A Detailed Comparison of Meta-Heuristic Methods for Optimising Wave Energy Converter Placements

Mehdi Neshat, Bradley Alexander, Markus Wagner, Yuanzhong Xia
Optimization and Logistics Group, School of Computer Science, The University of Adelaide, Australia
givenname.familyname@adelaide.edu.au

ABSTRACT

In order to address environmental concerns and meet growing energy demand the development of green energy technology has expanded tremendously. One of the most promising types of renewable energy is ocean wave energy. While there has been strong research in the development of this technology to date there remain a number of technical hurdles to overcome. This research explores a type of wave energy converter (WEC) called a buoy. This work models a power station as an array of fully submerged three-tether buoys. The target problem of this work is to place buoys in a size-constrained environment to maximise power output. This article improves prior work by using a more detailed model and exploring the search space using a wide variety of search heuristics. We show that a hybrid method of stochastic local search combined with Nelder-Mead Simplex direct search performs better than previous search techniques.

CCS CONCEPTS

• **Computing methodologies** → *Heuristic function construction*; *Randomized search*;

KEYWORDS

Renewable energy; Evolutionary Algorithms; Position Optimization; Wave Energy

ACM Reference Format:
Mehdi Neshat, Bradley Alexander, Markus Wagner, Yuanzhong Xia. 2018. A Detailed Comparison of Meta-Heuristic Methods for Optimising Wave Energy Converter Placements. In *GECCO '18: Genetic and Evolutionary Computation Conference, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3205455.3205492

1 INTRODUCTION

Wave Energy Converters (WECs) are of interest to governments and industry as a means of complementing other renewable energy sources such as solar and wind-power. WECs have advantages in terms of high availability of resource (over 90%, depending on the location) [5] and wave energy densities of up to 60kW per square meter of water surface in prime locations. Individual WECs in the form of buoys can also be produced to have a high capacity for each unit with current proposals for units with over 1MW each [7] – providing potential for economies of scale. Finally, WEC's have a low impact on aquatic life[13], comparing favourably with other generation technologies.

This study focuses on WECs in the form of fully-submerged buoys. Submerged buoys are one of the most promising and cost-effective technologies for extraction of energy from waves[16]. The buoys in this study are hollow metallic vessels, floating a few meters below the water surface and tethered to the sea floor. Energy is extracted from changes in tension on the tethers as waves propagate through water. Buoys are usually deployed in farms or arrays consisting of multiple buoys. This is done for the reason of amortizing fixed infrastructure cost but also to take advantage of constructive interference between buoys [4]. To maximise the energy returned by a WEC farm buoys must be placed to exploit prevailing wave conditions, maximise constructive interference between buoys, and minimise destructive interference.

The interactions between buoys in a farm are complex, extensive, and dependent on local conditions. As a consequence there is, as yet, no simple recipe for buoy placement. Research to date on farm design has primarily focused on the placement of semi-submerged arrays [2]. Research on placement of fully-submerged arrays [16] has applied two popular evolutionary algorithms, the (1+1)EA [1] and CMA-ES [8]. This found that a (1+1)EA with simple mutation performed better than CMA-ES. However, this earlier work used a greatly simplified environmental model with just one wave direction and few wave frequencies. The current paper improves on prior work substantially in the following ways: deploying a more realistic and practical model with 50 wave frequencies and seven different wave directions; comparing a much broader range of heuristic search techniques adapted to functioning with a small number of function evaluations; exploring the use of surrogate functions in a partial evaluation framework[3]; and conducting a preliminary investigation of the local landscape for buoy placement. As a fair means of comparison, we examine how various frameworks perform within the context of a limited (but realistic) computational budget. Through this comparison we show that a hybrid search consisting of stochastic local search combined with downhill search outperforms previously published methods in terms of performance for 16-buoy array layouts. We also describe layouts resulting from these runs.

The remaining sections of the paper are organised as follows. In the next section we describe the buoy model. The optimisation problem is defined in Section 4 and the search methods to be compared are briefly described in Section 3. Section 5 presents experimental results and finally, Section 6 discusses these results and canvases future work.

GECCO '18, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5618-3/18/07…$15.00
https://doi.org/10.1145/3205455.3205492

Table 1: Key parameters for WECs simulated in this work

Buoy number	4, 16
Buoy radius	5m
Submergence depth	3m
Water depth	30m
Buoy mass	376 tonnes
Buoy volume	$523.60m^3$
Tether angle	55°

2 MODEL FOR WAVE ENERGY CONVERTERS (WECS)

This research considers a model for a WEC consisting of a fully submerged three-tether buoy. Each tether is anchored to a generator placed on the sea floor. The anchors are assumed to be placed in a triangular pattern below each buoy in a configuration that optimises the transmission of energy from heave and surge wave motions in the waves, through to the generators [10]. Table 1 gives relevant details of the WECs modelled in this work.

2.1 System dynamics and parameters

The WEC model calculates the energy output of WEC based on a formula of dynamics[11] with three principle force components:

1. The force of wave excitation ($F_{exc,p}(t)$) incorporates the forces of incident and diffracted waves when all converters are in a fixed position. 入射波和衍射波的力
2. The force of radiation($F_{rad,p}(t)$) describes the force of an oscillating body independent of incident waves. 与入射波无关的振荡体的力
3. Power take off force($F_{pto,p}(t)$) is the force applied to the buoys through their tethers.

