Surrogate assisted optimization of particle reinforced metal matrix composites

Lorenzo Gentile
TH Köln
Cologne, Germany
lorenzo.gentile@th-koeln.de

Martin Zaefferer
TH Köln
Cologne, Germany
martin.zaefferer@th-koeln.de

Dario Giugliano
University of Strathclyde
Glasgow, United Kingdom
dario.giugliano@strath.ac.uk

Haofeng Chen
University of Strathclyde
Glasgow, United Kingdom
haofeng.chen@strath.ac.uk

Thomas Bartz-Beielstein
TH Köln
Cologne, Germany
thomas.bartz-beielstein@th-koeln.de

ABSTRACT

Surrogate Model Based Optimization (SMBO) is an established technique for handling computationally expensive optimization problems. One important application is the optimization of Particle Reinforced Metal Matrix Composites (PRMMCs). Multi-phase materials are gaining attention. Their performance is strongly affected by microscale properties. By optimizing the microscale structure, these materials can be tailored to satisfy specific requirements. Current manufacturing techniques have limited control over the distribution of reinforcing particles and are subject to considerable uncertainty. Moreover, the simulation and optimization of PRMMCs requires significant computational effort. We propose an approach that tackles the problem of optimizing the characteristics of PRMMCs subject to uniaxial load, by improving the particles' spatial distribution. The optimization problem is split into a bilevel problem: The upper-level optimization aims to find the particle distribution parameters which maximize the PRMMC limit load. Due to potentially infeasible distributions, the lower-level problem attempts to create a particle placement that reflects the specifications of an upper-level candidate solution.

We employ an SMBO approach that combines Kriging, Sequential Parameter Optimization, and a Genetic Algorithm. Experimental results indicate that our approach can find promising solutions within few evaluations, handles uncertainty, and allows insight into the most important effects on the limit load.

CCS CONCEPTS

• **Mathematics of computing → Continuous optimization**; • **Information systems → Uncertainty**; • **Theory of computation → Gaussian processes**; • **Computing methodologies → Modeling and simulation**;

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

ACM ISBN 978-1-4503-5618-3/18/07…$15.00
https://doi.org/10.1145/3205455.3205574

KEYWORDS

Optimization under uncertainty, Surrogate model based optimization, Parameter optimization, Finite Element Methods, Multilevel optimization

ACM Reference Format:
Lorenzo Gentile, Martin Zaefferer, Dario Giugliano, Haofeng Chen, and Thomas Bartz-Beielstein. 2018. Surrogate assisted optimization of particle reinforced metal matrix composites. In *GECCO '18: Genetic and Evolutionary Computation Conference, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3205455.3205574

1 INTRODUCTION

Metal Matrix Composites (MMCs) are being established for component design in applications where the property profile of conventional materials cannot keep up with performance requirements. A mix of metal and ceramic materials results into the improvement of specific thermo-mechanical properties such as stiffness, yield strength, tensile strength, creep resistance, fatigue strength, wear resistance, and performance at elevated temperatures while maintaining the maximum ductility [14].

To optimize the structure and the characteristics of Particle Reinforced MMCs (PRMMCs), different manufacturing processes have been developed over the last 20 years. In particular, promising forefront manufacturing processes have been proposed and are still under development [15, 17]. These processes aim for a better control over reinforcement distributions. The growing interest as well as increasing requirements lead to a need for a more thorough understanding of the behavior of PRMMCs on the basis of computational simulations. Numerous models have been proposed over the years.

Various studies revealed the accuracy of simulating representative material microstructures. They are able to predict the effective properties of particle reinforced composites subjected to elastic [12, 18, 25] and elasto-plastic deformation [2, 11, 24] with high fidelity. Finite Element Method (FEM) simulations of 3D representative volume elements (RVEs) appears to be widely recognized as state-of-the-art in the field.

The load that leads to the failure of the material, the so-called *limit load*, is a fundamental performance value of mechanical components, including composite materials. It will be referred to as σ_{LL} in the following. A good estimate of the limit load can be obtained through large scale finite elements calculations. This leads

Table 1: Material properties used in the numerical analyses.

Material	E[GPa]	ν	$\sigma_{p,0}$	UTS[MPa]
Al6061-T6	68.9	0.3	276	295.5
SiC	380	0.19	-	3450

to difficulties related to a demand for large run times. In spite of these issues, the recent development of Linear Matching Method (LMM) [4, 6] allows to accurately and efficiently determine the shakedown boundaries of composite materials [5, 9, 10]. The LMM is a direct method for the evaluation of limits associated with an isotropic, elastic-perfectly plastic body subjected to mechanical or thermal cyclic and monotonic loads.

In the last two decades, research has considered the volume fraction, the size, the shape, and the spatial distribution of the reinforcing phase(s) as the main parameters of optimal material design [2, 13, 16, 19, 20, 22, 26, 27]. In light of these considerations, our research concentrates on investigating the effect of the particles' spatial distribution and size on the limit load of PRMMCs. To that end, we employ surrogate assisted optimization techniques which means that, to deal with the large simulation times and considerable uncertainty, we use data driven surrogate models to estimate the performance of candidate solutions. Thus, an effective search for near-optimal solutions becomes possible despite the challenging problem characteristics.

This paper is structured as follows. Section 2 introduces the PRMMCs optimization problem. It explains details about the limit load evaluation method and the optimization structure. Our methodology is described in Sec. 3. It presents further information about the SBMO and the Genetic Algorithm (GA) that were used in the study. The experimental setup and the analysis of the results are presented in Sec. 4. Finally, the paper concludes with a discussion in Sec. 5.

2 PRMMCS OPTIMIZATION

2.1 Problem description

The basis of the considered design problem are idealized three-dimensional RVEs. They are formed by a variable number of elastic reinforcing particles with spherical shape that are inserted into an elasto-plastic matrix. Examples are shown in Fig. 1. The limit load along the x-axis of the RVEs has to be maximized.

