Research papers

Struct Multidisc Optim 23, 259–267
Digital Object Identifier (DOI) 10.1007/s00158-002-0188-0

The particle swarm optimization algorithm in size and shape optimization

P.C. Fourie and A.A. Groenwold

Abstract Shape and size optimization problems instructural design are addressed using the particle swarm optimization algorithm (PSOA). In our implementation of the PSOA, the social behaviour of birds is mimicked. Individual birds exchange information about their position, velocity and fitness, and the behaviour of the flock is then influenced to increase the probability of migration to regions of high fitness. New operators in the PSOA, namely the elite velocity and the elite particle, are introduced.

Standard size and shape design problems selected from literature are used to evaluate the performance of the PSOA. The performance of the PSOA is compared with that of three gradient based methods, as well as the genetic algorithm (GA). In attaining the approximate region of the optimum, our implementation suggests that the PSOA is superior to the GA, and comparable to gradient based algorithms.

Key words particle swarm optimization, size optimization, shape optimization

1 Introduction

In structural optimization, a number of efficient optimization algorithms mimicking natural phenomena and physical processes have been applied. Amongst others, notable formulations are the genetic algorithm (GA), simulated biological growth (SBG), simulated annealing (SA) and the particle swarm optimization (PSO).

SBG (Mattheck and Burkhardt 1990) mimics phenomena that have been observed in the mechanism of tree growth. This involves the self-optimization of living trees which always try to grow into shapes of constant surface stress. Another example is the phenomenon that has been observed in animal and bone tissue (Huiskes *et al.* 1987). This involves the addition of bone material in regions of high stress and conversely, the reduction of material in regions of low stress. Simulated annealing (Metropolis *et al.* 1953) is based on statistical thermodynamics and is used to simulate the behaviour of the atomic arrangements in solid material during an annealing process.

In all probability, the best known of the methods mentioned above is the GA (Goldberg 1989; De Jong 1975; Beasley *et al.* 1993a,b), which mimics natural selection and survival of the fittest. A population of genes is ranked based on the fitness of the individual genes, whereafter the genes with the best fitness are selected according to a given selection criterion to reproduce. Reproduction is affected by the cross-over operator. Genetic diversity is introduced in the population by means of mutation. Elitism is often employed where the gene with the best fitness is copied to the next generation.

As opposed to the well-established methods mentioned above, PSO is still in its infancy. This method was proposed by Kennedy and Eberhart (1995) and is based on the simulation of a simplified social model. Some aspects that intrigued scientists were the underlying rules that enabled large numbers of birds to flock synchronously, often changing direction suddenly, scattering and regrouping, etc. Since these initial observations, *bird flocking* and *fish schooling* were some of the behavioural patterns which were sought to be mimicked. It was noted that the social sharing of information among members offers an evolutionary advantage. This observation is fundamental to the development of the PSO algorithm. Birds and fish adjust their physical movement to avoid predators, seek food and mates.

Received December 18, 2000

P.C. Fourie[1] and A.A. Groenwold[2]

[1] Department of Mechanical Engineering, Technikon Pretoria, Private Bag X680, 0001, Republic of South Africa,
e-mail: `fouriec@techpta.ac.za`
[2] Department of Mechanical Engineering, University of Pretoria, 0002, South Africa
e-mail: `Albert.Groenwold@eng.up.ac.za`

Previously, the PSO algorithm has been applied to analytical test functions, mostly univariate or bivariate without constraints (Kennedy and Spears 1998; Shi and Eberhart 1998), and multimodal problem generators as described by Kennedy and Spears (1998). Kennedy (1997) used the PSOA as an optimization paradigm that simulates the ability of human societies to process knowledge. The algorithm models the exploration of a problem space by a population of individuals; individuals' successes influence their searches and those of their peers.

Lately, the PSOA was successfully applied to the optimal shape and size design of structures by Fourie and Groenwold (2000). An operator, namely craziness, was re-introduced, together with the use of dynamic varying maximum velocities and inertia. Fourie and Groenwold (2002) also applied the PSOA to generally constrained nonlinear mathematical programming problems. New operators were introduced, namely the elite particle and the elite velocity. An attempt was also made to optimize the parameters associated with the various operators in the case of generally constrained nonlinear mathematical programming problems.

In this paper we focus on the application of the PSOA to the optimal size and shape design problem. Close to optimal values for the various parameters, as obtained by Fourie and Groenwold (2002), are used in optimizing the structures. An important motivation for this line of research is that the PSOA can easily be parallelized on massive parallel processing machines. Furthermore, the PSOA is simpler, both in formulation and computer implementation, than the GA.

The development of our paper is as follows: Firstly, the optimal size and shape design problem is formulated, whereafter the PSOA is discussed. This is followed by the application of the PSOA to several benchmark problems in size and shape optimization. Finally, the performance of the PSOA is compared with that of the GA, and also with gradient based algorithms.