Because oscillating buoys exert a force on the surrounding water they can interact with each other at distance. Buoys can interact not only destructively but also constructively, depending on their relative angles and distances, and depending on the surrounding sea conditions. In a buoy array the power accruing to a buoy number p is characterised by Equation 1.

$$M_p\ddot{X}_p(t) = F_{exc,p}(t) + F_{rad,p}(t) + F_{pto,p}(t) \tag{1}$$

where M_p is the displacement of the p_{th} buoy, $\ddot{X}_p(t)$ is a vector of body acceleration in heave, sway and surge. The final term, describing the power take-off system, is simulated as a linear damper and spring. For each mooring line two control factors are applied: the coefficient of damping B_{pto} and stiffness K_{pto}. Thus the extended version of Equation (1) for all converters is:

$$((M_\Sigma + A_\sigma(\omega))j\omega + B_\sigma(\omega) - \frac{K_{pto,\Sigma}}{\omega}j + B_{pto,\Sigma})\ddot{X}_\Sigma = \hat{F}_{exc,\Sigma} \tag{2}$$

where $A_\Sigma(\omega))$ and $B_\Sigma(\omega)$ are hydrodynamic parameters which are derived from the semi-analytical model based on [15]. In addition,$K_{pto,\Sigma}$ and $B_{pto,\Sigma}$ are control coefficients which are tuned to provide the maximum level of isolated buoy power absorption. In the following, two performance measures are described. To compute the total power output of the layout, we utilise Equation (3):

$$P_\Sigma = \frac{1}{4}(\hat{F^*}_{exc,\Sigma}\ddot{X}_\Sigma + \ddot{X^*}_\Sigma\hat{F}_{exc,\Sigma}) - \frac{1}{2}\ddot{X^*}_\Sigma B\ddot{X^*}_\Sigma \tag{3}$$

The second important performance measure used here is the the q-factor (q) of the array. q measures the efficiency of an entire array of N as compared power output from each buoy taken in isolation. q is defined in Equation (4) as:

$$q = \frac{P_\Sigma}{N \cdot P_0} \tag{4}$$

In favorable circumstances $q > 1$ due to constructive interference, even though the buoys extract energy from the waves. In this work we aim to maximise the total power output: P_Σ of an array of a given size N within a constrained farm area. Because each buoy in the array is identical the corresponding q-factor is easily derived from the total output.

3 OPTIMISATION SETUP

The optimisation problem here can be stated as:

$$P_\Sigma^* = argmax_{\mathbf{x},\mathbf{y}} P_\Sigma(\mathbf{x}, \mathbf{y})$$

where $P_\Sigma(\mathbf{x}, \mathbf{y})$ is the average power obtained by placements of the buoys in a field at x-positions: $\mathbf{x} = [x_1, \ldots, x_N]$ and corresponding y positions: $\mathbf{y} = [y_1, \ldots, y_N]$. In the experiments here $N = 16$.

Constraints. All buoy positions (x_i, y_i) are constrained to a square field of dimensions: $l \times w$ where $l = w = \sqrt{N * 20000}\, m$. This gives $20000m^2$ of farm-area per-buoy. In addition buoys are required to maintain a safety distance of at least 50 metres from each other. For any layout $\mathbf{x}, \mathbf{y}$ the sum-total of the inter-buoy distance violations, measured in metres, is:

$$Sum_{dist} = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(dist((x_i, y_i), (x_j, y_j)) - 50),$$
$$\text{if } dist((x_i, y_i), (x_j, y_j)) < 50 \text{ else } 0$$

where $dist((x_i, y_i), (x_j, y_j))$ is the L2 (Euclidean) distance between buoys i and j. The penalty applied to the power output (in Watts) is $(Sum_{dist} + 1)^{20}$. This penalty is steep but continuous which allows better handling of constraint violations during search.

Buoy placements which are outside of the farm area are handled by repeating the placement process.

Computational Resources. This study aims to compare a diverse set of search methods in a realistic buoy-layout optimisation setting. The setting here assumes a limited computational time budget of three days on a moderately high performance shared-memory parallel platform. In this study the hardware platform are compute nodes with 2.4GHz Intel 6148 processors and with 128GB of RAM. In terms of software, the meta-heuristic frameworks as well as the evaluative function for $P_\Sigma(\mathbf{x}, \mathbf{y})$ were run in MATLAB R2017. The used Matlab license allows us to run 12 worker threads in parallel.

For each heuristic search method, we exploit parallel processing by either evaluating individual layouts in a population in parallel or by evaluating all wave frequencies in parallel. The dimension of parallelism chosen was determined according to which gave the best performance for each search method. In both cases, if there are enough frequencies or individuals to make use of the parallel worker threads, up to ten-fold speedups were achieved.

It should be emphasised that we are not comparing search methods for buoy placement by simply counting evaluations of $P_\Sigma(\mathbf{x}, \mathbf{y})$. This is because the computational cost of each evaluation varies greatly between search methods depending on the number of frequencies considered; similarly, run times vary with the number of buoys in the layout. We ran the experiments on dedicated compute resources to minimise the variance of the number of evaluations between runs of the same method and thus avoid bias due to noise or resource contention. In these experiments the standard deviation in the number of evaluations between trials of the same search method is less than 5%. While this deviation might seem substantial at first, we shall later see that the algorithms either tend to converge well before the computation budget is used up, or their performance variance is not significant.

The computation budget for each single optimisation run is three days (72 hours) using 12 worker threads. In practice, this can give engineers two rounds of what-if analyses per week.

4 META-HEURISTIC SEARCH METHODS

We list the search methods compared in this study in Table 3. All methods are run with the computational resources described in the previous section and each method is run for ten times, with the best output produced by each framework measured at the end of the trial. The bulk of experiments were run for $N = 16$ buoys, although we have also conducted experiments with $N = 4$ buoys. The dimension of parallelism used in each is specified in the second column of Table 2.

