Available online at www.sciencedirect.com
ScienceDirect
Computational Materials Science 42 (2008) 352–367
COMPUTATIONAL MATERIALS SCIENCE
www.elsevier.com/locate/commatsci

Reconstruction of periodic unit cells of multimodal random particulate composites using genetic algorithms

Natarajan Chennimalai Kumar [b], Karel Matouš [a,b,*], Philippe H. Geubelle [b]

[a] Computational Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA[1]
[b] Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Received 2 June 2007; received in revised form 14 July 2007; accepted 18 July 2007
Available online 17 September 2007

Abstract

We develop a procedure for characterization and reconstruction of periodic unit cells of highly filled, multimodal, particulate composites. *Rocpack*, a particle packing software, is used to generate the solid propellant microstructures and one- and two-point probability functions are used to describe its statistical morphology. The reconstruction is carried out using a parallel Augmented Simulated Annealing algorithm with a novel mutation operator based on a mass–spring system to eliminate overlap and improve the code performance. Results from the reconstruction procedure, for four-phase random particulate composites of 40–70% packing fraction, are detailed to demonstrate the capabilities of the reconstruction model. The presented results suggest good convergence and repeatability of the optimization scheme, even for high volume fractions, and good scalability of the algorithm.

Keywords: Microstructure reconstruction; Particle packing; Probability functions; Parallel genetic algorithm; Solid propellant

1. Introduction

It has long been recognized that the macroscopic properties of a heterogeneous material depend strongly on its microstructure and a wide range of approaches have been proposed that take this information into account [10,9,26,16,34,27]. Some of these approaches use simple statistics, such as volume fraction [16], while others try to simplify the problem by replacing a complex microstructure by a simple hexagonal array of phases [27] or by a periodic unit cell (PUC) [6,18]. It is essential, however, to realize the inherent randomness of the system to describe its macroscopic properties. Several procedures have been proposed to characterize and reconstruct the microstructure depending on an application: Povirk [21] suggested replacing the complex two-phase particulate microstructure with a simple, yet random periodic unit cell that has similar statistical characteristics as the original microstructure. Yeong and Torquato introduced a numerical "microstructure reconstruction" procedure [31,32]. Kumar et al. [14] used similar scheme within the simulated annealing (SA) framework for ductile iron metals. Sundararaghavan and Zabaras [24] developed a support vector machine framework for microstructure classification and reconstruction from limited statistical information. Quintanilla and Max Jones used convex quadratic programming to model random media with Gaussian random fields [22]. Zeman [33], Matouš et al. [19], and Zeman and Šejnoha [34] applied genetic algorithms to reconstruct PUC of two-phase fibrous composites.

Traditionally, the microstructures of random composites are studied from micrographs or topographic data at a high level of magnification. Although very efficient, such methods require sometimes costly or complicated experiments

[1] ATK/Thiokol, ATK-21316; US Department of Energy, B523819.
* Corresponding author. Address: Computational Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA. Tel.: +1 2173338448.
E-mail addresses: chennima@uiuc.edu (N.C. Kumar), matous@uiuc.edu (K. Matouš), geubelle@uiuc.edu (P.H. Geubelle).
URL: http://www.csar.uiuc.edu/~matous/ (K. Matouš).

0927-0256/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.commatsci.2007.07.043

or image processing. Therefore, computer-generated microstructures, which are statistically identical, started to become more popular. For example, Anderson et al. [1] proposed a grain growth model suitable for polycrystalline materials. For heterogeneous solid propellants, Webb and Lee Davis [30] developed a reduced-dimension particle packing algorithm based on a ballistic deposition model. Another packing algorithm that is dynamic in nature, which can pack spheres or discs of arbitrary sizes, called *Rocpack*, was introduced by Kochevets et al. [13] and Knott et al. [12]. *Rocpack* has been extensively tested and validated against available experimental data [12,18]. In order to be representative of an actual material, such computer-generated packs must be large. Two examples, one in three dimensions and one in two dimensions, are shown in Fig. 1. These domains are, however, too big to be fully numerically resolved, when complex nonlinear processes such as decohesion and matrix tearing are modeled. The aforementioned space dimension reduction techniques are, thus, important to reduce the problem size while preserving the cell statistics.

The objective of this work is to study multimodal, highly filled, random particulate composites, such as solid propellants, and to reconstruct the PUC, which is geometrically identical to the original packs generated by packing algorithm (such as *Rocpack*) or obtained experimentally. The high content of energetic crystals (~60–80%) and large particle size variations make the reconstruction complicated and most of the techniques discussed above are likely to fail. Therefore, we develop and implement a highly efficient and parallel reconstruction scheme based on the Augmented Simulated Annealing (AUSA) method, which attempts to reconstruct a PUC of a m-phase particulate random composite using one- and two-point probability functions. In order to eliminate the particle overlap and its influence on the overall performance of the optimization scheme, we develop the problem-specific mutation operator and incorporate it into the Genetic Algorithm (GA) strategy.

The paper is organized as follows: Section 2 details the formulation and numerical determination of the statistical descriptors (probability functions) used in the present study to characterize the microstructural statistics. One- and two-point probability functions are used to describe the statistical details of the particulate composite microstructure and the ergodicity, homogeneity and statistical isotropy assumptions are numerically investigated. The sampling template technique proposed in [25] is used here to numerically determine the two-point probability functions. Section 3 describes the optimization procedure to reconstruct the PUC. The optimization involves formulation of suitably selected objective functions. To account for the particle overlap, a penalty, or constraint function, is formulated and a new problem-specific mutation operator, based on a mass–spring system, is introduced. The parallel implementation, crucial for characterization and reconstruction of large domains with high particle volume fractions and particle size variations, is also discussed. Section 4 summarizes the reconstruction procedure results for two-dimensional, four-phase, random composites of different particle packing fractions (from 40% to 70%).

2. Statistical morphology of random particulate composites

Theoretical foundations to quantify the statistical characteristics of heterogeneous materials have been laid out in the past 40 years, starting from the seminal work of Beran [2]. A number of other works address this topic in detail as well [3,4,28,29,34]. For the sake of completeness, we summarize here the fundamentals of a class of statistical descriptors – the n-point probability functions, which are used in the present study to characterize the microstructure and to reconstruct the PUC. Consider a collection of a large number of micrographs (that form the ensemble space $\mathscr{S}$) describing the geometry of a random particulate material, as shown in Fig. 2. To provide an initial statistical description of a random material, a function called the

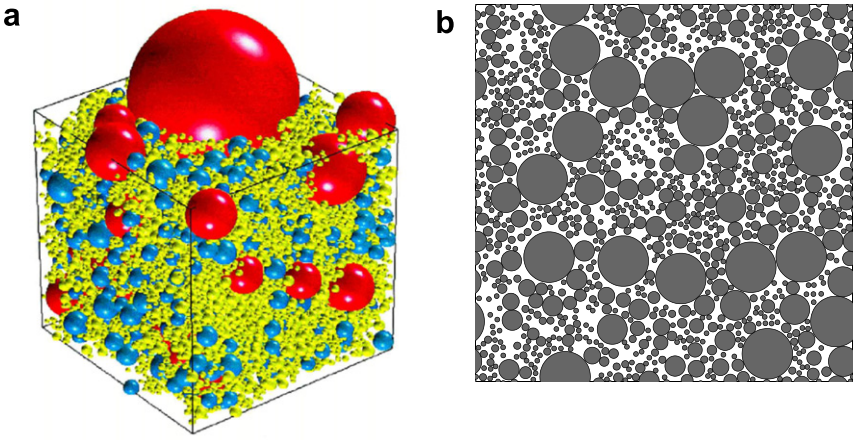

Fig. 1. Examples of (a) 3D and (b) 2D solid propellant packs generated by Rocpack.

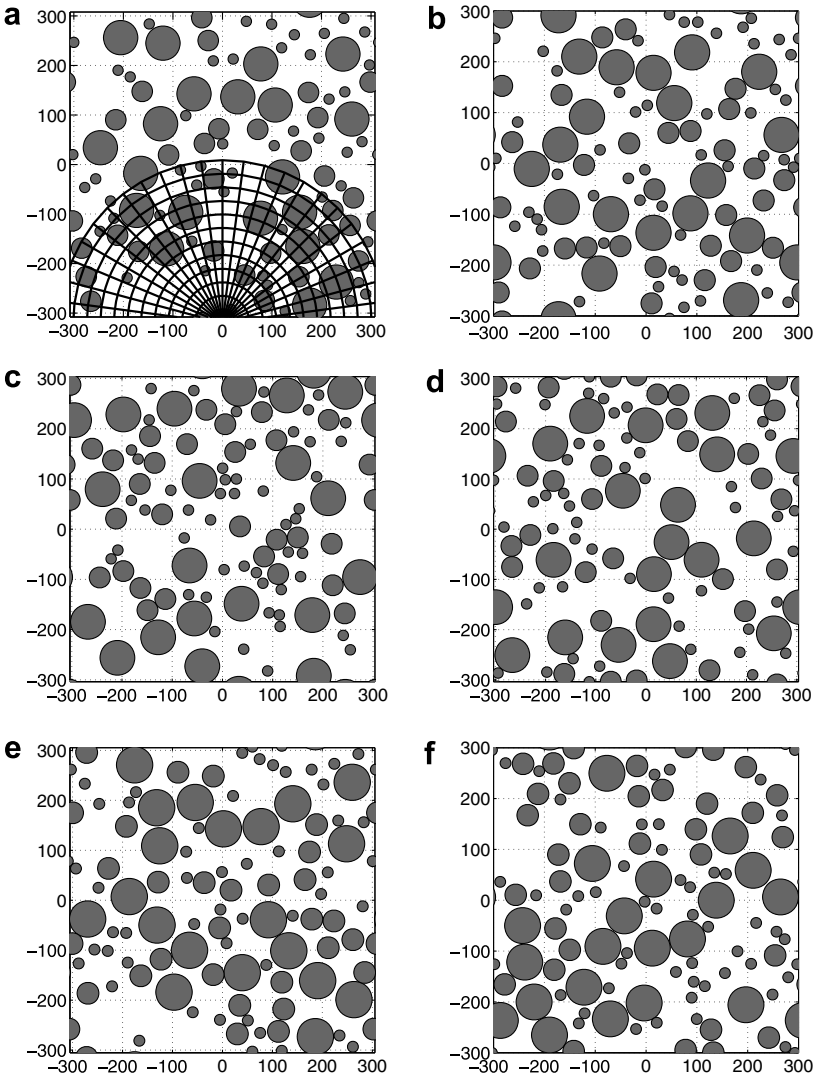

Fig. 2. Members of the ensemble space, $\mathscr{S}$, for a four-phase random particulate composite with ~40% packing fraction generated by *Rocpack*. The first pack also shows half of the sampling template used to extract the two-point probability functions S_{rs}.