2 Problem formulation

We consider two distinct problem classes in structural optimization, namely optimal sizing design and optimal shape design. In both cases, minimum weight is selected as the objective function f. The general optimal design problem is formulated as follows: Find the minimum weight f^* such that

$$f^* = f(\mathbf{x}^*) = \min f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}\,, \qquad (1)$$

subject to the general inequality constraints

$$g_j(\mathbf{x}) \le 0\,, \quad j = 1, 2, \dots, m\,, \qquad (2)$$

where $\mathbf{a}$ and $\mathbf{x}$ are column vectors in $\mathbb{R}^n$, and f and g_j are scalar functions of the design variables $\mathbf{x}$; $\mathbf{x}$ represents the member cross-sections or the geometry of the structure. The inequality constraints g_j may resemble stress, strain, displacement or linear buckling constraints. The finite element method (FEM) is used to approximate the objective function f and the constraint functions g_j.

To facilitate the inclusion of the constraints (2) in the GA and the PSOA, (1) is modified to become

$$\tilde{f} = f(\mathbf{x}) + \sum_{j=1}^{m} \lambda_j [g_j(\mathbf{x})]^2 \mu_j(g_j)\,, \qquad (3)$$

with

$$\mu_j(g_j) = \begin{cases} 0 & \text{if} \quad g_j(\mathbf{x}) \le 0 \\ 1 & \text{if} \quad g_j(\mathbf{x}) > 0 \end{cases}\,, \qquad (4)$$

and $\lambda_j > 0$, prescribed.

3 Particle swarm optimization (PSOA)

In our implementation of the PSOA, the social behaviour of birds is mimicked. Individual birds exchange information about their position, velocity and fitness, and the behaviour of the flock is then influenced to increase the probability of migration to regions of high fitness.

In flight, each bird[1] in a flock is considered to continuously process information about its current *position* and *velocity*. In addition, information regarding its position with respect to the flock is processed. In the optimal size and shape design problem, the position of each bird is represented by the design variables $\mathbf{x}$, while the velocity of each bird $\mathbf{v}$ influences the incremental change in the position of each bird, and hence the design variables.

Let us consider a flock of p particles[2]. For particle d, Kennedy and Eberhart (1995) originally proposed that the position $\mathbf{x}^d$ be updated as

$$\mathbf{x}^d_{k+1} = \mathbf{x}^d_k + \mathbf{v}^d_{k+1}\,, \qquad (5)$$

while the velocity $\mathbf{v}^d$ is updated as

$$\mathbf{v}^d_{k+1} = \mathbf{v}^d_k + c_1 r_1(\mathbf{p}^d_k - \mathbf{x}^d_k) + c_2 r_2(\mathbf{p}^g_k - \mathbf{x}^d_k)\,. \qquad (6)$$

Here, the subscript k indicates a pseudo-time increment. $\mathbf{p}^d_k$ represents the best previous position of particle d at time k, while $\mathbf{p}^g_k$ represents the global best position in the swarm at time k. r_1 and r_2 represent uniform random numbers between 0 and 1. Kennedy and Eberhart propose that $c_1 = c_2 = 2$, in order to allow a mean of 1 (when multiplied by the random numbers r_1 and r_2). The result

[1] For reasons of brevity, we restrict ourselves to bird flocking. Promising results based on bee swarms have recently been proposed. In particular, the swarm-and-queen approach (Clerc 1999) seems worthy of future perusal

[2] The terms *particle* and *bird* are used interchangeably

of using these proposed values is that birds *overfly* the target half the time.

Shi and Eberhart (1998) later introduced an inertia term w by modifying (6) to become

$$\mathbf{v}_{k+1}^d = w\mathbf{v}_k^d + c_1 r_1(\mathbf{p}_k^d - \mathbf{x}_k^d) + c_2 r_2(\mathbf{p}_k^g - \mathbf{x}_k^d)\,. \tag{7}$$

They proposed that w be selected such that $0.8 < w < 1.4$. In addition, they report improved convergence rates when w is decreased linearly during the optimization.

With the objective to improve the rate of convergence of the PSOA, Fourie and Groenwold (2002) proposed some modifications to the existing PSOA. These modifications relate to the use of best ever position, maximum velocity, inertia, craziness, elite particle and elite velocity.

3.1 Best ever position

Noting that the problems we consider are mostly convex in nature, we propose that $\mathbf{p}_g$ (indicating the best ever position in the swarm) replace the best position of the swarm $\mathbf{p}_k^g$ at time k. Hence, (7) is modified to become

$$\mathbf{v}_{k+1}^d = w\mathbf{v}_k^d + c_1 r_1(\mathbf{p}_k^d - \mathbf{x}_k^d) + c_2 r_2(\mathbf{p}_g - \mathbf{x}_k^d)\,. \tag{8}$$

Replacing $\mathbf{p}_k^g$ with $\mathbf{p}_g$ increases the pressure exerted on the particle to converge towards the global optimum without additional function evaluations. Numerical experimentation suggests that this approach improves the convergence rate of the algorithm (Fourie and Groenwold 2002).

