Chapter 5

Shape optimization problem

5.1 Introduction

Shape optimization involves the constrained minimization of a cost function. The cost function in turn typically involves the solutions of a system of partial differential equations, which depend on parameters that define a geometrical domain [59]. The continuum description of the geometrical domain is normally discretized. This allows for efficient solution of the system of partial differential equations, using for example the finite element method (FEM). Normally, the discretized geometric domain is defined by control variables with predefined freedom. The control variables in turn bound the geometrical domain through a predefined relationship, which may be piecewise linear, or based on B-splines, etc.

Furthermore, different meshing strategies can be used. These include fixed grid strategies [60, 61, 62], design element concepts [63], adaptive mesh strategies [64], and remeshing strategies. The first three methods imply an *a priori* mesh discretization with obvious limitations, for example when dealing with large shape changes in the geometry during optimization. Some of the drawbacks of remeshing strategies are the implementation expense, and the tendency of gradient based optimization methods to get trapped in local minima [65]. On the other hand, the (unstructured) remeshing strategies allow for generality in structural models and objective functions. Large shape changes can be accommodated using the remeshing strategy with minimum mesh distortion.

The cost function may be optimized using either a gradient free or gradient based optimization method. While the gradient free methods require only the relationship between the cost function and the discretized geometric domain to be specified, the gradient based optimization methods require additional sensitivity information. The sensitivities needed for the gradient optimization techniques can either be calculated numerically, semi-analytically or analytically. All these methods have merits and drawbacks. Numerical gradients using finite difference methods are computationally expensive, but are easily implementable. The semi-analytical and analytical methods are more complex to implement, but are computationally cheaper. The advantage of gradient free evolutionary strategies are their global optimization capability. They have been used with success by Xie and Steven [61, 62] in a fixed grid strategy. Related works of evolutionary strategies in shape optimization are the biological growth method of Mattheck and Burkhardt [66], and the genetic algorithm used by Garcia and Gonzales [60].

In shape optimization, mesh generation plays an important role and is in general an expensive aspect of a computational iteration when fixed grid strategies are not considered. Remeshing strategies in shape optimization accentuate the importance of robustness, computational speed, flexibility and accuracy of the mesh generator in discretizing the geometrical domain.

In this study a remeshing shape optimization environment is combined with the gradient free particle swarm optimization algorithm (PSOA) developed by Kennedy and Eberhart.

5.2 Problem formulation

The problem under consideration is now the general inequality constrained minimization problem stated as follows: Given a cost function $f\big(\Omega(\boldsymbol{x})\big)$, find the minimum f^* such that

$$f^* = f\big(\Omega^*(\boldsymbol{x}^*)\big) = \min_{\boldsymbol{x}\in\mathbb{R}^n}\{f\big(\Omega(\boldsymbol{x})\big) : \boldsymbol{g}\big(\Omega(\boldsymbol{x}),\boldsymbol{x}\big) \leq \boldsymbol{0}\}, \tag{5.1}$$

where $\mathbb{R}^n$ represents a set of n real numbers. The cost function $f(\boldsymbol{x})$ and the constraints $g_j(\boldsymbol{x}),\ j = 1, 2, \cdots, m$ are scalar functions of the control variables $\boldsymbol{x}$ and the geometrical domain $\Omega(\boldsymbol{x})$, which is also a function of the control variables $\boldsymbol{x}$. For the sake of brevity, the cost function and the constraints will respectively be denoted by $f(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$; this notation will however imply dependency on $\Omega(\boldsymbol{x})$. We choose to represent the geometrical domain boundary $\partial\Omega$ by a simple piecewise linear interpolation between the control variables. However, Bezier curves or B-splines, etc. may of course also be used.

In our case, the cost function $f(\boldsymbol{x}) = f\big(\boldsymbol{u}(\Omega(\boldsymbol{x}))\big)$ is an explicit function of the nodal displacements $\boldsymbol{u}$, which is obtained by solving the approximate finite element equilibrium equations for linear elasticity, formulated as

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}, \tag{5.2}$$

where $\boldsymbol{K}$ represents the assembled structural stiffness matrix and $\boldsymbol{f}$ the consistent structural loads.

5.2.1 Accommodation of constraints

Shape problems are subjected to an inequality constraint $g\big(\Omega(\boldsymbol{x}),\boldsymbol{x}\big) \leq \boldsymbol{0}$, being the maximum allowed volume.

