A Pilot Study into the Optimization of the Shape of a Wave Energy Collector by Genetic Algorithm

A. P. McCabe[1], G. A. Aggidis[2] and M. B. Widden[3]

Renewable Energy Research Group,
Engineering Department,
Lancaster University,
Lancaster, U.K., LA1 4YR.
http://www.engineering.lancs.ac.uk/REGROUPS/LUREG/home.htm
[1] E-mail: a.p.mccabe@lancaster.ac.uk
[2] E-mail: g.aggidis@lancaster.ac.uk
[3] E-mail: m.widden@lancaster.ac.uk

Abstract

This pilot study forms part of research into the optimization of the shape of a wave energy collector to improve energy extraction using genetic algorithms. Two main types of genetic algorithms exist, differentiated by the use of binary or real numbers as object descriptors. The study is intended to ascertain if one type is more suited to the specific problem by comparing the performance of two example algorithms. The algorithms optimize the shape of a bisymmetric wave energy collector moving in two degrees of freedom (surge and pitch). The collector is described by ruled surfaces in one quadrant, defined by the positions of seven vertices. The cost function is based upon a first-order model of the system, with the collector optimally tuned to a number of incident regular waves with a generalized occurrence distribution. High velocities and large collector volumes are penalized. An assessment of the performance of the two algorithms is made, looking at the improvement in value and change in diversity of the respective populations. A comparison is also made of the computational requirements of the different parts of the optimization process.

Keywords: Genetic algorithms, marine energy conversion, optimization methods, WEC design.

Nomenclature

$\mathbf{B}(T)$	= matrix of dissipative loss coefficients (Ns, Ns/m, Ns/rad, Nms/rad)
g	= acceleration due to gravity (m/s)
G	= genotype
N	= number of wave periods
$\mathbf{O}$	= origin of co-ordinate system
$\overline{P}_{IN,opt}(T)$	= mean power absorbed with optimal tuning (W)
Q_k	= probability of occurrence of wave period k
T	= wave period (s)
$U_{opt,1}(T)$	= surge velocity with optimum power absorption (m/s)
$\mathbf{U}_{opt}(T)$	= velocity vector with optimum power absorption (m/s, rad/s)
$\mathrm{Var}_R(x_n)$	= relative variance in gene x_n
$\mathrm{Var}(x)$	= variance in x
x	= axis, aligned with wave direction
x_n	= vertex co-ordinate (m) and gene
$\hat{x}$	= mean of range of x
$\mathbf{X}(T)$	= vector of relative excitation forces / moments (N, Nm)
y	= axis perpendicular to wave direction
y_n	= vertex co-ordinate (m) and gene
z	= axis, perpendicular to free surface
z_n	= vertex co-ordinate (m) and gene
α	= wave amplitude (m)
ρ	= density of fluid (kg/m^3)
$\forall$	= immersed volume (m^3)
$*T$	= conjugate transpose of matrix / vector

1 Introduction

Effective wave energy extraction requires the conversion of the energy within the waves to drive the power take off system. This conversion process involves the response of a principal interface, or collector, to the forces induced by the waves. A systematic method of optimising the collector shape is sought, so that energy capture rates are improved and the design of future wave energy converters can be better informed. The work described in this paper forms part of the research undertaken by the SuperGen Marine Research Consortium (*http://www.supergen-marine.org.uk*) towards the development of such a robust, systematic approach. Overall, the project will require the development of a parametric description of wave power devices with a solid/water prime interface, the identification of appropriate cost functions and constraints, and the utilisation of an advanced optimization procedure, in the form of a genetic algorithm. Genetic algorithms can be categorized into two main types by the numeric system in which they operate, namely binary numbers and real numbers. This pilot study is intended to compare and contrast the performance of two example algorithms, to ascertain if one type is more suited to the specific problem.

Many of the methods used to solve optimization problems in engineering design have been developed during the last forty years. 'Traditional' methods are based upon linear or nonlinear numerical programming techniques and prove useful in the solution of simple or ideal problems. However, many optimization problems can be very complex and hard to solve using such algorithms. The development of stochastic, or 'metaheuristic' optimization techniques, such as genetic algorithms, has been driven by an attempt to overcome the shortcomings of traditional numerical methods. A common aspect of metaheuristic algorithms is the aim of replicating natural phenomena by a mixture of rule and chance, a major inspiration being biological evolution. Evolutionary systems were first simulated in the 1960s [1], leading to the evolutionary algorithms proposed by Fogel et al. [2], De Jong [3], and Koza [4], and, subsequently, genetic algorithms.

