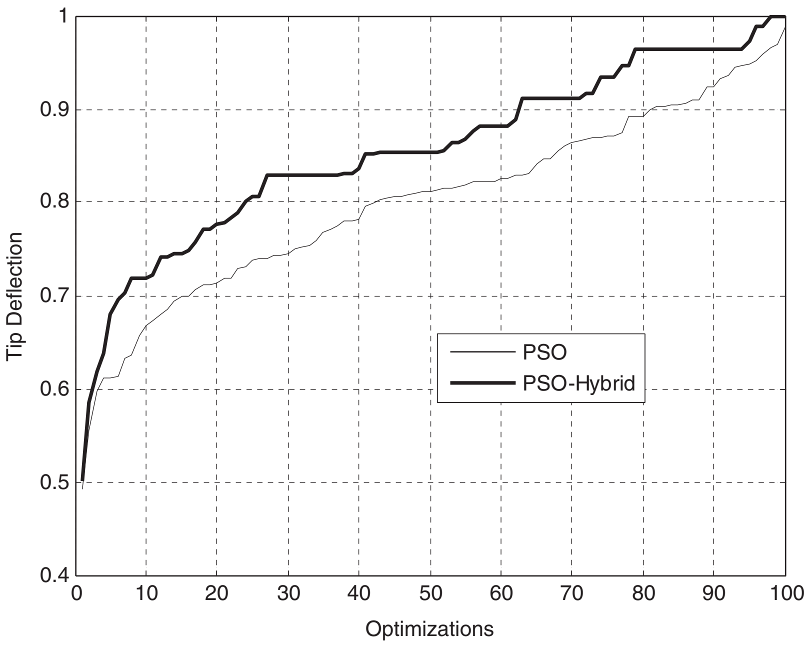

Figure 13. Tip deflection results for 100 PSO and PSO–fmincon hybrid optimizations. Ordinate normalized to global optimal tip displacement.

Table VI. Median solution out of 1000 fmincon optimizations using random starting points.

	Optimal tip deflection contributions	Median solution tip deflection contributions
Section 1	0.2579	0.0001
Section 2	0.2107	0.2107
Section 3	0.1588	0.0000
Section 4	0.1004	0.0599
Section 5	0.0267	0.0000
Total tip deflection	0.7545	0.2707

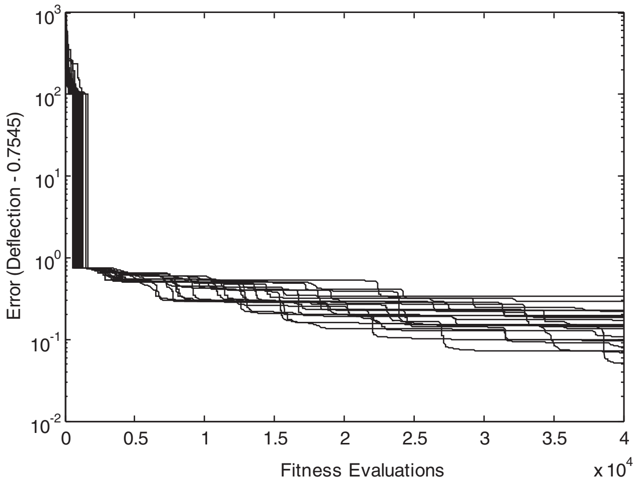

Figure 14. Stepped cantilever beam: sample of 20 fitness histories.

runs were processed by applying the rate of change criteria of 0.001 in the displacement over 2000 iterations. Among the 100 hybrid optimizations a minimum n_e value of 6743 fitness evaluations, a median of 13 685 and a maximum 27 662 were obtained. Applying Equations (18) and (19) to these results with a total budget of $n_b = 4000\,000$ fitness evaluations required $N = 592$, 291 and 143 optimizations for the budget strategy, each allocated $n_i = 6745$, 13 698 and 27 778 fitness evaluations, respectively. Performing these optimization runs resulted in 3, 8 and 14 optimization converging (out of 592, 291 and 143), with corresponding convergence ratios (using Equation (14)) of 0.51, 2.75 and 9.8%. These rapidly rising percentages level off, though, because 10 runs with 400 000 evaluations each did not yield a single optimal result, which suggests that the convergence rates levels off near 10%. With a reduced budget of 400 000 function evaluations, we can perform 59, 29, and 14 of the shorter-run optimization, respectively. Using Equations (16), 59 runs with 6745 function evaluations should give us 26% probability of finding the optimum, 29 runs with 13 698

