Using Genetic Algorithms to Evolve Three-Dimensional Microstructures from Two-Dimensional Micrographs

DAVID BASANTA, MARK A. MIODOWNIK, ELIZABETH A. HOLM, and PETER J. BENTLEY

The article describes work to bring together the topics of evolutionary computing and stereology and asks the reader to judge whether such an approach can be genuinely useful or just represents a clever application of computer science. The problem we address is that of constructing three-dimensional (3-D) microstructures from two-dimensional (2-D) micrographs. Our solution is a computer program called MicroConstructor that evolves 3-D discrete computer microstructures, which are statistically equivalent to the 2-D inputs in terms of the microstructural variables of interest. The core of MicroConstructor is a genetic algorithm that evolves the 3-D microstructure so that its stereological parameters match the 2-D data. MicroConstructor uses a general method of pattern construction, the EmbryoCA, that does not require intervention from the user and is highly evolvable. This article presents initial results from successful experiments to evolve 3-D two-phase microstructures from 2-D input microstructures. The advantages and disadvantages of the method are discussed, and we conclude that the method, though delightfully elegant and full of potential, has yet to prove itself capable of constructing 3-D microstructures that would interest experimentalists and computer modelers.

I. INTRODUCTION

THERE is a general consensus among computational material scientists of the need to incorporate experimental microstructures as starting configurations into computer models of various types.[1,2,3] This is because materials property prediction is based on one of the basic tenants of materials science, the structure-property relationship. Thus, without accurate measures of the structure, property prediction is flawed. Since almost all microstructures are three-dimensional (3-D) in nature, it follows that computer models need 3-D microstructures as inputs.

Measuring 3-D microstructural information is experimentally challenging. Serial sectioning, where material is removed layer by layer, was for a long time the only method available.[4] It presents a number of technical difficulties to the microscopist: first, how to reconstruct the 3-D structure from two-dimensional (2-D) layers; and, second, how to interpolate to reconstruct the material lost in removal. Recently, there have been a number of advances in mechanical sectioning automization.[5] Digital reconstruction techniques have also improved, making the 3-D reconstruction of the serial sectioning data routine enough to study systems other than model metal alloys.[6–11] A method of serial sectioning coupled with orientation mapping has been developed, allowing the crystallography of boundary planes to be studied.[12,13,14]

DAVID BASANTA, Ph.D. Student, and MARK A. MIODOWNIK, Lecturer, are with the Mechanical Engineering Department, King's College London, London WC2R 2LS, United Kingdom. Contact e-mail: mark.miodownik@kcl.ac.uk ELIZABETH A. HOLM, Distinguished Member Technical Staff, is with the Sandia National Laboratories, Albuquerque, NM 87185-1411. PETER J. BENTLEY, Research Fellow, is with the Computer Science Department, University College London, London WCIE 6BT, United Kingdom.

This article is based on a presentation made in the symposium entitled "Three Dimensional Materials Science" during the 2003 MS&T '03: Materials Science & Technology Conference 2003 in Chicago, Illinois, on November 11 & 12, 2003, under the auspices of the ASM/MSCTS: Materials Science Critical Technology Sector Committee and the TMS/SMD: Structural Materials Division Committee.

Three-dimensional microstructural information can be obtained nondestructively using X-ray synchrotron methods.[15,16] Despite technical challenges, the spatial resolution of 3-D X-ray diffraction microscopes has steadily been increasing. An attraction of the technique is that it can capture not just spatial information, but also temporal information, thus allowing microstructural evolution to be studied in 3-D.[17] At the moment, these techniques require access to rare X-ray synchrotron sources and expensive equipment, which limits their general use.

The inverse problem of reconstructing 3-D microstructures from limited information, such as a single 2-D micrograph, has been studied by a number of authors and has been reviewed by Torquato.[18,19] The early work was based on thresholding Gaussian random fields.[20] Recently, Rintoul and Torquato[21] introduced a stochastic optimization technique to reconstruct dispersoid microstructures. They used radial distribution functions to statistically characterize the microstructure and reconstructed 2-D two-phase dispersions using a simulated annealing approach. Yeong and Torquato extended this approach to reconstructing one-dimensional (1-D) and 2-D microstructures[22] and 3-D microstructures.[23] They introduced a set of correlation functions to characterize the microstructure, and were able to reconstruct 3-D pore structures from charaterizations of 2-D slices of Fontainebleau sandstone. The 3-D reconstructions were compared with experimentally obtained 3-D characterizations of the microstructure, and Yeong and Torquato showed that the reconstruction gave an accurate measure of the pore-size distribution and other physical properties of the sandstone. This work has been extended by Manwart *et al.*[24] to Berea sandstone, and by Sheehan and Torquato to aluminum-boron carbide composites.[25]

The Torquato approach to reconstructing 3-D two-phase microstructures is to discretize the microstructure into 3-D voxels, where each voxel can have one of two states corresponding to the two phases in the system. The initial configuration is determined by randomly allocating a state to each voxel, but conserving the volume fraction of each state to

correspond to the 2-D original input microstructure. The Hamiltonian "energy" of the system is calculated, which relates to how closely the characterization of the microstructure matches the correlation functions of a desired microstructure. Microstructure is then evolved by randomly swapping the states of pairs of voxels and accepting the change using a Metropolis probability scheme. Simulated annealing is then carried out to evolve the 3-D microstructure with the lowest energy. Since there is a finite (but very large) number of possible two-phase microstructures in a discrete system, one can think of simulated annealing as a method to navigate the large state space of possible microstructures (phenotypic space) and locate the one with the greatest correspondence with the desired correlation functions.[23,18]

The work presented in this article follows the Torquato approach in that we use stereological measures to characterize 2-D microstructures and reconstruct statistically equivalent but nonunique 3-D reconstructions. What is novel in our approach is both our method of reconstruction and our method of navigating phenotypic space. In the Torquato approach, the method of reconstruction is voxel swapping, and navigation of phenotypic space takes place by simulated annealing. In our method, we use cellular automata as our reconstruction method, and use a genetic algorithm to navigate phenotypic space. Currently, our work is limited to two-phase single-crystal microstructures, but the method presented here is general enough that it could be applied to all other types of microstructures such as polycrystals and multiphase materials.

This article is divided into a number of sections. First, the background to genetic algorithms is presented. Second, we describe our computational approach, called MicroConstructor, in which we bring together genetic algorithms with topics familiar to materials scientists, stereology and cellular automata, to describe how we evolve and grow 3-D microstructures. Third, we present and discuss the results of computer experiments, which investigate the current limits of MicroConstructor's ability to evolve 3-D two-phase microstructures from 2-D micrographs.

II. GENETIC ALGORITHMS

Evolutionary algorithms use ideas inspired by natural evolution[28] such as survival of the fittest and inheritance with variation in order to evolve a population of potential solutions to a problem, starting from individuals that have been randomly created.[29] Genetic algorithms (GAs) are one of the most popular form of evolutionary algorithms.[30]

The algorithm for a typical GA can be characterized as follows.[31]

(1) Randomly create initial population of genotypes.
(2) For each individual in the population,
    (a) decode the genotype into a phenotype of final representation, and
    (b) evaluate the fitness of the phenotype.
(3) While size of next generation $<$ threshold,
    (a) select two parents, choosing fitter individuals with increased probability;
    (b) use a selection and mutation to generate two offspring from the parents; and
    (c) place offspring into new population.
(4) If acceptable solution not yet found, repeat from step 2.