To calculate the σ^x_{LL} with Finite Element (FE) methods, geometries are meshed by Abaqus C3D10 tetrahedral quadratic elements. The resulting models are typically constituted of around 2e4 elements. The material properties considered in the numerical analyses are summarized in Table 1. They correspond to Aluminum 6061T6 that fully embeds silicon carbide (SiC) particles [3]. In order to evaluate the σ^x_{LL} properly, an analysis of the material's microscale 3D periodicity has been implemented. Periodic Boundary Conditions (PBC) are assigned to RVE surfaces as reported in [7, 27].

In the experiments, we fix the reinforcement fraction volume to 10%. This indicates that the volume of particles material in relation to the overall volume. Furthermore, particles are assumed to be normally distributed along the three axes. The number of particles in the RVEs and the characteristics of the normal distribution are to be optimized.

From an optimization perspective, the problem can be formulated as a bilevel, nonlinear, constrained optimization problem. The upper-level consists of determining the spatial distribution characteristics, specified by the number of particles and standard deviations in each dimension that maximizes the limit load. The lower-level consists of determining a feasible spatial distribution holding the properties specified by the candidate solutions of the upper-level problem.

2.2 Limit load simulation via LMM

2.2.1 Numerical description. In this article, the LMM is used to evaluate the PRMMC's limit load σ^x_{LL} as a special case of the shakedown procedure. Details on the numerical procedure can be found in [4]. In this section, we only attempt to provide the general background of the procedure using the same notation as [4]. Consider the following problem: A structure is subjected to a cyclic history of varying temperature $\lambda\theta(x_i, t)$ within the volume of the structure and it is subjected to a mechanical load $\lambda P(x_i, t)$. Here, λ denotes a load multiplier, which allows to consider whole classes of loading histories. Hence, there exists a linear elastic solution history, $\lambda\hat{\sigma}_{ij} = \lambda\hat{\sigma}^{\theta}_{ij} + \lambda\hat{\sigma}^{P}_{ij}$, where $\lambda\hat{\sigma}^{\theta}_{ij}$ and $\lambda\hat{\sigma}^{P}_{ij}$ are respectively the elastic solutions due to the thermal and mechanical stress. The aim of the LMM is to calculate the λ boundaries λ_{UB} and λ_{LB}, assuming that:

$$\lambda_{LB} \leq \lambda \leq \lambda_{UB}, \tag{1}$$

with an iterative process:

$$\lim_{iter\to\infty} \lambda^{iter}_{LB} = \lambda_{LB}, \qquad \lim_{iter\to\infty} \lambda^{iter}_{UB} = \lambda_{UB}. \tag{2}$$

Determining the λ value within this interval, it is possible to evaluate the effective load magnitude that represents the shakedown limit: $\tilde{P} = \lambda P$ and $\tilde{\theta} = \lambda\theta$.

The theoretical assumption behind the LMM is that the cyclic stress and the nonlinear strain can be described as a series of iterative linear elastic solutions in which the moduli are spatial and time-dependent. After a sufficient number of cycles, the stresses and strain rates can be approximated by cyclic states, i.e,

$$\sigma_{ij}(t) = \sigma_{ij}(t + \Delta t), \qquad \dot{\epsilon}_{ij}(t) = \dot{\epsilon}_{ij}(t + \Delta t). \tag{3}$$

Therefore, the asymptotic stress history may be decomposed into the sum of three terms. This leads to the expression:

$$\sigma_{ij}(x, t) = \lambda\hat{\sigma}_{ij}(x, t) + \bar{\rho}_{ij}(x) + \rho^{r}_{ij}(x, t), \tag{4}$$

where $\lambda\hat{\sigma}_{ij}$ represents the linear cyclic elastic stress solution, $\bar{\rho}_{ij}$ represents the residual stress field at the beginning and at the end of the cycle while ρ^{r}_{ij} represents the residual stress occurring within the cycle and holds:

$$\rho^{r}_{ij}(x, 0) = \rho^{r}_{ij}(x, \Delta t) = +\bar{\rho}^{r}_{ij}. \tag{5}$$

The shakedown condition imposes the time dependent component of the residual stress $\rho^{r}_{ij} = 0$ and consequently, the cyclic stress history at shakedown has the form [10]:

$$\sigma_{ij} = \lambda\hat{\sigma}_{ij}(x, t) + \bar{\rho^{r}_{ij}}. \tag{6}$$

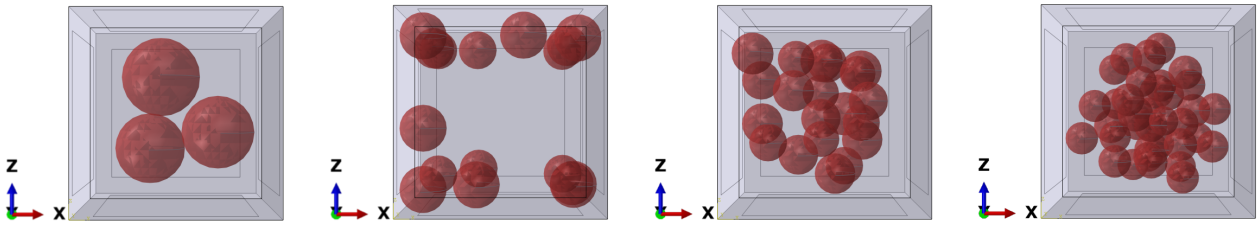


Figure 1: Examples of RVEs with different number of particles and particles' spatial distributions.