Describing the table in row-order: Random Search (R-S) places buoys at random across the search field; $\text{PE}_{50,\mu}$ and $\text{PE}_{f,\mu}$ are partial-evaluation searches (see Section 4.1) evaluating solutions (in tournaments) on randomly selected subsets of unique frequencies; TDA is an algorithm for placing wind-turbines described in [14]. CMA-ES applies CMA-ES to all the dimensions of the search problem with a population determined by the formula in that row; CMA-ES (2+2) is one of the two major buoy placement search methods in [16]. DE is differential evolution with population 50 and three different values for the Pcr parameter ($Pcr \in \{0.3, 0.5, 0.7\}$); $(1{+}1)\text{EA}_\sigma$ and $(1{+}1)\text{EA}_s$ mutate one buoy’s location at a time using either a normal distribution ($\sigma = 100m$) or a uniform distribution ($[0, s]$) respectively ; $(1{+}1)\text{EA}_{\text{Linear}}$ uses a mutation step size that decreases linearly [6][1]; $(1{+}1)\text{EA}_{1/5}$ uses a step size that becomes larger if more than 1/5th of the steps are successful in improving fitness and it reduces the step size if less than 1/5th of steps are successful; Iterative-(1+1)EA is an iterative algorithm (see Section 4.2) for one-at-a-time buoy placement; $\text{LS+NM}_{\text{allDims}}$ is a hybrid-search (see Section 4.3 for all hybrid methods) which follows stochastic buoy placement with optimisation by the Nelder-Mead (NM) simplex direct search [9]; NM_Norm_{2D} and NM_Unif_{2D} are the same as $\text{LS+NM}_{\text{allDims}}$ but it uses NM search to refine buoy positions one at a time rather than all-at-once; $LS_1 + NM_{2D}$ alternates stochastic placement and NM search; finally, $LS_3 + NM_{2D}$ conducts a three-sample local search for each buoy placement followed by NM search. Short descriptions of the more specialised search methods listed above follow.

4.1 Partial Evaluation

Partial Evaluation [3] saves evaluation time by evaluating the fitness of an individual just partially. In our work we applied partial evaluation with randomly selected subsets of frequencies in each generation, where the number of such frequencies is fixed for the duration of the run. We used the non-elitist $\mu + \lambda$EA in [3] as the framework for driving evolution. Note that, because fitness is assessed on partial information it is necessary to include a single generation at the end of the process where each individual layout is evaluated at all frequencies so the best-performing individual can be selected. The cost of this last generation depends on the population. For $\mu = 100$, this time is substantial and 12 hours must be allocated at the end, leaving 2.5 days to run the actual PE search algorithm. Proportionately less time is needed for smaller populations. In the meantime, two kinds of mutations are used. Firstly, the position of buoys are mutated based on uniformly distributed random numbers in a circle ($r = l/16$) with a radius of 18(m) and 35(m) for 4 and 16 buoys respectively. Secondly, a normal distribution is employed for resampling the buoys location with $\sigma = 10(m)$ $(PE - N)$.

4.2 Iterative 1+1EA

In contrast to the other (1+1)EA algorithms described in Table 2 the Iterative (1+1)EA method positions buoys one after the other. Each buoy is placed using a (1+1)EA-like search starting from the previously placed buoy. Step size decreases linearly during search (see Equation 5). For each buoy the search stops either when the new buoy has a q-factor of $\leq$ 1.0, or when a preset number of mutation steps is reached. The latter is done in order to limit the time spent in the local search as further buoys remain to be placed.

4.3 Hybrid Search

In pursuit of a more informed search heuristic, a brief study was conducted to sample the marginal energy gain resulting from adding a new buoy to the neighbourhood of buoys that have already been placed. Figure 1 shows the results of this landscape analysis for placing a fourth buoy near three previously placed buoys. Areas of high energy output are shown in yellow, while the blue chasms represent closeness constraint violations.[2]

Two important properties are apparent from these graphs. First, is that the landscape, though multi-modal, is smooth. This means that a search with a local search component may be beneficial. The second property is that, due to positive reinforcement effects, peak energy output is often in the neighbourhood of previously placed buoys. This indicates that it might be good to start the search near a previously placed buoy.

These observations have informed the design of the last five search methods in Table 2. The first of these is $\text{LS+NM}_{\text{allDims}}$, described in Algorithm 1.

This algorithm repeatedly adds buoys at random offsets from the previous one followed by a Nelder-Mead local search on all buoy positions. The Nelder-Mead local search is limited to 10 iterations so that the outer **while** loop has time to build and test repeated configurations until the time budget for buoy placement runs out. Inside the **for** loop the buoys are placed one at a time with each successive buoy being placed at a distance, sampled from a normal distribution, from the previous buoy. In this algorithm the normal

[1] $Mutation - step_{size} = (Initialstep_{size}) * (1 - 0.92 * iter/Max_{iter})$ (5)

[2] In fact, the actually underlying 4-buoy layout is the result of comprehensive 4-buoy layout optimisations. For each of the four figures, one buoy was removed and then the landscape mapped using a grid search. This figure confirms that the underlying layout was indeed a local optimum with respect to single-buoy mutations.

Table 2: Summary of the search methods used in this paper. All methods are given the same computational budget. Parallelism can be expressed as per-individual or per-frequency depending on the number of individuals in the population (see Section 3).