Although the algorithm starts from a randomly created initial population, that does not mean that the algorithm itself is random. Thanks to artificial selection (survival of the fittest), the best solutions tend to prevail and pass on their genes to the next generation. Genetic operators such as crossover and mutation make sure that new solutions are frequently introduced in the population. In GAs, solutions are not created as much as evolved, and this evolution is not specified in the algorithm but emerges from it.[31]

When designing a GA, it is necessary to pay attention to three features: the genetic operators, the fitness function, and the representation of the individuals. We discuss these topics in Sections A through C.

A. *Representation of Individuals*

One of the features of GAs that differentiate them from other evolutionary algorithms is that the solution and its encoding are different. This means that GAs work in two different spaces: the solution space and the search space.[31] The separation between search space (genetic space) and solution space (phenotypic space) means that the individuals of the population can represent complex solutions and still be encoded in a way that allows the GA to efficiently operate on the genotype.

Figure 1 shows an example of the mapping between the search space and the solution space. The representation of the individual in the search space is called genotype. In the example, the genotype is a binary string. The representation in the solution space is known as phenotype. In Figure 1, the phenotype is a bitmap in which all the white sites correspond to "0's" in the genotype, whereas the colored sites correspond to the "1's." This is an example of direct mapping, which is characterized by a one to one relationship between the genotype and the phenotype. Direct mapping is not the only way to map genotype to phenotype; nature chooses a developmental mapping between DNA and biological form,[32] and thus the phenotype of all biological organisms is established by growth from a single cell. We will address thus issue at greater length in Section III–A.

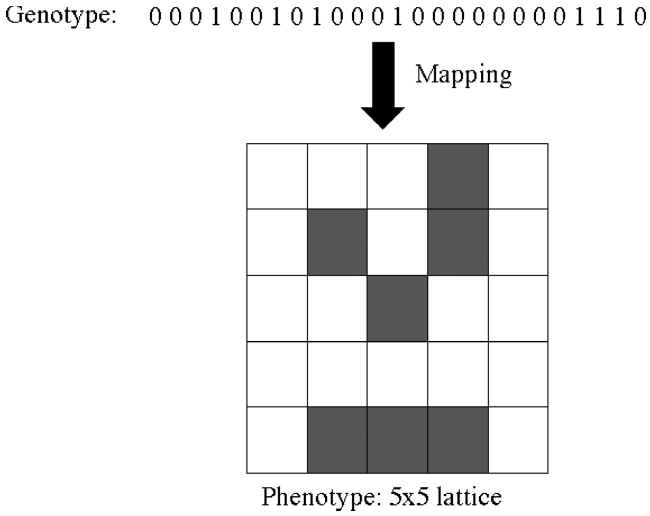


Fig. 1—Example of the separation of genotype and phenotype in GAs. The genotype is the representation of an individual in the search space, and the phenotype is the representation in the solution space. In the example, the genotype is a binary string and the phenotype is a 2-D lattice.

B. *Fitness*

It is the phenotypic representation that is used to judge the fitness of an individual. All the individuals in the population are measured and given a fitness value that measures the quality of the solution they represent. It is this fitness value that determines if the genes of the individual will be passed on to the next generation. For example, if we try to evolve a design for a car, then the fitness of different evolved solutions in a population might be estimated by measuring the aerodynamic drag. Clearly, there is more to a car design that its drag efficiency; this is why the choice of fitness function is critical to the success of a GA. It is also worth noting that GAs work best with fitness functions that vary smoothly in solution space. Thus, the choice of the form of the fitness function is also important.[31]

C. *Genetic Operators*

There are several different methods to select the parents of the individuals of the next generation, but one of the most widely used is called tournament selection. Using this selection method, a number of candidates are randomly drawn from the population and the fittest one is selected for reproduction. This procedure is repeated until there are enough parents selected to create a new population.[30]

Once the parents are chosen, there are many operations that can be performed on the genotypes of the individuals to produce the next generation. Two of the most common are crossover and mutation. The crossover operation is used to create new individuals that combine the genes of two selected parents. An example of crossover is the one point crossover illustrated in Figure 2. With this operation, two offspring are created after randomly selecting one point of the genotype and then swapping the segment of the genotype of those parents between the beginning and the selected point. A similar type of crossover operation is the two-point crossover, in which the genotype of the two parents is swapped between two randomly selected points, to create two offspring. The mutation operation is normally applied to the individuals of a recently created population. In most cases, the operation consists on randomly selecting a gene and changing its value.[31]

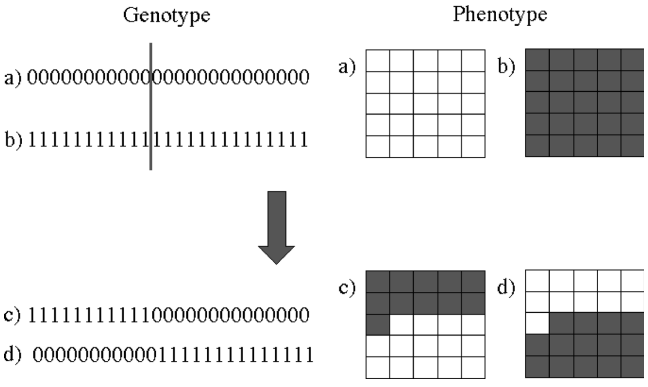


Fig. 2—Example of one-point crossover. (*a*) and (*b*) Two individuals have been selected for crossover. (*c*) and (*d*) The result of applying the operator to these directly mapped individuals.

III. MICROCONSTRUCTOR

MicroConstructor is a system comprised of a GA that evolves populations of cellular automata (CA), which develop 3-D two-phase microstructures. The aim of the GA is to find a 3-D microstructure with stereological properties that match those of the user-provided 2-D input. Figure 3 is a schematic overview of the MicroConstructor approach. It shows an example of a 2-D microstructure that is used as an input structure. It is a structure with two phases, an α matrix phase and a β phase, which in this case has a particle morphology. The experimental inputs are discretized and measured to establish a stereological characterization. The figure also shows schematically how MicroConstructor's GA evolves the genotypes, by using CA to map them onto 3-D microstructures and comparing stereological parameters of these microstructure with those of the 2-D input structure.

The most important attributes of MicroConstructor are the representation of the individuals in the population and the way these individuals are measured with a fitness function. We discuss these features in Sections A and B.