The upper bound shakedown limit λ_{UB}^{iter} is evaluated by:

$$\lambda_{UB}^{iter} = \frac{\int_V \int_0^{\Delta_t} \sigma_y \bar{\dot{\epsilon}}(\dot{\epsilon}_{ij}) dt dV}{\int_V \int_0^{\Delta_t} \hat{\sigma}_{ij}(\hat{\epsilon}_{ij}) dt dV}, \tag{7}$$

in which $\sigma_y$is the temperature-dependent yield, $\dot{\epsilon}_{ij}$ is a kinematically admissible strain rate and $\bar{\dot{\epsilon}} = \sqrt{\frac{2}{3}\dot{\epsilon}_{ij}\dot{\epsilon}_{ij}}$ is the effective strain rate. The definition of the shakedown condition enforces that no plastic strain accumulation will occur if the combination of the applied elastic stresses and a constant residual stress field satisfy the von Mises yield criterion at any location. Hence, the lower bound of the shakedown limit can be computed with:

$$\lambda_{LB}^{iter} = max \tilde{\lambda}_{LB}^{iter} \text{ subject to } f(\tilde{\lambda}_{LB}^{iter} \hat{\sigma}_{ij}(x_i, t) + \bar{\rho}_{ij}(x_i)) \leq 0, \tag{8}$$

at each integration point and for all load instances. At each iteration, λ_{UB}^{iter} is evaluated by the obtained kinematic field and λ_{LB}^{iter} is calculated by the obtained static field. The iterative procedure is halted when the desired level of convergence is reached arriving to consider $\lambda_{LB} = \lambda_{LB}^{iter}$ and $\lambda_{UB} = \lambda_{UB}^{iter}$. The load configuration used in this research is characterized by $\hat{\sigma}_{ij}^{\theta} = 0$ and a monotonic mechanical load, therefore σ_{LL}^{x} can be expressed by:

$$\sigma_{LL}^{x} = \hat{\sigma}_{ij} = \lambda \hat{\sigma}_{ij}^{P}.$$

For the scope of this study, we assume $\lambda = \lambda_{UB}$, arriving to the relation:

$$\sigma_{LL}^{x} = \lambda_{UB} \hat{\sigma}_{ij}^{P}. \tag{9}$$

Thus, the shakedown multiplier $\lambda = \lambda_{UB}$ can be named limit load multiplier λ^{LL}.

2.2.2 Implementation. The simulation is implemented in Abaqus. Since the optimization algorithms are implemented in the programming language R, an interface between R and Abaqus is required. This has been realized with the Abaqus Scripting Interface (ASI). With the ASI, Abaqus commands can be executed via Python code, which is interfaced with R via the command line interface.

Once the particle placement has been defined, a command line call is used to run Python code. The Python code generates the FE model which creates the matrix geometry and arranges the particles correctly. Meshes are created with Abaqus automesher. To avoid a decreased accuracy due to inadequacies of element shapes and dimension uniformity, the minimum distance between surfaces Δ is imposed equal to 0.1. Moreover, this pre-processing code automatically assigns material properties, generates the mesh, configures the LMM subroutine, runs the PBC code and submits the simulations. Once the analysis is completed, the limit load multiplier λ^{LL} is read from a file generated by Abaqus that contains the analysis summary. The value is used as an observation which is fed back to the optimization algorithm. The FE analyses, performed in parallel-mode on 3 cores/6 threads on a workstation presenting Intel i7-4770 processor with 12Gb of dedicated memory, lasted approximately 35m (wall time).

2.3 Optimization problem

The bilevel optimization process is split into two different optimization problems, one nested into the other. They will be referred to as *upper-level* and *lower-level* optimization.

2.3.1 Upper-level optimization. From Eq. 9, one can see that $\sigma_{LL}^{x} \propto \lambda^{LL}$, because $\hat{\sigma}_{ij}^{P}$ is constant. Due to this fact, optimizations of σ_{LL}^{x} or λ^{LL} are equivalent. We decided to maximize and report λ^{LL}.

The objective of the upper-level problem is hence to maximize λ^{LL} by trying to find the best combination of the variables of interest. The variables of interest describe a distribution of particles in the material (number of particles, standard deviations in each dimension). Since we consider distributions, the results will differ depending on each individual sample from a distribution. The exact positions of the particles are not defined. Thus, the objective function is non-deterministic. More formally, the upper-level optimization can be defined as:

$$\text{Find max} \quad f(Npart, \sigma_x, \sigma_y, \sigma_z) = \lambda^{LL} \tag{10}$$

$$\text{subject to} \quad \begin{cases} 2 < Npart < 50 \\ 0.7 < \sigma_x, \sigma_y, \sigma_z < 2.5 \end{cases}, \tag{11}$$

where $Npart$ is the number of particles per RVE, σ_x, σ_y, σ_z are respectively the standard deviations of the particle placements along the x, y, and z axes.

2.3.2 Lower-level optimization. To perform the FE analysis, the exact placement of all the N_{part} particles has to be addressed. The position of each particle along each axis constitutes the set of the problem design variables.

The upper-level problem has a fixed number of variables (four: $Npart, \sigma_x, \sigma_y, \sigma_z$). In contrast, the number of lower-level optimization variables is not fixed. In fact, this number depends on the

upper-level variables. That is, if another particle is added ($Npart$ is increased by one), then three new variables have to be optimized in the lower level (one for each coordinate of the new particle's position).

The goal of the lower-level optimization is to specify the exact positions of all particles, that respects the statistical properties specified by an upper-level candidate solution. Therefore, the objective function to be minimized is defined as the deviation between actual sample statistics and the desired distribution. Furthermore, constraints that limit the feasible region are imposed: Non-linear inequality constraints avoid overlapping particles while box-constraints ensure that particles are fully embedded in the material matrix. Formally, the lower-level optimization problem can be defined as:

$$\begin{aligned}\text{Find min } g(x_x^i, x_y^i, x_z^i, i = 1, ..., N_{part}) = |sd(x_x^1, ..., x_x^{N_{part}}) - \sigma_x| + \\ ||sd(x_y^1, ..., x_y^{N_{part}}) - \sigma_y| + |sd(x_z^1, ..., x_z^{N_{part}}) - \sigma_z|\end{aligned} \tag{12}$$

subject to

$$\begin{cases} x_x^i, x_y^i, x_z^i \in [l_b + r + \Delta, u_b - r - \Delta], i = 1, ..., N_{part}, \\ C_{ij} = \Delta - \sqrt{(x_x^i - x_x^j)^2 + (x_y^i - x_y^j)^2 + (x_z^i - x_z^j)^2)} \leq 0, \\ \quad i, j = 1, ..., N_{part}, j \neq i, \end{cases} \tag{13}$$

where x_x^i, x_y^i, x_z^i are the i-th particle's exact position along the three axes, $\sigma_x, \sigma_y, \sigma_z$ are the reference statistical properties of the distribution defined by the upper-level candidate solution, sd computes the sample standard deviation, and $l_b = 0, u_b = 10$ represent the coordinates of the matrix bounds. The particle radius r strictly depends on N_{part} through the expression

$$r = \frac{3 \times Volume \times Vf}{(4\pi \times N_{part})^{1/3}}$$

and Δ is the minimum distance between the particles and between the particles and the matrix bounds already mentioned in Sec. 2.2.2.