Abbreviation	Parallelism	Description
R-S	per-frequency	Random Search
$PE_{50,\mu}$	per-individual	Partial Evaluation[3], all frequencies (PE_{Full}), population $\mu \in \{10, 50, 100\}$
$PE_{f,\mu}$	per-individual	Partial Evaluation [3], partial frequencies, $f \in \{1, 4, 16\}$, $\mu \in \{10, 50, 100\}$
TDA	per-individual	Algorithm for optimising wind turbine placement from [14]
CMA-ES	per-individual	CMA-ES[8] all dimensions, $\mu =' 4 + int(3 * log(D))$ndim , $\sigma = 0.17 * Area$
CMA-ES (2+2)	per-individual	Setup for CMA-ES from [16], $\sigma = 20m$
$CMA-ES_{PF}$ (2+2)	per-frequency	All settings are based on [16]
DE_{Pcr}	per-individual	Differential evolution [12], $\mu = 50$, $F = 0.5$, $P_{cr} \in \{0.3, 0.5, 0.7, 0.9\}$
$1+1EA_{\sigma}$	per-frequency	1+1EA(all dimensions), mutation step size with $\sigma \in 3, 10, 30$(m)
$1+1EA_{s}$	per-frequency	1+1EA (all dimensions) with uniform mutation in range $[0, s]$ with $s = 30$ from [16]
$1+1EA_{Linear}$	per-frequency	1+1EA (all dimensions) with linearly decaying mutation step size [6]
$1+1EA_{1/5}$	per-frequency	1+1EA (all dimensions) with adaptive step size [6]
Iterative 1+1EA	per-frequency	Iterative local search - buoys are placed in sequence using best of local neighborhood search, $\sigma = 100(m)$ for inserting the new buoy, Mutation step size= $(l/10)$ decreased lineally (Eq.5), Stopping Criteria for optimising each buoy based on power and number of mutations
$LS+NM_{allDims}$	per-frequency	Local sampling + Nelder-Mead search in all Dimensions
NM_Norm_{2D}	per-frequency	Buoys placed in sequence using Nelder-Mead search, Initial placement normally distributed from last buoy position, MaxFunEvals=30, for inserting the new buoy $\sigma = 100(m)$
NM_Unif_{2D}	per-frequency	Buoys placed randomly and then refined using Nelder-Mead Initial placement uniformly distributed from last buoy position, MaxFunEvals=30.
$LS_1 + NM_{2D}$	per-frequency	Local Sampling + Nelder Mead search. Buoys placed at random offset from previous buoy and placement refined by Nelder-Mead search. [9], Stopping criteria for NM for optimising added buoy (Tolerance=0.1% $*$ $Power$), $\sigma = 100m$ (inserting buoys) and step size based on Equation 5
$LS_3 + NM_{2D}$	per-frequency	Repeated local sampling + Nelder Mead search. Placements sampled at three random offsets from previous location, best placement used as starting point for Nelder-Mead search.

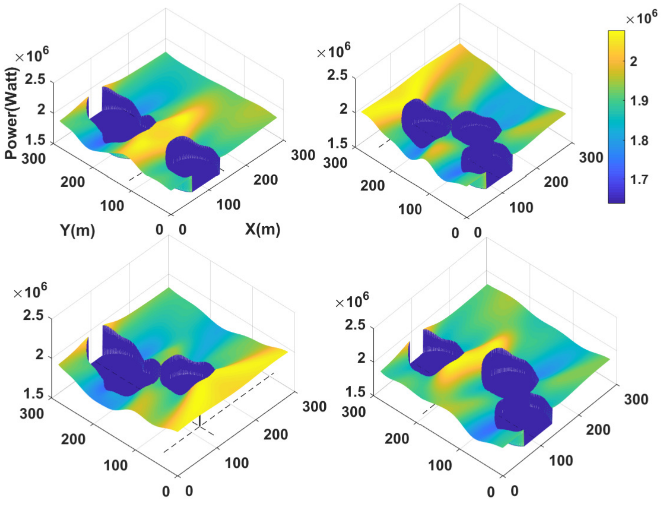

Figure 1: The wave farm's power landscape for the insertion of the last buoy of 4-buoy layout into locations across the farm area. Dashed lines show the locations of the local optima for adding a fourth buoy.

distribution has $\sigma = 100m$, which is an educated guess informed by the landscape mapping in Figure 1. Note that, for this algorithm, the *Eval* function is parallelised on a per-frequency basis.

The next two search methods in Table 2 are: NM_Norm_{2D} and NM_Unif_{2D} are greedy algorithms that, like $LS+NM_{allDims}$, place buoys one at a time at a random offset from the previous buoy. However, in these algorithms the *NM_Search* is run to optimise each buoy position before proceeding to the next buoy placement. The time budget for each *NM_Search* phase is: $3days/N$ so that there is equal time devoted to each buoy placement. Note that in this algorithm the call: $Eval_{([x_1,\ldots,x-i],[y_1,\ldots,y-1])}$ is implicitly passed the arguments for the buoys placed to date so that it can evaluate the new buoy position $[x_i, y_i]$ with respect to these. Also note that, due to the shorter evaluation time for smaller numbers of buoys this equal time allocation results in more search iterations for earlier buoys which serves as a good foundation for the rest of the search. The algorithm for NM_Norm_{2D} (normally-distributed offset $\sigma = 100m$) is shown in Algorithm 2. NM_Unif_{2D} (uniformly-distributed offset in range $[0, size]$) differs from this only in the sampling approach.

The last two search methods in Table 2 are: $LS_1 + NM_{2D}$ and $LS_3 + NM_{2D}$. The algorithm for $LS_3 + NM_{2D}$ is shown in Algorithm 3. This algorithm makes three samples of the neighbourhood surrounding the last buoy and conducts *NM_Search* from the sampled point giving the highest energy. The stopping condition for *NM_Search* is

Algorithm 1 LS+NM$_{\text{allDims}}$

```
procedure LOCAL SAMPLING + NELDER-MEAD SEARCH (ALL DIMS)
Initialization
    $size = \sqrt{N * 20000}$                          ▷ Farm size
    $\mathbf{x} = [x_1, \ldots, x_N] = \perp$          ▷ x-positions
    $\mathbf{y} = [y_1, \ldots, y_N] = \perp$          ▷ y-positions
    $lastx{=}size/2;\ \ lasty{=}0$                      ▷ first buoy position
    $bestEnergy = 0$                                    ▷ Best energy so far
    $bestLayout = [\mathbf{x}, \mathbf{y}]$             ▷ Best layout so far
search
    while $stillTime()$ do                              ▷ Iterative search
        for $i$ in $[1, .., N]$ do
            while not valid $(\mathbf{x}, \mathbf{y})$ do
                $x_i = randn(\sigma) + lastx$           ▷ new buoy position
                $y_i = randn(\sigma) + lasty$           ▷ new buoy position
            end while
            $lastx{=}\ x_i;\ lasty{=}\ y_i$             ▷ Update last buoy position
        end for
        $([\mathbf{x}, \mathbf{y}], energy){=}\ NM\_Search(Eval, [\mathbf{x}, \mathbf{y}])$ ▷ Local search
        if thenenergy $> bestEnergy$                    ▷ If better?
            $bestEnergy = energy$                       ▷ Update energy
            $bestLayout = \mathbf{x}, \mathbf{y}]$      ▷ Update layout
        end if
    end while
    return $bestLayout$                                 ▷ Final Layout
end procedure
```