A genetic algorithm is a global search technique based on natural selection and other processes found in population genetics, first proposed by Holland [5] and further developed by Goldberg [6] and others. It combines 'survival-of-the-fittest' selection mechanisms with information exchange that is both structured and randomized to create an extremely flexible search algorithm. The main feature which distinguishes genetic algorithms from numerical or other heuristic algorithms is the concurrent evaluation of multiple solutions, which permits the exploration of a larger search space and, because the search procedure is stochastic, a greater likelihood of convergence on a global optimum. In contrast to many numerical programming methods, a genetic algorithm does not require the cost function to be smooth (that is, continuously differentiable up to some desired order over some domain), since the operation of the algorithm is based only on the cost function values themselves. Efficiency in the search process of a genetic algorithm is a balance between the exploitation of the best current resources (by retention of the best individuals) and exploration for new, possibly better resources (by recombination or mutation). The relative emphasis on exploitation and exploration, therefore, determines the rate of convergence on the solution and the diversity of the population evaluated during the search.

Genetic algorithms and other evolutionary computation methods have recently been applied in several areas of structural design optimization. A popular subject for study is the optimal design of steel frame structures, for example [7–8], while some have concentrated upon optimizing the design of planar structures, for example [9–10]. Many have also studied the use of genetic algorithms in the optimization of solid structures, mainly to improve load-bearing characteristics. Various geometric representations have been used, both for defining the surface of the structure and for the cost function analysis. The definition of the search space, by selection of the right representation of the geometry, is a crucial factor in the success of the optimization [11]. In many cases, surfaces are discretized for analysis, in the form of meshes, which are then represented as a bit-array in the genetic algorithm. A major shortcoming of the bit-array representation is that, with fine surface discretization meshes, the array size and the associated computational effort required becomes impracticably large [12]. To alleviate this complexity, several methods which disconnect the representation from the surface mesh used in the computation of the cost function have been introduced. For example, boundary element analysis methods have been used in [13] with a free form deformation technique to control the mesh. Another example is the integrated approach for 2-D and 3-D structural shape optimization used in [14], employing a β-spline surface representation, along with a genetic algorithm optimization process and boundary element method to calculate the structural stresses.

A rare example of the use of a genetic algorithm in the optimization of the design of a wave energy converter is that of Babarit, Clément and Gilloteaux [15] for the SEAREV device. The procedure adopted is a hybrid genetic / gradient-descent method. Patel, et al. [16], suggest a method which may be applied to the control of devices similar to Pelamis (*Pelamis Wave Power Ltd.*: *http://www.pelamiswave.com*), using an artificial neural network, optimized by genetic algorithms.

Section 2 outlines the genetic algorithms used in this study. The definition of the collector geometry is covered in Section 3 and the cost function is described in Section 4. Results are presented and discussed in Section 5, and conclusions drawn in Section 6.

2 The Genetic Algorithms

The characteristic feature of genetic algorithms is the simultaneous evaluation of multiple solutions. In these solutions the 'real-world' physical variables (called phenotypes) are represented by sets of variables in the genetic algorithm search space called genotypes. The general genotype (or chromosome) consists of a sequence of genes. The basic genetic algorithm structure is depicted in Fig. 1. Each iteration of the algorithm, or 'generation', usually consists of three main operations: selection; recombination; and mutation. Selection introduces the Darwinian element to the optimization process by identifying and retaining the strong individuals, whilst eliminating the weak ones.

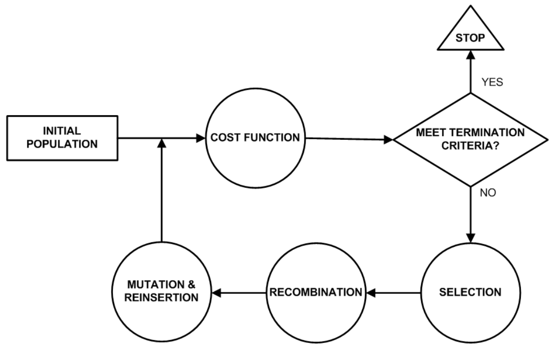

Figure 1: Genetic algorithm structure.