3 METHODS

Here, we describe the methods that have been employed to resolve the bilevel PRMMC design problem. A GA is chosen to solve the lower-level problem. Sequential Parameter Optimization (SPO) [1] has been used as a tuner for the lower-level GA optimizer, and also to solve the upper-level problem. The details are provided in the following.

3.1 Upper-level optimization via Kriging and SPO

To solve the upper-level optimization problem, we require a method that accounts for the stochastic and expensive nature of the problem. Hence, we used the Sequential Parameter Optimization Toolbox (SPOT), which is an implementation of the SPO in the programming language R. From the set of surrogate models provided by SPOT, we decided to use Kriging as a surrogate model. Kriging assumes that the data follows a multi-variate Gaussian distribution, where errors are spatially correlated. A more detailed and easy to follow description of Kriging is given by Forrester et al. [8]. Importantly, the spatial correlation of the data is encoded within a kernel function. A frequently employed correlation function that models the correlation between samples (or candidate solutions) is the Gaussian kernel $k(z, z') = \exp(-\sum_{i=1}^{n} \theta_i |z_i - z_i'|^2)$. Here, n is the number of modeled variables (search space dimension), θ_i is a parameter of the kernel (determined by Maximum Likelihood Estimation (MLE)). Furthermore, z as well as z' are potential candidate solutions (or samples). In our case, they are $z = \{Npart, \sigma_x, \sigma_y, \sigma_z\}$. Employing such a kernel, a Kriging model produces the following predictor:

$$\hat{y}(z^*) = \hat{\mu} + \mathbf{k}^T \mathbf{K}^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu}), \tag{14}$$

where $\mathbf{y}$ are the training observations, $\hat{y}(z^*)$ is the predicted function value of a new sample z^*, $\hat{\mu}$ represents the process mean determined by MLE, $\mathbf{1}$ is a vector of ones, $\mathbf{K}$ is the matrix that collects all pair-wise correlations of the training samples $\mathbf{Z}$, and $\mathbf{k}$ is the column vector of correlations between the set of training samples $\mathbf{Z}$ and the new sample z^*. After appropriate training, such a predictor may be employed to replace an expensive objective function.

In model based optimization, Kriging is a popular choice, as it also provides an estimate of its own prediction uncertainty. This estimate can be used to balance exploration and exploitation by computing the expected improvement (EI) of candidate solutions [21]. The uncertainty of the model is computed with

$$\hat{s}^2(z^*) = \sigma^2_{process}(1 - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}), \tag{15}$$

where $\sigma_{process}$ is the process variance, determined by MLE. If the uncertainty is zero, the EI is also zero. Else, the uncertainty is non-zero, EI is

$$\text{EI}(z^*) = \mathbf{y}_{imp} \Phi\left(\frac{\mathbf{y}_{imp}}{\hat{s}(z^*)}\right) + \hat{s}(z^*) \phi\left(\frac{\mathbf{y}_{imp}}{\hat{s}(z^*)}\right),$$

where $\mathbf{y}_{imp} = \min(\mathbf{y}) - \hat{y}(z^*)$. $\Phi()$ indicates the normal cumulative distribution function. Respectively, $\phi()$ is the probability density function.

It has to be noted, that the above description of Kriging presents an interpolating model, which assumes zero error at already observed locations. Clearly, this does not take noise or uncertainty into account. One way to account for noise is to introduce the so-called nugget effect. This essentially adds a constant value η to the diagonal of the kernel matrix $\mathbf{K}$. The parameter η is determined by MLE. The nugget effect enables the model to regress the observed data, and hence smoothen noisy observations. Furthermore, it may now produce a non-zero estimate of the uncertainty at observed locations.

Algorithm 1 Sequential parameter optimization

```
1: t = 0. P(t) = SetInitialPopulation().
2: Select one or several surrogate models 𝔐.
3: Evaluate(P(t)) on f.
4: while not TerminationCriterion() do
5:     Use P(t) to build a model M(t) using 𝔐.
6:     P'(t + 1) = GlobalSearch(M(t)).
7:     Evaluate(P'(t + 1)) on f.
8:     P(t + 1) = P(t) ∪ P'(t + 1).
9:     t = t + 1.
10: end while
```

By employing a surrogate model (e.g., Kriging), SPO finds improved solutions in the following way (see Algorithm 1). A set of candidate solutions is created by a design of experiment method (e.g., Latin-hypercube sampling). Then, the solutions are evaluated with the objective function. Next, the surrogate model is built. Then, an optimizer searches for the most promising candidate solution by optimizing an infill criterion (here: expected improvement). The new candidate solution is evaluated with the objective function, potentially with replications to account for uncertainty. These expensive objective function values are added to the previously evaluated ones. Then, the surrogate model can be updated to improve its accuracy. These steps are repeated until a satisfying solution has been found or the computational budget is exhausted.