The inequality constraints $\boldsymbol{g}\big(\Omega(\boldsymbol{x}),\boldsymbol{x}\big) \leq \boldsymbol{0}$ can be accommodated using a simple exterior penalty formulation. With the exterior penalty formulation the optimal design problem represented by (5.1) is now modified to become: Find the minimum f^* such that

$$\begin{aligned} f^* &= f\big(\Omega^*(\boldsymbol{x}^*)\big) = \min_{\boldsymbol{x}\in R^n}\left\{ f\big(\Omega(\boldsymbol{x})\big) + \sum_{j=1}^{m}\Lambda_j[g_j\big(\Omega(\boldsymbol{x}),\boldsymbol{x}\big)]^2\mu_j(g_j)\right\}, \quad \text{with} \\ \mu_j(g_j) &= \begin{cases} 0 & \text{if} \quad g_j\big(\Omega(\boldsymbol{x}),\boldsymbol{x}\big) \leq 0 \\ 1 & \text{if} \quad g_j\big(\Omega(\boldsymbol{x}),\boldsymbol{x}\big) > 0 \end{cases} \quad \text{and penalty parameters } \Lambda_j > 0, \text{ prescribed.} \end{aligned} \tag{5.3}$$

5.3 Mesh generation

In this section, we present triangulation based on the truss structure analogy proposed by Persson and Strang [17]. This incorporates a Delaunay strategy [67] which ensures good mesh quality, albeit at the cost of potentially introducing discontinuities between consecutive meshes due to the addition or removal of nodes. The continuous geometrical domain Ω is discretized by nodes $\boldsymbol{\mathcal{X}}$. The nodes $\boldsymbol{\mathcal{X}}$ are the union of the boundary nodes $\boldsymbol{\mathcal{X}}^{\partial\Omega}$ and the interior nodes $\boldsymbol{\mathcal{X}}^{\Omega}$.

5.3.1 Mesh generator based on a truss structure analogy

The mesh generator is based on a truss structure analogy that solves for the equilibrium position of a truss structure. The geometrical domain is defined by a signed distance function that signs the nodes outside the domain as positive, inside as negative and zero on the boundary. The distance function is a function of the control variables through the interpolation of the domain.

The truss force function $\mathcal{F}$ is defined with a force discontinuity, as no tensile forces are permitted in the truss elements. This allows the propagation of the nodes $\boldsymbol{\mathcal{X}}$ to the boundary $\partial\Omega$. The nodes are kept inside the geometrical domain by external forces acting on the boundary nodes $\boldsymbol{\mathcal{X}}^{\partial\Omega}$. The forces act perpendicularly to the boundary, keeping the nodes from moving outside the boundary while allowing movement along the boundary.

The truss force function $\mathcal{F}$ is defined as

$$\mathcal{F}(l, h_0) = \begin{cases} k(h_0 - l) & \text{if } l < h_0 \\ 0 & \text{if } l \geq h_0 \end{cases} \tag{5.4}$$

with k the spring (truss) stiffness, l the current spring length and h_0 the undeformed spring length (also referred to as the ideal element length). In Persson and Strang's [17] implementation, all springs are precompressed by 20%, which provides the driving force necessary to propagate nodes to the boundary.

The truss system $\boldsymbol{F}(\boldsymbol{\mathcal{X}}) = \mathbf{0}$ is transformed to a system of ordinary differential equations through the introduction of artificial time-dependence in the equations. The system is then solved by a forward Euler method

$$\boldsymbol{\mathcal{X}}_{n+1} = \boldsymbol{\mathcal{X}}_n + \Delta t \boldsymbol{F}(\boldsymbol{\mathcal{X}}_n). \tag{5.5}$$

The forward Euler method is essentially a matrix free method ideally suited to create meshes with a very large number of elements. This method exhibits linear convergence rates.

5.4 Numerical results for the cantilever beam problem

Consider the cantilever beam depicted in Figure 5.1. The domain has a predefined length of 30 units and a height of 10 units. The objective is to minimize the maximum vertical displacement of the structure, subject to a maximum volume constraint of 70%. The magnitude of F is 10 N.

Numerical results for the cantilever beam problem are presented, using linear elastic finite element analyses under plane stress conditions. All the simulations are performed using material property values of $E = 200 \times 10^3$ for Young's modulus, and $\nu = 0.3$ for Poisson's ratio. An elemental thickness of unity and ideal element length $h_0 = 1$ is used throughout. 21 control variables are used.

For the PSOA, a swarm size of $p = 20$ is used. The social and cognitive parameters are selected as $c_1 = c_2 = 2$. The constant inertia factors for PSOAF1 is $w = 0.7$, for PSOAF2 $w = 0.5$ and for PSOAF1* $w = 0.6$, together with $\alpha = 5$. Each run consists of 20000 function evaluations (1000 iterations). Results are obtained by conducting 10 independent runs for each formulation.

