fluids

MDPI

Article

Numerical Optimization of Drucker-Prager-Cap Model Parameters in Powder Compaction Employing Particle Swarm Algorithms

Sanaz Davarpanah [1,*], Madjid Allili [1] and Seyed Soheil Mousavi Ajarostaghi [2,*]

[1] Department of Computer Science, Bishop's University, Sherbrooke, QC J1M 1Z7, Canada; mallili@ubishops.ca
[2] Mechanical Engineering Department, Université de Sherbrooke (UDS), Sherbrooke, QC J1K 2R1, Canada
* Correspondence: sdavarpanah20@ubishops.ca (S.D.); seyed.soheil.mousavi.ajarostaghi@usherbrooke.ca (S.S.M.A.)

Abstract: A growing number of scholars are drawn to using numerical approaches powered by computer simulations as a potential solution to industrial problems. Replicating the compaction process in powder metallurgy with accuracy is one such issue. The Drucker-Prager-Cap model requires parameter calibration as the most used method for simulating powder compaction. This paper addresses this issue and presents a new technique for doing so. Utilizing Abaqus software 2020, the compaction process was simulated for the benchmark powder, which is the alloy Ag57.6-Cu22.4-Sn10-In10. The difference between simulation results and experimental data was reduced by applying the Particle Swarm Optimization technique in Python. The suggested approach may accurately forecast the Drucker-Prager-Cap model parameters, as demonstrated by comparing the optimized parameters utilizing the research's method with their experimental values. The findings revealed how well the suggested approach in this study calibrated the DPC model, yielding three parameters—Young's modulus, material cohesion, and hydrostatic pressure yield stress—with respective RMSEs of 1.95, 0.12, and 324.64 concerning their experimental values.

Keywords: model calibration; drucker-prager-cap; particle swarm optimization; powder compaction process; inverse optimization

check for updates

Citation: Davarpanah, S.; Allili, M.; Mousavi Ajarostaghi, S.S. Numerical Optimization of Drucker-Prager-Cap Model Parameters in Powder Compaction Employing Particle Swarm Algorithms. *Fluids* **2024**, *9*, 262. https://doi.org/10.3390/fluids9110262

Academic Editor: Andrew S. Rees

Received: 31 August 2024
Revised: 29 October 2024
Accepted: 30 October 2024
Published: 10 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Powder compaction is converting any type of powdered material into the desired shape. The process is done by compressing the powder particles into a die. It can be accompanied by additional operations such as re-compressing, heating, machining, and surface polishing if necessary. The powder compaction process offers several outstanding features. These include economic efficiency in large-scale production, low waste of raw materials, environmental and safety benefits, and the ability to achieve high dimensional accuracy in a single step. Additionally, it allows for mixing powder particles to produce composite parts. These advantages have made the method widely used in industries such as pharmaceuticals, ceramics, powder metallurgy, and the production of detergent and chemical tablets [1].

Along with the unique advantages that have caused interest in various industrial fields for the powder compaction method, the complexity of the modeling process has caused its application to be limited in some cases. The reason is that modeling the process requires costly and time-consuming experimental procedures to determine the material parameters in the constitutive model. One idea to overcome this limitation is to use the inverse optimization method to determine the parameters, which has attracted much attention recently. The present research seeks to implement the abovementioned idea to model the powder compaction process and thus expand its application in industry.

Modeling is a process that simulates the mechanical behavior of a material during a physical or chemical process, considering the influence of effectual parameters. Therefore,

modeling the phenomenon of powder compaction is a process that leads to predicting the behavior of powder particles during compaction and estimating the properties of the parts resulting from compaction. The research conducted in modeling the powder compaction process can be classified based on two general approaches: phenomenological and micromechanical. The phenomenological approach considers the powder as a continuous medium and thus uses the constitutive equations governed by continuum mechanics. In this approach, the parameters caused by the particulate nature of the powder environment (e.g., movement of particles, interaction between them, inter-particle bonds, and friction) are ignored, and it is assumed that the powder is compacted only due to the movement of its boundaries. Based on the given approach, researchers who modeled the behavior of powder during the compaction and predicted the final properties of the compacted parts used different yield criteria, including yield criteria in the field of metal powders [2], mechanics of porous materials [3], and even fluid mechanics [4]. Elastic properties (Young's modulus and Poisson's ratio), plastic properties (shear yield, strain hardening, and porosity), and interaction (friction between powder particles and die wall) are among the most common parameters that are considered in the phenomenological-based material models. Considering the complexity of the analytical method in powder compaction modeling, some researchers have applied empirical [5–7] and semi-empirical [8–11] relationships based on phenomenological models. These relationships are used in industrial centers to predict the compaction pressure necessary to achieve a specific density.

