On Design Mining: Coevolution and Surrogate Models

Richard J. Preen and Larry Bull

Abstract—Design mining is the use of computational intelligence techniques to iteratively search and model the attribute space of physical objects evaluated directly through rapid prototyping to meet given objectives. It enables the exploitation of novel materials and processes without formal models or complex simulation. In this paper, we focus upon the coevolutionary nature of the design process when it is decomposed into concurrent sub-design threads due to the overall complexity of the task. Using an abstract, tuneable model of coevolution we consider strategies to sample sub-thread designs for whole system testing, how best to construct and use surrogate models within the coevolutionary scenario, and the effects of access to multiple whole system (physical) testing equipment on performance. Drawing on our findings, the paper then describes the effective design of an array of six heterogeneous vertical-axis wind turbines.

Index Terms—3-D printing, coevolution, meta-models, shape optimisation, surrogate models, wind energy

I. INTRODUCTION

DESIGN MINING [1], [2] is the use of computational intelligence techniques to iteratively search and model the attribute space of physical objects evaluated directly through rapid prototyping to meet given objectives. It enables the exploitation of novel materials and processes without formal models or complex simulation, whilst harnessing the creativity of both computational and human design methods. A sample-model-search-sample loop creates an agile/flexible approach, i.e., primarily test-driven, enabling a continuing process of prototype design consideration and criteria refinement by both producers and users.

Computational intelligence techniques have long been used in design, particularly for optimisation within simulations/models. Recent developments in additive-layer manufacturing (3-D printing) means that it is now possible to work with over a hundred different materials, from ceramics to cells. In the simplest case, design mining assumes no prior knowledge and builds an initial model of the design space through the testing of 3-D printed designs, whether specified by human and/or machine. Optimisation techniques, such as evolutionary algorithms (EAs), are then used to find the optima within the data mining model of the collected data; the model which maps design specifications to performance is inverted and suggested good solutions identified. These are then 3-D printed and tested. The resulting data are added to the existing data and the process repeated. Over time the model—built solely from physical prototypes tested appropriately for the task requirements—captures the salient features of the design space, thereby enabling the discovery of high-quality (novel) solutions. Such so-called surrogate models have also long been used in optimisation for cases when simulations are computationally expensive. Their use with 3-D printing opens new ways to exploit optimisation in the design of physical objects directly, whilst raising a number of new issues over the simulation case.

This approach of constantly producing working prototypes from the beginning of the design process shares resemblance to agile software engineering [3]: requirements are identified at the start, even if only partially, and then corresponding tests created, which are then used to drive the design process via rapid iterations of solution creation and evaluation. The constant supply of (tangible) prototypes enables informed sharing with, and hence feedback from, those involved in other stages of the process, such as those in manufacture or the end user. This feedback enables constant refinement of the requirements/testing and also means that aspects of the conceptual and detailed design stages become blended. Moreover, due to the constant production of better (physical) prototypes, aspects of the traditional manufacturing stage become merged with the design phase. The data mining models created provide new sources of knowledge, enabling designers, manufacturers, or users, to do what-if tests during the design process to suggest solutions, the sharing of timely/accurate information when concurrent sub-design threads are being exploited, etc. Thereafter, they serve as sources of information for further adaptive designs, the construction of simulators/models, etc.

In contrast to human designers, who typically arrive at solutions by refining building blocks that have been identified in highly constrained ways, computational intelligence offers a much more unconstrained and unbiased approach to exploring potential solutions. Thus, by creating the designs directly in hardware there is the potential that complex and subtle physical interactions can be utilised in unexpected ways where the operational principles were previously unknown. These physical effects may simply be insufficiently understood or absent from a simulator and thus otherwise unable to be exploited. Design mining is therefore ideally suited to applications involving highly complex environments and/or materials.