This standard SPO procedure has to be modified to account for feasibility issues that are present in the PRMMC design problem. In fact, the upper-level optimization is a constrained problem with unknown feasible regions. It is possible to define particle distributions which are physically infeasible: If too many or too large particles are present, they may not respect the desired distribution parameters anymore. Then, the lower-level optimization cannot succeed in finding adequate particle placements that satisfy the desired properties of the upper-level candidate solutions. Hence, there will be some unknown difference between the sample statistics of the actual particle placements

$$\tilde{\sigma}_x = sd(x_x^1, ..., x_x^{N_{part}})$$
$$\tilde{\sigma}_y = sd(x_y^1, ..., x_y^{N_{part}})$$
$$\tilde{\sigma}_z = sd(x_z^1, ..., x_z^{N_{part}})$$

and the desired statistics of the candidate solution $\sigma_x, \sigma_y, \sigma_z$. Thus, the lower-level optimization may result into particle placements that follow a distribution that is different from the desired one. To overcome this problem, SPO was modified to allow for updating proposed candidate solutions *after* they were evaluated. In essence, the lower-level optimization may replace an infeasible candidate solution of the upper-level problem by a different candidate that is feasible. After each upper-level objective function evaluation, an update to the upper-level candidate variables is made by replacing $\sigma_x, \sigma_y, \sigma_z$ with $\tilde{\sigma}_x, \tilde{\sigma}_y, \tilde{\sigma}_z$.

3.2 Lower-level optimization via GA

Finding a feasible particle placement that is as close as possible to the desired distribution characteristics given by an upper-level candidate solution represents a challenging issue. This is due to the high dimensionality of the problem, its non-linearity and its constraints. For these reasons, this assignment has been treated as a complex black-box optimization problem. The interest in a globally optimal solution, the lack of preliminary information about the objective function features and the necessity of a robust optimizer lead us to adopt an evolutionary algorithm for this task. In detail, we chose the Genetic Algorithm (GA) available in the R package `GA` [23] because of its robustness and ease of use.

Choosing a GA produces an additional issue: the optimization performance strongly relies upon its parametrization. Finding an appropriate set of algorithm parameters is not a trivial problem and a good solution can lead to a considerable improvement of algorithm performance. Hence, we decided to tune the GA in a preliminary study. SPO is chosen to tune the GA.

4 EXPERIMENTS

4.1 Setup

4.1.1 Preliminary lower-level optimization tuning. A preliminary tuning process has been conducted on the GA design parameters in order to reduce the computation time required by each lower-level optimization run. Particularly, SPOT has been employed to perform a model based optimization of the GA parameters. The probability of crossover between pairs of chromosomes (`crossover`), the number of best fitness individuals to survive at each generation (`elitism`), the population size (`popSize`), the probability of performing a local search at each iteration of the GA (`poptim`), the probability to select the solutions with the largest fitness as starting point of the local search (`pressel`) and the maximum number of iterations of the local search (`maxiter`) have been tuned. Similarly to the upper-level optimization, Kriging has been adopted as surrogate model and has been optimized by a L-BFGS-B algorithm. The first model has been built using ten samples generated by Latin hypercube sampling. The number of expensive evaluations used to train the model has been limited to 150 and each candidate has been evaluated three times to account for uncertainty. Furthermore, the maximum number of function evaluations on the model has been fixed to 800. GAs are stochastic optimizers. To receive reliable performance estimates, the tuning procedure has been repeated with 40 different combinations of GA and SPOT random number generator seeds, for two possible upper-level candidate solutions.

4.1.2 Lower-level. The employed GA implementation has the capability of considering box-constraints but not non-linear inequality constraints. For this reason, a penalty approach has been used to direct the search to feasible regions. A constraint tolerance equal to ϕ = 1e-4 has been enforced. A penalty is added to the objective function value g as follows:

$$g_{\text{penalized}} = \begin{cases} g + \psi C & \text{if } C > \phi, \text{ unfeasible solution,} \\ g & \text{if } C \leq \phi, \text{ feasible solution,} \end{cases} \tag{16}$$

where ψ = 1e3. We define C^+ as the set of all the l positive values among all the C_{ij}. Then, C is calculated with:

$$C = \sum_l C_l^+. \tag{17}$$

Before every run, the box-constraints have to be calculated with respect to the upper-level candidate solution variable N_{part}. This study has been conducted with a fixed reinforcement fraction volume. That means, the combined volume of the particles stays constant. Consequently, particles sizes have to change when their number changes. The box-constraints are the maximum and minimum values that the particles' centers can assume to assure a minimum distance between particle and matrix surfaces of at least Δ, see Eq. (13). The maximum number of GA generations has been set to 2e4. All the other parameters shown in Table 2 have been fixed after the tuning process.

4.1.3 Upper-level. The upper-level optimization has been performed using the SPOT. An initial set of ten candidate solutions is

Table 2: The search bounds and spacing that has been employed for the tuning of the lower-level optimization algorithm, i.e., the GA. Results of the preliminary lower-level tuning. The depicted GA parameter values have been determined to be optimal in the preliminary tuning experiments.

Values	crossover	elitism	popSize	poptim	pressel	maxiter
Lower bound	0.5	1	25	0	0	100
Upper bound	0.8	10	80	0.001	1	500
Spacing	-	1	1	0.0001	0.1	1
Optimum configuration	0.59	3	32	0	-	-

created by Latin hypercube sampling using the `designLHD` function. We employ a Kriging model as implemented in the R package `SPOT`: `buildKriging`. The `noise` parameter has been set to (`TRUE`) due to the stochastic nature of the problem. The Kriging surrogate will be subject to optimization via L-BFGS-B for a maximum number of 700 evaluations. Initially, new candidate solutions will be evaluated three times to account for uncertainty. The variable σ_x, σ_y and σ_z have been treated as *continuous* variables, whereas N_{part} has been considered as an *integer* variable in SPOT.

4.1.4 LMM configuration. The LMM has been used in this context to evaluate the limit load. Therefore, a load configuration of the type $\lambda\hat{\sigma}_{ij} = \lambda\hat{\sigma}^P_{ij}$ has been imposed setting $\lambda\hat{\sigma}^\theta_{ij} = 0$. The reference value $\hat{\sigma}^P_{ij}$ has been set to 6e2 MPa, to simulate a force of 6e4 N applied to the whole matrix surface. Based on previous simulations, this value has been chosen to assure that the load multiplier always respects $0 < \lambda^{LL} < 1$. The maximum number of iterations has been limited to 300 and the convergence level has been fixed to 1e-4.