Another approach in powder compaction modeling is the micromechanical approach. With the given approach, the mold's powder environment is considered a set of discrete particles. Most researchers using the micromechanical approach to model the powder compaction have considered the powder particles as spheres without deformability (Discrete Element Method (DEM)). In fact, DEM aims to study powder compaction in the early stages of the process, where the mechanical behavior of the powder is mainly determined by the rearrangement of the particles and their movement (displacement, rotation, and sliding). Therefore, the results applied by the present research category are primarily used in the range of low compression pressures and powders with low deformability. Recently, the Multi-Particle Finite Element Method (MPFEM) has been introduced as a promising micromechanical-based modeling method for modeling materials composed of small deformable particles. Unlike the DEM, in the MPFEM, each particle is considered a deformable material. As a result, it will be possible to model the forming processes of particulate materials using the given method [12]. In fact, MPFEM combines two phenomenological and micromechanical approaches, in which the powder environment is considered a set of individual particles (micromechanical approach). Still, the simulation of the behavior of each of these particles is based on the yield criterion developed for continuum mediums (phenomenological approach).

Among the various phenomenological models researchers have applied to simulate the powder compaction process, the DPC model has received much more attention than the others, especially in recent years. The widespread acceptance of the DPC is because the effects of the three main characteristics of the mechanical behavior of the powder during compaction (i.e., elastic-plastic deformation, strain hardening, and inter-particle friction) are fully considered in this model [13]. That causes the results predicted by this model to be in good agreement with those obtained by experiments. The initial version of the DPC model, introduced in 1952 by Drucker and Prager, was a stress-dependent yield criterion for predicting the plastic yield of soil [14]. Although the Drucker-Prager yield criterion considered the effects of parameters like soil adhesion and friction, its application to other powder materials was accompanied by considerable errors arising from overlooking the hardening behavior of the powder during compaction. DiMaggio and Sandler modified the Drucker-Prager model to overcome this limitation, considering the powder strain-hardening effect by adding a cap yield surface instead of assuming a perfect plastic material for the powder [15]. The modified model was called DPC, and it

immediately attracted great attention. Since then, it has been used by many researchers for a wide range of powders [16–20].

The experimental methods of determining the coefficients used in the DPC material model include triaxial and uniaxial compression tests. The triaxial compaction test requires a specific die equipped with several sensors and an acquisition system for real-time data recording, making model calibration a complicated and challenging matter [21]. On the other hand, calibration of the DPC model using the uniaxial compression test requires special equipment called an instrumented die, which performs numerous experiments for different density levels and records the data measured by the sensors simultaneously. Also, the method requires performing several types of post-compaction tests on the ejected specimens, which makes its use time-consuming and expensive [22].

Due to the complexity of the experimental determination of the material parameters in the DPC model, many researchers tried to find simpler alternatives to calibrate the model. Some researchers [23–27] developed a method to determine material parameters for the DPC model by combining uniaxial compression tests, simulation, and optimization methods for metal powders. Particle Swarm Optimization (PSO) is one of the metaheuristic optimization algorithms invented by Eberhart and Kennedy in 1995 [23] based on the social behavior of fish and birds. Mathematicians and computer scientists initially noticed PSO, and after some modifications by them, it attracted the attention of researchers from various scientific fields, so that recently, it has been used in extensive and diverse areas such as medical disease detection [24], geophysics [25], estimation of the total number of confirmed COVID-19 cases [26], and computer science [27].

Researchers widely accept the PSO because the principles governing the search process for the optimal solution in this algorithm are based on the communication and simultaneous learning of the population (particles), making it simple and efficient [28]. These features have caused the PSO algorithm's limitations to be overcome and the introduction of modified versions to become an attractive topic for researchers in different fields [29]. For example, Zhang et al. [30] proposed a modified version of the PSO to enhance the algorithm's performance in multi-objective optimization problems. They replaced the global learning strategy with a dynamic neighborhood-based one to strengthen the diversity of the particles in the feasible space. Also, they used a competitive mechanism between particles to avoid getting trapped in the local optimum. In another study, Li et al. [31] proposed a multi-population cooperative Particle Swarm Optimization algorithm. They employed a dynamic segment-based mean learning strategy to construct learning exemplars and achieve information sharing and coevolution between populations. Also, they utilize a multidimensional comprehensive learning strategy to speed up convergence and improve the accuracy of the solutions. Additionally, they introduced a differential mutation operator to enhance the population diversity in the feasible space.