3.2 Maximum velocity

Shi and Eberhart (1998) experimented with fixed maximum velocities. They reported that a maximum velocity setting can be eliminated, but at the expense of a greater number of function evaluations. Additionally, the number of function evaluations can be lowered by employing a linear decrease in the inertia w. Based on numerical experimentation, we select a starting value $\mathbf{v}_0^{\max}$ and then decrease this value by the fraction β if no improved solution is obtained within h consecutive time steps, i.e.

$$\text{if } f(\mathbf{p}_g)|_k \;\geq\; f(\mathbf{p}_g)|_{k-h} \text{ then } \mathbf{v}_{k+1}^{\max} \;=\; \beta\mathbf{v}_k^{\max}\,, \tag{9}$$

with $0 < \beta < 1$. Numerical experimentation suggests that this approach improves the convergence rate of the algorithm (Fourie and Groenwold 2002). Additionally, we define γ as the fraction of the initial search space for each design variable which is allocated to $\mathbf{v}_0^{\max}$, the initial maximum velocity. The initial maximum velocity is then calculated according to

$$\mathbf{v}_0^{\max} = \gamma(\mathbf{x}_{UB} - \mathbf{x}_{LB})\,, \tag{10}$$

where $\mathbf{x}_{UB}$ indicates the upper bounds for the design variables, while $\mathbf{x}_{LB}$ indicates the lower bounds.

3.3 Inertia

The value of w is strongly problem dependent, as reported by Shi and Eberhart (1998). Based on numerical experimentation, we select a fixed starting value w^0 and then decrease this value by the fraction α if no improved solution is obtained within h consecutive time steps, i.e.

$$\text{if } f(\mathbf{p}_g)|_k \;\geq\; f(\mathbf{p}_g)|_{k-h} \text{ then } w_{k+1} = \alpha w_k\,, \tag{11}$$

with $0 < \alpha < 1$. Numerical experimentation suggests that this approach improves the convergence rate of the algorithm (Fourie and Groenwold 2002), as opposed to linearly decreasing w during the optimization.

3.4 Craziness

Kennedy and Eberhart (1995) introduced a craziness operator to mimic the random (temporary) departure of birds from the flock. However, this operator was superseded by the introduction of a cornfield vector, initially introduced for demonstration purposes. We reintroduce the concept of craziness, with the particles having a predetermined probability of craziness P_{cr}. The direction and magnitude of the velocity of influenced particles are then changed randomly, i.e.

$$\text{if } r < \mathrm{P}_{cr}\,, \text{ then randomly assign } \mathbf{v}_{k+1}\,, \tag{12}$$

with

$$0 < \mathbf{v}_{k+1} \leq \mathbf{v}^{\max}\,, \tag{13}$$

for all particles d, where r again represents a uniform random number between 0 and 1.

Craziness has some similarity to the mutation operator in the genetic algorithm, since it increases the directional diversity in the flock. "Crazy" birds explore previously uncovered ground, which in general increases the probability of finding the optimum, albeit at additional computational expense, since the optimal craziness in the flock cannot be predetermined.

3.5 Elite particle

Eberhart and Shi (1998) discussed the possibility of an *elite* particle replacing the worst particle (with regard to its fitness). The concept is borrowed from the GA where the gene with the best fitness never vanishes. In our implementation, we update the status of the elite particle $\mathbf{x}^{pe}$ in the swarm at time k as

$$\mathbf{x}^{pe} = \mathbf{p}_g\,, \tag{14}$$

where $\mathbf{x}^{pe}$ typically replaces the worst positioned particle in the swarm with regard to its fitness. Comparing the functionality of $\mathbf{x}^{pe}$ with that of $\mathbf{p}_g$, $\mathbf{x}^{pe}$ is a particle positioned at the best ever position, whereas $\mathbf{p}_g$ serves as a point of attraction for all the particles. Numerical results indicate that the elite particle improves convergence rates (Fourie and Groenwold 2002).

3.6 Elite velocity

Another proposal by Eberhart and Shi (1998) is the possibility of a particle continuing in its direction of movement if its velocity resulted in an improvement of the best ever function value. We have expanded on this proposal and introduce the following for nonlinear mathematical programming problems: if velocity $\mathbf{v}_k^d$ caused an improvement on $\mathbf{p}_g$, then

$$\mathbf{x}_{k+1}^d = \mathbf{p}_g + c_3 r \mathbf{v}^{pe}\,, \tag{15}$$

where $\mathbf{v}^{pe} = \mathbf{v}_k^d$, the velocity that caused an improvement on $\mathbf{p}_g$. r indicates a random number generated uniformly between 0 and 1. Once again, improved convergence rates were obtained when using the elite velocity operator.

3.7 PSO algorithm

In this section an outline of our implementation of the PSOA is presented. We introduce s, the number of steps in which no improvement in the objective function occurs within a prescribed tolerance ε.