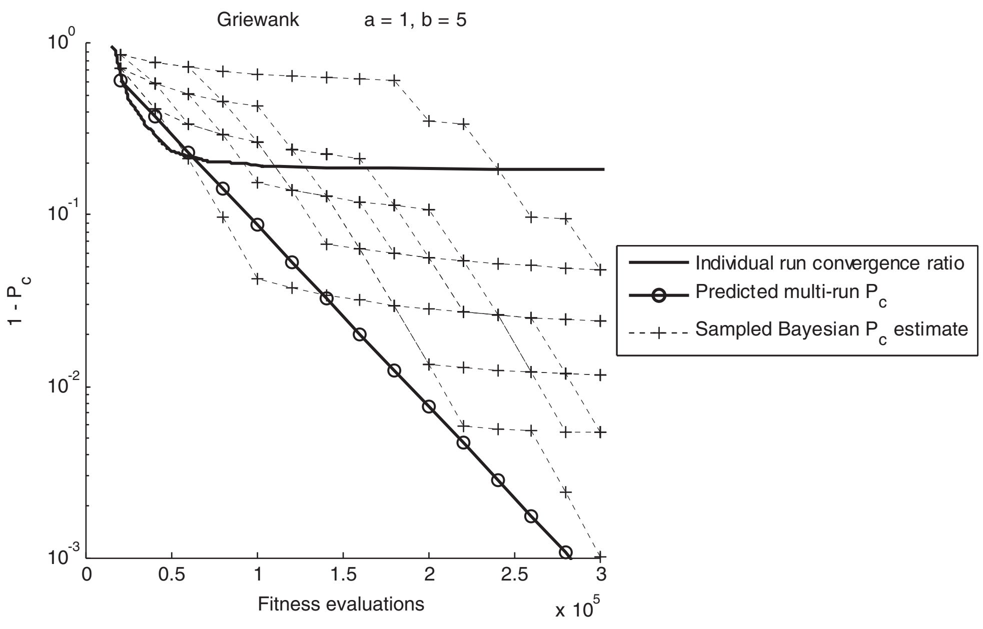

Figure 15. Bayesian P_c estimation in comparison to using extrapolated and randomly sampled optimizations out of pool of 1000 runs for Griewank problem.

function evaluations will give us 55% probability, and 14 runs with 27 778 function evaluations should give us 76% probability.

3.4. Bayesian convergence probability estimation

The Bayesian scheme (Equation (20)) indicates s consistently conservative estimation of the cumulative convergence probability for the Griewank (Figure 15), Hartman (Figure 16), and Shekel (Figure 17) problems. In order to show the accuracy of this method as the P_c values approach 1 the probability of failure $(1 - P_c)$ is indicated on a logarithmic ordinate axis. The Bayesian estimation method is sensitive to the problem and type of optimization algorithm under consideration, and values of a and b parameters in the Beta distribution require fine tuning to obtain a more accurate estimation. The parameters recommended by Groenwold and Hindley [26], however, yield a consistently conservative estimation of the confidence level. A possible exception to this may occur when several local minima are present which will satisfy the condition given in Equation (3) if ε_t is not chosen to be small enough to exclude these, as is the case for the Griewank problem. Any of these local optimum will then be considered a global optimum in the convergence probability estimation. The results suggest the Bayesian prediction of P_c may be useful when confidence in the solution is traded-off with the optimization time, on either a single or parallel processor machine.