4.2 Results and analysis

4.2.1 Preliminary lower-level optimization tuning. To ensure good performance of the most critical cases (large number of particles), the GA has been tuned. It was tuned for the two reference cases (which are potential upper-level solutions) reported in Table 3.

The results of the tuning are presented in Table 2. Sporadic local optimizations appear to not improve the GA's performance (i.e., `poptim` is zero). Since the values of `pressel` and `maxiter` do not matter when `poptim` is zero, they are omitted.

Table 3: The two different upper-level candidate solutions for which a tuning of the lower-level GA has been performed.

Configurations	σ_x	σ_y	σ_z	N_{part}
1	2	2	2	20
2	1.5	1.5	1.5	50

4.3 Upper-level optimization

The results showed that the search procedure was able to find a promising solution rather early. Overall, 500 iterations were performed but there was no improvement after the 120^{th} iteration, as shown in Fig. 2.

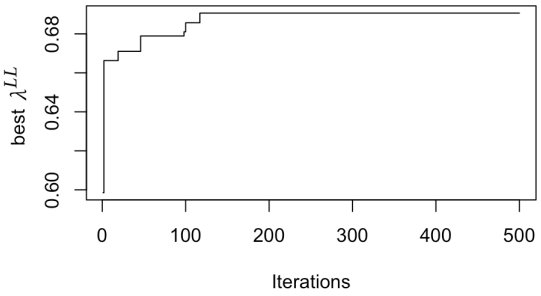


Figure 2: Evolution of the best observation over the iterations of the upper-level optimization.

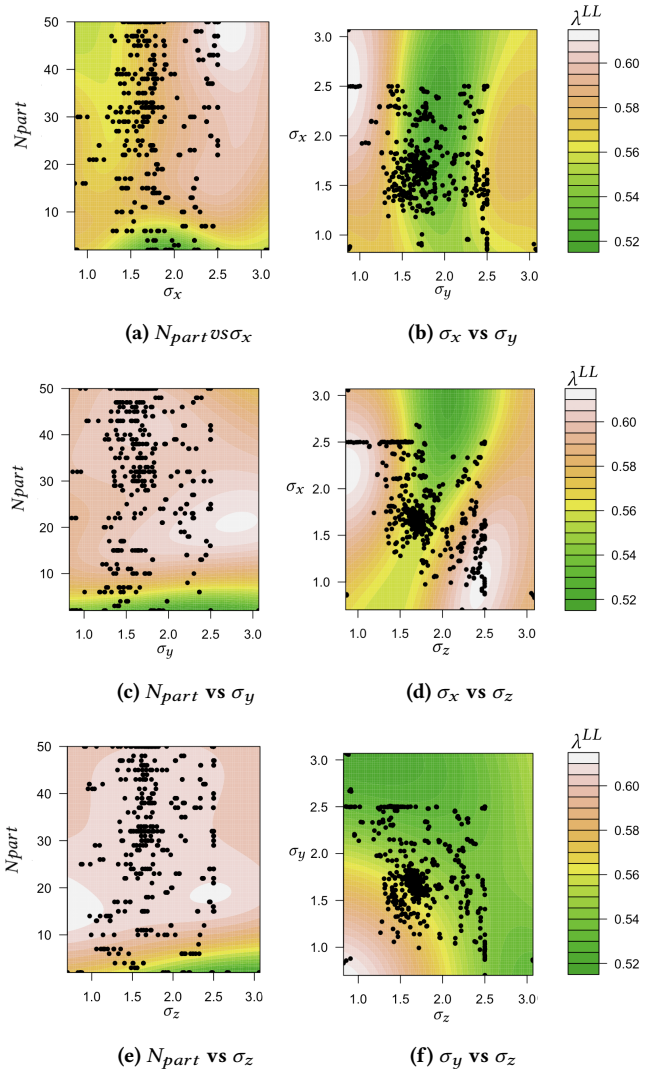
Table 4: Best-found upper-level problem design and variable importance ranking, based on the kernel parameter θ of the last Kriging model trained during the optimization process.

	$Npart$	σ_x	σ_y	σ_z
Best-found values	45	2.463	1.320	1.572
Importance (θ)	7.714	4.007	1.827	1.68

The results obtained by the upper-level optimization may help to understand the effect of the particles' spatial distribution on the limit load σ^x_{LL}. To that end, we use the last surrogate model that has been trained by SPOT. The importance of the different variables can be estimated from the activity parameters θ (which are determined for each problem variable) in that model. The θ values are shown in Table 4. The number of particles has the largest value and hence seems to have the strongest effect on the limit load. This is not an unexpected result: with a fixed reinforcement fraction volume, a change of the number of particles results in a change of their size. It is well-known from the literature that two things play a prevailing role in the interaction between the two PRMMC phases: the ratio between particle volume and their surface as well as the particles' absolute radius of curvature [28, 29]. Among the remaining variables, the most significant for this optimization is the standard deviation of the particle distribution along the x-axis σ_x. This result can easily be explained by the physical conditions of this study. The analysis has been conducted applying a distributed uni-axial load in the direction of the x-axis. This means that, although the composite structure is considerably inhomogeneous, the prevalent stress will be directed in the direction of the x-axis. Therefore, changing the distribution along that direction will have an immediate effect on the material's ability to resist stresses. This is confirmed by the importance of the other two axes' distribution characteristics σ_y and σ_z. These two directions are orthogonal to the stress applied, which explains their similar importance.

The best parameter values found during the optimization run are reported in Table 4.