Algorithm 2 NM_Norm_{2D}

```
procedure NELDER-MEAD SEARCH (2 DIMS)
Initialization
    $size = \sqrt{N * 20000}$                          ▷ Farm size
    $\mathbf{x} = [x_1, \ldots, x_N] = \perp$          ▷ x-positions
    $\mathbf{y} = [y_1, \ldots, y_N] = \perp$          ▷ y-positions
    $lastx{=}size/2;\ \ lasty{=}0$                      ▷ first buoy position
search
    for $i$ in $[1, .., N]$ do
        while not valid $(\mathbf{x}, \mathbf{y})$ do
            $x_i = randn(\sigma) + lastx$               ▷ new buoy position
            $y_i = randn(\sigma) + lasty$               ▷ new buoy position
        end while
        $([x_i, y_i], energy){=}$
            $NM\_Search(Eval_{([x_1, \ldots, x_{i-1}], [y_1, \ldots, y_{i-1}])}, [x_i, y_i])$
        $lastx{=}\ x_i;\ lasty{=}\ y_i$                 ▷ Update last buoy position
    end for
    return $[\mathbf{x}, \mathbf{y}]$                   ▷ Final Layout
end procedure
```

also different from previous algorithms with a stopping tolerance of 0.1% in the energy output. Compared to earlier approaches, this *NM_Search* configuration devotes relatively little time to the search for early buoy placements, which tend to converge fast, and more to the later buoy placements which converge slowly. Note that the stopping tolerance was tuned to make sure the algorithm's

Algorithm 3 $LS_3 + NM_{2D}$

```
procedure LOCAL SAMPLING + NELDER-MEAD SEARCH (2 DIMS)
Initialization
    $size = \sqrt{N * 20000}$                          ▷ Farm size
    $\mathbf{x} = [x_1, \ldots, x_N] = \perp$          ▷ x-positions
    $\mathbf{y} = [y_1, \ldots, y_N] = \perp$          ▷ y-positions
    $lastx{=}size/2;\ \ lasty{=}0$                      ▷ first buoy position
search
    for $i$ in $[1, .., N]$ do
        $iters = 3$                                     ▷ Number of local samples
        $bestx = 0;\ besty = 0;\ bestEnergy = 0$
        for $j$ in $[1, .., iters]$ do
            while not valid $(\mathbf{x}, \mathbf{y})$ do
                $x_i = randn(\sigma) + lastx$           ▷ new buoy position
                $y_i = randn(\sigma) + lasty$           ▷ new buoy position
            end while
            $energy = Eval([x_1, \ldots, x_{i-1}, x_i, y_1, \ldots, y_{i-1}, y_i])$
            if $energy > bestEnergy$ then
                $bestx = x_i;\ besty = y_i$
                $bestEnergy = energy$
            end if
        end for
        $([x_i, y_i], energy){=}$
            $NM\_Search(Eval_{([x_1 \ldots x_{i-1}], [y_1 \ldots y_{i-1}])}, [bestx, besty])$
        $lastx{=}\ x_i;\ lasty{=}\ y_i$                 ▷ Update last buoy position
    end for
    return $[\mathbf{x}, \mathbf{y}]$                   ▷ Final Layout
end procedure
```

running time is close to three days. The $LS_1 + NM_{2D}$ is identical to $LS_3 + NM_{2D}$ but with $iters = 1$.

5 EXPERIMENTS

In this section, we report on the results of our experiments. The search methodologies can be divided into single-solution and population-based methods. In the latter group the sizes of populations used vary from 2 to 100 depending on the algorithm.

Figures 2 and 3 show box-and-whiskers plots for the power output of the best individuals resulting from all the configurations of the all the search heuristics shown in Table 2 for determining well-performing 16-buoy layout. Note that, Figure 3 is a subplot of Figures 2 showing the outputs for all the variations of PE. The PE variations shown in Figure 2 are full-frequency evaluation variants of the $\mu + \lambda$ algorithm used for PE with uniform and normally distributed mutation, respectively.

The first observation from both figures is that the differences in ~~the mean output attained by all methods is less than 20%.~~ This shows that even the most naive search methods are able to obtain non-trivial power outputs. ~~The second observation is that with the limited number of function evaluations at hand highly adaptive search heuristics such as CMA-ES and DE only perform moderately well.~~ One potential reason for this is that small number of evaluations possible, in the order of 300 full evaluations of 16 buoy layouts in three days, gives little time for these methods to learn