indicator function, or the characteristic function $\chi_r(\boldsymbol{x},\alpha)$, is defined as follows:

$$\chi_r(\boldsymbol{x},\alpha)=\begin{cases}1 & \boldsymbol{x}\in D_r(\alpha),\\ 0 & \text{otherwise},\end{cases} \tag{1}$$

where $D_r(\alpha)$ denotes the domain occupied by rth phase, $\boldsymbol{x}$ is a point thrown into the random medium and α is the member of the ensemble onto which the point is thrown. From this definition, the *n-point probability function* $S_{r_1,r_2,\ldots,r_n}$, which gives the probability of finding phases r_1, $r_2,\ldots,r_n$ simultaneously at points $\boldsymbol{x}_1$, $\boldsymbol{x}_2,\ldots,\boldsymbol{x}_n$, is given by

$$S_{r_1,\ldots,r_n}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)=\overline{\chi_{r_1}(\boldsymbol{x}_1)\ldots\chi_{r_n}(\boldsymbol{x}_n)}, \tag{2}$$

where the overbar indicates the ensemble average:

$$\overline{\chi_r(\boldsymbol{x})}=\int_{\mathscr{S}}\chi_r(\boldsymbol{x},\alpha)p(\alpha)\mathrm{d}\alpha. \tag{3}$$

Here $p(\alpha)$ denotes the probability density of α in $\mathscr{S}$. In particular, the one-point and two-point probability functions read

$$\begin{aligned}S_r(\boldsymbol{x})&=\overline{\chi_r(\boldsymbol{x},\alpha)},\\ S_{rs}(\boldsymbol{x},\boldsymbol{x}')&=\overline{\chi_r(\boldsymbol{x},\alpha)\chi_s(\boldsymbol{x}',\alpha)}.\end{aligned} \tag{4}$$

In general, evaluation of probability functions can be prohibitively difficult, but can be simplified if the material under consideration is ergodic, statistically homogeneous and statistically isotropic. The Ergodic hypothesis requires that all states available to an ensemble of systems must be available to every member of the system as well [2]. If the material is ergodic, then the ensemble average of a given function, $\bullet(\boldsymbol{x},\alpha)$, (such as the characteristic function) is equal to its volume average,

$$\int_{\mathscr{S}}\bullet(\boldsymbol{x},\alpha)p(\alpha)\mathrm{d}\alpha=\frac{1}{V}\int_V\bullet(\boldsymbol{x},\alpha)\mathrm{d}V. \tag{5}$$

It follows that the probability functions can be evaluated on only one member of the ensemble instead of the costly proposition of evaluation of all the probability functions on each member of the ensemble. Statistical homogeneity results in the ensemble average of the characteristic functions being independent of translation, while statistical isotropy implies directional independence. For an ergodic, statistically homogeneous, and isotropic random heterogeneous medium, the one-point and two-point probability functions reduce to

$$\begin{aligned} &S_r(\boldsymbol{x}) = S_r = c_r, \\ &S_{rs}(\boldsymbol{x}, \boldsymbol{x}') = S_{rs}(\boldsymbol{x} - \boldsymbol{x}') = S_{rs}(\|\boldsymbol{x} - \boldsymbol{x}'\|), \end{aligned} \tag{6}$$

where c_r represents the volume fraction of phase r in the composite and $\|\boldsymbol{x} - \boldsymbol{x}'\|$ represents the distance between points $\boldsymbol{x}$ and $\boldsymbol{x}'$.

To numerically determine the one-point probability function, described by Eqs. (4) and (6), respectively, a simple Monte-Carlo like simulation can be utilized, i.e., we throw randomly generated points into the microstructure and count the number of successful "hits" into the rth phase. Then, the value of the one-point probability function can be estimated as

$$S_r \to \frac{n'}{n_t}, \quad \text{as } n_t \to \infty, \tag{7}$$

where n' is the number of successful hits and n_t denotes the total number of throws. To compute the two-point probability functions, the sampling template procedure proposed by Smith and Torquato [25] is used. Instead of tossing a line corresponding to $\boldsymbol{x} - \boldsymbol{x}'$ into a medium, a circular sampling template is formed. The center of such sampling template is thrown randomly into the medium under consideration and corresponding successful hits are counted. Furthermore, if the medium is statistically isotropic, successful hits found for points located on the same circumference can be averaged as well, which allows a large number of tests to be performed within one placement of the template. One half of a typical sampling template used in the analysis is shown in Fig. 2a.

2.1. Verification of ergodicity, homogeneity and statistical isotropy

First, we set out to verify the ergodicity of numerical generated, multimodal, particulate composites. We focus on packs consisting of particles of three different radii (10, 20 and 30 μm) with ~40% packing fraction. To test the procedure for establishing the ergodic hypothesis, it is necessary to form the ensemble space $\mathscr{S}$. When sampling individual members of the ensemble space $\mathscr{S}$, random packs were generated using *Rocpack* with an approximately same volume fraction of each phase, computed directly from packs. Fig. 2 shows six such individual packs with volume fraction of 0.37–0.39, obtained by modifying the sampling window (300 ± 20 μm). This will produce slightly perturbed packs, which are used here instead of experimental micrographs. Although the initial packs and boundary conditions used for computing the statistical descriptors are periodic, it has been shown by Gajdošík et al. [7] that the boundary conditions have negligible effect on the two-point probability functions. In the view of the ergodic hypothesis, we shall require only that

$$\overline{c_r} \approx \overline{S_r} \quad \text{with } \overline{S_r} = \frac{1}{\hat{n}} \sum_{i=1}^{\hat{n}} S_r^i, \tag{8}$$

where $\hat{n}$ is the number of members in the ensemble and $\overline{c_r}$ represents the mean volume fraction of each of the phases. The individual first-order probability functions, S_r^i, were computer using 50,000 random throws. Table 1 shows the evaluated S_r^i, $\overline{S_r}$ and $\overline{c_r}$. It is evident from Table 1 that the medium can be assumed ergodic within 1% based on the one-point probability functions. We opine that the presented results are sufficient for the medium to be considered ergodic, provided that the medium is statistically homogeneous, although, an ultimate justification of an ergodic assumption would require to prove equality of higher moments as well [28]. In the sense of the ergodic assumption, we suggest that a single micrograph be used for evaluation of the required statistical descriptors.

Next, we validate statistical homogeneity on the pack shown in Fig. 2a. Since statistical homogeneity implies translational invariance of a given function, the one-point probability function is evaluated on this pack for different coordinate systems. As in previous example, 50,000 random throws were used to compute the first-order statistical moment. Thus, we compute the one-point probability functions by translating the origin at $(0,0)$ to $(-L,-L)$, $(-L/2,-L/2)$, $(-L/4,-L/4)$, $(L/4,L/4)$, $(L/2,L/2)$ and (L,L), where $2L$ is the size of the pack. Table 2 shows the values of the one-point probability function evaluated for each of the aforementioned coordinate systems, and its standard deviation. We can see that the standard deviation of the evaluated one-point probability functions for different coordinate systems is approximately 3%. Even though it is necessary to establish homogeneity on higher-order probability

Table 1
Verification of ergodicity for an ensemble of six 40% packs with three different particle radii (P1 – Particle 1, P2 – Particle 2, P3 – Particle 3 and M – Matrix)

S_r^i	Pack 1	Pack 2	Pack 3	Pack 4	Pack 5	Pack 6	$\overline{S_r}$	$\overline{c_r}$
P1	0.221	0.210	0.214	0.196	0.210	0.218	0.21150	0.20496
P2	0.111	0.117	0.106	0.095	0.112	0.114	0.10916	0.11070
P3	0.042	0.055	0.044	0.048	0.048	0.047	0.04733	0.04612
M	0.626	0.618	0.636	0.661	0.630	0.621	0.63201	0.63821

Table 2
Verification of statistical homogeneity for a 40% pack with three different particle radii (P1 – Particle 1, P2 – Particle 2, P3 – Particle 3 and M – Matrix)

S_r	Origin $(-L,-L)$	Origin $\left(-\frac{L}{2},-\frac{L}{2}\right)$	Origin $\left(-\frac{L}{4},-\frac{L}{4}\right)$	Origin $\left(\frac{L}{4},\frac{L}{4}\right)$	Origin $\left(\frac{L}{2},\frac{L}{2}\right)$	Origin (L,L)	Standard deviation
P1	0.245	0.180	0.187	0.180	0.173	0.226	0.0325
P2	0.073	0.105	0.113	0.125	0.126	0.105	0.0194
P3	0.024	0.053	0.049	0.060	0.067	0.064	0.0156
M	0.657	0.657	0.651	0.636	0.635	0.605	0.0198

functions we take the medium to be statistically homogeneous within 3%. Note that for a periodic medium the statistical homogeneity is satisfied by construction. Thus, the error obtained here is purely numerical. Nevertheless, this procedure was presented for completeness and generality of the method, when different media are investigated.