Following from the earlier discussion of the parameter importance, it is clear that the particles will tend to assume a rather narrow distribution with respect to the x-plane and spread in the



(a) N_{part} vs σ_x (b) σ_x vs σ_y

(c) N_{part} vs σ_y (d) σ_x vs σ_z

(e) N_{part} vs σ_z (f) σ_y vs σ_z

Figure 3: Visualizations of the objective function landscapes in respect of all variable combinations. For each individual plots, variables that are not shown are fixed to the respective optimal values.

x-axis. This means that the arrangement tends to align the particles, to create a sort of fragmented fiber. It is well-known that the fiber reinforced metal matrix composites are among the most well performing materials in case of uniaxial stress, but they are usually more expensive to produce than PRMMCs. With this optimal configuration, a value of $\lambda^{LL}_{opt} = 0.691$ has been determined. The corresponding limit load σ^{x}_{LL} is equal to: $\lambda^{LL} \times \sigma^{P}_{ij} = 414.6MPa$.

It is worth to highlight that λ^{LL}_{opt} represents a significant improvement with respect to the worst value found in the optimization process, $\lambda^{LL}_{worst} = 0.519$. This constitutes a relative improvement of around 33%.

An additional analysis has been performed to validate that the best solution found during the optimization (final best) is actually better than the best solution from the initial design (initial best). To

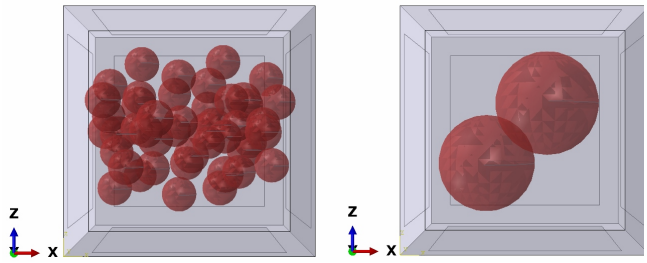


Figure 4: Comparison of examples of RVEs with the optimum particles distribution (left) and the worst (right) found during the optimization process.

that end, 100 replicates have been performed with the LMM analysis, for each candidate. The results are shown in the histogram in Fig. 5. The plot indicates that the solution improved considerably, even taking the uncertainty of the objective value into account. This is confirmed by statistical tests: the parametric Welch's t-test as well as the non-parametric Wilcoxon rank sum test report p-values $<$ 1e-16. That means, there is sufficient evidence to claim that the difference between these two solutions is significant.

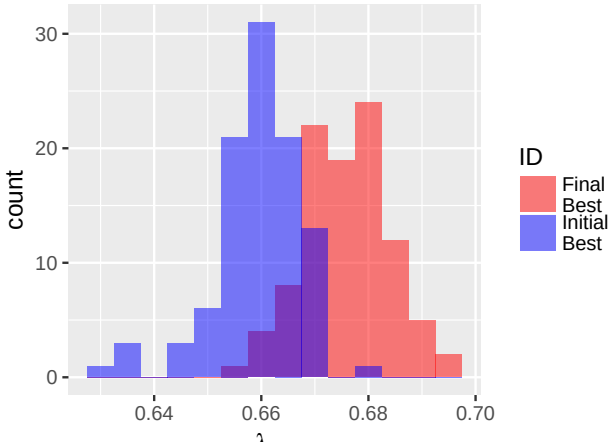


Figure 5: Validation: A comparison of histograms for the final best, λ^{opt}_{UB}, and the initial best solution. Histograms are based on 100 simulation replications.

5 CONCLUSIONS AND OUTLOOK

This study aimed to maximize the limit load multiplier of a PRMMC subjected to a monotonic mechanical load. To that end, we employed a bilevel SMBO approach coupled with an FE simulation solver and the LMM. For this investigation, maximizing the limit load multiplier results into the maximization of the limit load, which was the original property of interest. The upper-level optimization has been performed using `SPOT` with a Kriging surrogate model. The lower-level optimization has been performed using a GA. To improve the GA performance, its parameters were additionally tuned

by SPOT. This combined procedure (tuning plus lower and upper-level optimization) yields a promising solution to the PRMMC design problem. The results of this research clearly highlight that, with a fixed reinforcement fraction volume, both the number of particles per RVE and the particles' spatial distribution strongly influence the limit load. The optimum RVE configuration obtained through the proposed optimization process performs up to 33% better than a non-optimized solution.

The problem design variables do not deterministically define the value of the limit load multiplier. Therefore, an uncertainty analysis on the initial and final optimum configurations was used to address the effect of the exact particle placement. Indeed, the results revealed that the proposed method found significantly improved solutions.

However, to judge whether the algorithm in fact produces such results consistently, and to avoid possible bias due to stochastic procedures, a single optimization run is not sufficient. For future work, we propose to validate the performance of the proposed optimization framework by performing repeated runs of the algorithm. To avoid the obstacle of high evaluation times, such replicated runs may employ considerably fewer evaluations of the objective function. Less than 25% of the run simulations were sufficient to find the estimated optimum, hence, more effort could have been spent in uncertainty handling and replications.

Furthermore, an in-detail study should consider to tune the upper level optimization process. Similar to the GA, SPOT has parameters that affect its performance, for example, the chosen optimizer of the surrogate model (here: L-BFGS-B). This is a continuous, local optimizer. This is not necessarily problematic, since the surrogate model may help to smoothen the ruggedness produced by the integer variable ($Npart$). Also, the locality of L-BFGS-B may not be problematic, since each iteration of SPOT yields a restart of the L-BFGS-B algorithm, hence enabling a global search. Still, an alternative which is global and able to handle ordinal integers may be preferable for this test case. A mixed-integer evolutionary algorithm may hence be a promising choice.

Considering the results obtained, the authors also plan to leverage the flexibility of the framework, to extend the research to more load conditions and different PRMMCs configurations.

ACKNOWLEDGMENTS

This work is funded by the European Commission's H2020 programme, through the UTOPIAE Marie Curie Innovative Training Network, H2020-MSCA-ITN-2016, Grant Agreement number 722734.