The volume constraint is accommodated using the exterior penalty approach with the exterior penalty parameter selected as $\Lambda = 10^8$. When a particle's position violates a simple bound the position update is allowed but no function evaluation is conducted. The simple bounds for this problem are $1 \leq x_k^i(d) \leq 10, \quad d = 1, 2, \ldots, n$. No velocity restrictions are implemented and the initial velocities are set equal to $\mathbf{0}$.

The chosen settings for each formulation are based on judgment obtained from the results of the test sets in Chapter 4. The aim is not to determine or use optimal settings for each formulation, but merely to illustrate the applicability of the PSOA in shape optimization.

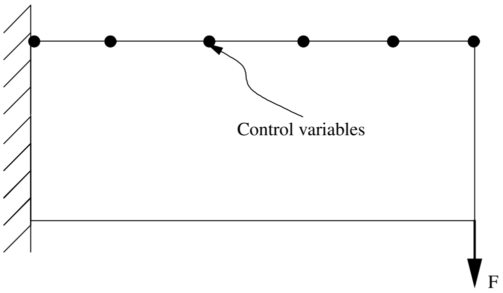


Figure 5.1: Initial structure and definition for the cantilever beam.

The results are summarized in Table 5.1 and Figure 5.2 illustrates the average convergence history over 10 runs. As shown, for the selected settings, PSOAF1* demonstrates the fastest initial convergence. PSOAF1 stagnates rapidly, due to the particle trajectories collapsing to line searches.

Selected shapes are depicted in Figures 5.3–5.11; this includes the best function value f^{best} obtained for each run, and the mean best function value $f^{\text{best}}_{\text{mean}}$ obtained over 10 runs. The statistical significance of using only 10 runs allows at most for only general observations. The performance of PSOAF1 is poor, as illustrated visually and shown numerically. PSOAF1* and PSOAF2 have comparable performance.

Table 5.1: Optimal results for the cantilever beam after only 1000 iterations.

Run	PSOAF1	PSOAF2	PSOAF1*
	$f^{\text{best}}(\times 10^{-2})$	$f^{\text{best}}(\times 10^{-2})$	$f^{\text{best}}(\times 10^{-2})$
1	1.981224	1.001660	1.000307
2	2.585177	1.001963	1.004114
3	1.814116	1.002475	1.014873
4	1.570862	1.002285	1.000066
5	1.578468	1.002799	1.004792
6	2.586973	1.000644	1.000284
7	1.488452	1.003684	1.000723
8	1.612004	1.000872	1.000594
9	2.136010	1.001666	1.001907
10	2.082606	1.000784	1.000346
$f^{\text{best}}_{\text{mean}}(\times 10^{-2})$	1.943589	1.001883	1.002800

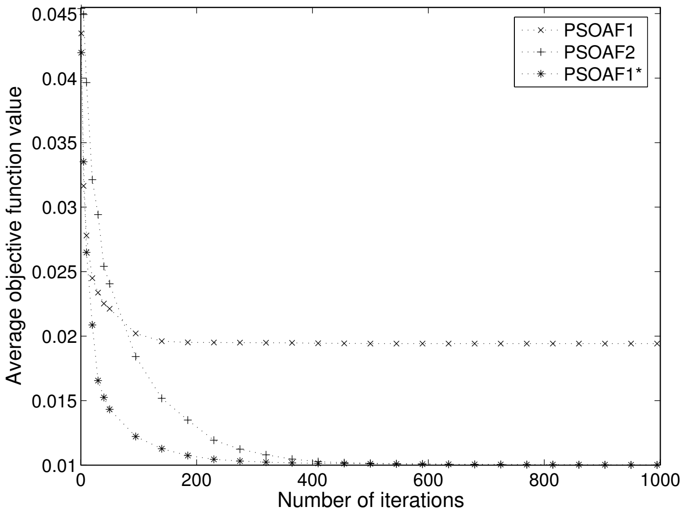
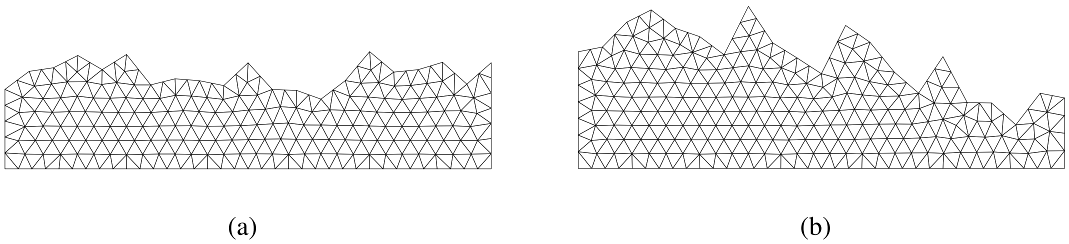
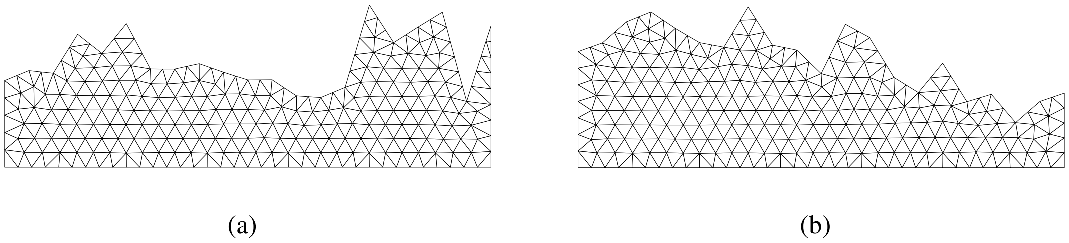