In our initial pilot study we used the design mining approach to discover a pair of novel, heterogeneous vertical-axis wind turbine (VAWT) designs through cooperative coevolution [1]. All but one known previous work considering the design...
of wind farms had used arrays of homogeneous turbines originally optimised to operate alone. Accurate and computationally efficient modelling of the inter-turbine interactions is extremely difficult and therefore the area is ideally suited to the design mining approach. More recently, we have begun to explore the performance of relevant techniques from the literature within the context of design mining. Following [4], the pilot study used multi-layered perceptrons (MLPs) [5] for the surrogate modelling. Using the data from that study, we have subsequently shown that MLPs appear a robust approach in comparison to a number of well-known techniques [2]. That is, MLPs appear efficient at capturing the underlying structure of a design space from the relatively small amount of data points a physical sampling process can be expected to generate. In this paper we begin by continuing the line of enquiry, here focusing upon the coevolutionary nature of the design process when it is decomposed into concurrent sub-design threads due to the overall complexity of the task. Using an abstract, tuneable model of coevolution we consider strategies to sample sub-thread designs for whole system testing, how best to construct and use surrogate models within the coevolutionary scenario, and the effects of access to multiple whole system (physical) testing equipment on performance. Drawing on our findings, the paper then describes the effective design of a more complex array of VAWT than our pilot study.

II. BACKGROUND
A. Cooperative Coevolution and Surrogates
Cooperative coevolution decomposes a global task into interacting sub-component populations and optimises each in parallel. In the context of a design process, this can be seen as directly analogous to the use of concurrent sub-threads. The first known use of cooperative coevolution considered a job-shop scheduling task [6]. Here solutions for individual machines were first evaluated using a local fitness function before being partnered with solutions of equal rank in the other populations to create a global solution for evaluation. Bull and Fogarty [7] subsequently presented a more general approach wherein the corresponding newly created offspring solutions from each population are partnered and evaluated. Later, Potter and De Jong [8] introduced a round-robin approach with each population evolved in turn, which has been adopted widely. They explored using the current best individual from each of the other populations to create a global solution, before extending it to using the best and a random individual from the other population(s). These two partnering strategies, along with others, were compared under their round-robin approach and found to be robust across function/problem types [9]. We used the round-robin approach and partnering with the best individual in our pilot study and return to it here with the focus on learning speed, i.e., how to minimise the number of (timely/costly) fitness evaluations whilst learning effectively.

As EAs have been applied to ever more complex tasks, surrogate models (also known as meta-models) have been used to reduce the optimisation time. A surrogate model, \( y = f(\vec{x}) \), can be formed using a sample \( D \) of evaluated designs \( N \), where \( \vec{x} \) is the genotype describing the design morphology and \( y \) is the fitness/performance. The model is then used to compute the fitness of unseen data points \( \vec{x} \notin D \), thereby providing a cheap approximation of the real fitness function for the EA to use. Evaluations with the real fitness function must continue to be performed periodically otherwise the model may lead to premature convergence on local optima (see Jin [10] for an overview). There has been very little prior work on the use of surrogates in a coevolutionary context: they have been shown capable of solving computationally expensive optimisation problems with varying degrees of epistasis more efficiently than conventional coevolutionary EAs (CEAs) through the use of radial basis functions [11] and memetic algorithms [12]. As we have reviewed elsewhere [1], there is also a small amount of prior work considering the evolutionary design of physical systems directly, stretching back to the origins of the discipline. Remarkably, in 1963 Dunham et al. [13], in describing the evolutionary design of physical logic circuits/devices, briefly note (without giving details): “It seemed better to run through many ‘generations’ with only approximate scores indicating progress than to manage a very few ‘evolutions’ with rather exact statements of position.”

Our aforementioned pilot study is the first known use of coevolutionary design without simulation. As noted above, we have recently compared different modelling techniques by which to construct surrogates for coevolution. In this paper we further consider how best to train and use such models.

B. The NKCS Model
Kauffman and Johnsen [14] introduced the abstract NKCS model to enable the study of various aspects of coevolution. In their model, an individual is represented by a genome of \( N \) (binary) genes, each of which depends epistatically upon \( K \) other randomly chosen genes in its genome. Thus increasing \( K \), with respect to \( N \), increases the epistatic linkage, increasing the ruggedness of the fitness landscape by increasing the number of fitness peaks, which increases the steepness of the sides of fitness peaks and decreases their typical heights. Each gene is also said to depend upon \( C \) randomly chosen traits in each of the other \( X \) species with which it interacts, where there are \( S \) number of species in total. The adaptive moves by one species may deform the fitness landscape(s) of its partner(s). Altering \( C \), with respect to \( N \), changes how dramatically adaptive moves by each species deform the landscape(s) of its partner(s). The model assumes all inter- and intragenome interactions are so complex that it is appropriate to assign random values to their effects on fitness. Therefore, for each of the possible \( K + (X \times C) \) interactions, a table of \( 2^{K+(X \times C)+1} \) fitnesses is created for each gene, with all entries in the range 0.0 to 1.0, such that there is one fitness for each combination of traits. The fitness contribution of each gene is found from its table; these fitnesses are then summed and normalised by \( N \) to give the selective fitness of the total genome for that species. Such tables are created for each species (see example in Fig. [1] for full details). This tuneable model has previously been used to explore coevolutionary optimisation, particularly in...
**C. Evolving Wind Farms**