Finally, we establish statistical isotropy from the determination of the two-point probability distribution functions, using a sampling template of 500 radial points and 100,000 random throws. The two-point probability function distribution and the coefficient of variation, computed as the standard deviation of the two-point probability function in the circumferential direction, is plotted in Fig. 3a and b, respectively. As one can see, the maximum variation is <2%.

After establishing ergodicity, statistical homogeneity and isotropy, the spatial distribution of the two-point probability functions is computed as a spatial average of the two-point probability functions in the circumferential direction and is shown in Fig. 4b. Fig. 4b also attests to the following limiting cases as applicable to an ergodic, homogeneous and isotropic medium:

$$\begin{aligned} &\text{for } \boldsymbol{x} \to \boldsymbol{x}' : S_{rs}(\boldsymbol{x},\boldsymbol{x}') = \delta_{rs} S_r = \delta_{rs} c_r, \\ &\text{for } \|\boldsymbol{x}-\boldsymbol{x}'\| \to \infty : \lim_{\|\boldsymbol{x}-\boldsymbol{x}'\|\to\infty} S_{rs}(\boldsymbol{x},\boldsymbol{x}') = c_r c_s. \end{aligned} \tag{9}$$

Eq. (9a) is a manifestation of ergodicity, while Eq. (9b) refers to the length scale of statistical independence of the heterogeneous material. From Fig. 4b, it can be seen that, for distances greater than ≈150 μm, the material points are statistically independent. This length scale forms the initial guess of the PUC size in the optimization problem described in Section 3.

Please note that a discrete sampling template is used to compute the two-point probability functions of a continuum

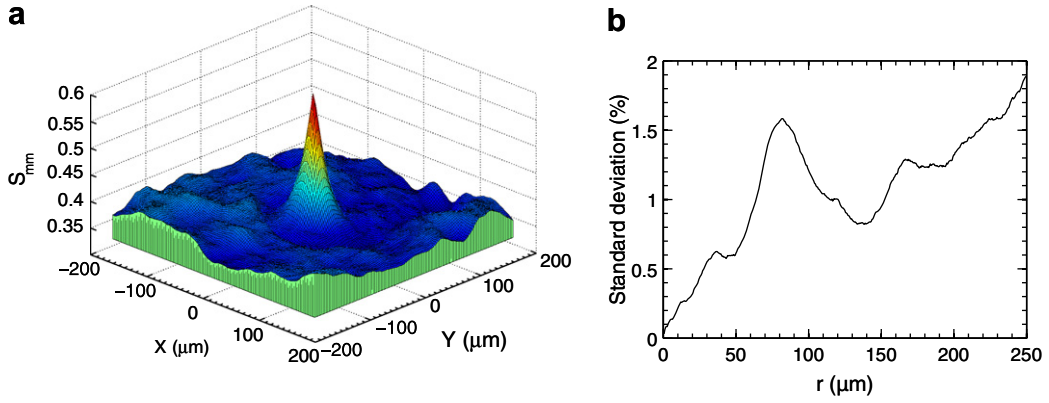

Fig. 3. (a) Two-point matrix–matrix probability function, $S_{mm}(\boldsymbol{x}-\boldsymbol{x}')$, and (b) standard deviation as a function of distance r.

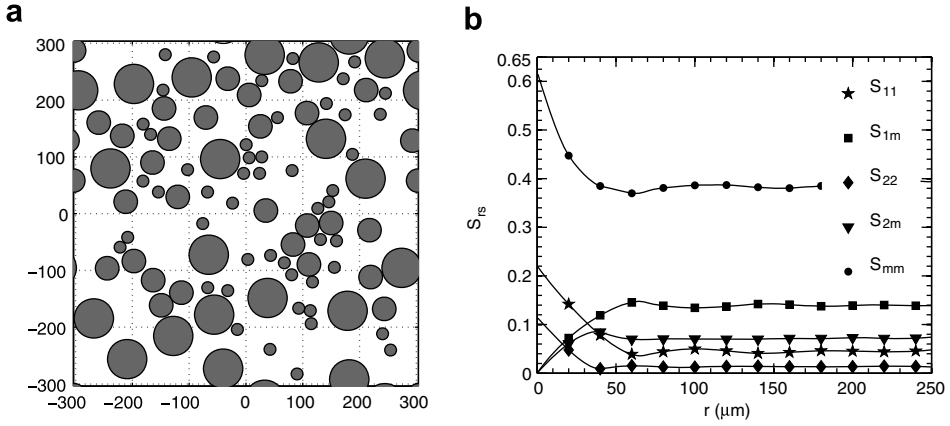

Fig. 4. (a) Solid propellant pack with 40% volume fraction. (b) Corresponding two-point probability functions, S_{rs}. Only selected probability functions are shown for clarity.

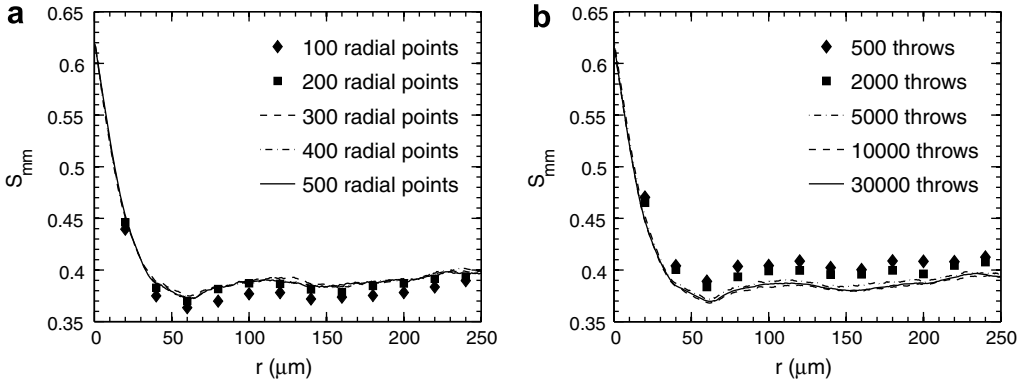

Fig. 5. Convergence of the two-point matrix–matrix probability function, $S_{mm}(r)$, with (a) number of radial points in the sampling template for fixed number of throws – 30,000 and (b) number of throws of the sampling template for fixed number of radial points – 500.

heterogeneous material. Thus, the accuracy of the evaluated statistics depends on: (a) the number of random throws of the sampling template and (b) the number of radial points in the sampling template, which can be related to the smallest particle radius. Fig. 5a and b show the dependence of the matrix–matrix two-point probability function on the number of radial points and the number of throws, respectively. It can be seen that a good convergence of the two-point probability function is obtained for 300 radial points (or a radial spacing of 1.5 μm) and 5000 throws of the sampling template.

3. Construction of the periodic unit cell (PUC)

3.1. Objective and penalty functions

Once the microstructural statistics of the heterogeneous material are established, the reconstruction of the PUC is undertaken. In this study a PUC is geometrically representative of the original microstructure provided that the one- and two-point probability functions of the PUC are similar to those from the original pack. To quantify this similarity, we introduce two optimization problems:

(a) For a given initial length scale of the PUC H_{init}, volume fractions c_i, particle radius r_i and number of particles N_i of phase i, find the PUC dimension H, such that the differences in the volume fractions are minimized using the following objective function:

$$G(H) = \sqrt{\sum_{i=1}^{n_p} \left(c_i - \frac{N_i \pi r_i^2}{H^2} \right)^2}, \tag{10}$$

where n_p is the number of particle phases in the composite. This optimization problem can be solved analytically to get the optimal dimensions of the PUC to be:

$$\frac{dG}{dH} = 0 \Rightarrow H = \sqrt{\frac{\pi \sum_{i=1}^{n_p} N_i^2 r_i^4}{\sum_{i=1}^{n_p} N_i r_i^2 c_i}}. \tag{11}$$

This step in the optimization procedure ensures the similarity of the one-point probability functions, since the size of the PUC is based on the respective particle volume fractions, which are the same as the corresponding one-point probability functions.

(b) After determining the optimal length scale of the PUC using Eq. (11), the optimal positions of the particle centers are found by minimizing the following objective function, based on the two-point probability distributions:

$$\begin{aligned} F(\boldsymbol{x}_N) &= \sum_{r=1}^{m} \sum_{s=1}^{m} \| S_{rs}^{\text{orig}} - S_{rs} \|_{L_2} \\ &= \sum_{r=1}^{m} \sum_{s=1}^{m} \sqrt{\int_0^H (S_{rs}^{\text{orig}} - S_{rs})^2 \, dr}, \end{aligned} \tag{12}$$

where $\boldsymbol{x}_N = \{x_1, y_1, \ldots, x_N, y_N\}^T$ is the vector of the positions of particle centers, x_i and y_i correspond to the x- and y-coordinates of the ith particle, respectively, m is the total number of phases in the composite that is equal to $n_p + 1$, S_{rs}^{orig} are the two-point probability functions computed for the original microstructure and S_{rs} are the two-point probability functions computed for the PUC. It is to be noted here that the objective function, as formulated in Eq. (12), does not contain any details on the overlap of particles inside the PUC. Hence, a constraint is added to ensure that there is no overlap inside the optimized PUC. Thus, the solution vector $\boldsymbol{x}_N$ should comply with,

$$\begin{aligned} &\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geqslant r_i + r_j, \quad \forall i = 1, \ldots, N, \\ &j = i + 1, \ldots, N, \end{aligned} \tag{13}$$

where N denotes the total number of particles, and r_i and r_j are the radii of ith and jth particles, respectively.