Figure 5.2: Mean function value history plot averaged over 10 runs for the cantilever beam shape optimization problem for PSOAF1 (with $w = 0.7$), PSOAF2 (with $w = 0.5$) and PSOAF1* (with $w = 0.6$ and $\alpha = 5$).



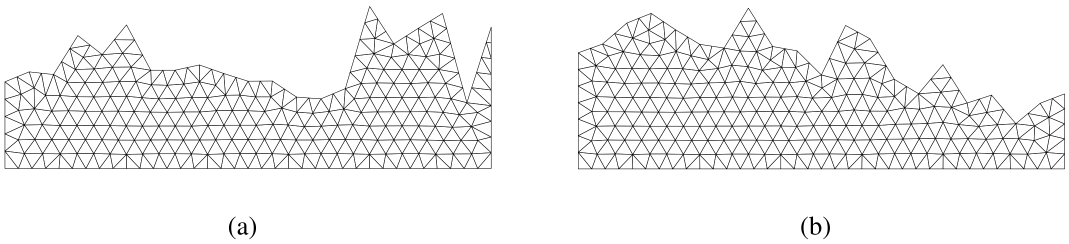
(a) (b)

Figure 5.3: Cantilever beam: results obtained after 100 iterations with PSOAF1 for a) the worst run (run 6), and b) the best run (run 7).



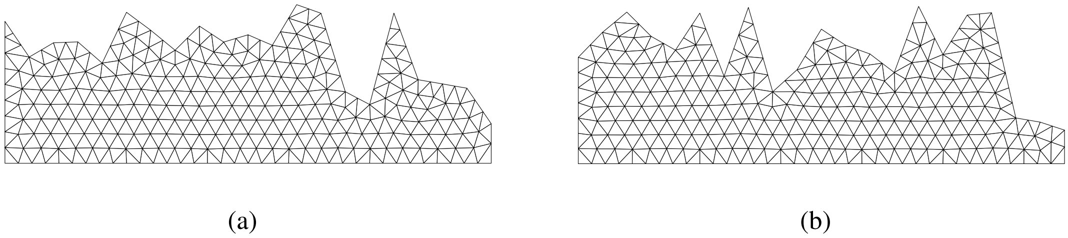
(a) (b)

Figure 5.4: Cantilever beam: results obtained after 500 iterations with PSOAF1 for a) the worst run (run 6), and b) the best run (run 7).



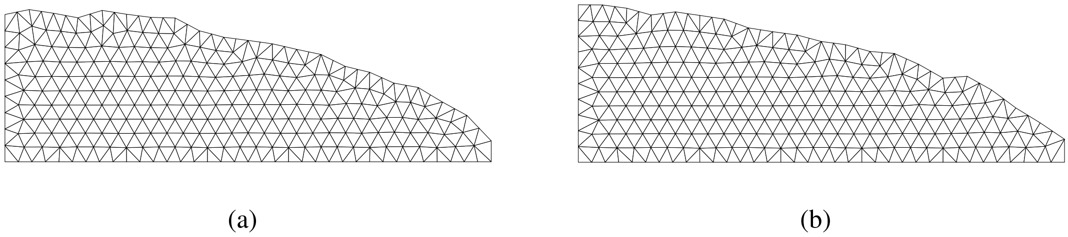
(a) (b)

Figure 5.5: Cantilever beam: results obtained after 1000 iterations with PSOAF1 for a) the worst run (run 6), and b) the best run (run 7).