For the stepped cantilever beam the Bayesian probability prediction yielded the curve reported in Figure 18. From this figure it can be seen that although the global solution value was found (indicated by the normalized dark curve) during the 27th optimization the convergence probability estimation only jumped to unity probability after 63 optimizations.

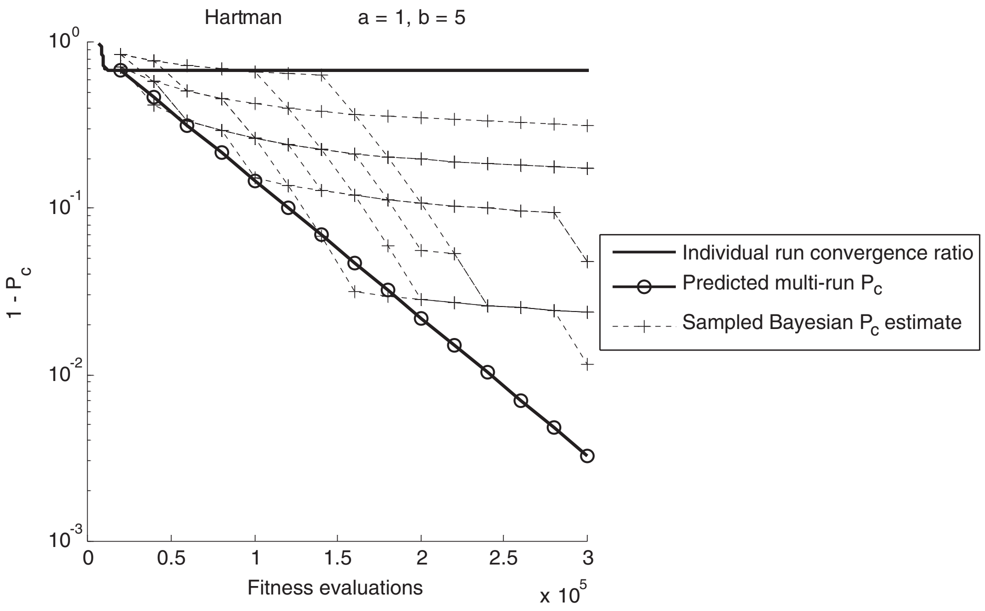

Figure 16. Bayesian P_c estimation in comparison to using extrapolated and randomly sampled optimizations out of pool of 1000 runs for Hartman problem.

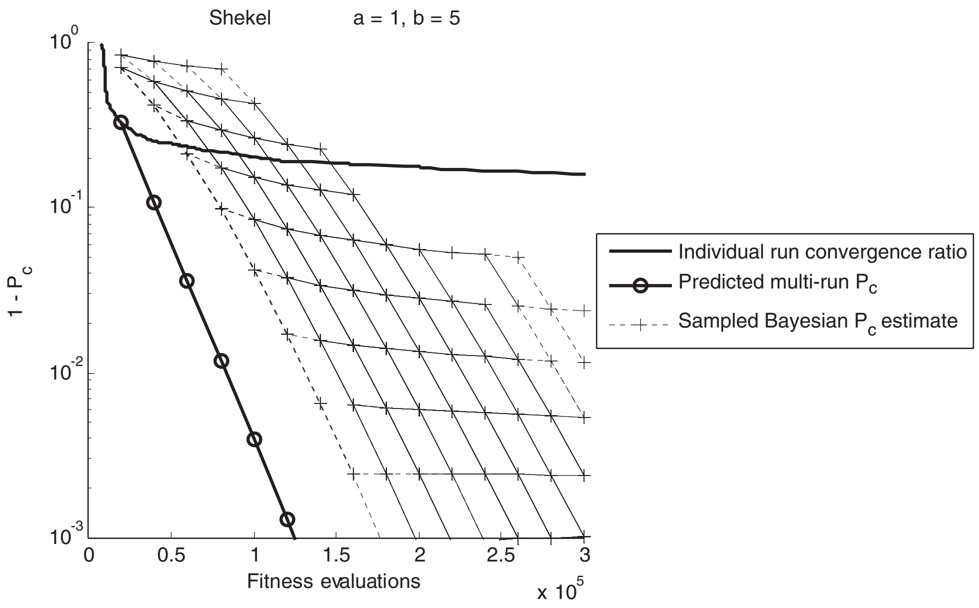

Figure 17. Bayesian P_c estimation in comparison to using extrapolated and randomly sampled optimizations out of pool of 1000 runs for Shekel problem.