Based on the distance between the centers of a pair of particles, we classify the overlap into two types, namely: partial overlap, if the distance between the centers is less than the sum of the radii, and complete overlap, if the distance between the centers is less than either of the radii of the particles (as shown schematically in Fig. 6a). Let the number of occurrences of partial overlap be denoted as N_p and the number of occurrences of complete overlap

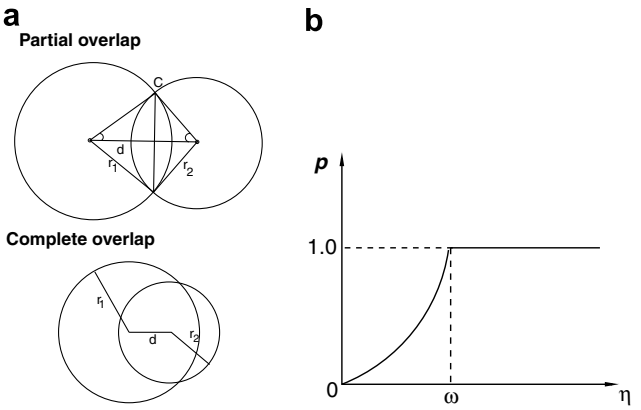

Fig. 6. (a) Two types of overlap encountered. (b) Penalty function used.

be denoted as N_c. The amount of overlap in a potential solution is described by the parameter η as

$$\eta = \begin{cases} \frac{2}{N_p}\sum_{i=1}^{N}\sum_{j=i}^{N}\frac{A_{overlap}}{(A_i+A_j)} & N_c = 0, \\ \frac{2}{N_p}\sum_{i=1}^{N}\sum_{j=i}^{N}\frac{A_{overlap}}{(A_i+A_j)} + 0.5 & N_c > 0. \end{cases} \tag{14}$$

If no instance of complete overlap occurs (i.e., $N_c = 0$), the penalty parameter is a global mean of the ratio of the area of overlap to the mean area of a pair of particles. The limiting case of this expression is encountered when a particle center is on the circumference of another particle. For this limiting case, the value of η is evaluated to be 0.45. Since a complete overlap is a highly undesirable situation, it is to be penalized heavily. Hence, on encountering a single occurrence of a complete overlap, an amount of 0.5 is automatically added to the overlap parameter η, so that the value of η lies within 0 and 1.

Based on the amount of overlap given by the parameter η, a constraint function (or a penalty function) is established as shown in Fig. 6b. The formulation of the constraint function is based on similar one used in [17] and follows a simple power law,

$$p = \begin{cases} 0 & \eta = 0, \\ \left(\frac{\eta}{\omega}\right)^{\beta} & 0 < \eta < \omega, \\ 1.0 & \eta \geqslant \omega, \end{cases} \tag{15}$$

where ω is the upper cutoff of overlap above which $p = 1$ and β is the exponent of the penalty function in the range $0 \leqslant \eta \leqslant \omega$.

From the objective function described by (12) and the constraint function (15), the overall objective function (or the fitness function) is defined as

$$f = \frac{\|S_{rs}^{orig} - S_{rs}\|_{L_2}}{\max_{P_0}\|S_{rs}^{orig} - S_{rs}\|_{L_2}} + p, \tag{16}$$

where $\max_{P_0}\|S_{rs}^{orig} - S_{rs}\|_{L_2}$ represents the value of the objective function of the worst individual of the starting population. The value of f always lies between 0 and 2.

It has been shown for a similar system (two-phase fibrous composites) that the fitness function (such as given by Eq. (16)) is multimodal and contains a number of local minima. Hence, an optimization using conventional gradient-based methods is not appropriate for this case. Therefore, stochastic optimization methods based on the principle of evolution such as genetic algorithms (GA) and simulated annealing (SA) have been used efficiently to solve such complex optimization problems [19,33]. In the present work, the *Augmented Simulated Annealing* (AUSA) technique, that effectively exploits the essentials of GA in combination with SA is employed to minimize the overall objective function. This algorithm is identical to that used by Matouš and Dvorak [17]. Please note that the emphasis of this paper is not on the development of a new generic algorithm, rather on the effective application of an established stochastic optimization procedure. Nevertheless, for the sake of completeness, we briefly introduce some of the important concepts of the GA and list the basic steps in Algorithm 1.

Algorithm 1

Principle of genetic algorithm

```
1  t = 0
2  generate and evaluate P_0
3  while (not termination-condition) {
4     select M_t from P_t (apply sampling mechanism)
5     alter M_t (apply genetic operators)
6     create P_{t+1} from M_t (insert new individuals into P_{t+1})
7  t = t + 1
8  }
```

For the present unit cell reconstruction, each individual in the population P_t, which is a set of potential solutions to the optimization problem, is a possible configuration of particles inside a unit cell, represented by a real-valued chromosome $\boldsymbol{Z} = \{z_1, \ldots, z_{2N}\}$, such that

$$z_{2i-1} = x_i \quad \text{and } z_{2i} = y_i \quad \text{for } i = 1, \ldots, N, \tag{17}$$

where x_i and y_i are the x- and y-coordinates of the ith particle center. As described in Algorithm 1, the three basic steps of the GA are as follows: First, an individual is selected for reproduction (line 4) based on the remainder stochastic sampling without replacement (RSSwoR) method, commonly known as the Roulette Wheel Sampling scheme [8]. The reproduction (or alteration) step (line 5) makes use of different types of crossover and mutation operators applicable to real-valued chromosomes. More details on these operators can be found in [20]. The replacement step (line 6) is controlled in the AUSA algorithm using the Metropolis criterion, which allows a worse child to replace its better parent with a certain probability. This probability measure is reduced as the number of iterations increase, thereby driving the solution to converge to a global minimum. Readers are referred to [8,15,17,19] for detailed explanation on each of the different steps of a GA cycle.

3.2. Problem-specific mutation operator

Based on different numerical experiments presented in Section 4, we observed that with increasing volume fractions the overlap in the optimized PUC is not eliminated completely and the rate of convergence is slow, especially for packs of packing fraction 60% or more. Restart of the AUSA algorithm and reoptimization, as suggested in [8], can be attempted, but do not yield satisfactory results. Other possibility is construction of genetic operators that meet *impenetrability constraint* directly as done in [34]. Such an approach might, however, limit the random information in the population and slow down the convergence of GA. It might be also difficult to implement for densely packed multimodal systems. Thus, in order to eliminate the overlap in the PUC and to improve the convergence rate, we propose a problem-specific mutation operator, which is based on a mass–spring system. For a pair of overlapping particles, we introduce a stiff repulsive spring to push the particles away from each other and for a pair of non-overlapping particles, we introduce a soft attractive spring to keep particles close together, such that the statistics are not altered substantially by the new operator. We further assume that the mass of the particle is directly proportional to its area, and that the force–displacement relation for the springs is linear, with spring stiffnesses of κ or $\bar{\kappa}$ for repulsive or attractive springs, respectively, as shown in Fig. 7a. The separation, d_{ij}, between the particles i and j is computed simply as

$$d_{ij} = \|\boldsymbol{y}_i - \boldsymbol{y}_j\| - (r_i + r_j), \tag{18}$$

where $\boldsymbol{y}_i$ and r_i are the position vector of the center and radius of particle i, respectively.

The placement of springs in a potential PUC is shown schematically in Fig. 7b. First, we create the Delaunay triangulation of the set of particle centers. Then, particles sharing a triangle are identified as neighbors and only neighbors interact through springs. For example in Fig. 7b, particles 1 and 2 are neighbors, while particles 1 and 4 are not. Overlapping neighbors interact through a stiff repulsive spring, as shown by solid lines in Fig. 7b, while the soft attractive springs between non-overlapping neighbors are shown in broken lines. Based on standard spring dynamics, we construct the equation of motion as follows:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = 0, \tag{19}$$

where $\boldsymbol{u}$ is the displacement vector, $\boldsymbol{M}$ is the diagonal mass matrix and $\boldsymbol{K}$ represents the global stiffness matrix. The initial displacement conditions are given by the initial spring excitations due to the overlap. The vector of initial displacements reads

$$\boldsymbol{u}_0 = \frac{1}{2}\left\{ \sum_{i=2}^{\overline{N}_1} \boldsymbol{d}_{1i}, \sum_{\substack{i=1 \\ i\neq 2}}^{\overline{N}_2} \boldsymbol{d}_{2i}, \ldots, \sum_{\substack{i=1 \\ i\neq N}}^{\overline{N}_N} \boldsymbol{d}_{Ni} \right\}^{\mathrm{T}} \tag{20}$$

where N denotes the number of particles and $\overline{N}_i$ represents the number of neighbors for ith particle. The vector $\boldsymbol{d}_{ij}$ is computed as

$$\boldsymbol{d}_{ij} = d_{ij}\{\cos(\theta_{ij}), \sin(\theta_{ij})\}^{\mathrm{T}} \tag{21}$$

where θ_{ij} is the angle between the spring ij and the y_1-axis (Fig. 7b). The global stiffness and mass matrices $\boldsymbol{K}$ and $\boldsymbol{M}$ are constructed from their standard element counterparts by usual assembly procedure.

The standard implicit Newmark predictor–corrector scheme is used for the time integration of (19) because of its stability and simple implementation [11]. The SuperLU library is utilized for the LU decomposition [5]. The numerical integration is carried out until the overlap inside a potential PUC goes to zero or a prescribed number of time-steps is reached. Please note that in the applications presented in Section 4 the stiffness of the soft attractive springs between non-overlapping particles is set to zero.