A. *Representation of Individuals*

Representing the 3-D images of microstructures in a linear way, like the direct mapping example shown in figure 1, is an approach to microstructure representation that has a number of problems associated with scalability.[26,33] Nature creates organisms of extraordinary complexity and sophistication using a developmental approach.[34] Biological development is the mapping between the DNA (genotype) of a biological organism and the complex pattern of cells that represents their structure (phenotype). It is a process of construction and growth in which pattern and structure emerges from the interactions between proteins and genes and cells, with the environment. As a consequence of the interplay of these elements, structures emerge from a simple group of cells that divide, grow, and change shape.[35] Development is the key to understanding how complex systems evolve.[33]

MicroConstructor uses CA to model the development of the genotype into a 3-D microstructure. The CA were

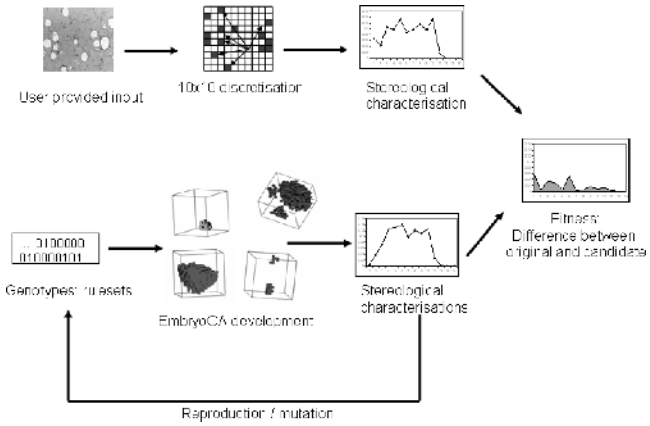


Fig. 3—A schematic overview of MicroConstructor. (top) 2-D input micrographs are discretized and measured stereologically. (bottom) A binary genotype is mapped onto a 3-D microstructure using a CA. A GA evolves this system to achieve 3-D microstructures than match the 2-D experimental microstructures stereologically.

introduced by Von Neumann and Ulam for the study of self-replication;[36] they are dynamical systems made of a discrete number of elements arranged into a lattice (normally 1-D, 2-D, or 3-D ones are used). These elements, or automata, can be in a number of discrete states and change according to a finite number of rules that determine the state of an automaton given the state of its neighbors. Time is also discrete and is divided into time-steps. The CA are interesting because even though they are simple, they can show very complex behaviors and patterns. They have been used to study electrostatic self-assembly processes,[37] pattern formation,[38] and to model microstructural evolution.[39]

There are several models of CA. Differences between CA models lie in the definition of neighborhood, whether the rules are applied synchronously or asynchronously, and the number and type of states in which an automaton can be at any given time-step. One model of CA is the effector automata (EfA). The EfA is a model of CA designed and created to evolve self-replication.[40] In the EfA model, automata are autonomous elements capable of moving, creating copies of themselves, and dying, in an otherwise empty lattice. The output of a rule in an EfA is the action to be performed by the automaton when its internal state and its configuration of neighbors are the ones specified in the rule.

The individuals in the population of the GA are rule sets of a type of EfA whose design has been inspired by developmental biology and that are called EmbryoCA. The main aims of the EmbryoCA model are to be able to grow binary 3-D spatial patterns and to be evolvable. In this work, the evolvability of a CA model is considered as the capability of the model to be effectively and efficiently modified by evolution through gradual change, a property that most CA models lack. An EmbryoCA is specified with a list of rules that have the following format:

$$\text{if (variable} = \text{number) then do consequence}$$

where variable can be either the internal variable that keeps track of the number of divisions that the automaton has gone through, or the number of neighbors in one of the six directions of a semitotalistic Moore neighborhood (north, south, east, west, up, and down).[41] There are two types of consequences in a rule: actions (move, divide, and die) and antiactions (inhibiting the automaton from either moving, dividing, or dying), as shown in Figure 4. At a given time, an automaton may have more than one applicable rule and a conflict resolution mechanism will decide what action to follow. For each time-step, every automaton follows the following algorithm.

(1) Get list of rules whose precondition is true.
(2) For every applicable rule,
    (a) if the consequence is an action, increase the counter associated with the action; and
    (b) if the consequence is an antiaction, decrease the counter associated with the action.
(3) Pick the action with the higher counter.
(4) If selected action's counter is higher than threshold, execute action.

Figure 5 shows an example of how the rule set is used with the automata in a 2-D lattice. In the example, the rule set contains only six rules (an unusually low size for a rule set). With

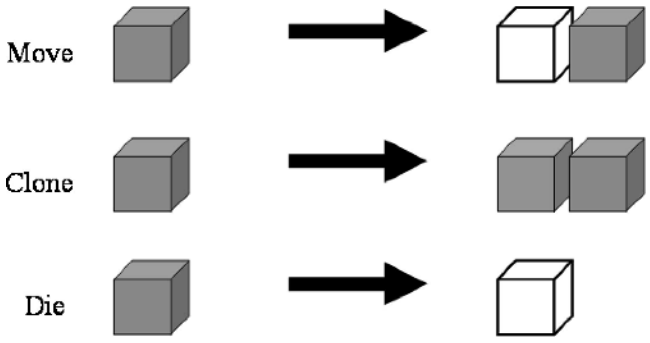


Fig. 4—The behavior of every cell in the CA is determined by its genome, a set of n rules of behavior each of which combines a condition (environmental or historical) with an action (move, clone, die).

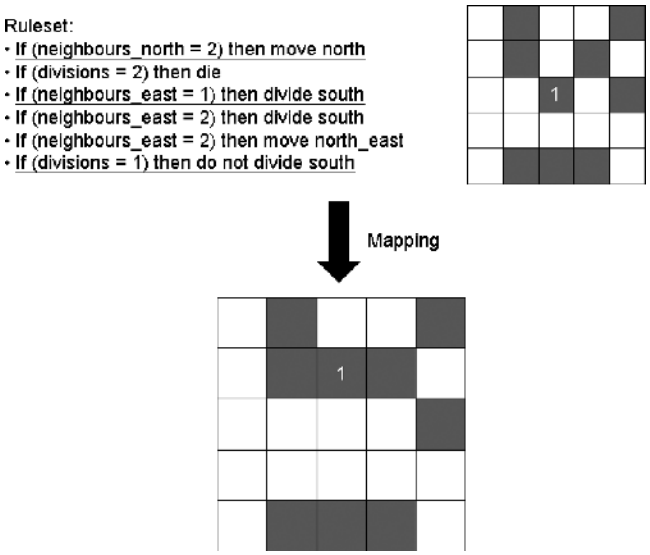


Fig. 5—Example of how rules are applied. The rule set, shown in the upper left corner, contains six rules and its effect on the automaton in the center of the 2-D lattice, upper right corner, is shown in the bottom. The rules that are active for that automaton at that time-step and given its internal variable and configuration of neighborhood are underlined. The action move north is selected after applying the conflict resolution mechanism.

that rule set and the configuration of the neighborhood and the internal variable shown in the upper right side, there are three rules that could be applied. The conflict resolution mechanism has to decide between the action move north and the action divide south, but since the third active rule cancels the action of the second one, the action of the first one is applied.

The evaluation of an EmbryoCA takes place once it has self-organized into a 3-D pattern after a number of time-steps. Figure 6 shows how the final pattern grows from a zygote (the initial starting cell). Even though the starting configuration is always the same, different rule sets grow different patterns.