1. **Initialization step:** Set $k_{\max}$, c_1, c_2, c_3, P_{cr}, h, α, β, γ, $\mathbf{v}_0^{\max}$, w^0, s and ε.
   Randomly generate $\mathbf{x}_0^d \in D$ in $\mathbb{R}^n$, and $0 < \mathbf{v}_0^d < \mathbf{v}_0^{\max}$, for $d = 1, \ldots, p$.
   Set $k := 1$.
2. **Optimization steps:**
   (a) Evaluate the function $\tilde{f}$ according to (3) for each particle. Record the best function value $\tilde{f}_k$ for time step k.
   (b) If $k > s$ and $(|\tilde{f}_k - \tilde{f}_{k-s}|)/|\tilde{f}_k| < \varepsilon$, go to 3.
   (c) If $k > h$, conditionally reduce the inertia w and velocity $\mathbf{v}^{\max}$, using (11) and (9).
   (d) Update the best position $\mathbf{p}_k^d$ for particle d and the best ever position $\mathbf{p}_g$. Update the elite particle $\mathbf{x}^{pe}$ according to (14) when $\mathbf{p}_g$ is updated. Update $\mathbf{v}^{pe}$ according to (15), the velocity that resulted in obtaining $\tilde{f}_k$, when $\mathbf{p}_g$ is updated.
   (e) Set $k := k+1$. If $k = k_{\max}$, go to 3.
   (f) Update the velocity $\mathbf{v}$, according to (8).
   (g) Stochastically implement craziness, using (12).
   (h) Update the position $\mathbf{x}$, according to (5).
   (i) Go to 2a.
3. **Termination:** Stop

4 Numerical results

Four well-known benchmark size and shape optimization problems from literature are used to evaluate the performance of the PSOA.

A comparison is drawn between our implementation of the PSOA and some gradient based methods, the GA implementation by Carroll (1996) and our implementation of the GA[3]. The population size for the GAs are chosen to be equal to three times the number of design variables. Furthermore, the maximum number of generations used in both implementations of the GA is limited to 50 and 100 generations.

The values assigned to each of the parameters in the PSOA are tabulated in Table 1. A short summary of the various algorithms used in our study is given in Table 2 and Table 3. We report, for all the benchmark problems,

[3] Our GA is based on a binary representation. Eletism is used to ensure that the current best individual never vanishes, while selection pressure is applied similar to that in the canonical GA of Whitley (1994). A uniform crossover strategy is used in conjunction with a primitive crossover strategy as proposed by Imam and Al-Shihri (2000). Jump mutation is employed to protect against loss of genetic diversity. In addition, the probability of mutation is increased as the generations start to converge

Table 1 PSOA parameter values

Parameter	Value
w^0	1.40
h	3
α	0.99
β	0.95
P_{cr}	0.22
c_1	0.50
c_2	1.60
c_3	1.30
γ	0.40
p	4

Table 2 Summary of different algorithms

Acronym	Brief Description
PLBA	Modified Pschenichny's linearization method
NPSOL	SQP
RQP	SQP based on Pschenichny's descent function. Potential constraint strategy used. (Available in IDESIGN3)
GAC	GA implemented by Carroll
GAF	Our implementation of GA
PSOA	Current Study

Table 3 Summary of references to algorithms

Acronym	Reference
PLBA	Lim and Arora (1986)
NPSOL	Thanedar *et al.* (1985, 1986)
RQP	Lim and Arora (1986)
	Belegundu and Arora (1985)
GAC	Carroll (1996)

the number of function evaluations N_{fe} required to find f^* to a tolerance of 10^{-3}. The *compound cost* N_c, which we define as $N_c = N_{fe} + nN_{ge}$, where N_{ge} indicates the number of gradient evaluations, is also reported. N_c represents a pseudo cost which allows direct comparison between the efficiency of the gradient based algorithms and the derivative free algorithms. Furthermore, a finite difference scheme is used to perform the sensitivity analyses required by the RQP method, while the other gradient methods employ analytical derivatives.

The algorithms were all coded in FORTRAN77 and numerical results were obtained using a 500 MHz personal computer. All the results that are reported, unless otherwise indicated, are the best obtained from 5 different runs.

4.1 Two-bar plane truss

The geometry and loading conditions of this simple truss structure are depicted in Fig. 1. The member cross-sections represent the design variables. Arora (1989) gives an analytical solution of $f^* = 8.046 \times 10^3$. Numerical results are tabulated in Table 4, while the convergence history is depicted in Fig. 2.

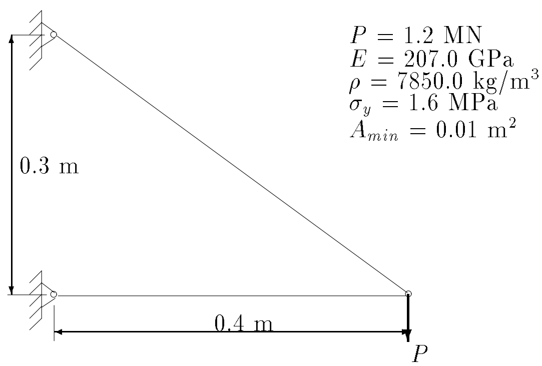

Fig. 1 Two-bar plane truss structure

The PSOA is more expensive than the RQP algorithm, but notably less expensive than the GAs. In addition, the quality of the solution found using the PSOA is superior to the solution found using the GAs. The convergence history (depicted in Fig. 2) reveals that most of the function values in the PSOA and GAs are associated with refinement of the minimum to the (reasonably stringent) tolerance of 10^{-3}. The neighborhood of the minimum is found quite quickly.