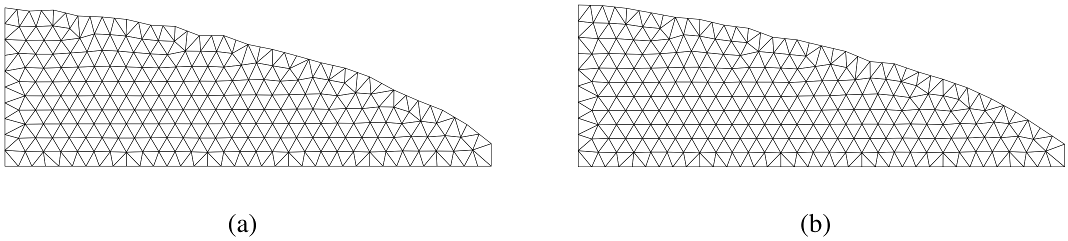
(a) (b)

Figure 5.6: Cantilever beam: results obtained after 100 iterations with PSOAF2 for a) the worst run (run 7), and b) the best run (run 6).



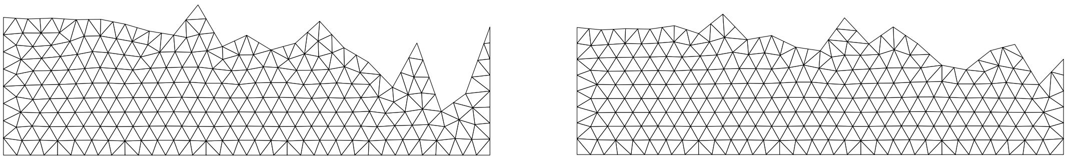
(a) (b)

Figure 5.7: Cantilever beam: results obtained after 500 iterations with PSOAF2 for a) the worst run (run 7), and b) the best run (run 6).



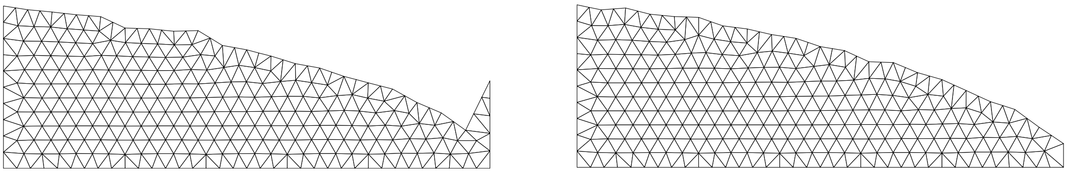
(a) (b)

Figure 5.8: Cantilever beam: results obtained after 1000 iterations with PSOAF2 for a) the worst run (run 7), and b) the best run (run 6).



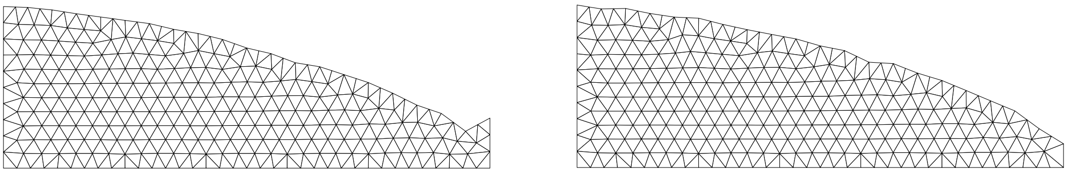
(a) (b)

Figure 5.9: Cantilever beam: results obtained after 100 iterations with PSOAF1* for a) the worst run (run 3), and b) the best run (run 4).



(a) (b)

Figure 5.10: Cantilever beam: results obtained after 500 iterations with PSOAF1* for a) the worst run (run 3), and b) the best run (run 4).



(a) (b)

Figure 5.11: Cantilever beam: results obtained after 1000 iterations with PSOAF1* for a) the worst run (run 3), and b) the best run (run 4).

5.5 Closure

In Chapters 3 and 4 it was demonstrated with a popular analytical test function set that the performance of PSOAF1 is poor. In turn, it was demonstrated that PSOAF2 and PSOAF1* demonstrate comparable performance.

In this chapter, the three different formulations of the PSOA were applied to a real engineering optimization problem, namely the optimal shape design of the cantilever beam. This was done using an unstructured remeshing optimization strategy, based on a truss structure analogy. Although the statistical significance of using only 10 runs at most allows for only general observations, it was again shown that PSOAF1 demonstrates poor performance. In turn, comparable performance for PSOAF1* and PSOAF2 was again demonstrated.

Chapter 6

Conclusions and recommendations

6.1 Conclusions

The main conclusions of this study are:

1. Implementation subtleties due to ambiguous notation have resulted in two distinctly different formulations of the PSOA, respectively denoted by PSOAF1 and PSOAF2 herein.

2. PSOAF1 as given by Eq. (3.1) is strictly *observer independent*, but the particle trajectories *collapse* to line searches.

3. PSOAF2 as given by Eq. (4.2) is *observer dependent*, although the particle trajectories are *space filling*.

4. Even though PSOAF2 is observer dependent, it outperforms PSOAF1 for both the rotated objective functions and unrotated objective functions (or decomposable multimodal objective functions).