Every individual is assigned a fitness value in relation to either its cost function value, or to its ranking in the generation. The fitness value determines the probability of an individual's survival into or contribution of genetic information toward the next generation. Recombination (often called 'crossover', although this refers to a specific type of recombination) is the exchange of genetic information from the parents to form offspring for the next generation. The underlying assumption of recombination is that parts of an individual's representation (or schemata) contribute to the overall fitness value of that individual, and that the reassembly of schemata from good individuals can lead to the production of a better individual. Mutation is the sporadic random variation of an element in the representation of a new individual. The rationale behind mutation is to increase the genetic diversity of the population by introducing non-recursive elements into the pool of genetic information. Reinsertion is the assembly of the next generation of candidate solutions from those newly created by the algorithm, or retained from the previous generation. The termination criteria are generally problem-specific and typically comprise the completion of a maximum number of iterations, or the completion of a fixed number of iterations in which the same solution produces the best cost function performance.

Two algorithms are used in this study, one with binary-coded genotypes, the other with real-coded genotypes. There are common features to both algorithms. In both cases the population consists of 50 candidate solutions and selection is ranking-based. Baker's linear ranking algorithm [17] with a selective pressure of 2 is used to determine fitness values, and the progenitors of the next generation are selected by the Roulette-Wheel method with stochastic universal sampling [17]. Twenty parents are selected from each generation and are paired randomly, each pair producing four offspring. An elitist reinsertion approach is used wherein the 10 fittest members of each generation are retained to form the next generation along with the 40 offspring produced by recombination. The binary-coded and real-coded algorithms differ in their recombination and mutation methods. Uniform Crossover [18] is used with the binary-coded genotypes, whilst Intermediate Recombination [19] is used with the real-coded genotypes. Mutation in the binary-coded algorithm consists of changing a randomly-selected gene to its inverse state (0 to 1, or vice versa), whilst the mutation operation for the real-coded algorithm is that proposed for the Breeder Genetic algorithm [19], essentially changing a randomly-selected gene to a different value in its possible range. The probability of mutation, or the mutation rate, is set to be equal to the reciprocal of the number of genes in both cases [19–20]. Five runs are made with the optimization being terminated after 50 generations. The optimization is then continued for a further 50 generations with the population containing the genotype with the best cost function value. The same course of action is taken for both algorithms.

3 From Geometry to Genotype

The geometric description of the body to be optimized must be put into a format suitable for the genetic algorithm. The body is defined in a three-dimensional Cartesian co-ordinate system, (x, y, z), which is stationary relative to the undisturbed position of the free surface and the body. The orthogonal axes, x and y, define a plane parallel to the free surface, and the third axis, z, is perpendicular to it. The origin is located on the free surface, designated 'O' in Fig. 2. The collector body is symmetric in both the x-z and y-z planes. The body geometry is taken to consist of ruled surfaces, which, for one quadrant, comprise three adjoining faces that can be varied by adjusting the positions of seven vertices, designated V_m (m = 1, 2, …, 7) in Fig. 2. The co-ordinates of the vertices are listed in Table 1.

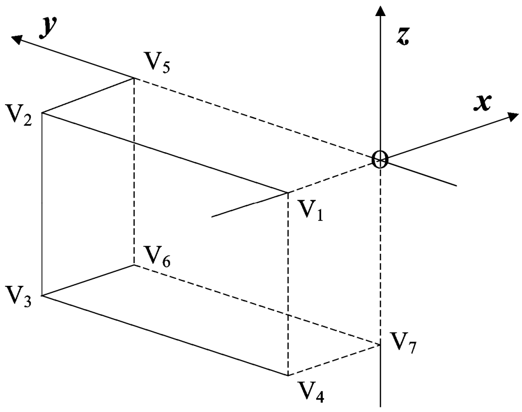

Figure 2: Collector body vertices.