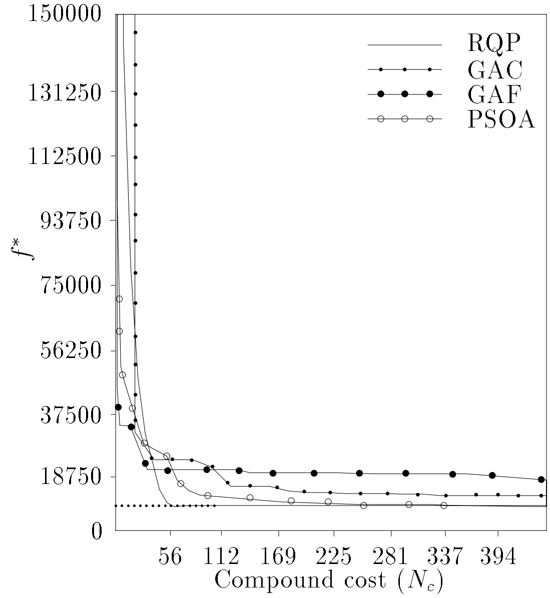

Fig. 2 Two-bar plane truss convergence history (stress constraints)

Table 4 Two-bar plane truss: comparative data

Algorithm	It./Gen.	N_{fe}	N_{ge}	N_c	f^*
RQP	10	58	9	76	8.046×10^3
GAC	50	300	0	300	9.662×10^3
GAC*	100	600	0	600	8.041×10^3
GAF	50	300	0	300	1.111×10^4
GAF*	100	600	0	600	9.115×10^3
PSOA	111	333	0	333	8.046×10^3
Arora (1989)	–	–	–	–	8.046×10^3

4.2 10-bar plane truss

The geometrical data of the nonconvex 10-bar plane truss structure is depicted in Fig. 3. The member cross-

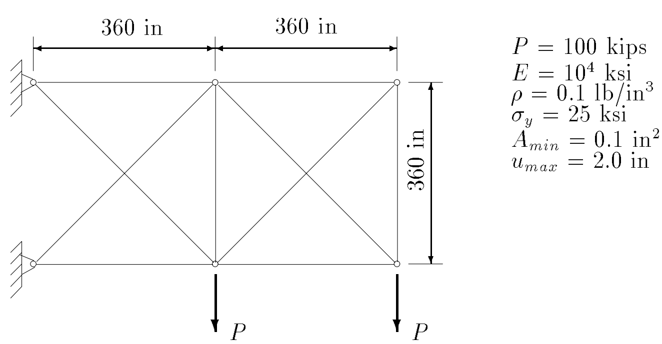

Fig. 3 10-Bar plane truss structure

Table 5 10-Bar plane truss: comparative data (sp 1)

Algorithm	It./Gen.	N_{fe}	N_{ge}	N_c	f^*
PLBA	15	17	127	1287	1.665×10^3
NPSOL	10	13	100	1013	1.665×10^3
RQP	23	497	84	1337	1.665×10^3
GAC	50	1500	0	1500	4.922×10^3
GAC*	100	3000	0	3000	4.276×10^3
GAF	50	1500	0	1500	4.214×10^3
GAF*	100	3000	0	3000	2.571×10^3
PSOA	278	1111	0	1111	2.244×10^3

Table 6 10-Bar plane truss: comparative data (sp 2)

Algorithm	It./Gen.	N_{fe}	N_{ge}	N_c	f^*
PLBA	33	51	215	2201	5.061×10^3
NPSOL	32	50	576	5810	5.061×10^3
RQP	32	683	93	1613	5.077×10^3
GAC	50	1500	0	1500	6.888×10^3
GAC*	100	3000	0	3000	4.833×10^3
GAF	50	1500	0	1500	6.933×10^3
GAF*	100	3000	0	3000	6.853×10^3
PSOA	276	1101	0	1101	5.112×10^3

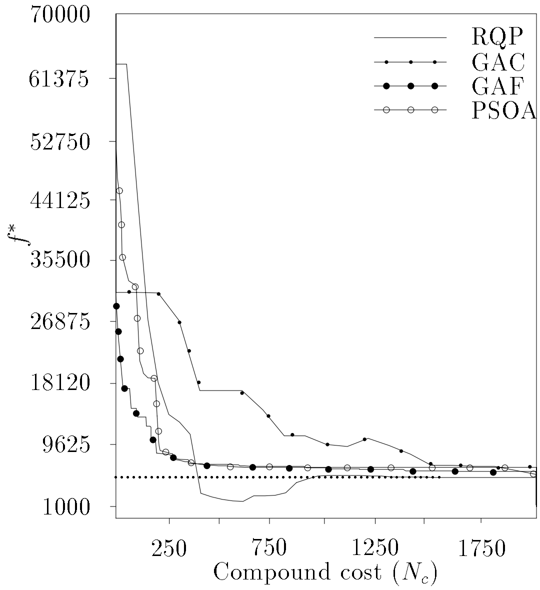

Fig. 5 10-Bar plane truss convergence history (sp 2)

sectional areas represent the design variables. Two distinct cases are considered, namely stress constraints only (sp 1), and stress and displacement constraints (sp 2). For the two subproblems, the results are tabulated in Tables 5 and 6 respectively, while the convergence histories are plotted in Figs. 4 and 5 respectively.