As we have reviewed elsewhere [11], techniques such as EAs have been used to design wind turbine blades using computational fluid dynamics (CFD), some in conjunction with surrogate models. Our pilot study considered wind farm design, more specifically the design of heterogeneous turbines that exploit the inter-turbine effects of close physical spacing. Homogeneous arrays were first explored in [16]. Heterogeneous wind farms have been almost completely unexplored within the wind energy literature. However, Chamorro et al. [17] recently explored a 3 × 8 horizontal-axis wind turbine (HAWT) farm with large and small turbines positioned alternately. They found that size heterogeneity has positive effects on turbulent loading as a result of the larger turbines facing a more uniform turbulence distribution and the smaller turbines operating under lower turbulence levels. Our pilot study found that asymmetrical pairs of VAWTs can be more efficient than similar symmetrical designs. In this paper, we extend our initial work to the heterogeneous design of an array of 6 closely positioned VAWT, which is currently effectively beyond the capabilities of accurate 3-D CFD simulation; the approach performs optimisation in the presence of non-uniform wind velocity, complex inter-turbine wake effects, and multi-directional wind flow from nearby obstacles, which is extremely difficult to achieve accurately under high fidelity CFD simulation. In addition, previously the combined rotational speed was simply used as the objective measure, whereas here we use the total angular kinetic energy of the array, which includes both mass and speed of rotation, and we use a more flexible spline representation that enables the potential exploitation of both drag and lift forces in conjunction with inter-turbine flow and turbulence from nearby obstacles.

**III. Surrogate-assisted Coevolution**

The basic coevolutionary genetic algorithm (CGA) is outlined in Algorithm 1. Initially all individuals in each of the species/populations must be evaluated. Since no initial fitness values are known, a random individual is chosen in each of the other populations to form a global solution, however if there is a known good individual then that individual can be used instead. The CGA subsequently cycles between populations, selecting parents via tournament and creating offspring with mutation and/or crossover. The offspring are then evaluated using representative members from each of the other populations. At any point during evolution, each individual is assigned the highest fitness value received as a member of any team in which it is a participant.

For the basic surrogate-assisted CGA (SCGA) used in this paper, the CGA runs as normal except that each time a parent is chosen, \( \lambda_m \) number of offspring are created and then evaluated with an artificial neural network surrogate model; the single offspring with the highest approximated fitness is then evaluated with the real fitness function in collaboration with the fittest solution (best partner) in each other population.

**IV. NKCS Experimentation**

**A. Methodology**

For NKCS simulations, \( S = 3, X = 2, P = 20, N = 20 \), per allele mutation probability \( \mu = 5\% \), and crossover probability is 0%. The model parameters: \( N \) input neurons, \( H = 10 \) hidden neurons, 1 output neuron, \( \lambda_m = 1000 \), \( T = 50, \beta = 0.1 \). All results presented are an average of 100 experiments consisting of 10 coevolutionary runs on 10 randomly generated NKCS functions.
Algorithm 1: Coevolutionary genetic algorithm

1. for each species do
   2. initialise population;
   3. select a random representative for each other species;
   4. for each individual in population do
      5. evaluate;
   6. end
   7. end
8. while evaluation budget not exhausted do
   9. for each species do
      10. create an offspring using genetic operators;
      11. select a representative for each other species;
      12. evaluate the offspring;
      13. add offspring to species population;
   14. end
15. end

Algorithm 2: SCGA with enhanced local search

1. for each species do
   2. initialise population;
   3. select a random representative for each other species;
   4. for each individual in population do
      5. evaluate;
      6. archive;
   7. end
   8. end
9. while evaluation budget not exhausted do
   10. for each species do
      11. initialise model;
      12. train model on species archive;
      13. select parent(s) using tournament selection;
      14. for $\lambda_m$ number of times do
      15. create an offspring using genetic operators;
      16. predict offspring fitness using the model;
      17. end
      18. select the offspring with the highest model predicted fitness;
      19. select a representative for each other species;
      20. evaluate the offspring;
      21. add offspring to species population;
      22. archive offspring;
   23. end
24. end