V_1	:	x_1	0	0
V_2	:	x_2	y_1	0
V_3	:	x_3	y_2	z_1
V_4	:	x_4	0	z_2
V_5	:	0	y_3	0
V_6	:	0	y_4	z_3
V_7	:	0	0	z_4

Table 1: Vertex co-ordinates.

Because of the symmetry of the body, only 12 of the 21 co-ordinates can be moved by the optimization routine, these being denoted x_n, y_n and z_n (n = 1, 2, 3, 4). So that an exact binary coding of the search space can be made, these co-ordinates are constrained to exist in a finite space, with the limits in each direction listed in Table 2. The resulting minimum collector volume is 124 m^3, and the maximum collector volume is 6,174 m^3. Each range may then be discretized to millimetre intervals.

	Maximum [m]	Minimum [m]
x_n	–2.905	–7.000
y_n	10.500	2.309
z_n	–4.617	–21.000

Table 2: Vertex constraints.

The genotype, therefore, consists of the vector of 12 variable co-ordinates, in either binary-coded or real-coded form, and the resulting mutation rate is 1/12. The initial populations consist of mutations of a body with vertices at the mean points of each range, but not the body itself.

$$G = \begin{bmatrix} x_1 & x_2 & \dots & z_4 \end{bmatrix} \tag{1}$$

4 The Cost Function

The cost function defines the meaning of the optimization and much work is planned over the lifetime of the SuperGen Marine project in its development. A fundamental part of the cost function lies in predicting the responses of a candidate device model to a set of sea conditions. Due to the generic nature of the research the device model needs to be realistic, but must avoid being too specific to any particular device. Similarly, the incident wave conditions also need to be realistic, but avoid being too specific to a particular site.

In this study an elementary cost function, based upon a first-order model of the system, is used. The model responses have two degrees of freedom: translation along the x-axis (surge motion), and rotation around the y-axis (pitch motion). The sea conditions consist of a number of incident regular waves, and the candidate collectors are optimally tuned to absorb the greatest power from each one. It is assumed that some wave periods will occur more often than others, so a generalized occurrence distribution is used. Should the optimal tuning result in high velocities, hence large displacements, this is penalized. Large collector volumes are also penalized.

It is shown in [21; 22, p.216] that, if higher-order effects are disregarded, the mean power absorbed, $\overline{P}_{IN,opt}(T)$, by an optimally tuned, unconstrained point absorber in a regular wave of amplitude, α, and period, T, is given by

$$\overline{P}_{IN,opt}(T) = \frac{\alpha^2}{8}[\mathbf{X}(T)]^{*T}\mathbf{B}(T)^{-1}\mathbf{X}(T) \tag{2}$$

where $\mathbf{X}(T)$ is the vector of unit-wave-amplitude excitation forces and moments, and $\mathbf{B}(T)$ the matrix of radiation and other dissipative loss coefficients, both varying with the wave period. The superscript, $*T$, denotes conjugate transpose of the matrix. The vector of translational and angular velocities when optimum power absorption occurs is

$$\mathbf{U}_{opt}(T) = \frac{\alpha}{2}\mathbf{B}^{-1}(T)\,\mathbf{X}(T) \tag{3}$$

The values of $\mathbf{B}(T)$ and $\mathbf{X}(T)$ were obtained numerically using the hydrodynamic analysis software package, WAMIT™ [23]. High velocities and large collector volumes are penalized by dividing the absorbed power by the product of the surge velocity, $U_{opt,1}(T)$, the wave amplitude, α, and the buoyancy force, $\forall\rho g$, where $\forall$ is the immersed volume, ρ the density of the water, and g the acceleration due to gravity. The resulting quantity is non-dimensional. In the case of a single-degree-of-freedom system, multiplying the ratio of absorbed power to velocity by 4 gives the excitation force magnitude. The cost function, therefore, is

$$C = \frac{1}{\forall\rho g}\sum_{k=1}^{N}\left\{\frac{4\overline{P}_{IN,opt}(T_k)}{\alpha U_{opt,1}(T_k)}Q_k\right\} \tag{4}$$

in which N is the number of wave periods and Q_k is the probability of occurrence of wave period k. The number of wave periods used in this study is 17, from 3 to 19 seconds, and their respective probabilities are determined by a normal distribution centred upon a wave period of 11 seconds, shown in Fig. 3.