The multiplicity of effective parameters (e.g., compaction force, temperature, large plastic deformations, compaction speed, friction, and hardening of the material), lack of knowledge about the effect of each parameter, and the dependency of the material properties on the density are three main factors that limit analytical modeling of the powder compaction process [32]. To overcome this limitation, FEM simulation attracted the attention of researchers in the field of powder compaction as an efficient alternative for modeling the process in the early 2000s [33]. Among the material models applied for FEM simulation of the powder compaction process, the DPC model is the most widely used due to its simplicity, considering the influence of the most influential parameters and the accuracy of the predicted results, so that today it has been implemented in most commercial FEM software.

At the same time, the time-consuming and expensive tests required for experimental determination of the material coefficients in the DPC model have caused these coefficients to be provided for a limited number of commercial powders, and they are still not available for many powders [34]. However, in recent years, the application of a novel combined experimental/numerical/optimization technique is expanding that enjoys the accuracy of the

DPC material model in the FEM numerical simulation as well as the ability of optimization algorithms to find the optimal DPC coefficients. Regarding that, Hrairi et al. [14] studied the FE simulation of die compaction of metal powders using the DPC model implemented in Abaqus software 5.7, along with an inverse optimization algorithm to calibrate the coefficients. They defined the difference in the density distribution data between the prediction of the FE simulation and the experiment as the objective function for optimizing DPC coefficients. Also, the modified Levenberg–Marquardt algorithm was used to optimize the objective function. Using their proposed combined method, they succeeded in predicting the density in the compacts with a maximum absolute error of 2.3% between densities. Majzoobi and Jannesari [35] used this method to calibrate the coefficients of the DPC model for aluminum, iron, and copper powders. They used the powder force-displacement curve obtained from the compaction test as the experimental data. They also performed a numerical simulation of powder compaction using Ls-Dyna FE software 2015. Their research defined the difference between numerical and experimental force-displacement curves as the objective function. The obtained results showed that the simulation of aluminum and iron powders using this method leads to results with reasonable accuracy. However, it was found that the method is not suitable for copper powder.

Atrian et al. [36] used this method to determine the DPC coefficients of aluminum alloy powder (Al 7075). They used the uniaxial compression test as the experimental data, the FE simulation in Abaqus software 2010 as the numerical data, and the artificial neural network as the optimization method. Their research trained the artificial neural network to determine the DPC coefficients to minimize the difference between the experimental and numerical curves. The results agreed with the experimental data, showing the method's efficiency. Buljak et al. [37] used a new calibration method called inverse analysis methodology to determine the DPC coefficients of alumina powder. The results obtained from this research showed that the proposed method can evaluate the DPC coefficients more accurately than the conventional experimental methods. They concluded that the inverse analysis methodology may be advantageous from an industrial perspective since it is more robust and economical. At the same time, it provides data for a broader range of relative densities. Zhou et al. [38] proposed an integrated method of modeling using inverse optimization to simulate the compaction process of metal powders. They mentioned avoiding numerous tests and high accuracy as features of this method. In their study, the downhill simplex optimization method was used to optimize the difference between experimental and numerical data of Distaloy AE powder. The results showed that considering DPC coefficients as constant values reduces the accuracy of the simulation. So, the coefficients should be regarded as density-dependent to achieve sufficient accuracy.

An enhanced Particle Swarm Optimization (IPSO) algorithm was presented by Wu et al. [39] to improve dependability problem-solving performance. The IPSO algorithm used two location update strategies: particles followed the track of the most successful particle in later iterations. In contrast, they primarily relied on their best experiences in early iterations. After every position update, a mutation operator was also included to enhance the algorithm's exploration capabilities and keep it from getting stuck in local optima. According to experimental results, IPSO exceeded the most well-known results in recent literature in terms of convergence and stability when compared to four other PSO variations. A new framework for reliability-based design optimization (RBDO), PS2, was presented by Yang and Hsieh [40] to lower computational costs without sacrificing model flexibility. PS2 addressed RBDO problems with discrete design parameters and intricate, non-differentiable limit state functions by combining Particle Swarm Optimization (PSO), Support Vector Machine (SVM), and Subset Simulation (SS). PSO managed the discrete optimization process, SVM categorized potential solutions for reliability evaluation, and SS effectively estimated minor failure probabilities. SVM and PSO worked together to improve PSO's efficiency by assessing the solution's viability, and PSO-assisted SVM in retraining for increased classification accuracy. The ideal design of a ten-bar truss with reliability restrictions for yield and buckling stresses was used to test this structure.