B. Results

The performance of four different CGA collaboration schemes are shown on four different $K$ and $C$ values in Fig. 2 each representing a different point in the range of inter- and intra-population dependence. Following Potter and De Jong [8], in CGA-b, each offspring is evaluated in collaboration with only the current best individual in each of the $X$ other species populations. In CGA-br, each offspring is additionally evaluated with a random member in each of the other populations. We also introduce a scheme similar to CGA-b, where each offspring is evaluated with only the best member in each of the other populations but re-evaluates all populations when one makes a progress (CGA-re). That is, we explore any benefits to overall learning speed from refreshing the population fitness values as the fitness landscapes potentially shift: all individuals in the $X$ other species populations are re-evaluated in collaboration with the current elite members each time a new fittest individual is found.

The results show that during the early stages of evolution, performing additional evaluations (CGA-br and CGA-re) results in a lower fitness compared with the approach of only collaborating with the elite members (CGA-b). As shown in Table I after 360 evaluations, the mean fitness of the best solutions discovered by CGA-b is significantly greater than CGA-re and CGA-br on all four $K,C$ combinations. After 1800 evaluations, the three approaches generally reach approximately similar performance, suggesting that there is no penalty for this increase in early learning speed. For the case of both lower inter- and intra-population epistasis CGA-br performs better, which supports findings reported elsewhere [8], [9].

As noted above, traditionally (after Potter and De Jong [8]), CEAs consider each population in turn. Thus, if $S = 10$ then 10 evaluations are required for the whole system. However, at the other end of this scale each population generates a new individual and evaluates all offspring at once [7], therefore requiring only one evaluation for the whole system. Varying this number is much like varying the system-level mutation rate. Therefore, in CGA-o, an offspring is created in each species simultaneously and evaluated together. Interestingly, during the early stages CGA-o provides no increase in performance compared with CGA-b, however after 1800 evaluations the mean fitness is significantly greater on all four $K,C$ values, showing that the extra exploration can be beneficial in the long-run.

In Fig. 3 the performance of CGA-b can be compared with the surrogate-assisted version (SCGA-b). The use of the surrogate model to identify more promising offspring clearly increases learning early in the search. For example, after 360 evaluations the mean fitness of the SCGA-b best individuals is significantly greater than CGA-b, while still reaching similar optima at the end of the experiment; see Table I. The benefit of the divide-and-conquer strategy to model building can be

| $K2C2$ | $K2C8$ | $K6C2$ | $K6C8$ |
|-------|-------|-------|-------|
| 1.9934 | 1.8982 | 1.9678 | 1.9069 |
| 2.0361 | 2.0988 | 2.0648 | 2.1178 |
| 2.0235 | 2.0258 | 2.0257 | 2.0648 |
| 2.0578 | 2.0703 | 2.0728 | 2.0988 |
| 2.0042 | 2.0077 | 2.0150 | 2.0361 |

TABLE I: CGA best fitnesses after 360 and 1800 evaluations (averages of 100). The mean is highlighted in bold font where it is significantly different to CGA-b using a Mann-Whitney U test at the 95% confidence interval.
seen by comparing SCGA-b (where the models are presented only the N genes from their own species) with the case where the models are presented all $N \times S$ partner genes (SCGA-a). On all four $K, C$ values, the SCGA-a mean best fitness is significantly less than SCGA-b after 360 evaluations; see Table II. That is, purely local models are both efficient and scalable.

It is also shown that the simple method of using the model we presented in our pilot study to find the most promising of $\lambda_m$ offspring stemming from a single parent is quite robust as there is no significant difference with searching the same number of offspring where $\lambda_m$ tournaments are performed to select parents that each create a single offspring (SCGA-p).