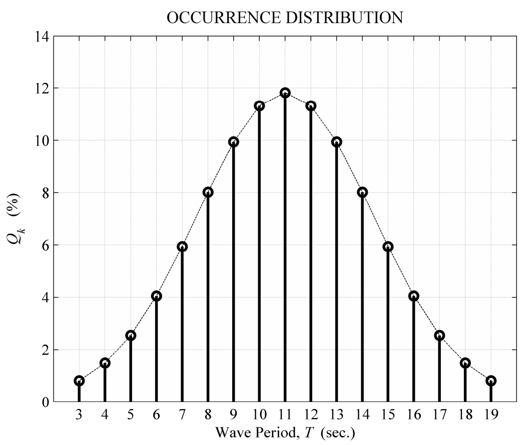

Figure 3: Wave period probability of occurrence.

5 Results

After the five preliminary optimization runs were made, similar results were obtained with both algorithms. The final top cost function values varied over a relatively large range with comparable variance. Using the binary-coded algorithm, the final top cost function values vary over a range of 0.75, from 1.1 to 1.85 with a mean value of 1.47 and a variance of 0.1. Using the real-coded algorithm, the final top cost function values vary over a range of 0.77, from 0.88 to 1.65 with a mean value of 1.15 and a variance of 0.12. The optimization was then resumed for the best cases with each algorithm and the progress of the optimization, in terms of best and mean cost function values in each generation, is shown in Fig. 4.

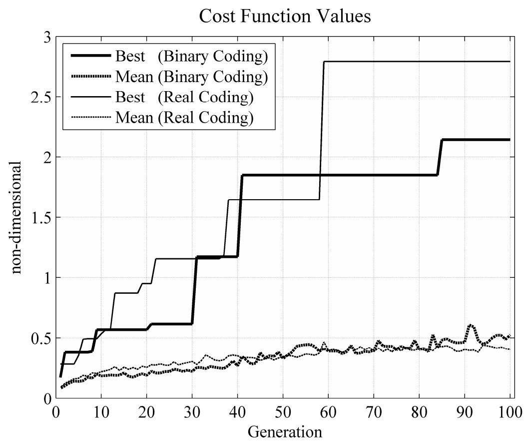

Figure 4: Best cost function values and mean cost function values in successive generations.

Both best cost function value characteristics show a tendency to remain constant over a number of generations, interspersed with sometimes large jumps to a higher value. The succession of generations in which the best cost function value remains the same is due to the elitist reinsertion approach, with the best candidate solutions retained from the previous generation and no better candidates created by recombination or mutation. The jumps in the characteristic result from either random gene mutations or serendipitous random choices of crossover mask during recombination. In both cases the mean cost function values increase by generation, showing a general improvement in the population as the optimization progresses.

Part of the search process of a genetic algorithm is the exploration for possibly better solutions, achieved by maintaining the diversity of the population evaluated during the search, thereby reducing the likelihood of convergence on a local optimum. A measure of the diversity of the population can be made using the mean of the relative variances in each gene over the generations. The relative variance in gene x_n is

$$\mathrm{Var}_{\mathrm{R}}\,(x_n) = \frac{\mathrm{Var}\,(x_n - \hat{\mathrm{x}})}{\hat{\mathrm{x}}} \tag{5}$$

where

$$\hat{\mathrm{x}} = \frac{(x_{n,\max} + x_{n,\min})}{2} \tag{6}$$

The change in mean relative gene variance during the optimization is shown in Fig. 5

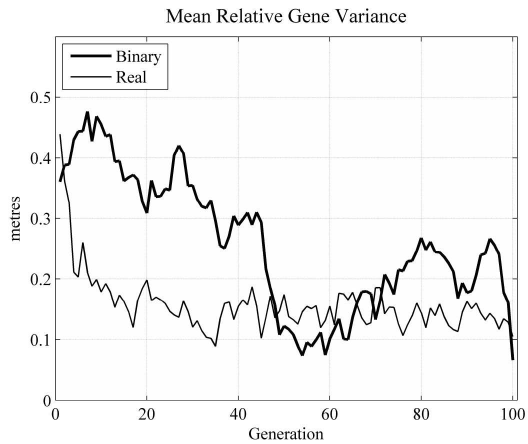

Figure 5: The mean relative gene variances by generation.