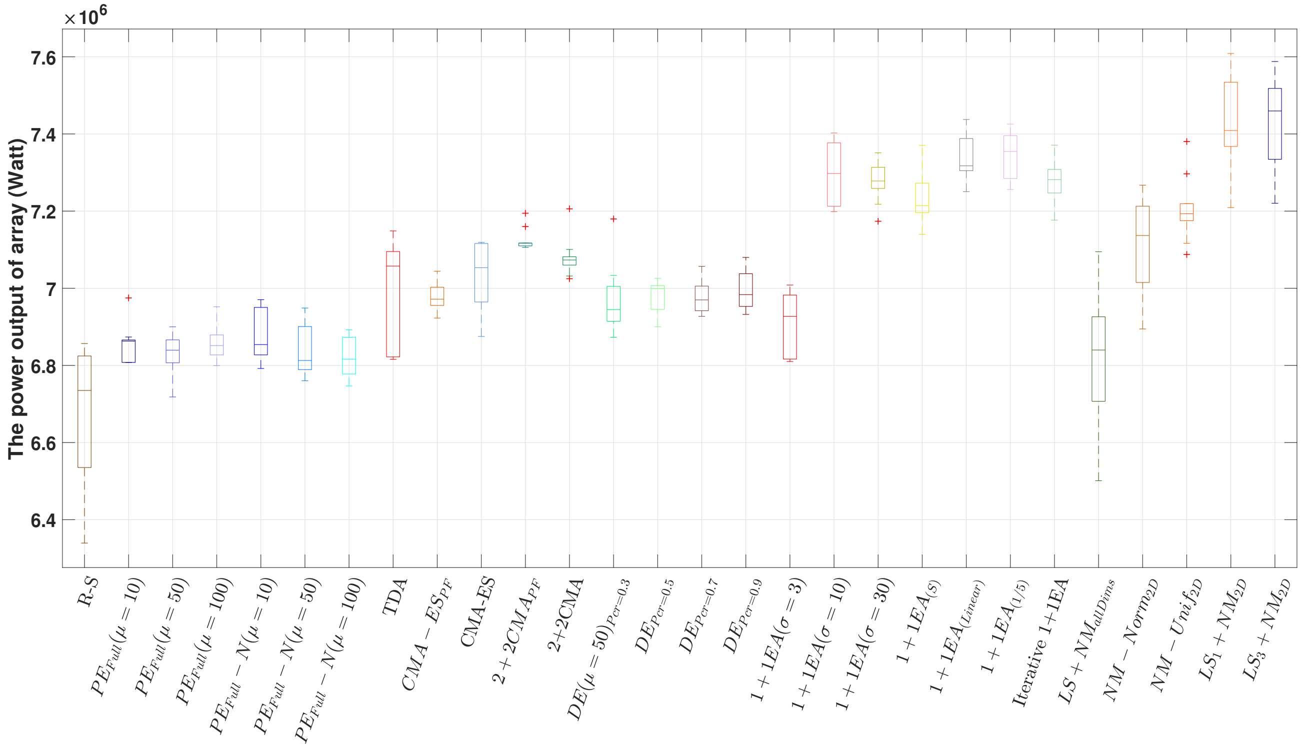

Figure 2: The comparison of the all proposed ideas results from 16-buoy layout in terms of the best layout per each experiment. With regard to the median performance , $LS_3 + NM_{2D}$ can overcome other methods .

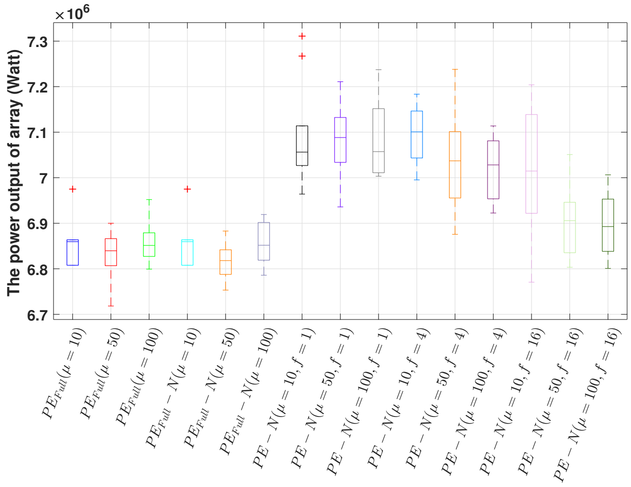

Figure 3: The optimisation results of Partial Evaluation method with three population sizes:$\mu = 10, 50, 100$ and different wave frequencies are used (1, 4, 16 and 50(f)) for 16-buoy layout.

the search landscape.[3] Another observation is that the (1+1)EAs and the buoy-at-a-time placement algorithms (with local search) all perform well. The best performing algorithms are the $LS_1 + NM_{2D}$ and $LS_3 + NM_{2D}$ which are hybrid searches with settings informed by the landscape. Of these two, the $LS_3 + NM_{2D}$, which does the local sampling appears to have a slightly higher mean performance but the difference is not significant with this sample size. The best performance overall of 7608600 Watts is given by one of the runs of $LS_1 + NM_{2D}$.

Examining the PE methods in Figure 3, it appears that variants with lower number of frequencies sampled seem to perform better. These variants are able to perform many more evaluations than those sampling higher numbers of frequencies, at the cost of having a less informed and more noisy evaluative function. From both figures it appears that there is no clear advantage accruing to methods with larger population sizes. This is likely to be a product of the limited number of evaluations available. Overall there seems to be an advantage in evaluating on fewer frequencies and using a smaller population.

To examine how the various search methods converged the average fitness of the best individuals in each population were recorded for each method. These results are plotted in Figures 4 for partial evaluation and 5 for all others. Note that, in both sets of plots the averages were obtained by fully evaluating the population at the sampled time and extracting the best performing individual for that run — in case of PE, this happened in post-processing. The top row of Figure 4 is ordered by the number of frequencies. As can

[3] Early experiments with four buoy layouts – which allow thousands of evaluations – show CMA-ES performing at least as well as other methods.

be seen there is a clear decrease in the speed of optimisation as the number of sampled frequencies increases. Moreover the relative advantage in speed of optimisation for small populations becomes more marked for more evaluated frequencies. In the second row, ordered by population, the speed of evolution is highest for the lowest population but starts off a lower base.