Where parallel evaluations of the whole physical system can be performed there exists the potential to decrease the wall-clock time of the design process by enabling better exploration of the search space. This raises the question of how to best select the extra individuals for evaluation. Fig. 4 illustrates the performance of SCGAs where 5 evaluations are executed in parallel; time $t$ is the number of GA invocations. In pSCGA-b, the 5 (unique) best of $\lambda_m$ offspring are evaluated in parallel instead of the usual 1 (SCGA-b). As one would expect, performing additional evaluations in the same amount of time discovers solutions with a greater fitness; e.g., as shown in Table III after $t = 180$ the pSCGA-b mean best fitness is significantly greater than SCGA-b. In pSCGA-p, each time the GA invokes, 5 parents each create $\lambda_m/5$ offspring and the best of each are selected for evaluation in collaboration with the elite members in the other species. From Table III it can be seen that using extra parents to increase the search diversity results in no significant difference to pSCGA-b. In pSCGA-o, 5 parents each create $\lambda_m/5$ offspring in each species simultaneously and the best of each species parent are evaluated together in 5 different teams. As can be seen, the benefits observed previously by the CGA approach (CGA-o) are lost in the surrogate-assisted version (pSCGA-o).

V. VAWT WIND FARM DESIGN

A. Methodology

A single 2-stage 2-blade VAWT candidate with end plates is here created as follows. End plates are drawn in the centre of a Cartesian grid with a diameter of 35 mm and thickness of 1 mm. A central shaft 70 mm tall, 1 mm thick, and with a 1 mm hollow diameter is also drawn in the centre of the grid in order to mount the VAWT for testing. The 2-D profile
TABLE II: SCGA best fitnesses after 360 and 1800 evaluations (averages of 100). The mean is highlighted in bold font where it is significantly different to SCGA-b using a Mann-Whitney U test at the 95% confidence interval.

|                | SCGA-b | CGA-b | SCGA-a | SCGA-p |
|----------------|--------|-------|--------|--------|
| **K=2 C=2**    |        |       |        |        |
| After 360 evals: | 2.0446 | 1.9934 | 1.9723 | 2.0187 |
| After 1800 evals: | 2.0927 | 2.0831 | 2.0650 | 2.0768 |
| **K=6 C=2**    |        |       |        |        |
| After 360 evals: | 1.9718 | 1.9300 | 1.9508 | 1.9807 |
| After 1800 evals: | 1.9974 | 1.9624 | 1.9688 | 2.0006 |
| **K=6 C=8**    |        |       |        |        |
| After 360 evals: | 1.9452 | 1.9272 | 1.9299 | 1.9507 |
| After 1800 evals: | 1.9718 | 1.9300 | 1.9508 | 1.9807 |

TABLE III: pSCGA best fitnesses after time intervals 180 and 600 (averages of 100). The mean is highlighted in bold font where it is significantly different to pSCGA-b using a Mann-Whitney U test at the 95% confidence interval.

|                | pSCGA-b | SCGA-b | pSCGA-a | pSCGA-p |
|----------------|---------|--------|---------|---------|
| After time 180: |         |        |         |         |
| K2C2           | 2.0689  | 1.9142 | 2.0450  | 2.0629  |
| K2C8           | 2.0108  | 1.8699 | 1.9970  | 1.8941  |
| K6C2           | 2.0557  | 1.8994 | 2.0457  | 2.0245  |
| K6C8           | 1.9981  | 1.8419 | 2.0205  | 1.8522  |
| After time 600: |         |        |         |         |
| K2C2           | 2.0929  | 2.0627 | 2.0664  | 2.0984  |
| K2C8           | 2.0371  | 2.0286 | 2.0272  | 2.0172  |
| K6C2           | 2.0814  | 2.0499 | 2.0765  | 2.0739  |
| K6C8           | 2.0314  | 2.0015 | 2.0205  | 1.9443  |

of a single blade on 1 stage is represented using 5 \((x, y)\) co-ordinates on the Cartesian grid, i.e., 10 genes, \(x_1, y_1, \ldots, x_5, y_5\). A spline is drawn from \((x_1, y_1)\) to \((x_3, y_3)\) as a quadratic Bézier curve, with \((x_2, y_2)\) acting as the control point. The process is then repeated from \((x_3, y_3)\) to \((x_5, y_5)\) using \((x_4, y_4)\) as control. The thickness of the spline is a fixed size of 1 mm. The co-ordinates of the 2-D blade profile are only restricted by the plate diameter; that is, the start and end position of the spline can be located anywhere on the plate.