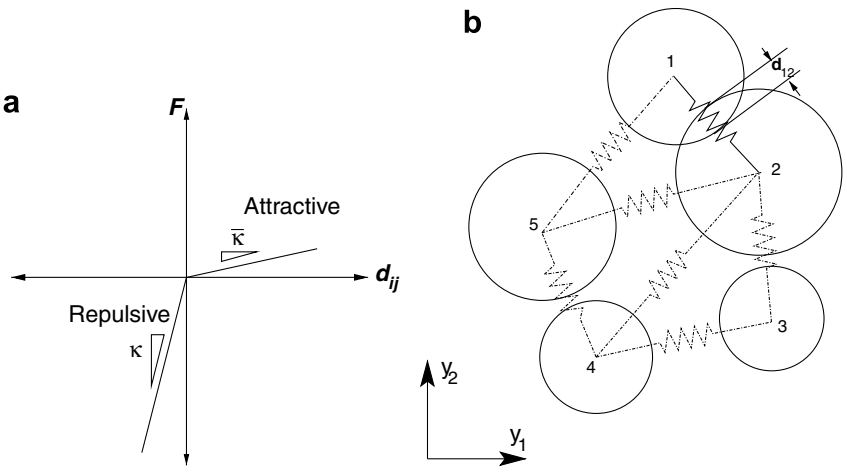

Fig. 7. (a) Force–displacement (spring) relation. (b) Delaunay triangulation and position of springs in a potential cell. Symbol d_{ij} denotes the separation between particles i and j, and symbols κ and $\bar{\kappa}$ represent repulsive and attractive spring constants, respectively.

The aforementioned process of changing the positions of particles inside a potential PUC based on the equation of motion is implemented as a mutation operator. This mutation operator is applied to those population members having only partial overlap, and is activated with a small probability of 0.05.

3.3. Parallel implementation

The most costly part of the AUSA algorithm is the evaluation of the fitness of each individual (either in the population at the start of GA or in the mating pool). All the other steps (sampling, genetic operations, simulated annealing and evaluation of constraints) do not involve intense computations. Hence, to accelerate the algorithm, parallelism is introduced in the evaluation of the fitness function.

Initially, N_{proc} processors are allocated, where N_{proc} is a multiple of number of individuals in the mating pool M_t ($N_{\text{proc}} = a \times M_t$, where a is an integer greater than or equal to 1). As seen from Fig. 8, the input parameters for the PUC and the AUSA algorithm are first read on the master processor (processor 0). Using the input information, a population of individual unit cells is generated (Step 2 of Algorithm 1). A set of a processors constitute a sub-communication world, as indicated by the processors enclosed inside the oval (Fig. 8). Every P/M_t individuals of the initial population are sent to a processors (sub-worlds) for evaluation of two-point probability functions and each processor carries out N_{hits}/a number of throws of the sampling template on each individual. This information of two-point probability functions, evaluated for N_{hits}/a throws, is then sent to the sub-master processor (master for sub-communication world). At the sub-master processor, all the two-point probability functions are added and averaged out (explicit assumption of isotropy), and sent subsequently to the global master (processor 0). Thus, the fitness for all the members in population P_0 is calculated.

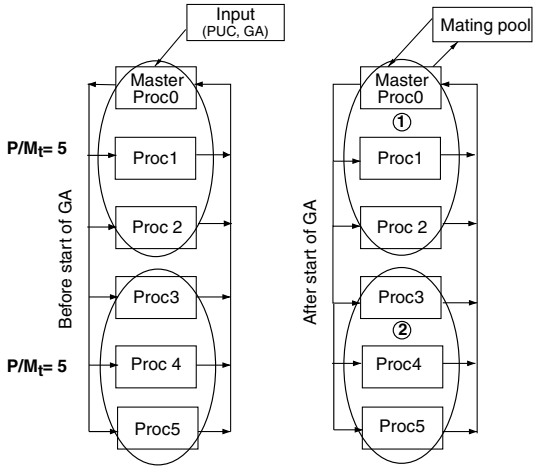

Fig. 8. Parallel implementation of AUSA algorithm.

Once the genetic algorithm has started, the master processor executes the sampling mechanism and genetic operations inside the mating pool. Once new offsprings are created in the mating pool each offspring is sent to a processors (corresponding sub-communication world) and two-point probability functions are evaluated in the same manner as for the individuals in the initial population, as described in the previous paragraph. This procedure is continued throughout the execution of the genetic algorithm.

4. Examples

4.1. Packs with 40% particle volume fractions

We now turn our attention to a set of unit cell reconstruction problems illustrating the capabilities of the reconstruction procedure. First, we start with packs containing 40% packing fraction and three different radii (10, 20 and 30 μm), as shown in Fig. 2. Only five of the six micrographs are selected as the initial packs. All five starting packs have particle packing fractions around 0.37–0.39 and the ensemble average (mean) of the two-point probability functions for all five packs, with error bars indicating the spread in the probability functions, is plotted in Fig. 9. There are two sources of error in this plot: (a) the different probability distributions for each member of the ensemble (arising from error in the ergodic assumption) and (b) the numerical error associated with the use of optimal sampling parameters. The sampling template parameters are calibrated to be 1.5 μm radial spacing (300 radial points) and 2500 throws of the sampling template. It should also be noted that, for the present 40% packs, statistical homogeneity and isotropy are assumed explicitly based on the verification study in Section 2.1, and the ergodicity assumption will be re-confirmed below. The initial size H_{init} of the PUC for this problem is chosen as 300 μm, from the results in Fig. 9, and the optimal size is determined to be 296 μm from Eq. (11).

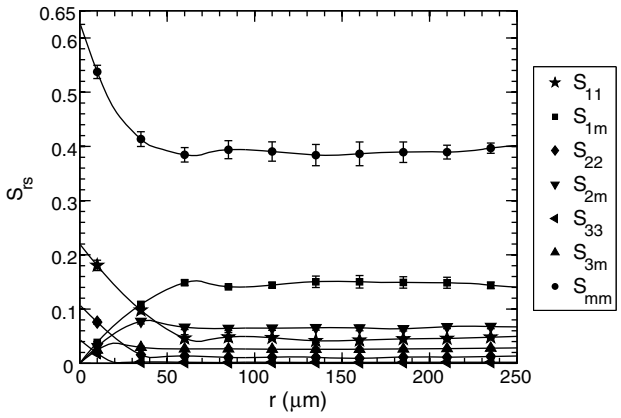

Fig. 9. Two-point probability function distributions of the initial 40% pack with error bars representing both numerical and statistical errors. The radial independence of all S_{rs} for $r \gtrsim 150$ μm suggests the use of a 300 μm initial size for the reconstructed PUC.

The AUSA algorithm described in Section 3 is utilized to reconstruct the PUC. All the AUSA runs presented hereafter consist of a population of 600 individuals and 20 individuals in the mating pool. Table 3 lists the values used for the GA and SA parameters in the present study and shows that the probability of mutation operators is greater than crossover operators, as was the case in [17,33]. We use two different crossover operators: simple and arithmetic crossover, and four different mutation operators: uniform, boundary, non-uniform and problem-specific mutation (introduced in Section 3.2). The probability of mutation is uniformly distributed between the mutation operators, except for the problem-specific mutation, which is invoked with a probability of 0.05. The simulation is stopped when the maximum number of iterations reaches 10,000 or the tolerance (defined by the difference of the fitness of the worst individual and the best individual) reaches 0.015 and stays there for at least 1000 iterations. The upper cutoff of overlap ω for the evaluation of constraint (see Eq. (15)) is taken to be 0.05, while the exponent β is set to 5.0.

The fitness evolution of one of the AUSA runs is given in Fig. 10. As seen there, the worst, mean and the best fitness values have converged for around 8000 iterations. The final fitness value attained is argued to be the global minimum, since no new offsprings are accepted into the population after 7000 iterations. Similarly, in all the other 40% pack reconstruction runs the minimum is reached within 7000–9000 iterations.

One of the five reconstructed unit cells is plotted in Fig. 11, showing no overlap and substantial reduction in the size of the PUC. Fig. 12 depicts the comparison of the mean of the two-point probability functions of the five original packs and the corresponding five reconstructed PUCs. Only selected probability functions are plotted for clarity there, but similar trend would be observed for the remaining ones. The thin error bars indicate the spread in the evaluation of the two-point probability functions in the five original packs and the thick error bars indicate the spread in the evaluation of the two-point probability functions in the corresponding reconstructed PUCs. In both cases, these error bars arise from ergodic assumption and from the numerical evaluation of the two-point probability functions.

The final values of the fitness function given by Eq. (12) of the optimized PUC for each of the five independent runs and the number of iterations needed to reach the tolerance limit of 0.015 are tabulated in Table 4. It can be seen that the minima reached for all five runs are consistent. Table 4 also shows that the initial best population, generated randomly, is optimized to about 75% by the end of the AUSA run, which is a substantial amount of improvement on the initial guess. Also, the final PUC is reduced to about 25%

Table 3
Parameters used in the implementation of the AUSA algorithm

Description	Parameter	Value
Population size	P_t	600
Mating pool size	M_t	20
Probability of mutation	p_{mut}	0.7
Probability of crossover	p_{cross}	0.3
Maximum annealing temperature	T_{max}	0.05
Minimum annealing temperature	T_{min}	1E−6
Ratio of cooling	T_{mult}	0.999

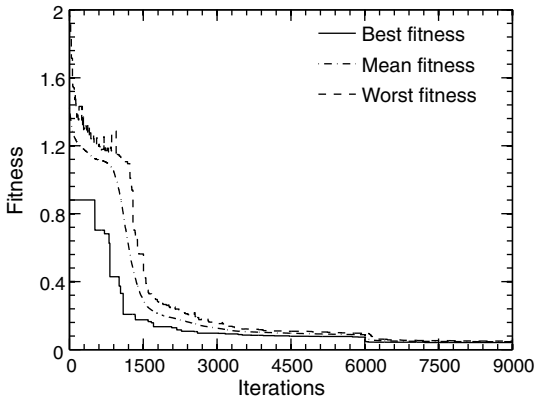

Fig. 10. Fitness evolution during a AUSA run for the 40% pack.