As a consequence of the fact that more than one rule may be applicable at any given time by an automaton, the actions executed are not, in most cases, determined by a single rule and, therefore, a change of a few rules in the ruleset does not carry the same weight as the same change in a conventional CA. Thanks to this, small changes in the genotype of an individual translate into comparatively small changes

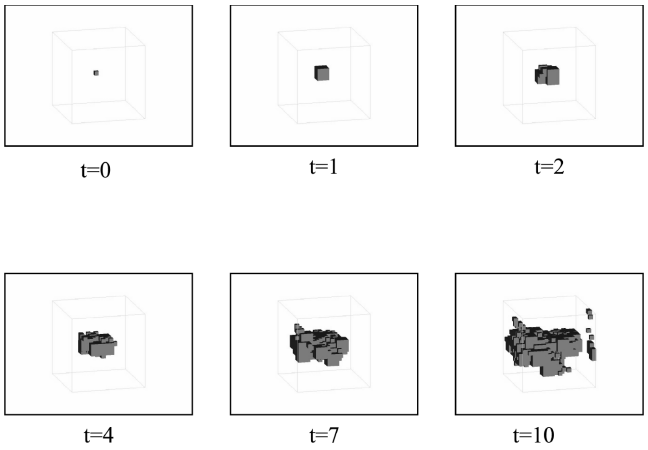


Fig. 6—An example of the development of a 3-D microstructure using an EmbryoCA. The behavior of every cell in the CA is determined by a set of rules of behavior each of which combines a condition with an action. In this case, the result of the collective behavior of the cells results in the growth of a large particle and several particle nuclei.

in the phenotype, making the EmbryoCA model more evolvable than other CA models and, therefore, better suited to be used by a GA.

B. *Fitness Function*

The fitness function of MicroConstructor is a multiobjective fitness function in which several stereological tests are performed to numerically characterize the user-provided 2-D input and the 3-D characterizations grown by the individuals of the GA population. The results of these tests provide a measure of the stereological closeness of the 2-D and 3-D microstructural characterizations. The tests that are performed use five different stereological measures: volume and area fraction, surface to volume/area fraction, two-point correlation, number of particles, and particle size distribution.

1. *Area and volume fraction*

In traditional stereology, the area fraction of a microstructural section taken from a material is calculated using a grid of points that are placed in a 2-D section to be analyzed.[42] In MicroConstructor, measuring area and volume fraction is easier since both 2-D and 3-D characterizations are in digital format. Measuring the area fraction for the β phase of the 2-D user-provided inputs consists of counting the number of pixels in the lattice that represent a β phase and dividing that number by the total number of pixels in the area represented by the image. The volume fraction of the 3-D characterizations evolved in MicroConstructor is computed using the same procedure on the 3-D lattice.

2. *Two-point correlation*

Two-point correlation functions are widely used in materials science to characterize microstructures.[43] The two-point correlation function is described in the following equation:

$$f(d) = N_s^{\frac{1}{2}} \sum_{i=0}^{N_s} n_d \qquad [1]$$

where d is the correlation distance, Ns is the total number of β-phase cells, and n_d is the number of cells of β phase that are separated at distance d from particle i.

The distributions obtained using the two-point correlation function on the 2-D input and the 3-D candidates are normalized (distance in the range [0,30]), and a new distribution, with the differences between the two, is created. The smaller the sum of the differences, the better the value of the fitness.

3. *Surface to area and surface to volume fraction*

This stereological value is a measure of the surface of the particles that belong to a specific phase against the volume (or area if the test is done on a 2-D image) of the microstructure under study. These measurements are computed using a similar method to the one described for the area and volume fraction. For the surface to area fraction, the pixels representing the α/β are counted. The total is divided by the total number of β pixels. The surface to volume fraction is computed in a similar way.

4. *Particle size distribution*

A particle size distribution is a distribution of sizes of particles per unit area. The test involving particle size distributions, also known as the Saltykov test, is more complex than other traditional stereological tests, since, using conventional stereologic methods, it involves making assumptions about the shapes of the particles in the microstructure (in most cases, it is assumed that all the particles are spheres of different radii).[44] In MicroConstructor, this test is easier to perform due to the fact that the characterizations are in digital discrete format. A list of the different β-phase particles is computed and used to create a distribution of sizes of particles in the lattice. This distribution is normalized (in the range [0,30]). The fitness is computed summing the differences between the 2-D input and the 3-D candidates distributions.

5. *Number of particles*

The list of β-phase particles in the lattice is computed, and the size of this list is used to compare 2-D and 3-D characterizations.

6. *Multiobjective fitness*

Using a multiobjective fitness function raised a number of issues about how the different criteria should be compared and weighted. The sum of weighted global ratios method has been shown to be an effective method of leading the convergence toward a solution in a GA when multiple criteria are involved.[45] Using this technique, the fitness of an individual is not measured immediately after the phenotype is developed but after having developed all the individuals in the population. The fitness of the individual for each of the objectives is normalized using the maximum and the minimum found during the run of the GA, as shown in Eq. [2].

$$\text{Norm}(\text{Fitness}_i) = \frac{\text{Fitness}_i - \text{minFitness}}{\text{maxFitness} - \text{minFitness}} \qquad [2]$$

where $\text{Norm}(\text{Fitness}_i)$ is the normalized fitness for a given objective that will be used in subsequent stages of the GA, Fitness_i is the temporal fitness for the objective calculated after measuring different features of the lattice associated to an individual, maxFitness is the maximum value for the

objective found so far in the current run of the GA, and minFitness is the minimum.

The total fitness of an individual is the weighted sum of the values of the individual in all the five criteria. All of them are considered equally important so they are all weighted by 1 to 5 in the total fitness. The overall fitness is measured in a scale of 0 to 1, where 1 would represent an exact stereological match between the 2-D input and 3-D evolved microstructures.

IV. EXPERIMENTS

We used six examples of two-phase microstructures to test MicroConstructor: (1) a-phase matrix with a single large β-phase particle; (2) a-phase matrix with a single small β-phase particle; (3) a-phase matrix with two β-phase particles separated by a large distance; (4) α-phase matrix with two β-phase particles separated by a small distance; (5) α-phase matrix with three β-phase particles; and (6) α-phase matrix with seven β-phase particles of different radii. The inputs and their stereological parameters are shown in Figures 7(a), 8(a), 9(a), 10(a), 11(a), and 12(a).