In Figure 5 the distinct groups of algorithms are observable. The PE full frequency heuristics start with relatively good performance but have relatively flat fitness curves. Next the CMA-ES variants progress quickly from a low base and then flatten out in performance. The DE and 1+1EA variants, respectively, follow smoother and higher curves. Finally, the $LS_1 + NM_{2D}$ starts off a very low base (below the x-axis) and steps up steeply with initial buoy placements followed by Nelder-Mead search (the shallow-sloping steps). The overall result of this hybrid algorithm is slightly better overall than the other methods.

Finally, the layout of wave-buoy's produced by the algorithms offers some interesting insight into the features of these highly productive individuals. Figure 6 shows the most productive individual layout found in all the search runs. This layout is built by the algorithm from the x-axis upwards with buoys numbered in the figure in order of placement. It is clear that the initial placement order forms an almost straight diagonal line from the bottom sloping upwards to the right. The buoys then start to slope leftwards toward the front. These placement make sense in terms of placement of adjoining buoys in the peaks of the power landscape. Note that buoy 8 is placed in front of the others which reduces the energy output of the buoys behind before buoy 9 and 10 are placed in the original diagonal pattern. At this point, options that do not interfere negatively with other buoys in this layout are exhausted so a second front of buoys has started to form that alternates in the y-dimension with the original front so as to minimise the impact of negative interference. It should be noted that this zig-zag pattern of farm layout is observable in results of many of the high-performing runs. Another feature common to many runs is the formation of the second row of buoys, often started before first row is complete. It is not clear if the early formation of this second row is an artifact of stochastic nature of the hybrid search heuristics or there are fundamental properties of the problem that drive this behaviour, at least in constrained environments.[4]

6 CONCLUSIONS

In this investigation, several evolutionary optimisation algorithms are applied and evaluated for maximising the total captured power of 16 buoy layouts using an improved and detailed evaluative function. The optimisation environment is challenging, with a very limited number of full evaluations possible within the evaluation budget. Because the algorithms explored have diverse behaviour in terms of evaluative costs algorithms were compared in the realistic scenario of searching within a generous time budget on a multi-core machine.

The methods that performed best were hybrids of stochastic buoy placements and uphill local search. One advantage of these search strategies is the one-at-a-time buoy placements reduced the dimensionality of the search space to just the next buoy. A potential disadvantage of this greedy placement approach is that it allows no backtracking to improve the positions of previously placed buoys. However, preliminary experiments with global optimisation of these best buoy layout have yielded very little improvement, indicating that substantial improvement will involve more than simply tuning the discovered layout.

This work also explored partial evaluation by frequency and showed that a small number of frequencies and a small population yielded the best results in terms of search but still less effective overall than other methods.

Finally from many observations of different optimal layouts and analysing the landscapes of the farms, it appears that a positive hydrodynamic interaction can be obtained if buoys are placed at a relative angle of approximately 45 degrees. This observation might be exploited in the initialisation phase.

This work can be carried in several potential directions. First new, more informed hybrid algorithms can be developed. It may be possible to combine smarter initialisation with iterative local search. Variants of partial evaluation can be used that evaluate on the energy from a sample of buoys rather than frequencies. If carefully designed such an algorithm may allow the productive use of crossover as a way of combining individuals with complementary partial fitnesses. There is also scope to apply this work to an even more refined model with more wave directions and non-uniform water depth. Finally, the optimisation can be extended to incorporate a cost model based on sharing tether points, accounting for the different tether angles and tether lengths that this analysis would entail.

Our code, layouts, and auxiliary material are publicly available: https://cs.adelaide.edu.au/~optlog/research/energy.php

Acknowledgements. We would like to thank Boyin Ding, Frank Neumann, and Nataliia Sergiienko for their input. Our work was supported by the Australian Research Council project DE160100850.

REFERENCES

[1] Hans-Georg Beyer and Hans-Paul Schwefel. 2002. Evolution strategies–A comprehensive introduction. *Natural computing* 1, 1 (2002), 3–52.
[2] BFM Child and Vengatesan Venugopal. 2010. Optimal configurations of wave energy device arrays. *Ocean Engineering* 37, 16 (2010), 1402–1417.
[3] Duc-Cuong Dang and Per Kristian Lehre. 2016. Runtime analysis of non-elitist populations: From classical optimisation to partial information. *Algorithmica* 75, 3 (2016), 428–461.
[4] AD De Andrés, R Guanche, L Meneses, C Vidal, and IJ Losada. 2014. Factors that influence array layout on wave energy farms. *Ocean Engineering* 82 (2014), 32–41.
[5] Benjamin Drew, Andrew R Plummer, and M Necip Sahinkaya. 2009. A review of wave energy converter technology. (2009).
[6] Aguston Eiben, Zbigniew Michalewicz, Marc Schoenauer, and Jim Smith. 2007. Parameter control in evolutionary algorithms. *Parameter setting in evolutionary algorithms* (2007), 19–46.
[7] Carnegie Clean Energy. 2017. CETO 6 Design Update @ONLINE. (2017). https://s3-ap-southeast-2.amazonaws.com/website-sydney-1/media/2017/11/13134305/1738756.pdf
[8] Nikolaus Hansen. 2006. The CMA evolution strategy: a comparing review. *Towards a new evolutionary computation* (2006), 75–102.
[9] Jeffrey C Lagarias, James A Reeds, Margaret H Wright, and Paul E Wright. 1998. Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM Journal on optimization* 9, 1 (1998), 112–147.
[10] JT Scruggs, SM Lattanzio, AA Taflanidis, and IL Cassidy. 2013. Optimal causal control of a wave energy converter in a random sea. *Applied Ocean Research* 42

[4] In experiments with four buoys there is no formation of a second front.