5. A novel formulation, denoted PSOAF1*, was proposed, which is *observer independent*, while the particle trajectories are *space filling*.

6. The performance of PSOAF1* is comparable to the performance of PSOAF2 for the test function considered, with the added advantage of being objective.

7. The three different formulations were applied to a real engineering problem, namely the shape optimization of a cantilever beam.

8. For both the analytical test functions and the shape optimization problem, PSOAF1* and PSOAF2 outperformed PSOAF1, with the performance of PSOAF1* and PSOAF2 being comparable.

6.2 Recommendations

It is recommended that PSOAF1* is used in practice. Other formulations may of course also be formulated, but care should be taken to ensure that these formulations are objective and diverse, viz. the search trajectories should be space filling in n-dimensional space, and not collapse to line searches.

Secondly, when reporting on algorithms, it is of importance that the following points are reported upon, to ensure that research is reproducible:

1. General:
   (a) the selected velocity rule (using unambiguous notation),
   (b) the updating strategy (whether synchronous or asynchronous), and
   (c) the stopping criteria used.
2. The heuristics used (with a detailed description thereof):
   (a) minimum velocity restriction (and arithmetic precision),
   (b) maximum velocity restriction,
   (c) the implemented inertia strategy,
   (d) position restriction,
   (e) etc.
3. The parameters used:
   (a) the population size p,
   (b) the initial inertia and velocity values, and
   (c) the social and cognitive scaling factors c_1 and c_2.

Furthermore, in order to ensure robustness (objectivity) of an algorithm, decomposable functions should be used in both the rotated and the unrotated reference frames.

Bibliography

[1] J.A. Snyman. *Practical Mathematical Optimization*, volume 97 of *Applied Optimization*. Springer, New York, USA, 2005.

[2] R. Horst and P.M. Pardalos. *Handbook of Global Optimization*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1995.

[3] J. Kennedy and R.C. Eberhart. Particle swarm optimization. In *Proc. IEEE Int. Conf. Neural Networks*, volume 4, pages 1942–1948, Perth, Australia, November 1995.

[4] R.C. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Proc. 6th Int. Symp. Micro Machine and Human Science*, pages 39–43, Nagoya, Japan, October 1995.

[5] J. Salerno. Using the particle swarm optimization technique to train a recurrent neural model. In *Proc. 9th IEEE Int. Conf. Tools with Artificial Intelligence*, pages 45–49, Newport Beach, CA, USA, November 1997.

[6] J. Kennedy and R.C. Eberhart. A discrete binary version of the particle swarm algorithm. In *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, volume 5, pages 4104–4108, Orlando, FL, USA, October 1997.

[7] P.J. Angeline. Using selection to improve particle swarm optimization. In *Proc. IEEE Int. Conf. Evolutionary Computation*, pages 84–89, Anchorage, AK, USA, May 1998.

[8] Y. Shi and R.C. Eberhart. A modified particle swarm optimizer. In *Proc. IEEE Int. Conf. Evolutionary Computation*, pages 69–73, Anchorage, AK, USA, May 1998.

[9] J. Kennedy and W.M. Spears. Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator. In *Proc. IEEE Int. Conf. Evolutionary Computation*, pages 78–83, Anchorage, AK, USA, May 1998.

[10] V. Tandon, H. El-Mounayri, and H. Kishawy. NC end milling optimization using evolutionary computation. *Int. J. Machine Tools and Manufacture*, 42(5):595–605, 2002.

[11] G. Ciuprina, D. Ioan, and I. Munteanu. Use of intelligent-particle swarm optimization in electromagnetics. *IEEE Trans. Magnetics*, 38(2):1037–1040, 2002.

[12] W.H. Slade, H.W. Ressom, M.T. Musavi, and R.L. Miller. Inversion of ocean color observations using particle swarm optimization. *IEEE Trans. Geoscience and Remote Sensing*, 42(9):1915–1923, 2004.

[13] E. Ozcan and C.K. Mohan. Particle swarm optimization: surfing the waves. In *Proc. IEEE Congr. Evolutionary Computation*, volume 3, pages 1939–1944, Washington D.C., USA, July 1999.

[14] M. Clerc and J. Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evolutionary Computation*, 6(1):58–73, 2002.

[15] I.C. Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Proccessing Letters*, 85(6):317–325, 2003.

[16] Y.-L. Zheng, L.-H. Ma, L.-Y. Zhang, and J.-X. Qian. On the convergence analysis and parameter selection in particle swarm optimization. In *Proc. 2nd Int. Conf. Machine Learning and Cybernetics*, volume 3, pages 1802–1807, Xi-an, China, November 2003.