In both cases, the mean relative gene variance decreases by generation, indicating loss of diversity as the population improves. This happens earlier with the real-coded algorithm, reflecting the earlier improvement of the population displayed by the mean cost function value characteristic in Fig. 4. After about 60 generations, however, the mean relative gene variance of the binary-coded population increases again, but not sufficiently to find a solution to match the best solution produced by the real-coded algorithm. The best cost function value after 100 generations (C = 2.79) is produced by the real-coded algorithm solution depicted, in front elevation, in Fig. 6, with vertices listed in Table 3. For concision, this solution will be referred to as ‘Solution R’.

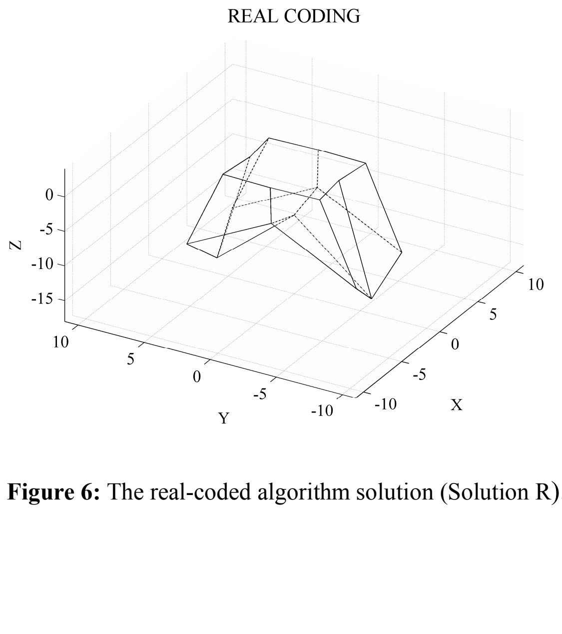

Figure 6: The real-coded algorithm solution (Solution R).

REAL CODING ($\forall = 510.56\text{m}^3$)				
V_1	:	–3.1378	0	0
V_2	:	–3.0102	3.6608	0
V_3	:	–2.9797	6.4162	–11.4639
V_4	:	–3.0168	0	–5.2371
V_5	:	0	3.3618	0
V_6	:	0	5.8442	–15.8005
V_7	:	0	0	–6.6863

Table 3: The real-coded algorithm solution vertices.

The best cost function value after 100 generations produced by the binary-coded algorithm (C = 2.15) is by the solution depicted, in front elevation, in Fig. 7, with vertices listed in Table 4. For concision, this solution will be referred to as 'Solution B'.

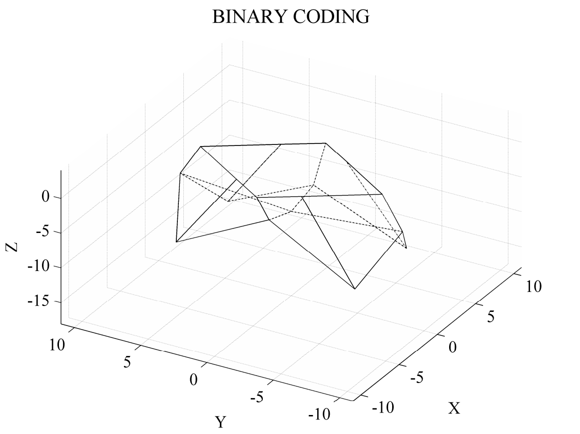

Figure 7: The binary-coded algorithm solution (Solution B).

BINARY CODING ($\forall = 482.71\text{m}^3$)				
V_1	:	–4.4990	0	0
V_2	:	–2.9050	2.4560	0
V_3	:	–3.3770	6.7310	–10.7410
V_4	:	–2.9050	0	–4.6170
V_5	:	0	6.8490	0
V_6	:	0	8.3800	–4.6170
V_7	:	0	0	–5.9380

Table 4: The binary-coded algorithm solution vertices.