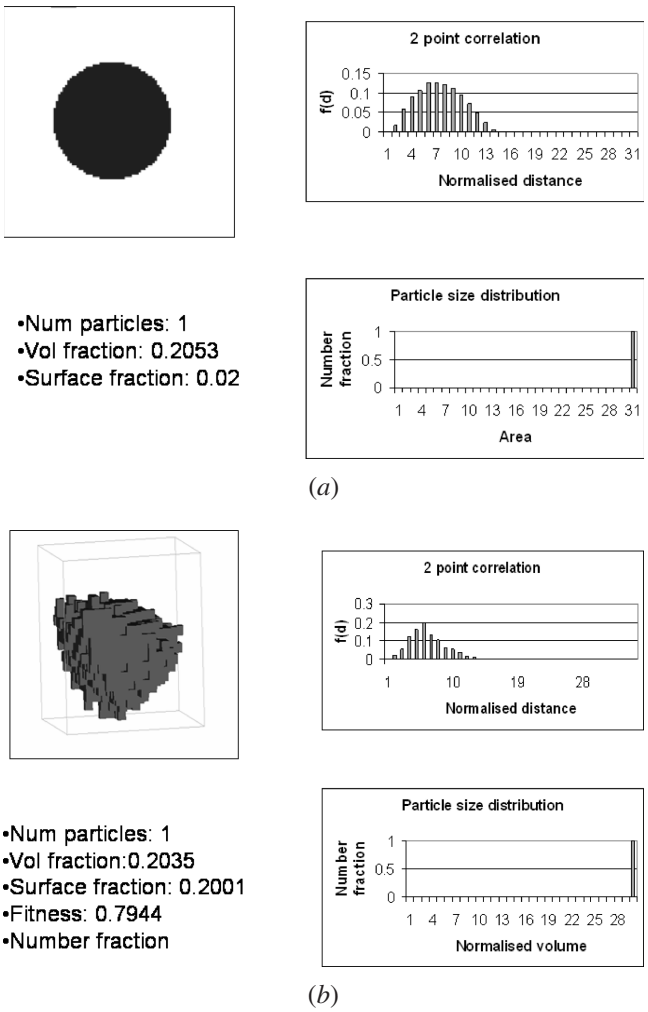


Fig. 7—Input and output structures, and stereological parameters for experiment 1: (*a*) 2-D input and (*b*) 3-D evolved structure.

In each case, MicroConstructor evolved EmbryoCA with 500 rules that grow 3-D microstructures of size $20 \times 20 \times 20$. The initial 500 rules were randomly generated at the beginning of each experiment. In each generation, 90 pct of the individuals were created by crossing over the fittest individuals of the previous generation, whereas the remaining 10 pct were taken directly from the 10 pct best candidates of the previous generation (elitism). The full parameter specifications for each experiment are detailed in Table I.

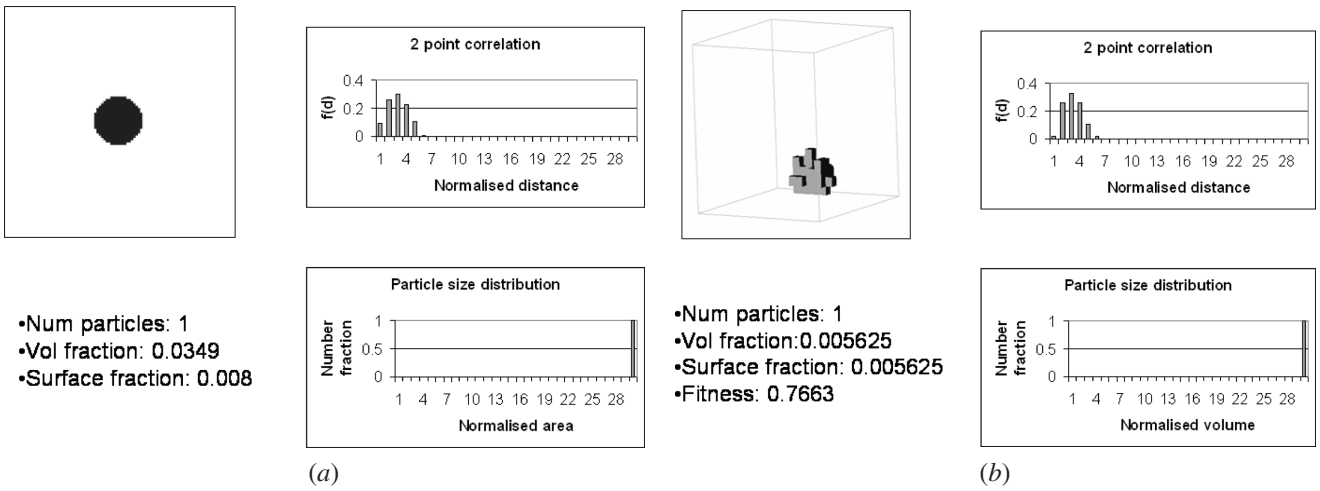
V. RESULTS

For each of the six experiments, 100 solutions were evolved for 500 generations. The best solutions are shown in Figures 7(b), 8(b), 9(b), 10(b), 11(b), and 12(b). In each case, MicroConstructor managed to reconstruct a 3-D microstructure that is stereologically similar to the inputs provided, although in none of the cases was an exact stereological match obtained.

a. In the first experiment (Figure 7(b)), the best output has a single β-phase particle, with the same particle size as the input, an identical surface area fraction, and a similar two-point correlation function. It did not match the surface fraction. The overall fitness was 0.7944.
b. In the second experiment (Figure 8(b)), MicroConstructor constructed a 3-D microstructure with a similar two-point correlation function, equal particle size distribution, and perfect match in the number of particles. The match in the volume fraction criterion was not as good and it failed to match the surface to area/volume fraction. The overall fitness in this case was 0.7663 (note that fitness value cannot be compared between experiments because we use weighted global ratios).
c. In the third experiment (Figure 9(b)), MicroConstructor fared less well, providing two β-phase particles, and a similar two-point correlation function, but the other three stereological criteria did not match. The overall fitness was 0.7632.
d. In the fourth experiment (Figure 10(b)), MicroConstructor managed to evolve a solution with reasonable two-point correlation function, particle size distribution, and equal number of particles. It did not match the volume/area fraction and the surface to area/volume fraction. The total fitness was 0.7837.
e. In the fifth experiment (Figure 11(b)), MicroConstructor evolved a solution that matched the two-point correlation

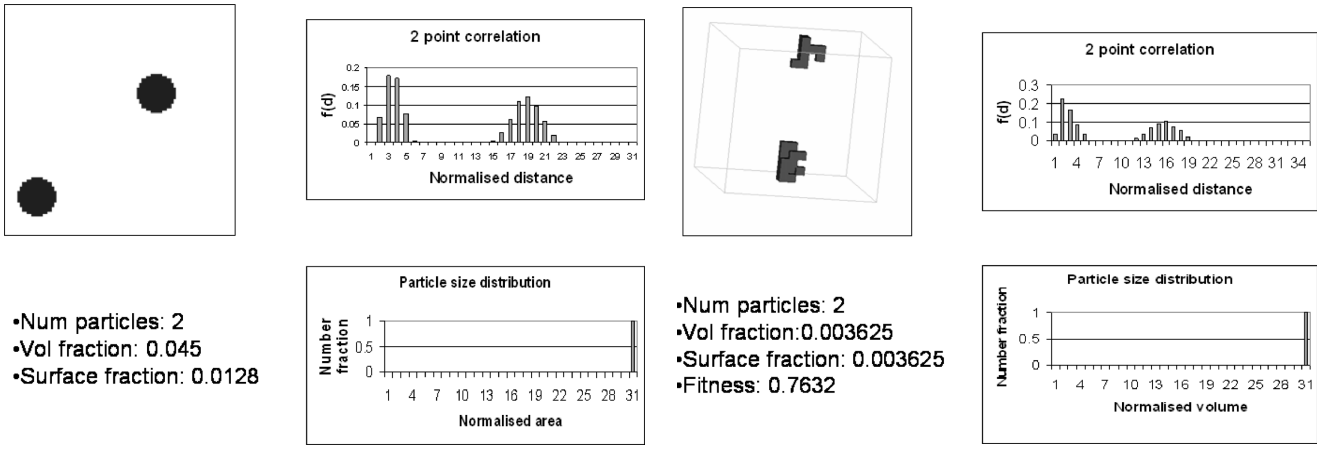
Table I. Setup Used for the Experiments

Feature	Value
CA dimensions	$20 \times 20 \times 20$
Mutation rate	0.05
Rule set size	500
Population size	100
Selection	tournament (3)
Maximum number generations	500
Crossover	2 point
Time-steps to evaluate a CA	30
Elitism	10 pct of population