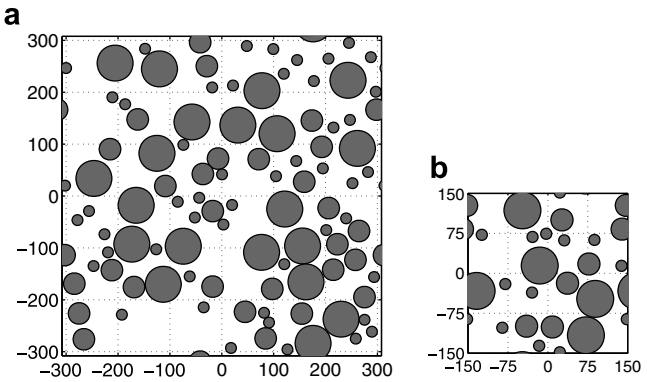

Fig. 11. (a) A member of the ensemble for 40% packs and (b) the reconstructed periodic unit cell plotted on the same scale (all dimensions in μm).

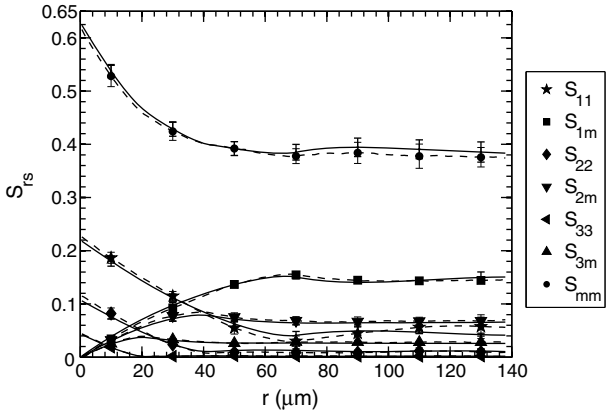

Fig. 12. Comparison of the two-point probability functions of the original pack (solid line) and the reconstructed PUC (dashed line) for the 40% packs shown in Fig. 11.

Table 4
Final values of the objective function F (Eq. (12)) and number of iterations needed to attain the required tolerance for the 40% pack

Pack #	Fitness F						# of iteration
	Before GA			After GA			
	Min	Mean	Max	Min	Mean	Max	
Pack 1	2.102	3.708	8.451	0.616	0.675	0.730	8430
Pack 2	1.081	3.474	8.583	0.571	0.634	0.684	8512
Pack 3	1.938	4.106	9.569	0.545	0.605	0.656	7542
Pack 4	4.447	5.868	13.870	0.574	0.648	0.699	7970
Pack 5	1.746	3.215	8.462	0.608	0.653	0.712	7850

of the size of the original pack. Even though five different packs are used as initial packs, the consistency of the final fitness values and the good agreement of the two-point probability functions, between the original packs and the reconstructed unit cells, reinforce the validity of ergodicity of the random particulate composite under study. Hence, for further reconstruction simulations (50%, 60% and 70% packing simulations), only one solid propellant pack is used as the initial pack.

4.2. Packs with 50% particle volume fractions

Now we concentrate on reconstructing a PUC of 50% packing fraction. The initial pack and the two-point probability function distributions are shown in Fig. 13a and b, respectively. The error bars now represent only the numerical error introduced by the sampling procedure. The parameters for the AUSA algorithm are the same as for the 40% packing simulations. However, the maximum number of iterations is set to 20,000 and the tolerance is increased to 0.02 to accommodate for the increase in the packing fraction. Since the maximum number of iterations is increased, the value of $T_{\min}$ is reduced to 1E−8. The sampling template parameters used for the current simulations are 2.0 μm radial spacing (225 radial points) and 3000 throws of the sampling template. Since the particle packing fraction is higher, we relax the overlap constraint by increasing the upper cutoff of overlap ω to 0.3 and selecting the exponent β equal to 1.0, leading to a linear penalty function. These particular values are selected after experimenting with

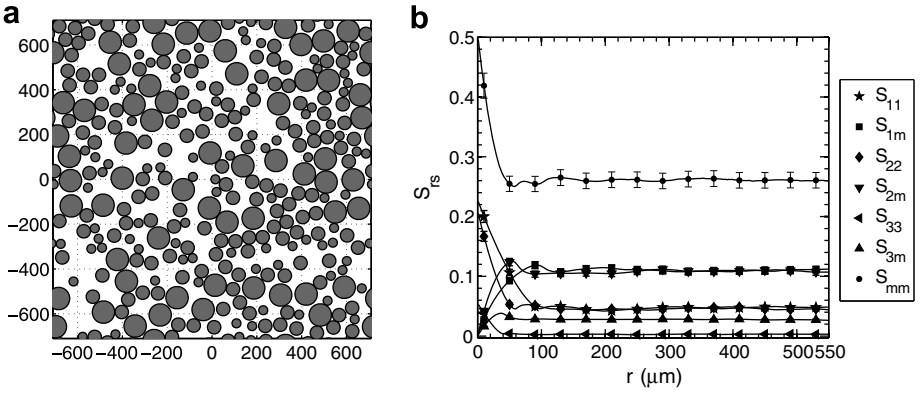

Fig. 13. Initial 50% pack considered for reconstruction (left) and associated two-point probability function distributions (right).

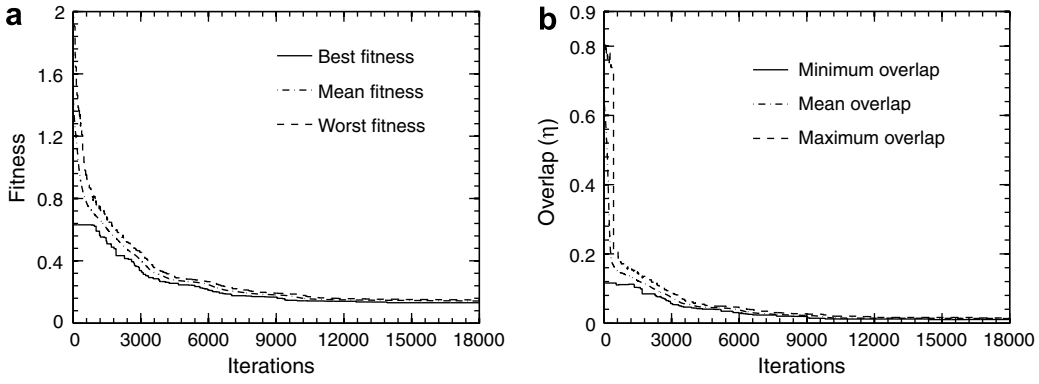

Fig. 14. (a) Evolution of fitness values and (b) evolution of the overlap parameter η during GA run for the 50% pack. Although significantly reduced, the overlap is not completely eliminated and converges to about 0.01.

several combinations of ω and β. Ergodicity, statistical homogeneity and isotropy are assumed explicitly for the 50% pack simulations. From the two-point probability function distributions (Fig. 13b), the initial size of the PUC is chosen to be 450 μm and the optimized size H given by (11) yields 448 μm.

The fitness evolution during one of the AUSA runs is shown in Fig. 14a. As expected, because of the higher packing fraction, this run involves a larger number of iterations than the 40% pack runs. From Fig. 14b, it can be seen that the value of overlap (η) converges to about 0.01 towards the end of the simulation. Fig. 15 shows the initial 50% pack and the reconstructed PUC from one of the AUSA runs, emphasizing the size reduction. Fig. 16 shows the comparison of the two-point probability function distributions of the original pack and the reconstructed PUC. Good agreement can be again observed. Here, the thin error bars correspond to the scatter in the two-point probability function distribution of the original pack due to the numerical errors, while the thick error bars correspond to the scatter in the two-point probability function distributions of the optimized PUC due to both numerical and statistical errors (average of five runs). The final value of the objective function of the optimized PUC and the number of iterations needed to reach the tolerance limit of 0.02 are tabulated in Table 5.

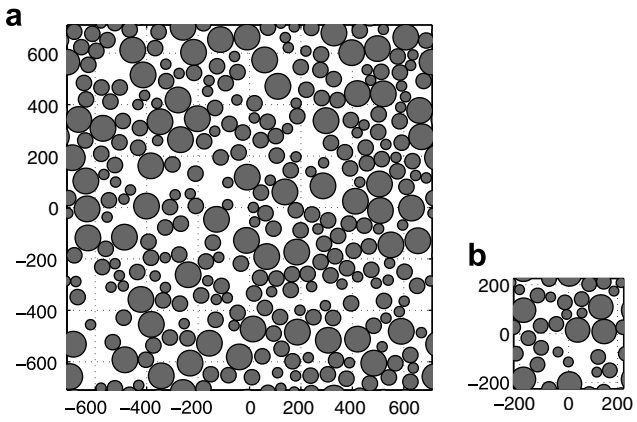

Fig. 15. (a) Initial 50% pack considered for reconstruction and (b) the reconstructed periodic unit cell plotted on the same scale (all dimensions in μm).

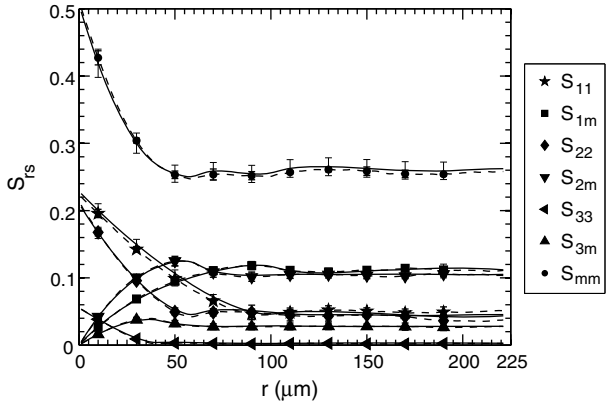

Fig. 16. Comparison of the two-point probability functions of the original pack (solid line) and the reconstructed PUC (dashed line) for the 50% pack.

Table 5 shows that the initial best population, generated randomly, is optimized to about 70% by the end of the AUSA run. However, it is also observed that out of the five runs, four of them resulted in unit cells with a small overlap ($\tilde{\eta}$). Here we use the physical overall overlap $\tilde{\eta}$, rather than its computational counterpart η given by (14), defined as

$$\tilde{\eta} = \frac{A_o}{A_t}, \tag{22}$$

where A_o is the area of overlap and A_t is the total particle area. Although the physical value of overlap $\tilde{\eta}$ is relatively small for all five runs, it becomes more and more pronounced as the packing fraction increases.