[17] P.-O. Persson and G. Strang. A simple mesh generator in Matlab. *SIAM Review*, 46(2):329–345, 2004.

[18] A.R. Cockshott and B.E. Hartman. Improving the fermentation medium for Echinocandin B production part II: Particle swarm optimization. *Process Biochemistry*, 36:661–669, 2001.

[19] B. Brandstätter and U. Baumgartner. Particle swarm optimization – mass-spring system analogon. *IEEE Trans. Magnetics*, 38(2):997–1000, 2002.

[20] U. Baumgartner, Ch. Magele, and W. Renhart. Pareto optimality and particle swarm optimization. *IEEE Trans. Magnetics*, 40(2):1172–1175, 2004.

[21] M.A. Abido. Optimal power flow using particle swarm optimization. *Int. J. Electrical Power and Energy Systems*, 24(7):563–571, 2002.

[22] H. Yoshida, Y. Fukuyama, S. Takayama, and Y. Nakanishi. A particle swarm optimization for reactive power and voltage control in electric power systems considering voltage security assessment. In *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, volume 6, pages 497–502, Tokyo, Japan, October 1999.

[23] P.C. Fourie and A.A. Groenwold. The particle swarm optimization algorithm in size and shape optimization. *Structural and Multidisciplinary Optimization*, 23(4):259–267, 2002.

[24] P.C. Fourie and A.A. Groenwold. The particle swarm algorithm in topology optimization. In *Proc. 4th World Congr. Structural and Multidisciplinary Optimization*, Dalian, China, May 2001. Paper no. 154.

[25] J. Kennedy. The particle swarm: social adaptation of knowledge. In *Proc. IEEE Int. Conf. Evolutionary Computation*, pages 303–308, Indianapolis, IN, USA, April 1997.

[26] P.N. Suganthan. Particle swarm optimiser with neighbourhood operator. In *Proc. IEEE Congr. Evolutionary Computation*, volume 3, pages 1958–1962, Washington D.C., USA, July 1999.

[27] Y. Shi and R.C. Eberhart. Empirical study of particle swarm optimization. In *Proc. IEEE Congr. Evolutionary Computation*, volume 3, pages 1945–1950, Washington D.C., USA, July 1999.

[28] Y. Shi and R.C. Eberhart. Fuzzy adaptive particle swarm optimization. In *Proc. IEEE Congr. Evolutionary Computation*, volume 1, pages 101–106, Seoul, Korea, May 2001.

[29] M. Clerc. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Proc. IEEE Congr. Evolutionary Computation*, volume 3, pages 1951–1957, Washington D.C., USA, July 1999.

[30] J. Robinson and Y. Rahmat-Samii. Particle swarm optimization in electromagnetics. *IEEE Trans. Antennas and Propagation*, 52(2):397–407, 2004.

[31] A. Carlisle and G. Dozier. An off-the-shelf PSO. In *Proc. Workshop on Particle Swarm Optimization*, Purdue School of Engineering and Technology, Indianapolis, IN, USA, April 2001.

[32] M.M. Noel and T.C. Jannett. Simulation of a new hybrid particle swarm optimization algorithm. In *Proc. 36th Southeastern Symp. System Theory*, pages 150–153, Atlanta, GA, USA, 2004.

[33] T. Aruldoss, A. Victoire, and A.E. Jeyakumar. Hybrid PSO-SQP for economic dispatch with valve-point effect. *Electric Power Systems Research*, 71(1):51–59, 2004.

[34] S. Baskar and P.N. Suganthan. A novel concurrent particle swarm optimization. In *Proc. IEEE Congr. Evolutionary Computation*, volume 1, pages 792–796, Portland, OR, USA, June 2004.

[35] R. Mendes, J. Kennedy, and J. Neves. The fully informed particle swarm: simpler, maybe better. *IEEE Trans. Evolutionary Computation*, 8(3):204–210, 2004.

[36] F.J. Solis and R.J.-B. Wets. Minimization by random search techniques. *Mathematics of Operations Research*, 6(1):19–30, 1981.

[37] K.A. de Jong. *An analysis of the behaviour of a class of genetic adaptive systems*. PhD thesis, Univ. Michigan, Ann Arbor, MI, USA, 1975.

[38] D. Beasley, D. R. Bull, and R. R. Martin. An overview of genetic algorithms: Part 1, Fundamentals. *University Computing*, 15(2):58–69, 1993.

[39] I. Rechenberg. *Evolutionsstrategie - optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog, Stuttgart, DE, 1973.

[40] R. Storn and K. Price. Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optimization*, 11(4):341–359, 1997.

[41] S. Yang, M. Wang, and L. Jiao. A quantum particle swarm optimization. In *Proc. IEEE Congr. Evolutionary Computation*, volume 1, pages 320–324, Portland, OR, USA, June 2004.