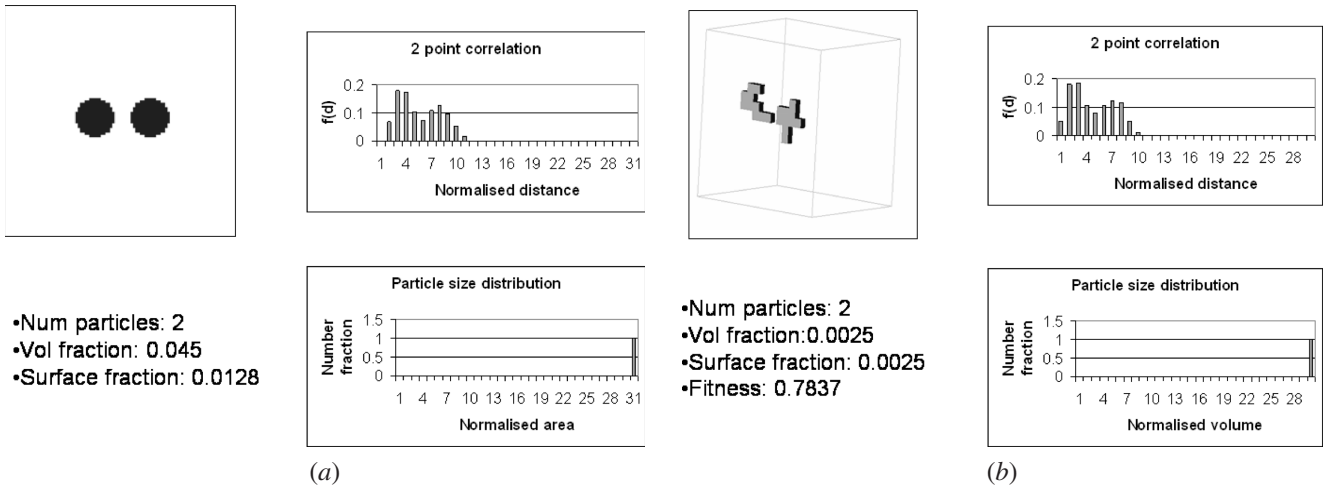
(*a*) (*b*)

Fig. 8—Input and output structures, and stereological parameters for experiment 2: (*a*) 2-D input and (*b*) 3-D evolved structure.



(*a*) (*b*)

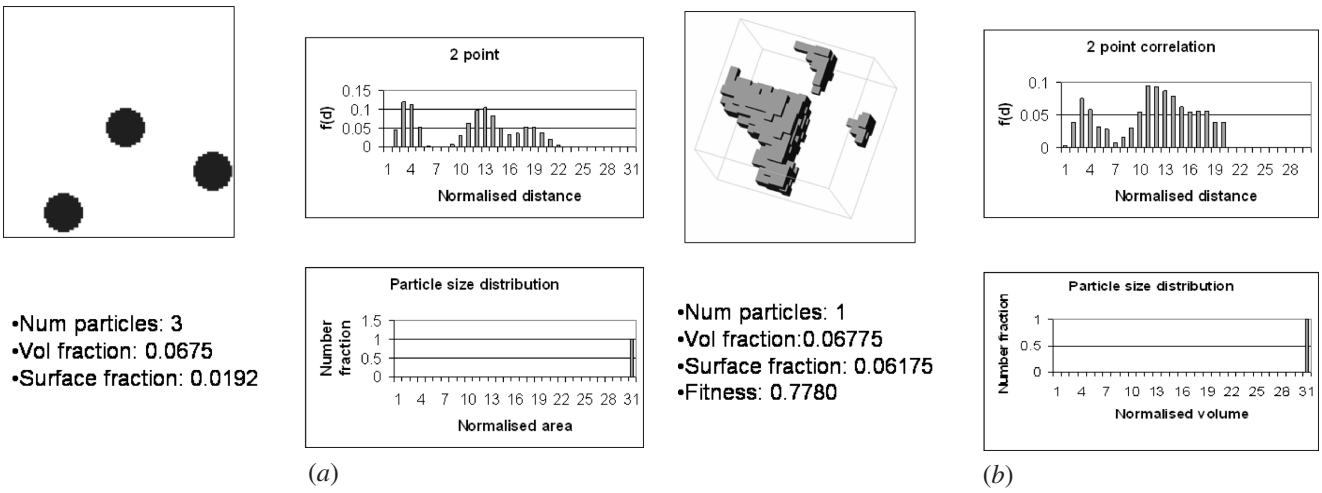
Fig. 9—Input and output structures, and stereological parameters for experiment 3: (*a*) 2-D input and (*b*) 3-D evolved structure.



(*a*) (*b*)

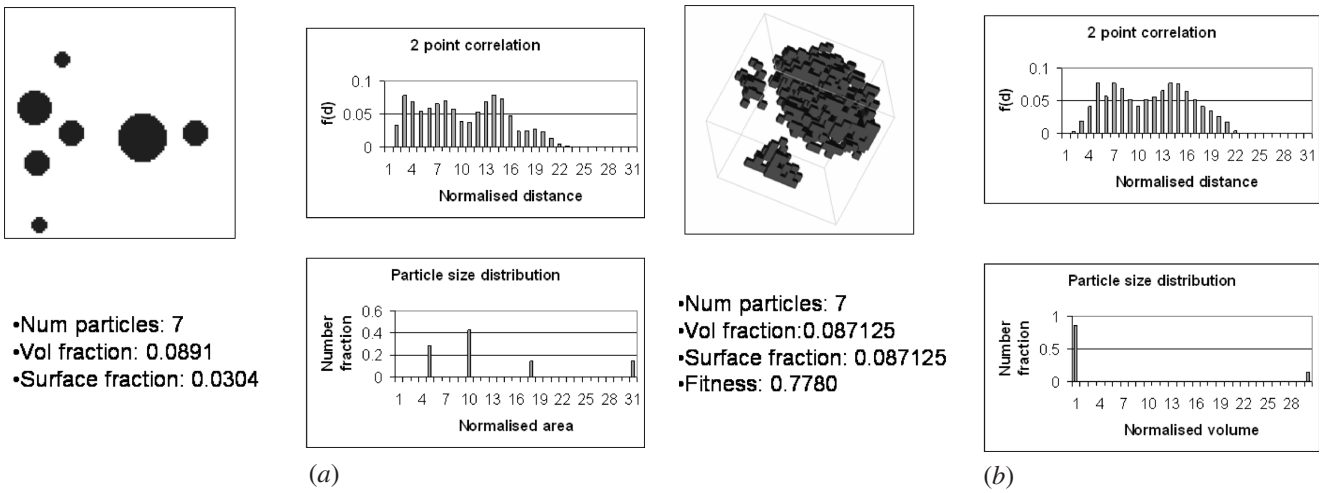
Fig. 10—Input and output structures, and stereological parameters for experiment 4: (*a*) 2-D input and (*b*) 3-D evolved structure.