Once again, the results indicate that the rate of convergence of the PSOA is superior to that of the GA implemented by Carroll (1996). However, the PSOA compares well with our implementation of the GA. Furthermore, the rate of convergence of the PSOA is comparable with that of the gradient method. The proximity of the optimal solution for both subproblems is reached relative quickly, after which significant effort is spent on refinement of the solution.

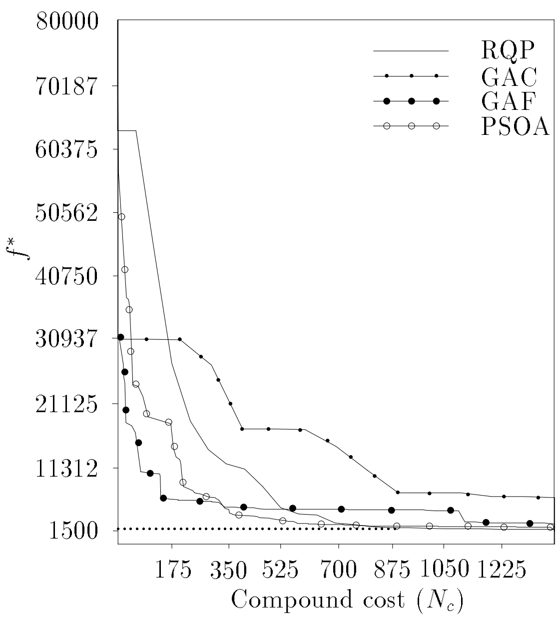

Fig. 4 10-Bar plane truss convergence history (sp 1)

4.3 25-bar space truss

The geometry of the 25-bar space truss structure is depicted in Fig. 6, and the truss is subjected to stress and displacement constraints. Complete descriptions of this problem can be found elsewhere in literature (i.e. Schmit and Farshi 1974; Haftka and Gürdal 1992).

The results are tabulated in Table 7, while the convergence histories are plotted in Fig. 7. Similar observations

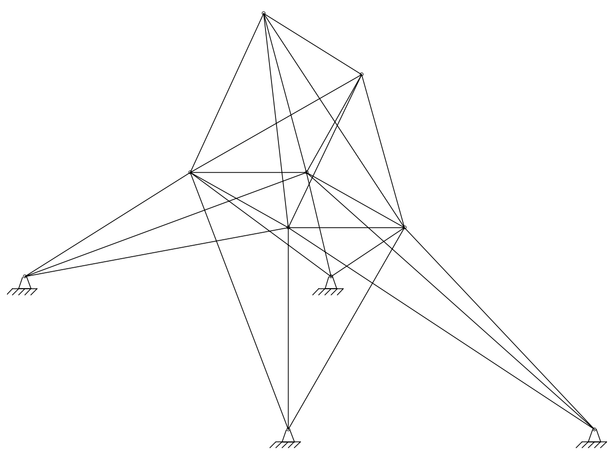

Fig. 6 25-Bar space truss structure

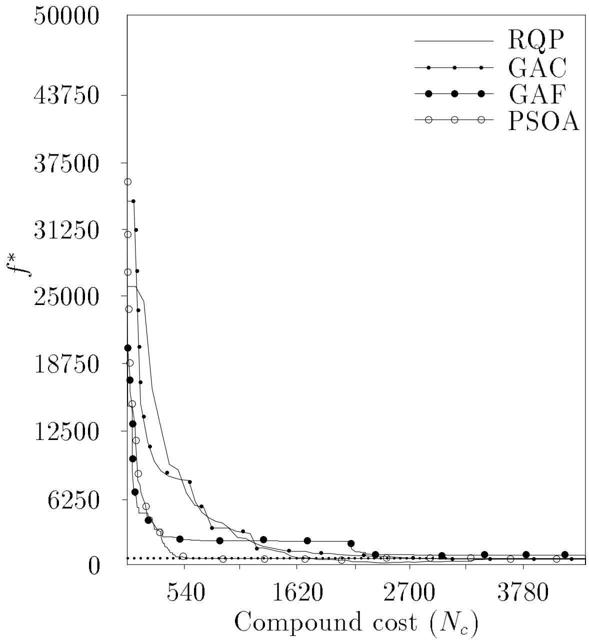

Fig. 7 25-Bar space truss convergence history (stress and displacement constraints)

Table 7 25-Bar space truss: comparative data

Algorithm	It./Gen.	N_{fe}	N_{ge}	N_c	f^*
PLBA	14	17	56	409	545.00
NPSOL	19	36	1634	11 474	545.03
RQP	22	385	58	849	545.05
GAC	50	1200	0	1200	3223.5
GAC*	100	2400	0	2400	726.78
GAF	50	1200	0	1200	693.07
GAF*	100	2400	0	2400	652.09
PSOA	297	1185	0	1185	572.29

can be made from these results as with the previous sizing problems studied. However, the PSOA outperforms the gradient method in the early iterative stages, but tends to slow down when refining the solution.