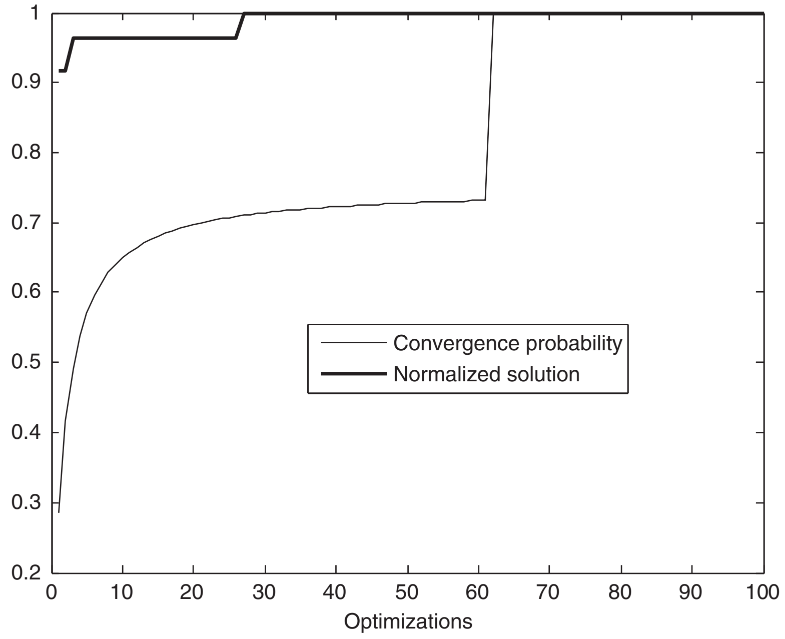

Figure 18. PSO-hybrid-stepped beam convergence probability estimation using Bayesian criteria.

4. CONCLUSIONS

The basic premise of this paper is that for a given problem and global optimizer settings, different runs may have substantially different outcomes, but they all start with initial rapid progress followed by much slower progress later on. Furthermore, we assume that the number of function evaluations to the end of the rapid progress stage does not change drastically from one run to another for a given problem and algorithmic setting. Therefore we propose that the number of function evaluations required for this rapid-progress stage be estimated with one or two runs, and then the same number of function evaluations be allocated to multiple searches. Three often used analytical problems were used to demonstrate the utility of this multi-run strategy for substantially increasing the probability of convergence to the global optimum for PSO. The multi-run strategy was then used for an engineering design problem with 32 local optima, which challenges the PSO. For the set of large scale optimization problems evaluated with the PSO, the multi-run strategy with small PSO populations delivers higher global convergence probability than a single run with a large population and an equal number of fitness evaluations. On both serial and parallel machines, a fraction of the allocated budget of fitness evaluations or computer time is required to evaluate the optimizer/problem behaviour. This exploratory optimization is terminated using a rate of change stopping criteria. The number of fitness evaluations required by exploratory run is used to calculate the total number of runs and the remainder of the budget of evaluations is divided among them. This approach allows the strategy to utilize the computational resources efficiently, preventing premature termination or wasted fitness evaluations on already converged optimizations. Close correlation between theoretically predicted cumulative convergence probability and the experimentally sampled probability is obtained for the strategy on a single processor machine.

When applied to the challenging engineering problem of maximizing the tip deflection of a stepped cantilever beam, the PSO and a PSO–fmincon hybrid only had limited success using an equally divided budget of 4 000 000 allowed fitness evaluations. Using the proposed budgeting strategy, however, yielded an equivalent average performance, with up to three times the convergence probability in cases where the rate of change stopping criteria estimated conservatively. It is recommended that more than one exploratory optimization should be performed when using this strategy, and the more conservative fitness evaluations per run value be used.