Perez and Behdinan [41] discussed analyzing, using, and enhancing a Particle Swarm Optimization method that works well for constraint optimization problems. They investigated the impact of the various PSO configuration settings in the context of traditional structural optimization issues and demonstrated the algorithm's operation and efficacy. Li et al. [42] suggested an integrated multi-objective cultural-based particle swarm method to tackle the double-loop reliability-based design optimization. In the inner optimization loop, the crowding distance ranking was introduced to update the global and local optimum and control the maximum number of solutions in elitism knowledge. The hybrid mean value approach was enhanced to conduct reliability analysis in the outer loop to accommodate both concave and convex performance function types. According to the results, the reliability-based design optimization issues may be effectively and practically solved by the suggested cultural-based multi-objective particle swarm optimizer.

A flexible computational method frequently used in performance reliability design and structural model optimization is PSO. It reduces structural performance-related objective functions like weight and cost while adhering to design limitations like stress limits. PSO enables creative material distributions and layouts in topology optimization, enhancing dynamic responses for improved seismic performance. Furthermore, PSO facilitates reliability analysis by estimating the probability of structural failure under various circumstances and uncertainties, making robust designs less susceptible to changes in material properties. PSO benefits engineers looking to improve structural performance while skillfully managing limits and uncertainties because of its versatility, global search capabilities, and simplicity [43–45].

The application of Optimization optimization algorithms to the determination of material coefficients has emerged as an essential topic of study in computer science. Optimization problems are increasingly seen as urgently needed solutions in various domains, including engineering, mathematics, computer science, and economics. Researchers can identify the comparatively best option from several legitimate solutions by employing optimization algorithms, which is critical for modeling the behavior of materials in industrial processes. As a result, optimization algorithms are becoming increasingly crucial in advancing materials science and engineering. Our research aims to provide a framework based on the Particle Swarm Optimization (PSO) algorithm and use it to determine the Drucker-Prager-Cap (DPC) material model coefficients for modeling the industrial powder compaction process. This work requires providing experimental data for training the optimization algorithm, finite element modeling of the process in commercial software (Abaqus 2020), coding the PSO in Python, establishing a connection between the algorithm and the finite element software (Abaqus 2020), and validating the results. The following section explains the general concept of each of the above items.

2. Methodology for Calibrating the DPC Model Parameters

This research's primary goal is to calibrate the coefficients of the DPC material model. Because in the DPC material model, most of the characteristics of powder particles, such as adhesion, displacement, deformation, friction, hardening, and volumetric plastic strain, are taken into account, DPC is currently the most efficient model in the FE simulation of the powder compaction process in industrial applications. The method used in the present study is a combination of experimental measurement, FEM simulation, and inverse optimization, which has recently been proposed as a new alternative that has received much attention due to the avoidance of the experimental determination of coefficients of material models. This section provides a detailed, comprehensive explanation of the method.

2.1. Experimental Setup

In this research, we are trying to determine the coefficients of the DPC model using the proposed method for a mixed metal powder. Based on the powder metallurgy route, the powder has been selected according to its importance and application in producing

industrial parts. For this purpose, Ag57.6-Cu22.4-Sn10-In10 (ACSI) has been chosen as the mixed metal powder.

Ag-Cu alloys have been widely used to join metallic components in various industries. Usually, mineral elements such as Sn and In are mixed into the Ag-Cu base to reduce consolidation temperature and melting point and improve flowability. The mass percentage of additives has a significant influence on the properties of the final alloy. Adding too much of the minerals can cause chemical reactions with the Ag-Cu base, reduce ductility, and increase the brittleness of the alloy. Among the Ag-Cu-based alloys with mineral additives, ACSI has the most widespread industrial applications in the automobile, household appliances, and electronics industries. Its favorable performance, high conductivity, corrosion resistance, and superior mechanical strength are the reasons for its effectiveness.

The force-displacement curve of ASCI powder has been measured experimentally by Zhou et al. [2] and is used in the present research. It is crucial to remember that the authors did not do any experiments in this work; instead, the emphasis is on evaluating and expanding on the results from the published work by Zhou et al. [2]. They measured this curve from the uniaxial compression test into an instrumented cylindrical die with an external diameter of 20 mm, an internal diameter of 10 mm, and a height of 50 mm. To ensure the elastic behavior of the setup during powder compaction, the die and punches are made of 60Si2Mn steel, with E = 206 GPa, v = 0.3, and yield strength of σ_Y = 1180 MPa.