Similar reconstruction is attempted for a pack of 60% packing fraction. Table 6 shows the number of iterations needed for convergence for each of the 60% PUC reconstruction runs and the final overlap. Reoptimization or a restart of the GA run can be carried out to further improve the solution, but when attempted, it did not show any

Table 6
Number of iterations needed for convergence and final value of the overall overlap parameter for 60% pack reconstruction runs

Run#	# of iteration	Overall overlap ($\tilde{\eta}$)
Run 1	26,700	5.140E−2
Run 2	23,300	5.451E−2
Run 3	25,224	3.102E−1
Run 4	23,715	1.302E−1
Run 5	25,366	8.524E−2

Maximum number of iterations was increased for this study to 30,000 due to the convergence problems.

Table 5
Final values of the objective function F, number of iterations needed to attain the required tolerance and overall overlap for 50% pack

Run #	Fitness F						# of iteration	Overall overlap ($\tilde{\eta}$)
	Before GA			After GA				
	Min	Mean	Max	Min	Mean	Max		
Run 1	1.700	3.240	7.579	0.873	0.901	0.942	15,678	2.2868E−4
Run 2	2.042	3.472	8.170	0.824	0.916	0.973	15,532	1.141E−4
Run 3	2.295	4.147	8.690	0.877	0.973	1.034	16,243	3.323E−4
Run 4	2.325	3.698	8.130	0.901	0.959	1.047	16,216	0.000
Run 5	1.886	3.834	7.611	0.861	0.888	0.948	14,561	1.235E−5

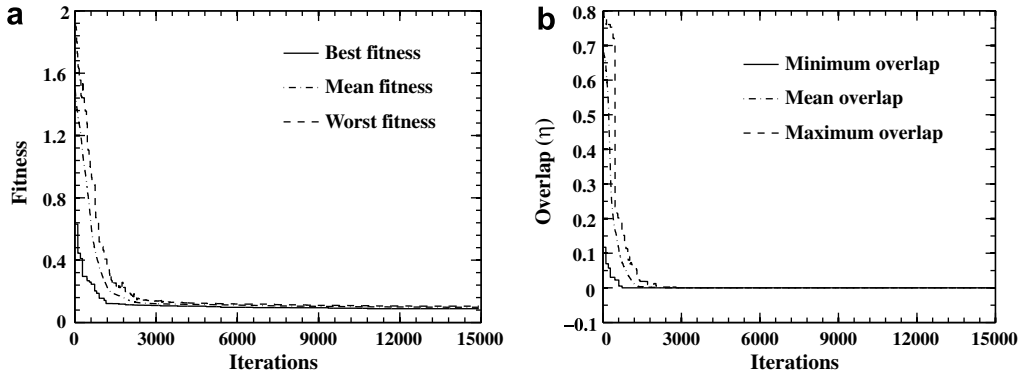

Fig. 17. (a) Evolution of fitness value and (b) evolution of the overlap parameter η during GA run for 60% pack with new problem-specific mutation operator.

marked improvements. When reconstruction is attempted on a 70% pack, GA run did not converge at all, as the overlap plays a dominant role in the optimization procedure. Hence, we employ the new problem-specific mutation operator described in Section 3.2 for the 60% and 70% packing fractions. This is described next.

4.3. Packs with 60% and 70% particle volume fractions

For the reconstruction of PUC of 60% and 70% packs, all the parameters of the AUSA algorithm and the penalty function are the same as in the 50% pack reconstruction simulations. As mentioned in Section 3, the new mutation operator is activated with a small probability of 0.05. The convergence results for one of the 60% pack AUSA runs, with the new mutation operator, are shown in Fig. 17. One can readily see, from Fig. 17b, that the overlap in the PUC solution is completely eliminated and from Fig. 17a that the optimization procedure has converged in around 7500 iterations. Fig. 18a shows the original 60% pack considered for reconstruction and Fig. 18b shows the reconstructed PUC. The two-point probability functions of the original pack and the reconstructed unit cell are compared and plotted in Fig. 19. One can observe

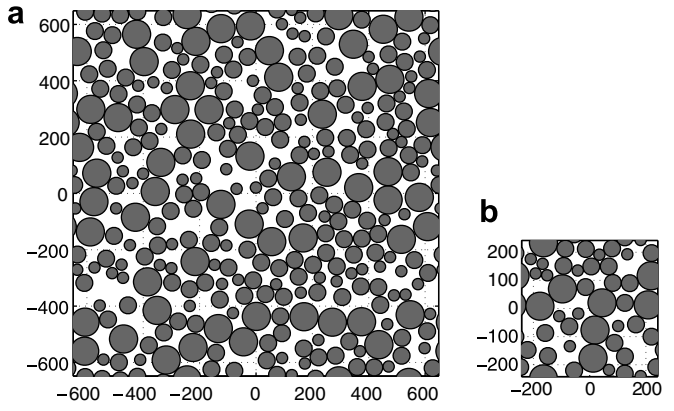

Fig. 18. (a) Initial 60% pack considered for reconstruction and (b) reconstructed periodic unit cell plotted on the same scale (all dimensions in μm).

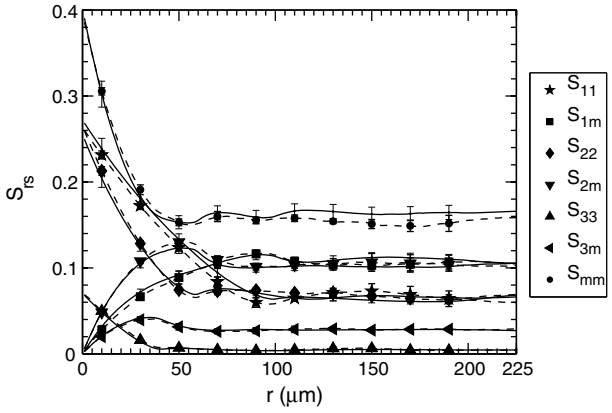

Fig. 19. Comparison of the two-point probability functions of the original pack (solid line) and the reconstructed PUC (dashed line) for 60% pack with the new mutation operator.

that the two-point probability functions of the PUC compare well with those of the original pack, even with the new mutation operator. This fact is reiterated in Table 7, which shows the initial and final values of the objective function (Eq. (12)), the number of iterations needed for convergence and the value of overall overlap $\tilde{\eta}$ for the 60% reconstruction. From the final values of the fitness function, we can see that the best population member is optimized to about 70% by the end of the AUSA run. It should also be noted that the overlap is completely eliminated in all five reconstruction attempts.

Since the effectiveness of the new mutation operator is established for the reconstruction of the 60% PUC, we now apply it to reconstruct the PUC of 70% packing fraction. The new mutation operator is shown to be effective for the 70% reconstruction simulations also, as observed from Table 8. We can see that the initial randomly generated best member is optimized to about 70% by the end of the AUSA run, similar to the previous reconstruction trials. The overlap is eliminated completely in four of the five AUSA runs and the fifth one contains a small overlap only. Fig. 20a and b show the initial 70% pack and the reconstructed PUC of one of the five AUSA runs, while

Table 7
Final objective function values, number of iterations needed for convergence and overall overlap for 60% pack

Run #	Fitness F						# of iteration	Overall overlap ($\tilde{\eta}$)
	Before GA			After GA				
	Min	Mean	Max	Min	Mean	Max		
Run 1	3.477	5.651	9.827	0.985	1.084	1.157	8885	0.000
Run 2	3.053	4.885	9.230	0.817	0.901	0.970	10,870	0.000
Run 3	2.468	4.837	9.312	1.113	1.208	1.258	9565	0.000
Run 4	2.860	4.852	9.591	0.957	1.035	1.088	9825	0.000
Run 5	2.540	4.901	9.490	1.025	1.115	1.178	10,920	0.000

Table 8
Final objective function values, number of iterations needed for convergence and overall overlap for the 70% pack

Pack #	Fitness F						# of iteration	Overall overlap ($\tilde{\eta}$)
	Before GA			After GA				
	Min	Mean	Max	Min	Mean	Max		
Run 1	3.450	6.967	12.219	1.168	1.320	1.450	19,672	0.0000
Run 2	4.258	7.122	11.246	1.375	1.496	1.565	23,211	0.0000
Run 3	4.211	7.070	11.950	1.157	1.351	1.467	20,080	0.0000
Run 4	3.260	7.053	11.210	0.946	1.030	1.061	23,065	0.0000
Run 5	3.783	7.064	11.412	1.296	1.351	1.452	24,090	1.0E−4

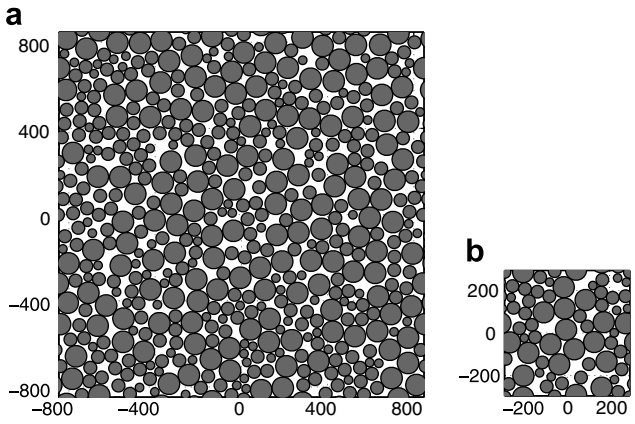

Fig. 20. (a) Initial 70% pack considered for reconstruction and (b) reconstructed periodic unit cell plotted on the same scale (all dimensions in μm).