To enable \(z\)-axis variation, 3 additional co-ordinates (i.e., 6 genes) are used to compute cubic Bézier curves in the \(xz\) and \(yz\) planes that offset the 2-D profile. The \(xz\)-plane offset curve is formed from an \(x\) offset=0 on the bottom plate to an \(x\) offset=0 on the top plate using control points \((z_1, z_1)\) and \((z_2, z_2)\). The \(yz\)-plane offset curve is formed in the same way with \(zy_1\) and \(zy_2\) control points, however reusing \(z_1\) and
Furthermore, an extra gene, $r_1$, specifies the degree of rotation whereby the blades of 1-stage are rotated from one end plate to the next through the $z$-axis to ultimately $0^\circ - 180^\circ$. Thus, a total of 17 genes specify the design of a candidate VAWT. The blade is then duplicated and rotated $180^\circ$ to form a 2-bladed design. The entire stage is then duplicated and rotated $90^\circ$ to form the second stage of the VAWT; see example design in Figure 5a. When physical instantiation is required, the design is fabricated by a MakerBot 2 Replicator 3-D printer using a polylactic acid (PLA) bioplastic at 0.3 mm resolution. Figure 5b shows the VAWT after fabrication.

In order to provide sufficient training data for the surrogate model, initially the CGA proceeds for 3 generations before the model is used, i.e., a total of 360 physical array evaluations with 60 evaluated individuals in each species. $S = 6$ species are explored, each with $P = 20$ individuals, a per allele mutation probability, $\mu = 25\%$ with a mutation step size, $\sigma_1 = 3.6$ (mm) for co-ordinates and $\sigma_2 = 18^\circ$ for $r_1$, and a crossover probability of $0\%$. Each species population is initialised with the example design in Figure 5a and 19 variants mutated with $\mu = 100\%$. The individuals in each species population are initially evaluated in collaboration with the seed individuals in the other species populations. Thereafter, the CGA runs as normal by alternating between species after a single offspring is formed and evaluated with the elite members from the other species (as in Algorithm 1). After 3 generations the SCGA (as in Algorithm 2) is used for an additional generation. The model parameters: 17 input neurons, $H = 10$ hidden neurons, 1 output neuron, $\lambda_m = 1000$, $T = 1000$, $\beta = 0.1$. Each VAWT is treated separately by evolution and approximation techniques, i.e., heterogeneous designs could therefore emerge.

The fitness, $f$, of each individual is the total angular kinetic energy of the collaborating array,

$$f = \sum_{i=1}^{S} KE_i$$

where the angular kinetic energy ($J$), $KE$, of an individual VAWT,

$$KE = \frac{1}{2} I \omega^2$$

with angular velocity (rad/s), $\omega = \frac{rpm}{60} \cdot 2\pi$, and moment of inertia (kg·m$^2$), $I = \frac{1}{2} mr^2$ with $m$ mass (kg), and $r$ radius (m).

The rotational speed (rpm) is here measured using a digital photo laser tachometer (PCE-DT62; PCE Instruments UK Ltd).
(a) Seed design; genome: $x_1 = 15.1, y_1 = 15.1, x_2 = 22.1, y_2 = 15.1, x_3 = 25.7, y_3 = 15.9, x_4 = 32.1, y_4 = 16.1, x_5 = 32.1, y_5 = 27.1, z x_1 = 0, z x_2 = 0, z y_1 = 0, z y_2 = 0, z_1 = 20, z_2 = 27.2, r_1 = 0.$

(b) Seed design printed by a 3-D printer; 35 mm diameter; 70 mm tall; 7 g; 45-min printing time at 0.3 mm resolution.

Fig. 5: Example VAWT.

by placing a $10 \times 2$ mm strip of reflecting tape on the outer tip of each VAWT and recording the maximum achieved over the period of $\sim 30$ s during the application of wind generated by a propeller fan.

Fig. 6 shows the test environment with the 30 W, 3500 rpm, 304.8 mm propeller fan, which generates 4.4 m/s wind velocity, and 6 turbines mounted on rigid metal pins 1 mm in diameter and positioned 42.5 mm adjacently and 30 mm from the propeller fan. That is, there is an end plate separation distance of 0.2 diameters between turbines. It is important to note that the wind generated by the fan is highly turbulent with non-uniform velocity and direction across the test platform, i.e., each turbine position receives a different amount of wind energy from different predominant directions and wind reflecting from the test frame may cause multi-directional wind flow. Thus, the designs evolved under such conditions will adapt to these exact environmental pressures.