A Bayesian convergence probability estimation method may be used to stop a serial or parallel optimization when optimization reliability is traded-off with time for optimization. This Bayesian prediction of the cumulative convergence probability is consistently conservative for all the problems tested when using parameters recommended in the literature.

Very high global convergence probabilities can be achieved in a limited time span on a massively parallel machine using the multi-run strategy. Although it may sometimes be impossible to find the global solution to an intractable large-scale engineering problem this strategy will allow the user to obtain high quality approximate solutions.

APPENDIX A

A.1. Griewank [7]

Objective function

$$f(\mathbf{x}) = \sum_{i=1}^{n} \frac{x_i^2}{d} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{A1}$$

with $n = 10$ and $d = 4000$.

Search domain $D = \{(x_1, x_2, \ldots x_{10}) \in R^{10} : -600 \leqslant x_i \leqslant 600, i = 1, 2, \ldots, 10\}$.

Solution: $\mathbf{x}^* = (0.0, 0.0, \ldots, 0.0)$, $f^* = 0.0$.

A.2. Hartman 6 [8]

Objective function

$$f(\mathbf{x}) = -\sum_{i=1}^{m} c_i \exp\left(-\sum_{j=1}^{n} a_{ij}(x_j - p_{ij})^2\right) \tag{A2}$$

Search domain $D = \{(x_1, \ldots, x_6) \in R^6 : 0 \leqslant x_i \leqslant 1, \ i = 1, \ldots, 6\}$.

Solution (*with* $m = 4$): $\mathbf{x}^* = (0.2017, 0.1500, 0.4769, 0.2753, 0.3117, 0.6573)$, $f^* = -3.322368$.

See Table A1 for values of a_{ij}, c_i, and$p_{i,j}$.

A.3. Shekel 10 [8]

Objective function

$$f(\mathbf{x}) = -\sum_{i=1}^{m} \frac{1}{(\mathbf{x} - a_i)^{\mathrm{T}}(\mathbf{x} - a_i) + c_i} \tag{A3}$$

Table A1. Hartman problem constants.

a_{ij}						c_i
10.0	3.0	17.0	3.5	1.7	8.0	1.0
0.05	10.0	17.0	0.1	8.0	14.0	1.2
3.0	3.5	1.7	10.0	17.0	8.0	3.0
17.0	8.0	0.05	10.0	0.1	14.0	3.2
p_{ij}						
0.1312	0.1696	0.5569	0.0124	0.8283	0.5886	
0.2329	0.4135	0.8307	0.3736	0.1004	0.9991	
0.2348	0.1451	0.3522	0.2883	0.3047	0.6650	
0.4047	0.8828	0.8732	0.5743	0.1091	0.0381	

Table A2. Shekel problem constants.

i	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

Search domain $D=\{x_i \in R^4 : 0 \leqslant x_i \leqslant 10, i=1,\ldots,4\}$.

Solution (*with* $m=10$): $\mathbf{x}^*=(4.00074671,\ 4.00059326,\ 3.99966290,\ 3.99950981)$, $f^*=-10.536410$.

See Table A2 for values of a_i and c_i.

NOMENCLATURE

P_i	individual optimization run global convergence probability
P_c	combined multiple optimization global convergence probability
C_r	convergence ratio
N	number of optimization runs
N_c	number of globally converged optimization runs
n_fe	number of fitness evaluations
n_b	budget of fitness evaluations allocated to solving problem
n_i	allowed fitness evaluations for each independent optimization
n_e	number of fitness evaluations required in exploratory optimization
s_e	standard error