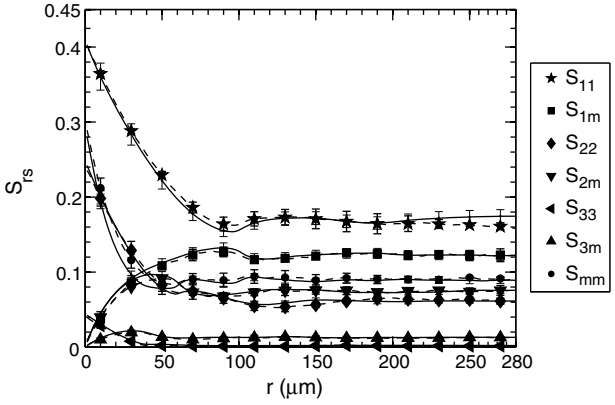

Fig. 21. Comparison of the two-point probability functions of the original pack (solid line) and the reconstructed PUC (dashed line) for the 70% pack with the new mutation operator.

Fig. 21 presents a comparison of the two-point probability functions. Very good agreement between the statistics of the original pack and the PUC is observed again.

4.4. Performance and scalability

In this section, we discuss the performance of the optimization scheme with respect to the packing fraction and scalability of the parallel algorithm. As apparent in Fig. 22a, there is a strong dependence of the number of iterations required for convergence on the packing fraction (solid line in Fig. 22a). Since a substantial part of the computational effort is spent on reducing the particle overlap, the adoption of the new mutation operator for the higher volume fraction cases (60% and 70%) significantly reduces the number of iterations needed to achieve convergence of the GA optimization, as apparent from the dashed line in Fig. 22a.

Finally, the speedup characteristics of the parallel implementation (described in Section 3.3) of the AUSA algorithm are investigated by measuring the CPU time needed for 1000 iterations on 40, 80, 120 and 160 processors for the 50% PUC reconstruction. The time required for completion of 1000 iterations on each set of processors is plotted in Fig. 22b (line with circles). The speedup factor is calculated from the reference time taken on 40 processors and is also plotted in Fig. 22b (line with squares). We can see that the parallel code exhibits almost linear scalability with increasing number of processors. Because of the almost ideally parallel nature of the problem, this

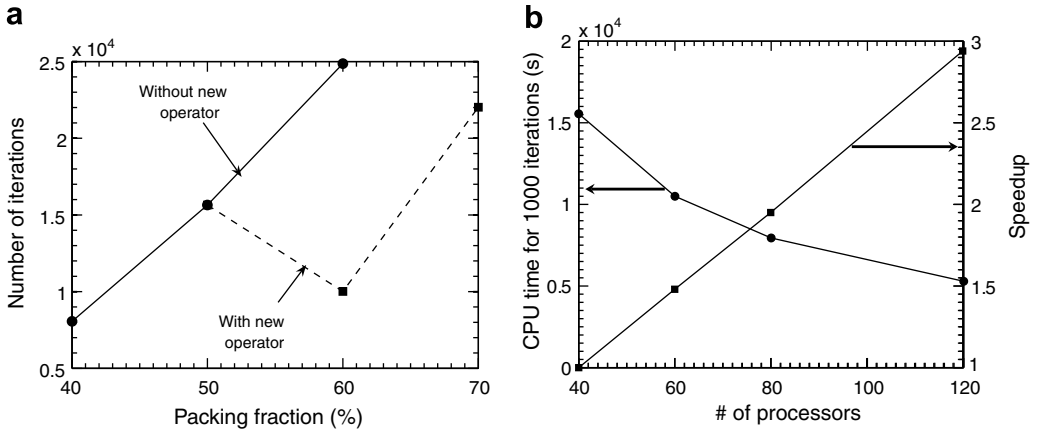

Fig. 22. (a) Relation between the volume fraction of the pack being reconstructed and the number of iterations needed to achieve convergence (solid line: simulations without the new mutation operator; dashed line: simulations with new mutation operator). (b) Speedup of parallel AUSA algorithm for 50% packing fraction packs.

scalability is expected for reconstruction of PUCs of different packing fractions as well.

5. Conclusions

An effective method has been presented to characterize and reconstruct the complex microstructure of a random highly packed, multimodal, particulate composite by a simplified periodic unit cell that is statistically (geometrically) similar to the original microstructure. It is important to note that the reconstructed periodic unit cells are only representative from a geometrical statistics point of view and that the representativity of the PUC must also account for the physical processes of interest (Swaminathan and Ghosh [23]). However, the construction of a geometrically equivalent periodic unit cell is an important first step in describing behavior of complex particulate materials, such as solid propellants. In this work, the micrographs have been computationally generated using a packing software called *Rocpack*, which has been tested and compared to available experimental data. For the present study, one- and two-point probability functions have been identified as the suitable statistical descriptors and the assumptions of ergodicity, homogeneity and statistical isotropy have been numerically assessed.

A stochastic optimization method called Augmented Simulated Annealing has been used to optimize the positions of particles inside the periodic cell, such that probability functions are similar to those from the original pack. The optimization scheme has been implemented in parallel allowing for the study of large data sets with large particle size variations and high packing content. A new mutation operator, based on a mass–spring system, has been developed to eliminate the particle overlap and to speed up the computations.

Reconstruction of periodic unit cells has been performed on four-phase random particulate composite packs of 40–70% packing fractions. The reconstruction results show good convergence and repeatability of the genetic algorithm and the statistics of the reconstructed cells compare well with those of the original packs.

Acknowledgements

The authors would like to gratefully acknowledge the support from ATK/Thiokol (ATK-21316), with J. Thompson and Dr. I.L. Davis serving as program monitors, and from the Center for Simulation of Advanced Rockets (CSAR) under contract number B523819 by the U.S. Department of Energy as a part of its Advanced Simulation and Computing (ASC) program. We would also like to thank Dr. T.L. Jackson and his team for providing *Rocpack* and for helpful discussions.

References

[1] M.P. Anderson, D.J. Srolovitz, G.S. Crest, P.S. Sahni, Acta Metallurgica 32 (5) (1984) 783–791.
[2] M.J. Beran, Statistical Continuum Theories, Interscience Publishers, 1968.
[3] J.G. Berryman, Journal of Applied Physics 57 (7) (1984) 2374–2384.
[4] P.B. Corson, Journal of Applied Physics 45 (7) (1974) 3159–3164.
[5] J.W. Demmel, S.C. Eisenstat, J.R. Gilbert, X.S. Li, J.W.H. Liu, SIAM Journal of Matrix Analysis and Applications 20 (3) (1999) 720–755.
[6] J. Fish, Q. Yu, International Journal for Numerical Methods in Engineering 52 (1–2) (2001) 161–192.
[7] J. Gajdošík, J. Zeman, M. Šejnoha, Probabilistic Engineering Mechanics 21 (2006) 317–329.
[8] D. Goldberg, Genetic Algorithms in Search Optimization and Machine Learning, Addison-Wesley, 1989.
[9] Z. Hashin, S. Shtrikman, Journal of the Mechanics and Physics of Solids 10 (1962) 335–342.
[10] R. Hill, Journal of the Mechanics and Physics of Solids 11 (1963) 357–372.
[11] T.J.R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Dover, 2000.
[12] G.M. Knott, T.L. Jackson, J. Buckmaster, AIAA Journal 39 (4) (2000) 678–686.
[13] S.V. Kochevets, J. Buckmaster, T.L. Jackson, A. Hegab, Journal of Propulsion and Power 17 (4) (2001) 883–891.

[14] H. Kumar, C.L. Briant, W.A. Curtin, Mechanics of Materials 38 (2006) 818–832.
[15] S. Mahfoud, D.E. Goldberg, Parallel Problem from Nature (1992) 301–310.
[16] K. Matouš, International Journal of Solids and Structures 40 (2003) 1489–1503.
[17] K. Matouš, G.J. Dvorak, IEEE Transactions on Magnetics 39 (3) (2003) 1827–1835.
[18] K. Matouš, H.M. Inglis, X. Gu, D. Rypl, T.L. Jackson, P.H. Geubelle, Composites Science and Technology 67 (2007) 1694–1708.
[19] K. Matouš, M. Lepš, J. Zeman, M. Šejnoha, Computer Methods in Applied Mechanics and Engineering 190 (2000) 1629–1650.
[20] Z. Michalewicz, T.D. Logan, S. Swaminathan, Evolutionary operators for continuous convex parameter spaces, in: Proceedings of the Third Annual Conference on Evolutionary Programming, 1994, pp. 84–97.
[21] G.L. Povirk, Acta Metallurgica et Materialia 43 (1995) 3199–3206.
[22] J.A. Quintanilla, W. Max Jones, Physical Review W 75 (4) (2007) 046709-1–046709-9.
[23] S. Swaminathan, S. Ghosh, Journal of Composite Materials 40 (7) (2006) 605–621.
[24] V. Sundararaghavan, N. Zabaras, Computational Materials Science 32 (2004) 223–239.
[25] P. Smith, S. Torquato, Journal of Computational Physics 76 (1988) 176–191.
[26] D.R.S. Talbot, J.R. Willis, IMA Journal of Applied Mathematics 35 (1985) 39–54.
[27] J.L. Teply, G.J. Dvorak, Journal of Mechanics and Physics of Solids 36 (1988) 29–58.
[28] S. Torquato, Random Heterogeneous Media, first ed., Springer Publishers, New York, 2002.
[29] S. Torquato, G. Stell, Journal of Chemical Physics 77 (1982) 2071–2077.
[30] M.D. Webb, I. Lee Davis, Powder Technology 167 (2006) 10–19.
[31] C.L.Y. Yeong, S. Torquato, Physical Review E 57 (1) (1998) 495–506.
[32] C.L.Y. Yeong, S. Torquato, Physical Review E 58 (1998) 224–233.
[33] J. Zeman, Analysis of mechanical properties of fiber-reinforced composites with random microstructure, M.S. Thesis, Czech Technical University, Prague, 2000.
[34] J. Zeman, M. Šejnoha, Journal of Mechanics and Physics of Solids 49 (2001) 69–90.