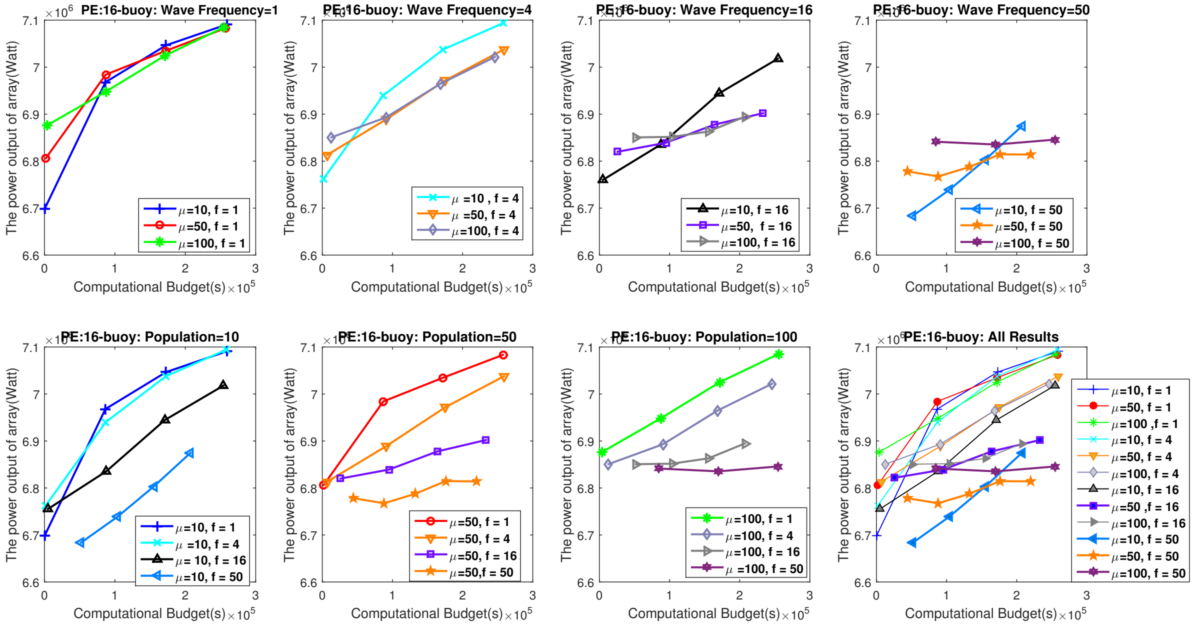

Figure 4: The PE performance comparison of different number of wave frequency (1, 4, 16 and 50) with three size of populations (μ =10, 50 and 100) results from 16-buoy layout based on the average computational time.

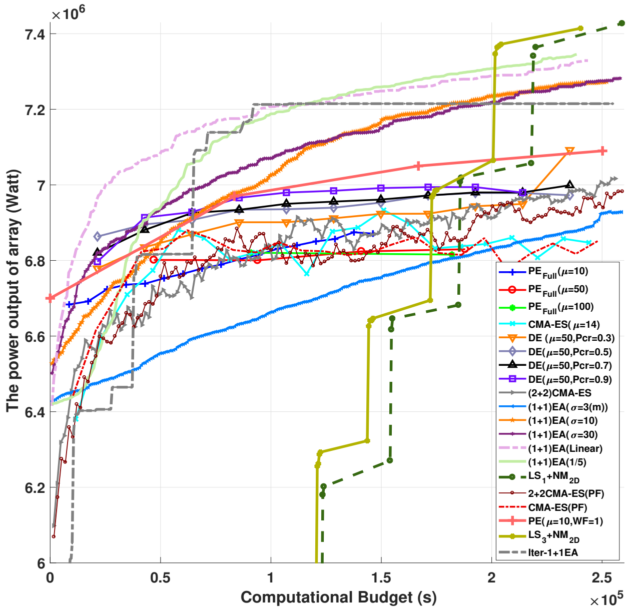

Figure 5: A comparison of the average computational budget of proposed methods for 16-buoy layout over 72 hours.

(2013), 1–15.

[11] N Yu Sergiienko. 2016. Frequency domain model of the three-tether WECs array. (2016).

[12] Rainer Storn and Kenneth Price. 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* 11, 4 (1997), 341–359.

[13] Tom W Thorpe et al. 1999. *A brief review of wave energy.* Harwell Laboratory, Energy Technology Support Unit.

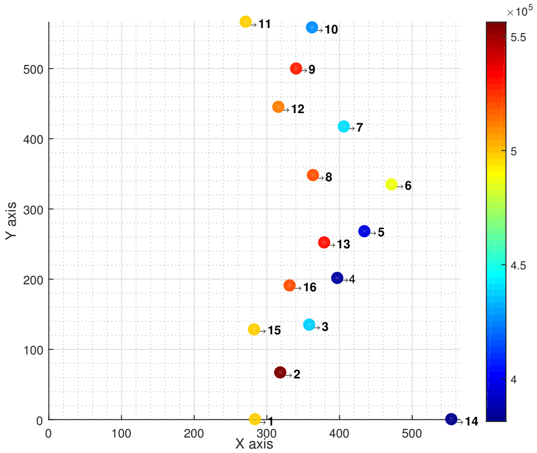

Figure 6: The best layout of $LS_1 + NM_{2D}$ for 16 buoy research. The area size is $566m^2$, the q-factor=0.956, total power output 7608600 Watts, and energy generated by each converter is shown by a range of colors. The order of inserting a new buoy is numbered.

[14] Markus Wagner, Jareth Day, and Frank Neumann. 2013. A fast and effective local search algorithm for optimizing the placement of wind turbines. *Renewable Energy* 51 (2013), 64–70.

[15] GX Wu. 1995. Radiation and diffraction by a submerged sphere advancing in water waves of finite depth. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, Vol. 448. The Royal Society, 29–54.

[16] Junhua Wu, Slava Shekh, Nataliia Y Sergiienko, Benjamin S Cazzolato, Boyin Ding, Frank Neumann, and Markus Wagner. 2016. Fast and effective optimisation of arrays of submerged wave energy converters. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference*. ACM, 1045–1052.