### B. Results

Each generation consisted of 120 fabrications and array evaluations. After evaluating all individuals in the initial species populations, no mutants were found to produce a greater total $KE$ than the seed array. After 1 evolved CGA generation, the fittest array combination generated a greater total $KE$ of 7.6 mJ compared with the initial seed array, which produced 5.9 mJ. A small increase in total $m$ of 42 g to 44.8 g was also observed. After 2 evolved CGA generations, the fittest array combination generated a total $KE$ of 10 mJ with a further small increase in total $m$ to 45.6 g. The surrogate model was then used for 1 additional generation and produced a total $KE$ of 12.2 mJ with a further increase in $m$ to 49.3 g. Fig. 7 shows the total angular $KE$ of the fittest arrays each generation; Fig. 8 shows the total $m$, and Fig. 9 the total rpm. The cross sections of the fittest array designs can be seen in Figs. 10–13.

It is interesting to note the similarity of some of the evolved VAWTs with human engineered designs. Bach [18] performed one of the earliest morphological studies of Savonius VAWT and found increased aerodynamic performance with a blade profile consisting of a $2/3$ flattened trailing section and a larger blade overlap to reduce the effect of the central shaft, which is similar to the fourth species design in Fig. 12d. The evolved VAWT in the second species, e.g., Fig. 12b overall appear more rounded and similar to the classic S-shape Savonius design [19]. There appears to be little twist rotation along the $z$-axis of the evolved designs, which may be a consequence of the initial seeding or due to the test conditions having strong and persistent wind velocity from a single direction; that is,
VI. CONCLUSION

Design mining represents a methodology through which the capabilities of computational intelligence can be exploited directly in the physical world. Our previous pilot study [1] considered the parallel design of two interacting objects. In this paper we have used a well-known abstract model of coevolution to explore and extend various techniques from the literature, with an emphasis on reducing the number of fitness function evaluations. Our results suggest the round-robin evaluation process using the best solution in each other population is robust [8], as is our previously presented sampling method of surrogate models built using strictly population specific data. It has also been shown that the same techniques remain effective where more than one physical test setup exists, enabling parallel evaluation and thus reducing time.

These findings were then applied to a more complex version of the wind turbine design task considered in our pilot study, primarily moving from designing a pair of heterogeneous VAWT to a system of 6 turbines. As noted above, the SCGA remains robust to an increasing number of turbines since the number of inputs to the models remains constant. Indeed, we are unaware of a real-world coevolutionary optimisation task of this scale with or without surrogate models.

The VAWT spline representation used here is also much more flexible than the simple integer approach used previously, enabling the exploration of designs where the blades are not attached to the central shaft. This has enabled designs to emerge that not only exploit or compensate for the wind interaction with the central shaft, but also the effect of mass and vibrational forces as the turbines freely rotate around the mounted pins at high speed. That is, it has been shown possible to exploit the fan-generated wind conditions in the environment, including the complex inter-turbine turbulent flow conditions, and position-specific wind velocity, to design an array of 6 different turbines that work together collaboratively to produce the maximum angular kinetic energy. Note that the starting designs for the turbines was based upon a standard commercial design and performance in the array was seen to double over the time allowed.

3-D printing provides a very flexible way to rapid-prototype designs for testing. One of the most significant benefits of the technology is a ‘complexity paradox’ wherein the complexity of the printed shape is highly uncorrelated with production time/cost. With conventional manufacturing, the more complex an object is, the more it costs to fabricate (especially when sub-components require complex assembly processes). However, with 3-D printing, the time and cost to fabricate an object mostly depends on the amount of material used. Moreover, the more complex a shape is, the more numerous the spaces (voids) that exist between components, and thus the smaller the quantity of material required. There is thus a synergy between computational intelligence techniques that can search a wide variety of complex shapes in a complex environment whilst...
also exploring the effects of novel materials. Here only PLA plastic was used to fabricate designs, however there are now over a hundred different materials that 3-D printers can use, ranging from cells to titanium. Future work will explore the use of flexible materials and multi-material designs, which may result in very different designs of future wind farms. In addition, 3-D printing can produce designs at different fidelity, such as slower more accurate prints for subtle optimisation or rapid coarse designs for quick evaluation. Fabrication can also be parallelised with multiple printers, e.g., a different printer for each species.