function and volume/area fraction but did not match the other stereological criteria. Notably, despite the appearance of the figure, it failed to evolve the right number of particles (periodic boundary conditions are used in the CA, and as a result, the particles all touch *via* boundary wrap). The fitness of the solution is 0.7780.



(*a*) (*b*)

Fig. 11—Input and output structures, and stereological parameters for experiment 5: (*a*) 2-D input and (*b*) 3-D evolved structure.



(*a*) (*b*)

Fig. 12—Input and output structures, and stereological parameters for experiment 6: (*a*) 2-D input and (*b*) 3-D evolved structure.

f. In the sixth experiment (Figure 12(b)), MicroConstructor managed to create a solution with exactly the same number of β-phase particles, a similar distribution of the β phase (as measured by the two-point correlation criterion), and very close match in the area/volume fraction. Despite the good results obtained by the solution in these three criteria, it fared less well in terms of surface fraction and particle size distribution. The fitness of the solution is 0.7780.

VI. ANALYSIS

Experiments 1 and 2 show that MicroConstructor can evolve the ability to construct particles of different sizes. It is clear there is a great deal of degeneracy in the stereological characterization of fitness (there being the possibility of many 3-D structures that match). More importantly, from an experimental point of view, the problem is ill defined because it relies on a single 2-D input, which by definition excludes information from two other independent orthogonal sections that characterize a real microstructure. We anticipate that multiple inputs will be required, such as longitudinal and transverse sections, to improve the performance of the algorithm. In the future, we also hope to systematically investigate the range of particle sizes and shapes that the algorithm is capable of constructing.

Experiments 3 and 4 show that MicroConstructor can control interparticle distance. It has performed this task using a fitness function in which interparticle distance is not explicitly defined. Instead, by matching the two-point correlation function, it arrives at a solution that matches the interparticle distance, showing that interparticle distance is implicitly encoded in the two-point correlation function. It is interesting to note that within the EmbryoCA, cells have no measure of their distance from each other and there is no internal variable that measures distance. Thus, interparticle distance is controlled within the CA solely by controlling cell death and cloning direction.

Experiments 5 and 6 explore the ability of MicroConstructor to evolve arrays of particles. The limited success of these simulations shows one aspect of the scalability and generality of the method. In the current experiments, we used a small simulation volume of $20 \times 20 \times 20$, but since the EmbryoCA rule sets are independent of simulation size, increasing the size of the evolved structures will

not increase the genome size. This is a major advantage over the direct mapping methods where each 3-D voxel must be encoded in the genome, and genome size varies with the cube of simulation size. Small genome sizes are useful because they require less computer memory and thus enable larger populations of solutions to be evolved. Further scalability of the EmbryoCA can be obtained by distributing initial cells (zygotes) within the simulation volume. Thus, a particle distribution in a very large (*e.g.*, $1000 \times 1000 \times 1000$) simulation volume may be evolved using the same 500-rule genome, but applied to n zygotes distributed in the simulation volume.

VII. DISCUSSION

Clearly, there are two key elements to evolving 3-D microstructures from 2-D sections. The first involves calculating a representative fitness function that characterizes a 3-D microstructure from a 2-D section. The second concerns the means of construction, in our case, a CA that will grow a microstructure whose stereological properties match the fitness function. We will comment on both of these features of the current work.

From the analysis of these experiments, it is clear that some of the stereological measures used in the fitness function conflict when translated directly from 2-D to 3-D, for instance, the number of particles and the volume fraction. Assuming a random microstructure and a random 2-D section, the area fraction should exactly equal the 3-D volume fraction. However, the number of particles in a random 2-D section need not equal the total number in the 3-D volume. In fact, one would expect that the 3-D volume should contain more particles than any representative 2-D section. This means that it is not reasonable to expect to match the volume fraction, number of particles, and particle size distribution simultaneously. We see this in the results in that 100 pct fitness is never obtained. This issue highlights the problem not including the interdependence of stereological measures in the multiobjective fitness function.

Yeong and Torquato have also shown that two-point correlation functions are not sufficient to reconstruct 3-D random media.[23] We have observed this is our previous work,[26,27] and this is why we combined the five stereological parameters in a multiobjective fitness function in this current work. However, it is clear that our multiobjective fitness function needs improving and we intend to investigate adding the lineal-path functions used by Yeong and Torquato to our multiobjective fitness function.

It should be stressed that the initial EmbryoCA rules are randomly created; thus, the algorithm starts with no inherent ability to construct particles or to change their size, or to arrange them at different distances from each other. Only evolution guided by the fitness function produced these CA that can grow two-phase microstructures with different morphologies. The EmbryoCA in itself is very simple; like many CA, it can produce complex patterns from simple local rules. The key feature of the EmbryoCA that distinguishes it from other CA is that the rule set (genome) was designed specifically with the feature that small changes in the rule set lead to small changes in the developed microstructure. It is this feature that gives the EmbryoCA the ability to evolve complexity.[46] This "insensitivity" of the EmbryoCA is primarily due to the use of the multiple rule set with conflict resolution (Section III–A).

Despite the sophistication of our EmbryoCA approach, the results we have shown here fall short of those demonstrated by Yeong and Torquato.[23] This is disappointing, but we feel we have not yet fully explored the potential of our approach to navigate phenotypic space for a wide range of microstructures. We are particularly optimistic about the ability of the EmbryoCA approach to generate polycrystalline microstructures, an important category of microstructure that is difficult to reconstruct by a voxel swapping approach. Genetic algorithms are well known to be efficient search algorithms for multiobjective problems, especially for complex search spaces such as the phenotypic search space of microstructures. One of their advantages is that they evolve populations of solutions and so have less propensity to get trapped in areas of phenotypic search space that might be characterized as local minima.[24]

VIII. CONCLUSIONS

We have shown with these preliminary results that MicroConstructor can evolve 3-D two-phase microstructures, in which particle size, interparticle distance, and particle distribution can all be controlled. These are all parameters that not only characterize many different types of microstructures, but they are also known to influence materials properties in a wide variety of materials, *e.g.*, particle-strengthened aluminum alloys. Thus, we have shown that this novel method has great potential to provide a tool for materials modelers to create 3-D microstructures that are stereologically equivalent to experimental 2-D micrographs.