4.4 Torque arm

This problem (Fig. 8) is an adaption of the problem studied by Bennett and Botkin (1984). For simplicity, we do not include (or allow the generation of) any additional holes in the centre of the torque arm. The outer boundaries of the torque arm are represented by a spline function, described by the seven design variables. No topological changes are allowed, and geometric symmetry is enforced. To prevent excessive distortion of the finite element mesh, move limits are included, as well as limits on the design variables (see Table 8).

The stress distributions in the original and optimized geometry are depicted in Fig. 9 (dark areas indicate high stresses). Comparative results are given in Table 9, while the convergence history for this problem is depicted in Fig. 10. Our PSOA outperformed our genetic algorithm. Furthermore, our PSOA indicates a steeper initial rate of convergence than the gradient based method, but

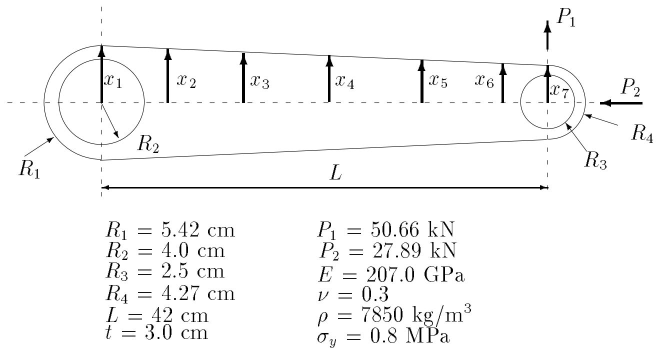

Fig. 8 Positioning of design variables for torque arm

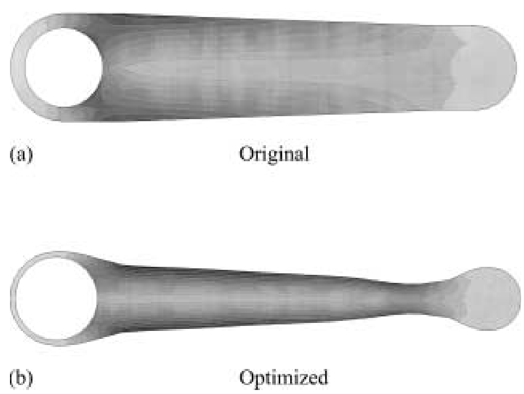

Fig. 9 Stress distribution in torque arm, (a) original geometry and (b) optimized geometry

Table 8 Torque arm: limits on design variables

Design variable	Lower bound	Upper bound
1	4.5	10.0
2	1.1	10.0
3	1.1	10.0
4	1.1	10.0
5	1.1	10.0
6	1.1	10.0
7	3.0	10.0

Table 9 Torque arm: comparative data with stress constraints

Algorithm	It./Gen.	N_{fe}	N_{ge}	N_c	f^*
RQP	12	192	175	1417	4.615×10^4
GAF	50	1050	0	1050	7.630×10^4
GAF*	100	2100	0	2100	6.585×10^4
PSOA	259	1033	0	1033	4.629×10^4

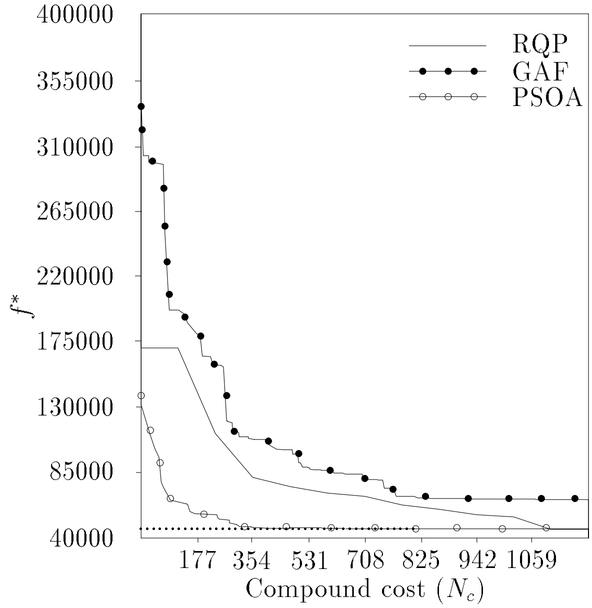

Fig. 10 Torque arm convergence history (stress constraints)

tends to appear stagnant when the proximity of the optimal solution is reached. All algorithms, however, reach the proximity of the optimal solution relatively quickly, after which computational effort is spent on refining the solution.

5 Conclusions

We have applied the particle swarm optimization algorithm (PSOA) to the optimal design of structures with sizing and shape variables. Our PSOA mimics the social behaviour of birds, and some new operators, namely elite particle and elite velocity, have been used.