REFERENCES

1. Törn A, Zilinskas A. *Global Optimization*. Springer: New York, 1989.
2. Horst R, Pardalos PM (eds). *Handbook of Global Optimization*. Kluwer Academic Publishers: Dordrecht, 1995.
3. Pardalos PM, Romeijn E (eds). *Handbook of Global Optimization*: *Heuristic Approaches*, vol. 2. Kluwer Academic Publishers: Dordrecht, 2002.
4. Kemenade CHM. A two-level evolution strategy (balancing global and local search). *Technical Report CS-R9559 1995*, Centrum voor Wiskunde en Informatica, The Netherlands, 1995.
5. Jones DR, Perttunen CD, Stuckman BE. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications* 1993; **79**:157–181.
6. Srinivas M, Patnaik LM. Genetic algorithms: a survey. *Computer* 1994; **27**(6):17–26.
7. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. *Science* 1993; **220**:621–680.
8. Geman S, Hwang CR. Diffusions for global optimization. *SIAM Journal on Control and Optimization* 1986; **24**:1031–1043.
9. Kan AHGR, Timmer GT. Stochastic global optimization methods. Part I: Clustering methods. *Mathematical Programming* 1987; **39**(1):27–56.
10. Kennedy J, Eberhart RC. Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4. Perth, Australia, 1995; 1942–1948.
11. Lourenço HR, Martin O, Stützle T. Iterated local search. In *Handbook of Meta-Heuristics*, Glover F, Kochenberger G (eds), International Series in Operations Research and Management Science, vol. 57. Kluwer Academic Publishers: Norwell, MA, 2002; 321–353.
12. Migdalas A, Toraldo G, Kumar V. Nonlinear optimization and parallel computing. *Parallel Computing* 2003; **29**:375–391.
13. Pardalos PM. Parallel search algorithms in global optimization. *Applied Mathematics and Computation* 1989; **29**:219–229.
14. Le Riche R, Haftka RT. Optimization of laminate stacking sequence for buckling load minimization by genetic algorithm. *AIAA/ASME/AHS/ASCE/ASC 33rd Structures*, *Structural Dynamics and Materials Conference*, Dallas, TX (Also *AIAA Journal* 1993; **31**(5):951–956).
15. Schutte JF, Reinbolt JA, Fregly BJ, Haftka RT. Parallel global optimization with the particle swarm algorithm. *International Journal for Numerical Methods in Engineering* 2004; **61**:2296–2315.
16. Shi Y, Eberhart RC. *Parameter Selection in Particle Swarm Optimization*. Lecture Notes in Computer Science, vol. 1447. Springer: Berlin, 1998; 591–600.
17. Fourie PC, Groenwold AA. The particle swarm optimization algorithm in size and shape optimization. *Journal of Structural and Multidisciplinary Optimization* 2002; **23**(4):259–267.
18. Carlisle A, Dozier G. An off-the-shelf PSO. *Proceedings of the Workshop on Particle Swarm Optimization*, Purdue School of Engineering and Technology, Indianapolis, U.S.A., 2001.
19. Griewank AO. Generalized descent for global optimization. *Journal of Optimization Theory and Applications* 1981; **34**:11–39.
20. Dixon LCW, Szegö GP. *Towards Global Optimization*. North-Holland: Amsterdam, 1975.
21. Schutte JF, Koh BI, Reinbolt JA, Fregly BJ, Haftka RT, George AD. Evaluation of a particle swarm algorithm for biomechanical optimization. *Journal of Biomechanical Engineering* 2005; **127**:465–474.
22. Muselli M. A theoretical approach to restart in global optimization. *Journal of Global Optimization* 1996; **10**:1–16.
23. Muselli M, Rabbia M. Parallel trials versus single search in supervised learning. *Proceedings of the Second International Conference on Artificial Neural Networks*, The Institution of Electrical Engineers, Bournemouth, 1991; 24–28.
24. Törn AA. Probabilistic global optimization, a cluster analysis approach. *Proceedings of the Second European Congress on Operations Research*, Stockholm, 1976; 521–527.
25. Groenwold AA, Snyman JA. Global optimization using dynamic search trajectories. *Journal of Global Optimization* 2002; **24**:51–60.
26. Groenwold AA, Hindley MP. Competing parallel algorithms in structural optimization. *Journal of Structural Multidisciplinary Optimization* 2002; **24**:343–350.