[42] T. Krink, J.S. Vesterstrøm, and J. Riget. Particle swarm optimisation with spatial particle extension. In *Proc. IEEE Cong. Evolutionary Computation*, volume 2, pages 1474–1479, Honolulu, HI, USA, May 2002.

[43] J.S. Vesterstrøm, J. Riget, and T. Krink. Division of labor in particle swarm optimisation. In *Proc. IEEE Cong. Evolutionary Computation*, volume 2, pages 1570–1575, Honolulu, HI, USA, May 2002.

[44] G. Venter and J. Sobieszczanski-Sobieski. Particle swarm optimization. *AIAA J.*, 41(8):1583–1589, 2003.

[45] B.R. Secrest and G.B. Lamont. Visualizing particle swarm optimization - Gaussian particle swarm optimization. In *Proc. IEEE Swarm Intelligence Symposium*, pages 198–204, Indianapolis, IN, USA, April 2003.

[46] D.N. Wilke and A.A. Groenwold. On constrained optimisation of structural systems using particle swarms. In *Proc. 5th World Congr. Structural and Multidisciplinary Optimization*, Lido di Jesolo, Italy, May 2003. Paper no. 84.

[47] Y. Shi and R.C. Eberhart. Parameter selection in particle swarm optimization. In *Evolutionary Programming VII: Proc. 7th Annual Conf. Evolutionary Programming*, pages 591–600, San Diego, CA, USA, March 1998.

[48] D.N. Wilke, S. Kok, and A.A. Groenwold. Particle swarm searches and adaptive unstructured meshing in shape optimization. In *Proc. 4th Int. Conf. Engineering Computational Technology*, Lisbon, Portugal, September 2004. Paper 97.

[49] J. Kennedy. Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In *Proc. IEEE Congr. Evolutionary Computation*, volume 3, pages 1931–1938, Washington D.C., USA, July 1999.

[50] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill Kogakusha, Ltd., Tokyo, Japan, 1965.

[51] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Trans. Evolutionary Computation*, 1(1):67–82, 1997.

[52] T.M. English. Evaluation of evolutionary and genetic optimizers: no free lunch. In *Evolutionary Programming V: Proc. 5th Annual Conf. Evolutionary Programming*, pages 163–169, San Diego, CA, USA, February 1996.

[53] G.A. Holzapfel. *Nonlinear Solid Mechanics*. John Wiley and Sons Ltd., Chichester, UK, 2001.

[54] R. Salomon. Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39(3):263–278, 1996.

[55] R. Salomon. Some comments on evolutionary algorithm theory. *Evolutionary Computation*, 4(4):405–415, 1996.

[56] N. Metropolis and S. Ulam. The Monte Carlo method. *J. American Statistical Association*, 44(247):335–341, 1949.

[57] G. E. Mase and G. T. Thomas. *Continuum Mechanics for Engineers*. CRC Press, Boca Raton, FL, USA, 1992.

[58] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49, 2003.

[59] E. Laporte and P. Le Tallec. *Numerical Methods in Sensitivity Analysis and Shape Optimization*. Modeling and Simulation in Science, Engineering and Technology. Birkhäuser, Boston, USA, 2003.

[60] M.J. Garcia and C.A. Gonzales. Shape optimization of continuum structures via evolution strategies and fixed grid finite element analysis. *Structural and Multidisciplinary Optimization*, 26(2):92–98, 2004.

[61] Q. Li, G.P. Steven, O.M. Querin, and Y.M. Xie. Evolutionary shape optimization for stress minimization. *Mechanics Research Communications*, 26(6):657–664, 1999.

[62] Y.M. Xie and G.P. Steven. A simple evolutionary procedure for structural optimization. *Computers and Structures*, 49(5):885–896, 1993.

[63] M.H. Imam. Three-dimensional shape optimization. *Int. J. Numerical Methods in Engineering*, 18(5):661–673, 1982.

[64] A.D. Belegundu and S.D. Rajan. A shape optimization approach based on natural design variables and shape functions. *Computer Methods in Applied Mechanics and Engineering*, 66(1):87–106, 1988.

[65] G. Allaire, F. Jouve, and A.-M. Toader. Structural optimization using sensitivity analysis and a level-set method. *J. Computational Physics*, 194(1):363–393, 2004.

[66] C. Mattheck and S. Burkhardt. A new method of structural shape optimization based on biological growth. *Int. J. Fatigue*, 12(3):185–190, 1990.

[67] H. Edelsbrunner. *Geometry and Topology for Mesh Generation*. Number 7 in Cambridge Monographs on Applied and Computational Mathematics. Cambridge Univ. Press, Cambridge, UK, 2001.