Both solutions are similar in being deeper at the sides and shallower along the longitudinal axis, and there is only about 5% difference in the volumes of the two forms. However, solution R is deep and narrow, in contrast to solution B, which is shallower and wider. Also, solution R is deepest along the entire transverse axis, whilst solution B is only deeper on the transverse axis near the centre, being much shallower at the sides. Another major difference is that the sides of solution R are slightly concave, giving it an almost rectangular surface perimeter, whereas the sides of solution B are highly convex, resulting in an almost lozenge-shaped surface perimeter.

Although the rudimentary geometric description and cost function will have an influence on the similarities in, and differences between, the 'best' solutions, the aim of this study is to assess the performance of the two genetic algorithms in this problem. It cannot be ruled out that no overall 'best' solution actually exists, especially in the discrete search space of the binary-coded algorithm, and that the algorithms may only be able to converge on a set of solutions with only insignificantly different cost function values.

No significant difference can be ascertained between the performances of the binary- and real-coded algorithms. In both cases the populations exhibit a tendency to stagnate around one superior solution for a number of generations, until the spontaneous creation of a better solution leads to a change of 'best' solution and a further period of stagnation. The periods of stagnation tend to get longer as the optimization progresses and the diversity of the population declines. Using a ranking method to determine the fitness of the population maintains a certain amount of diversity by reducing the bias towards those members of the population with relatively high cost function values during selection. However, the likelihood of the higher-ranked members being selected as one, or even both, parents appears to remain high. This can be lowered by reducing the selective pressure. An increase in the mutation rate could also lead to greater diversity, but, because of the random nature of the process, this cannot be guaranteed. A major contributor to the reduction in diversity is an elitism strategy in which too many high-ranking members of the population are retained in the subsequent generation. Here the balance between the exploitation and exploration noted in the introduction is to be made.

Since the major part of the computational burden is the derivation of the hydrodynamic parameters for the cost function, the two algorithms are substantially the same in this respect. The binary-coded algorithm can only work in a discrete search space, although, because this is defined in terms of millimetres, there is no real loss in practical terms. Also, the binary-coded algorithm requires number conversion to translate genes into co-ordinates to enable the mesh generation for the hydrodynamic analysis part of the cost function. In comparison, the real-coded algorithm requires a more complicated mutation routine, although, on average, this is only used on a fraction of the genes according to the mutation rate. Even though binary-coded algorithms are by far the most popular form of genetic algorithm in general usage, no discernible advantage over real-coded algorithms was found in the type of problem described in this study.

6 Conclusions

A pilot study into the optimization of the shape of a wave energy collector has been described. The collector moves in two degrees of freedom (surge and pitch), and is symmetric in both the *x-z* and *y-z* planes. Its geometry consists of ruled surfaces, with one quadrant comprising three adjoining faces defined by the positions of seven vertices. Two genetic algorithms are used to find the optimal vertex positions in this study, one with binary-coded genotypes, the other with real-coded genotypes. An elementary cost function, based upon a first-order model of the system, is used to determine the performance of each candidate solution. The candidate collectors are optimally tuned to absorb the greatest power from a number of incident regular waves, the results being weighted according to a generalized occurrence distribution. High velocities and large collector volumes are penalized. Five independent optimization runs of fifty iterations each were made with both algorithms. The optimization was then resumed for a further fifty iterations with the 'best' solutions found with each algorithm. A total of one hundred iterations of each algorithm seems to have been insufficient for convergence on a global optimum (assuming that one exists), as the resulting best solutions displayed more differences than similarities. An appraisal of the performance of the two genetic algorithms in this problem could be made, however.

Generally, the performance of both algorithms during the optimization was essentially the same, each producing the best solution at different stages of the process. The 'improvement' in the respective populations as the optimizations progressed was demonstrated by the increase in the mean cost function values of the entire populations, although at the cost of reduced diversity in the candidate solutions. Both algorithms exhibited a tendency to produce a solution which would remain the best for several generations, with sporadic replacement by progressively better solutions. The intervals between replacements get longer as the optimization progresses, along with the decrease in the diversity of the population. Although the choice of selection method was made with the intention of maintaining population diversity, adjustments to some parameters may be made to increase this. It is more likely that the elitist strategy which retains too many members of the previous generation is the main cause of the problem.