Using comparative studies and benchmark problems, the suitability of the PSOA for problems in structural optimization is demonstrated. Although implemented in a simple form, the PSOA appears superior to the GA. In attaining the region of the optimum, the computational effort is comparable to that of the gradient based recursive quadratic programming algorithm.

An important motivation for this line of research is the ease with which the PSOA can be parallelized on massive parallel processing machines.

References

Arora, J.S. 1989: *Introduction to optimum design*. New York: McGraw-Hill

Beasley, D.; Bull, D.R.; Martin, R.R. 1993a: An overview of genetic algorithms. Part 1, fundamentals. *University Computing* **15**, 58–69

Beasley, D.; Bull, D.R.; Martin, R.R. 1993b: An overview of genetic algorithms. Part 2, research topics. *University Computing* **15**, 170–181

Belegundu, A.D.; Arora, J.S. 1985: A study of mathematical programming methods for structural optimization. Part 1: Theory. *Int. J. Numer. Meth. Engrg.* **21**, 1583–1599

Bennett, J.A.; Botkin, M.E. 1984: Structural shape optimization with geometric description and adaptive mesh refinement. *AIAA J.* **23**, 458–464

Carroll, D.L. 1996: Chemical laser modelling with genetic algorithms. *AIAA J.* **34**, 338–346

Clerc, M. 1999: The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. *Cong. on Evolutionary Computation*. Washington D.C.: IEEE

De Jong, K. 1975: *An analysis of the behavior of a class of genetic adaptive systems*. Ph.D. Dissertation, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor

Eberhart, R.; Shi, Y. 1998: Comparison between genetic algorithms and particle swarm optimization. *7-th Annual Conf. on Evolutionary Programming*

Fourie, P.C.; Groenwold, A.A. 2000: Particle swarms in size and shape optimization. In: Snyman, J.A. (ed.) *Proc. Workshop on Multidisciplinary Design Optimization* (held at University of Pretoria, South Africa), pp. 97–106

Fourie, P.C.; Groenwold, A.A. 2002: Particle swarm optimization in generally constrained problems. (submitted)

Goldberg, D.E. 1989: *Genetic algorithms in search, optimization, and machine learning*. New York: Addison-Wesley

Haftka, R.T.; Gürdal, Z. 1992: *Elements of structural optimization*. Dordrecht: Kluwer

Huiskes, R.; Weinans, H.; Grootenboer, H.J.; Dalstra, M.; Fudala, B.; Sloof, T.J. 1987: Adaptive bone-remodeling theory applied to prosthetic-design analysis. *J. Biomech.* **20**, 1135–1150

Imam, M.H.; Al-Shihri, M.A. 2000: A primitive crossover for improving the reliability of genetic algorithms for structural optimization. In: Topping, B.H.V. (ed.) *Computational engineering using metaphors from nature*, pp. 91–97. Edinburgh: Civil-Comp Press

Kennedy, J. 1997: The particle swarm: social adaptation of knowledge. *Proc. Int. Conf. on Evolutionary Computation*, pp. 303–308

Kennedy, J.; Eberhart, R. 1995: Particle swarm optimization. *Proc. IEEE Int. Conf. on Neural Networks*, pp. 1942–1948

Kennedy, J.; Spears, W.M. 1998: Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator. *Proc. Int. Conf. on Evolutionary Computation*, pp. 78–83

Lim, O.K.; Arora, J.S. 1986: An active set RQP algorithm for engineering design optimization. *Comp. Meth. Appl. Mech. Engrg.* **57**, 51–65

Mattheck, C.; Burkhardt, S. 1990: A new method of structural shape optimization based on biological growth. *Int. J. Fatigue* **12**, 185–190

Metropolis, N.; Rosenbluth, A.; Rosenbluth, M.; Teller, A.; Teller, E. 1953: Equation of state calculations by fast computing machine. *J. Chem. Phys.* **21**, 1084–1092

Schmit, L.A.; Farshi, B. 1974: Some approximation concepts for structural synthesis. *AIAA J.* **12**, 692–699

Shi, Y.; Eberhart, R.C. 1998: Parameter selection in particle swarm optimization. *7-th Annual Conf. on Evolutionary Programming*

Shi, Y.H.; Eberhart, R.C. 1998: A modified particle swarm optimizer. *Proc. Int. Conf. on Evolutionary Computation*, pp. 69–73

Thanedar, P.B.; Arora, J.S.; Tseng, C.M.; Lim, O.K.; Park, G.J. 1986: Performance of some SQP algorithms on structural design problems. *Int. J. Numer. Meth. Engrg.* **23**, 2187–2203

Thanedar, P.B.; Park, G.J.; Arora, J.S. 1985: Performance of two superlinearly convergent RQP optimization algorithms on some structural design problems. *Technical Report*, Optimal Design Laboratory, College of Engineering, The University of Iowa

Whitley, D. 1994: A genetic algorithm tutorial. *Statics and Computing* **4**, 65–85