ACKNOWLEDGMENTS

The principal researcher, programmer, and problem solver was David Basanta. The research was directed by Mark Miodownik and Peter Bentley, and produced by Elizabeth Holm. The authors thank all of the management, administration, and IT staff of the Engineering Division, King's College London; the Computer Science Department, University College London; and Sandia National Laboratories for providing the research infrastructure and support. This work was performed in part at Sandia National Laboratories, a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract No. DE-AC04-94AL85000.

REFERENCES

1. D. Raabe: *Computational Materials Science: The Simulation of Materials, Microstructures and Properties*, Wiley, Weinheim, 1998.
2. *Proc. 1st Joint Int. Conf. on Recrystallization and Grain Growth*, G. Gottstein, ed., Trans Tech Publications, Aedermannsdorf, Switzerland, 2001.
3. M.A. Miodownik, A. Godfrey, E.A. Holm, and D.A. Hughes: *Acta Mater.*, 1999, vol. 47, p. 2661.
4. M.V. Kral, M.A. Mangan, G. Spanos *et al.: Mater. Characterization*, 2000, vol. 45, pp. 17-23.
5. J.E. Spowart, H.M. Mullens, and B.T. Puchala: *JOM-J. Min. Met. Mater.*, 2003, vol. 55, pp. 35-37.

6. M.S. Li, S. Ghosh, T.N. Rouns *et al.: Mater. Characterization*, 1998, vol. 88, pp. 73-83.
7. M. Li, S. Ghosh, O. Richmond, H. Weiland, and T.N. Rouns: *Mater. Sci. Eng. A-Struct*., 1999, vol. 265, pp. 153-73.
8. M.V. Kral and G. Spanos: *Acta Mater*., 1999, vol. 265, pp. 711-24.
9. C.Y. Hung, G. Spanos, R.O. Rosenberg, and M.V. Kral: *Acta Mater*., 2002, vol. 50, pp. 3781-88.
10. N.C.W. Kuijpers, J. Tirel, D.N. Hanlon *et al.: Mater. Characterization*, 2002, vol. 48, pp. 379-92.
11. T. Yokomizo, M. Enomoto, O. Umezawa, G. Spanos, and R.O. Rosenberg: *Mater Sci. Eng. A-Struct*., 2003, vol. 344, pp. 261-67.
12. V. Randle and R.O. Engle: *Introduction to Texture Analysis: Macrotexture, Microtexture and Orientation Mapping*, Gordon and Breach, London, 2000.
13. V. Randle and H. Davies: *Ultramicroscopy*, 2002, vol. 90, pp. 153-62.
14. M.A. Wall, A.J. Schwartz, and L. Nguyen: *Ultramicroscopy*, 2001, vol. 88, pp. 73-83.
15. S.F. Nielsen, E.M. Lauridsen, D. Juul Jensen, and H.F. Poulsen: *Mater. Sci. Eng. A*, 2001, vols. 319–321, pp. 179-81.
16. X. Fu, H.F. Poulsen, S. Schmidt *et al.: Scripta Mater*., 2003, vol. 49, pp. 1093-96.
17. E.M. Lauridsen, S.F. Nielsen, L. Margulies *et al.: J. Phys. IV*, 2003, vol. 104, pp. 495-98.
18. S. Torquato: *Random Heterogeneous Materials*, Springer-Verlag, Berlin, 2001.
19. S. Torquato: *Ann. Rev. Mater. Res*., 2002, vol. 32, pp. 77-111.
20. P.M. Adler: *Porous Media—Geometry and Transports*, Butterworth-Heinemann, London, 1992, p. 24.
21. M.D. Rintoul and S. Torquato: *J. Coll. Interface Sci*., 1997, vol. 186, pp. 467-76.
22. C.L.Y. Yeong and S. Torquato: *Phys. Rev. E*, 1998, vol. 57, pp. 495-506.
23. C.L.Y. Yeong and S. Torquato: *Phys. Rev. E*, 1998, vol. 58, pp. 224-33.
24. C. Manwart, S. Torquato, and R. Hilfer: *Phys. Rev. E*, 2000, vol. 62, pp. 893-99.
25. N. Sheehan and S. Torquato: *J. Appl. Phys*., 2001, vol. 89, pp. 53-60.
26. D. Basanta, M.A. Miodownik, E.A. Holm, and P.J. Bentley: *Computer-Based Design*, Engineering Design Conf. 2002, Tamer Shahin, ed., Professional Engineering Publishing Ltd., London, 2002, pp. 349-55,
27. D. Basanta, P.J. Bentley, M.A. Miodownik, and E.A. Holm: *Genetic Programming*, Proc. 6th Eur. Conf., EuroGP 2003, Springer, Berlin, 2003, pp. 1-10.
28. Charles Darwin: *The Origin of Species by Means of Natural Selection*, Penguin, London, 1868.
29. David E. Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Berlin, 1989.
30. Melanie Mitchell: *An Introduction to Genetic Algorithms*, The MIT Press, Cambridge, MA, 1998.
31. P. J. Bentley: *Evolutionary Design by Computers*, Morgan Kaufmann, San Francisco, CA, 1999.
32. S. Kumar and P.J. Bentley: On Growth, Form and Computers, Elsevier Academic Press, London, 2003.
33. S. Kumar and P.J. Bentley: in *On Growth, Form and Development*, S. Kumar and P.J. Bentley, Academic Press, New York, NY, 2003.
34. P. Ball: *The Self Made Tapestry*, Oxford University Press, Oxford, United Kingdom, 1999.
35. L. Wolpert: *Principles of Development*, 2nd ed., Oxford University Press, Oxford, United Kingdom, 2002.
36. J. von Neumann: *Theory of Self-Reproducing Automata*, University of Illinois Press, 1966.
37. J.B. Spillman, T. Zeng, and R.O. Claus: *Smart Mater. Struct*., 2002, vol. 11, pp. 623-30.
38. Julian Miller: *ECAL 2003*, Dortmund, Germany, 2003.
39. D. Raabe: *Ann. Rev. Mater. Res*., 2002.
40. J. Lohn and J. Reggia: *1995 IEEE Int. Conf. on Evolutionary Computing*, 1995.
41. Edward F. Moore: *American Mathematical Society Proc. Symp. in Applied Mathematics*, 1962, vol. 14, pp. 17-33.
42. John C. Russ and Robert T. Dehoff: *Practical Stereology*, 2nd ed., Kluwer Academic, New York, NY, 2000.
43. T.A. Mason and B.L. Adams: *Metall. Mater. Trans. A*, 1999, vol. 30A, pp. 969-79.
44. Ervin E. Underwood: *Quantitative Stereology*, Addison-Wesley, Reading, MA, 1970.
45. P.J. Bentley and J.P. Wakefield: *Soft Computing in Engineering Design and Manufacturing*, P.K. Chawdhry, R. Roy, and R.K. Pant, eds., Springer-Verlag London Limited, London, 1997, Part 5, pp. 231-40.
46. D. Basanta, M.A. Miodownik, P.J. Bentley, and E.A. Holm: "Investigating the Evolvability of Biologically Inspired Cellular Automata," 2004, in press.