Computationally, the main burden is in calculating the cost function, which is the same for both algorithms, so no apparent benefit is gained from the extra numerical conversions required by the binary-coded algorithm. For problems of the kind described in this study, real-coded genetic algorithms perform equally well and are easier to implement.

Acknowledgements

The work described in this paper was supported by the EPSRC SUPERGEN Marine Energy Research Consortium.

References

[1] A. S. Fraser. Simulation of genetic systems, *Journal of Theoretical Biology*, 2: 329-346, 1962.

[2] L. J. Fogel, A. J. Owens, M. J. Walsh. *Artificial Intelligence Through Simulated Evolution*, John Wiley, Chichester, UK, 1966.

[3] K. De Jong. *Analysis of the behavior of a class of genetic adaptive systems*, Ph.D. Thesis, University of Michigan, Ann Arbor, MI, USA, 1975.

[4] J. R. Koza. Genetic programming: a paradigm for genetically breeding populations of computer programs to solve problems, Rep. No. STAN-CS-90-1314, Stanford University, CA, USA, 1990.

[5] J. H. Holland. *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, USA, 1975.

[6] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, New York, USA, 1989.

[7] S. Rajeev and C. Krishnamoorthy. Genetic algorithms-based methodologies for design optimization of trusses, *Journal of Structural Engineering*, 123(3): 350–358, 1997.

[8] M. Ohsaki. Genetic algorithm for topology optimization of trusses, *Computers and Structures*, 57(2): 219-25, 1995.

[9] C. Camp, S. Pezeshk and G. Cao. Optimized design of two-dimensional structures using genetic algorithm, *Journal of Structural Engineering*, 124(5): 551–559, 1998.

[10] F. Erbatur, O. Hasancebi, I. Tutuncu and H. Kilic. Optimal design of planar and space structures with genetic algorithms, *Computers and Structures*, 75: 209–224, 2000.

[11] M. Schoenauer. Shape representations and evolutionary schemes. In *Evolutionary Programming: Proceedings of the Fifth Annual Conference on Evolutionary Programming*, 121–129, San Diego, USA, 1996.

[12] C. Kane and M. Schoenauer. Topological optimum using genetic algorithm, *Control and Cybernetics*, 25: 1059–1088, 1996.

[13] E. Kita and H. Tanie. Shape optimization of continuum structures by genetic algorithm and boundary element method, *Engineering Analysis with Boundary Elements*, 19: 129-136, 1997.

[14] W. Annicchiarico and M. Cerrolaza. A 3D boundary element optimization approach based on genetic algorithms and surface modeling, *Engineering Analysis with Boundary Elements*, 28: 1351–1361, 2004.

[15] A. Babarit, A. H. Clément and J-C. Gilloteaux. Optimization and time-domain simulation of the SEAREV wave energy converter, In *Proceedings of OMAE2005: The 24th International Conference on Offshore Mechanics and Arctic Engineering (OMAE*

2005), Halkidiki, Greece, June 12th-16th 2005. Ref. No. OMAE2005-67286.

[16] L. N. Patel, A. Murray and J. Hallam. Super-lampreys and wave energy: Optimised control of artificially-evolved, simulated swimming lamprey, *Neurocomputing*, 70: 1139-1154, 2007.

[17] J. E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms and their application*, 14-21, Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1987.

[18] G. Syswerda. Uniform crossover in genetic algorithms. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, 2-9, Morgan Kaufmann Publishers, San Mateo, California, USA, 1989.

[19] H. Mühlenbein and D. Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm, *Evolutionary Computation*, 1(1): 25-49, 1993.

[20] T. Bäck. Optimal Mutation Rates in Genetic Search. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, 2-8, Morgan Kaufmann Publishers, San Mateo, California, USA, 1993.

[21] D. J. Pizer. Maximum wave-power absorption of point absorbers under motion constraints, *Applied Ocean Research*, 15 (4): 227-234, 1993.

[22] J. Falnes. *Ocean Waves and Oscillating Systems*, Cambridge University Press, Cambridge, UK, 2002. ISBN: 0-521-78211-2.

[23] *WAMIT User Manual*, WAMIT Inc., Chestnut Hill, Massachusetts 02467-2504, USA, 2002.