The design mining approach outlined here provides a general and flexible framework for engineering design with applications that cannot be simulated due to the complexity of materials or environment. 3-D printing is a rapidly emerging area with an ever growing number of materials available for fabrication and with decreasing production times. We anticipate that in the future, design mining will be used to create highly unintuitive yet efficient bespoke solutions for a wide range of complex engineering design applications.

REFERENCES

[1] R. J. Preen and L. Bull, “Toward the coevolution of novel vertical-axis wind turbines,” IEEE Trans. Evol. Comput., vol. 19, no. 2, pp. 284–294, Apr. 2015.
[2] ——, “Design mining interacting wind turbines,” Evol. Comput., vol. 24, no. 1, Mar. 2016.
[3] J. Martin, Rapid Application Development. New York, NY, USA: Macmillan, 1991.
[4] L. Bull, “Model-based evolutionary computing: a neural network and genetic algorithm architecture,” in Proc. IEEE Int. Conf. Evol. Comput., 1997, pp. 611–616.
[5] F. Rosenblatt, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Washington DC, USA: Spartan Books, 1962.
[6] P. Husbands and F. Mill, “Simulated coevolution as the mechanism for emergent planning and scheduling,” in Proc. 4th Int. Conf. Genetic Algor., R. K. Belew and L. B. Booker, Eds. San Francisco, CA, USA: Morgan Kaufmann, 1991, pp. 264–270.

[7] L. Bull and T. Fogarty, “Coevolving communicating classifier systems for tracking,” in Artif. Neural Nets and Genetic Algor., R. F. Albrecht, C. R. Reeves, and N. C. Steele, Eds. Berlin, Germany: Springer, 1993, pp. 522–527.

[8] M. A. Potter and K. A. De Jong, “A cooperative coevolutionary approach to function optimization,” in PPSN III, ser. LNCS, Y. Davidor, H.-P. Schwefel, and R. Manner, Eds. Berlin, Germany: Springer, 1994, vol. 866, pp. 249–257.

[9] L. Bull, “Evolutionary computing in multi-agent environments: Partners,” in Proc. 7th Int. Conf. Genetic Algor., 1997, pp. 370–377.

[10] Y. Jin, “Surrogate-assisted evolutionary computation: recent advances and future challenges,” Swarm Evol. Comput., vol. 1, no. 2, pp. 61–70, Jun. 2011.

[11] Y. S. Ong, A. J. Keane, and P. B. Nair, “Surrogate-assisted coevolutionary search,” in Proc. 9th ICONIP, vol. 3, 2002, pp. 1140–1145.

[12] C. K. Goh, D. Lim, L. Ma, Y. S. Ong, and P. S. Dutta, “A surrogate-assisted memetic co-evolutionary algorithm for expensive constrained optimization problems,” in Proc. IEEE Congr. Evol. Comput., 2011, pp. 744–749.

[13] B. Dunham, R. Fridshal, and J. H. North, “Design by natural selection,” Synthese, vol. 15, no. 2, pp. 254–259, Jun. 1963.

[14] S. A. Kauffman and S. Johnsen, “Coevolution to the edge of chaos: Coupled fitness landscapes, poised states and coevolutionary avalanches,” J. Theor. Biol., vol. 149, no. 4, pp. 467–505, Apr. 1991.

[15] S. A. Kauffman, The origins of order: Self-organisation and selection in evolution. New York, NY, USA: Oxford University Press, 1993.

[16] A. F. Charwat, “Performance of counter- and corotating arrays of Savonius turbines,” J. Energy, vol. 2, pp. 61–63, Jan. 1978.

[17] L. P. Chamorro, N. Tobin, R. E. A. Arndt, and F. Sotiropoulos, “Variable-sized wind turbines are a possibility for wind farm optimization,” Wind Energy, vol. 17, no. 10, pp. 1483–1494, Oct. 2014.

[18] G. Bach, “Untersuchungen über Savonius-rotoren und verwandte strömungsmaschinen,” Forschung auf dem Gebiet des Ingenieurwesens A, vol. 2, no. 6, pp. 218–231, Jun. 1931.

[19] S. J. Savonius, “Wind rotor,” US Patent 1 766 765, June 24, 1930.