REFERENCES

[1] Thomas Bartz-Beielstein, Christian WG Lasarczyk, and Mike Preuß. 2005. Sequential parameter optimization. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, Vol. 1. IEEE, 773–780.

[2] Helmut J Böhm and Wensheng Han. 2001. Comparisons between three-dimensional and two-dimensional multi-particle unit cell models for particle reinforced metal matrix composites. *Modelling and Simulation in Materials Science and Engineering* 9, 2 (2001), 47.

[3] RD Carnahan. 1968. Elastic properties of silicon carbide. *Journal of the American Ceramic Society* 51, 4 (1968), 223–224.

[4] Haofeng Chen. 2010. Lower and upper bound shakedown analysis of structures with temperature-dependent yield stress. *Journal of Pressure Vessel Technology* 132, 1 (2010), 011202.

[5] Haofeng Chen and Alan RS Ponter. 2005. On the behaviour of a particulate metal matrix composite subjected to cyclic temperature and constant stress. *Computational materials science* 34, 4 (2005), 425–441.

[6] Haofeng Chen and Alan RS Ponter. 2010. A direct method on the evaluation of ratchet limit. *Journal of Pressure Vessel Technology* 132, 4 (2010), 041202.

[7] ASM Handbook Committee et al. 1978. Metals Handbook: Vol. 2, Properties and selection–nonferrous alloys and pure metals. *American Society for Metals, Metals Park, OH* (1978).

[8] Alexander Forrester, Andy Keane, et al. 2008. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons.

[9] Dario Giugliano, Daniele Barbera, and Haofeng Chen. 2017. Effect of fiber cross section geometry on cyclic plastic behavior of continuous fiber reinforced aluminum matrix composites. *European Journal of Mechanics-A/Solids* 61 (2017), 35–46.

[10] Dario Giugliano and Haofeng Chen. 2016. Micromechanical modeling on cyclic plastic behavior of unidirectional fiber reinforced aluminum matrix composites. *European Journal of Mechanics-A/Solids* 59 (2016), 155–164.

[11] Caroline González, Javier Segurado, and Javier Llorca. 2004. Numerical simulation of elasto-plastic deformation of composites: evolution of stress microfields and implications for homogenization models. *Journal of the Mechanics and Physics of Solids* 52, 7 (2004), 1573–1593.

[12] Andrei A Gusev. 1997. Representative volume element size for elastic composites: a numerical study. *Journal of the Mechanics and Physics of Solids* 45, 9 (1997), 1449–1459.

[13] Wensheng Han, Anton Eckschlager, and Helmut J Böhm. 2001. The effects of three-dimensional multi-particle arrangements on the mechanical behavior and damage initiation of particle-reinforced MMCs. *Composites science and technology* 61, 11 (2001), 1581–1590.

[14] Karl U Kainer. 2006. *Metal matrix composites: custom-made materials for automotive and aerospace engineering*. John Wiley & Sons.

[15] Xaiopeng Li, Lachlan Astfalack, Patrick Drew, Yor Schliwa, and Timothy B Sercombe. 2016. Selective Laser Melting Of Aluminium Metal Matrix Composites. (2016).

[16] Javier Llorca and Javier Segurado. 2004. Three-dimensional multiparticle cell simulations of deformation and damage in sphere-reinforced composites. *Materials Science and Engineering: A* 365, 1 (2004), 267–274.

[17] S Masurtschak and RA Harris. 2011. Enabling techniques for secure fibre positioning in ultrasonic consolidation for the production of smart material structures. In *Proc. of SPIE Vol*, Vol. 7981. 79810M–1.

[18] JC Michel, H Moulinec, and P Suquet. 1999. Effective properties of composite materials with periodic microstructure: a computational approach. *Computer methods in applied mechanics and engineering* 172, 1-4 (1999), 109–143.

[19] Leon L Mishnaevsky and S Schumacher. 2001. Continuum mesomechanical finite element modeling in materials development: A state-of-the-art review. *Applied Mechanics Reviews* 54, 1 (2001), 49–68.

[20] Leon L Mishnaevsky Jr. 1998. *Damage and fracture of heterogeneous materials*. CRC Press.

[21] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. 1978. *Towards Global Optimization 2*. North-Holland, Chapter The application of Bayesian methods for seeking the extremum, 117–129.

[22] Olivier Pierard, Caroline Gonzalez, Javier Segurado, Javier Llorca, and Issam Doghri. 2007. Micromechanics of elasto-plastic materials reinforced with ellipsoidal inclusions. *International Journal of Solids and Structures* 44, 21 (2007), 6945–6962.

[23] Luca Scrucca. 2013. GA: A Package for Genetic Algorithms in R. *Journal of Statistical Software* 53, 4 (2013), 1–37. http://www.jstatsoft.org/v53/i04/

[24] Javier Segurado, Caroline Gonzalez, and Javier Llorca. 2003. A numerical investigation of the effect of particle clustering on the mechanical properties of composites. *Acta Materialia* 51, 8 (2003), 2355–2369.

[25] Javier Segurado and Javier Llorca. 2002. A numerical approximation to the elastic properties of sphere-reinforced composites. *Journal of the Mechanics and Physics of Solids* 50, 10 (2002), 2107–2121.

[26] Javier Segurado and Javier Llorca. 2006. Computational micromechanics of composites: the effect of particle spatial distribution. *Mechanics of materials* 38, 8 (2006), 873–883.

[27] Javier Segurado, Javier Llorca, and Caroline González. 2002. On the accuracy of mean-field approaches to simulate the plastic deformation of composites. *Scripta Materialia* 46, 7 (2002), 525–529.

[28] Siamak Soleymani Shishvan and Amir-Hossein Asghari. 2016. Particle size effect in metal matrix composites: a study by the continuum theory of stress gradient plasticity. *Journal of Composite Materials* 50, 13 (2016), 1717–1723.

[29] Wei Yueguang. 2001. Particulate size effects in the particle-reinforced metal-matrix composites. *Acta Mechanica Sinica* 17, 1 (2001), 45.