Renewable Energy 51 (2013) 274–284

Contents lists available at SciVerse ScienceDirect

Renewable Energy

journal homepage: www.elsevier.com/locate/renene

Constrained optimization of the shape of a wave energy collector by genetic algorithm

A.P. McCabe*[1,2]

Renewable Energy Research Group, Engineering Department, Faculty of Science & Technology, Lancaster University, Lancaster LA1 4YR, UK

ARTICLE INFO

Article history:
Received 28 June 2012
Accepted 26 September 2012
Available online 25 October 2012

Keywords:
Genetic algorithms
Marine energy conversion
Optimization methods
Wave energy converter design

ABSTRACT

Wave energy extraction requires the conversion of the energy within the waves to drive the power take off system, often by means of a principal interface, or collector. This paper describes part of the development of a robust, systematic method of optimizing the collector shape to improve energy extraction using a genetic algorithm. The collector geometry uses a parametric description based upon bi-cubic B-spline surfaces, generated from a relatively small number of control points to reduce the dimensionality of the search space. The collector shapes that are optimized have one plane of symmetry and move in one degree of freedom (surge). Each candidate shape is assessed in a wave climate based upon data from a site in the North-East Atlantic Ocean. Three cost functions, distinguished by the severity of the penalty put on the size of the candidate collectors, and four constraint regimes, defined by two displacement and two power rating limits, are the governing influences on the twelve optimization procedures described. The selected collector shapes from each optimization run are appraised in terms of size, complexity and their performance compared to that of 'benchmark' box-shaped collectors.

1. Introduction

Effective wave energy extraction requires the conversion of the energy within the waves to drive the power take off system. In a surging wave energy converter, 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 formed part of the research undertaken by the SuperGen Marine Research Consortium (http://www.supergen-marine.org.uk) for 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. In contrast to the largely intuitive process used for most current wave power devices, the procedure is to be as free of human intervention and preconception as possible.

Traditional methods that are used to solve optimization problems in engineering design 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

* Tel.: +44 (0) 1524 592782; fax: +44 (0) 1524 381707.
E-mail address: a.p.mccabe@lancaster.ac.uk.
[1] http://www.engineering.lancs.ac.uk/lureg/people/.
[2] http://www.supergen-marine.org.uk/.

0960-1481/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.renene.2012.09.054

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 the bearing of loads. 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 et al. [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 [17] using an artificial neural network, optimized by genetic algorithms.

This paper describes the use of a genetic algorithm to optimize the shape of a wave energy collector with different cost functions under a range of constraint conditions. The type and structure of the genetic algorithm used is given in Section 2. The conversion of the geometric description of the collector shapes into a form suitable for use by the genetic algorithm is described in Section 3. Section 4 covers the formulation of the cost functions. The results are described and discussed in Section 5, and conclusions are presented in Section 6.

2. The genetic algorithm

The principal characteristic of the genetic algorithm is the concurrent manipulation of a group, or population, of solutions. In many cases the genetic algorithm does not operate directly on the design variables which define the optimization problem, but on coded equivalents which are more amenable to the processes of the algorithm. Many terms have been borrowed from natural genetics to describe the ‘real world’ environment and its representation used by the algorithm.

2.1. Genetic algorithm overview

The ‘real-world’ physical variables are called *phenotypes* and the corresponding variables in the genetic algorithm search space are called *genotypes*. The general genotype form, also called a *chromosome*, consists of a sequence of units, or *genes*. The size of the genotype defines the dimension of the search space. The stability of convergence of the genetic algorithm is unaffected by the size of the genotype [18]. However, in general, the larger the number of genes, the longer is the computation time. To reduce computation times, reduced implementations have been proposed [6,19], but prove less successful when there are a significant number of design variables. In such cases, the performance of the algorithm is improved by the use of a more complicated representation, with real-valued genes. Genotype formulation need not be restricted to just binary-valued or real-valued genes, since the genetic algorithm can handle both types of genes concurrently [20].

The basic genetic algorithm consists of three operations: selection; recombination; and mutation. Selection introduces the Darwinian element to the optimization process. In each iteration of the algorithm, or ‘generation’, every surviving individual is assigned a ‘fitness’ value, which determines the probability of its survival into, or contribution of genetic information toward the next generation of individuals. The fitness value may be determined by the cost function, or by the individual’s ranking within the population. Recombination is the exchange of genetic information from the parents to form offspring for the next generation. 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. The formation of successive generations may include members of previous generations, retained because of their fitness. This part of the management of the population dynamics is called ‘reinsertion’. The random processes built into genetic algorithms mean that there is an almost complete lack of repeatability. The strategy that has been adopted, therefore, is to perform several independent runs of the algorithm for a fixed number of generations, then take the ‘best’ candidates to form the initial population of a final stage, also with a fixed number of generations.

2.2. Genetic algorithm parameters

The algorithm used in this study employs real-number genotypes. The amount of computation required by the cost function far outweighs that required by the genetic algorithm itself, so it is preferable to reduce the number of calls made on the cost function. This leads to the use of a small population with a relatively large mutation rate to increase the population diversity. Recommended population sizes [21] are between one and two times the dimension of the search space, so, taking the lower limit, the population size was set to 22 candidate solutions (see Section 3.2). Selection is ranking-based, using Baker’s linear ranking algorithm [22], with the recommended selective pressure of 2, to determine fitness values. The progenitors of the next generation are selected by the Roulette-Wheel method with Stochastic Universal Sampling [22]. An ‘elitist’ reinsertion approach is used wherein the two fittest members of each generation are retained to form the next generation along with the offspring produced by recombination. The retention of a larger number of elite individuals into the next generation could cause the optimization to stagnate around the elite individuals. Twenty new members of the population are produced by recombination. Ten parents are selected from each generation, each undergoing two random pairings to produce four offspring. Intermediate Recombination [23] is used to produce the