Figure 1 illustrates the instrumented die compaction setup used in the uniaxial compaction test and the force-displacement response of the ASCI powder. As one can see, the compaction curve of a powder can be easily measured by performing a simple uniaxial compression test. At the same time, experimental determination of DPC material constants requires a large number of compression tests for different densities, so it is not time- and cost-effective. In the experimental part of the proposed method of this research, we only need to perform a uniaxial compression test to measure the compaction curve of the powder, which is easy and cheap.

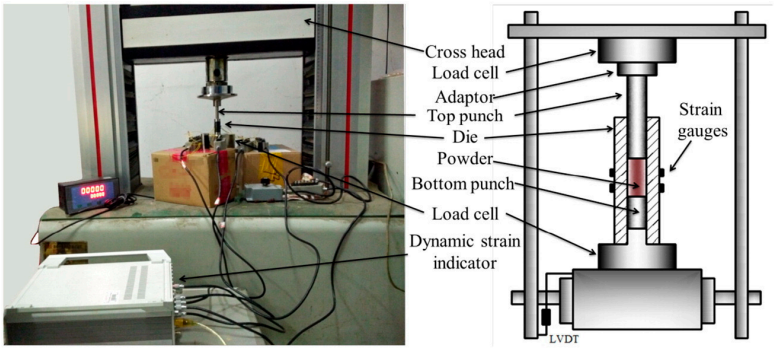


Figure 1. Experimental measurement of the force-displacement curve for ASCI powder; instrumented die under uniaxial compaction test [2] (This figure is not unique to the present study; it is an adaptation of the work of [2]).

2.2. FEM Simulation of the Powder Compaction Process

With the identification of the capabilities of the FEM for simulating problems that cannot be solved using conventional analytical methods, the FEM has gained a special place in scientific and industrial centers as a powerful modeling tool. The FEM is taught as a prerequisite in many engineering courses, and its mastery is considered one of the most critical skills for engineers. Today, the FEM has been implemented in many commercial software in which users can simulate their desired problem in different scientific fields. Commercial FE software package Abaqus 2020 is a powerful yet simple method for investigating the structural modeling of powder compaction processes [46–48]. Accessibility and ease of use make Abaqus 2020 a suitable software for our purpose. As explained earlier, the FE simulation of a process has different steps. In the following sections, the FE simulation performed by Zhou et al. [2] will be repeated, and the compaction curve

obtained from the developed FE model will be compared with the experimental data to check the validity of the FE modeling in the present study. In other words, to present and employ a validated numerical model, we use Abaqus software 2020 to directly recreate the FE model proposed by Zhou et al. [2]. In the framework of our investigation, this method is crucial to obtaining consistent data required for comparison and additional analysis. No changes were made to the original simulation parameters to maintain the authenticity of the published technique and concentrate on the relevance of these results in our current investigation. In this regard, we quantify the size and cross-sectional shape of the die, corresponding powder particles, and initial density; also, the weight of the punch and its speed can be ignored as the compaction process is assumed to be quasi-static. After that, the quantified features are transferred to a customized Python-based modeling framework.

The FE model of the powder compaction process usually consists of three parts: the die, punch, and powder. A part can be modeled as deformable or discrete rigid. Also, the axisymmetric model means that the model has symmetry in terms of geometry and loading; as a result, the model can be sketched in 2D to a symmetry line. Table 1 lists the type of modeling of the three parts and their dimensions. The friction between the surfaces is also defined using the penalty formulation [46–48] with a friction coefficient 0.8.

Table 1. Geometric modeling of the punch, die, and powder in part module.

Part	Modeling Space	Type	ASCI Powder [2]	
			Diameter [mm]	Height [mm]
Punch	Axisymmetric	Discrete Rigid	10	16.02
Die	Axisymmetric	Discrete Rigid	10	17
Powder	Axisymmetric	Deformable	10	15.82

It should be noted that in the powder compaction model, the loading is applied through the vertical movement of the punch on the powder, and this movement is equal to 50 mm for ASCI powder. Figure 2 and Table 2 show the meshed model and the meshing details for two types of powders. Abaqus has a library of different elements from which, in the analysis of FE problems, the user must choose the appropriate kind of element according to the nature of the problem. As can be seen, CAX4R element is used for the powder, and the RAX2 element is used for both the die and punch.

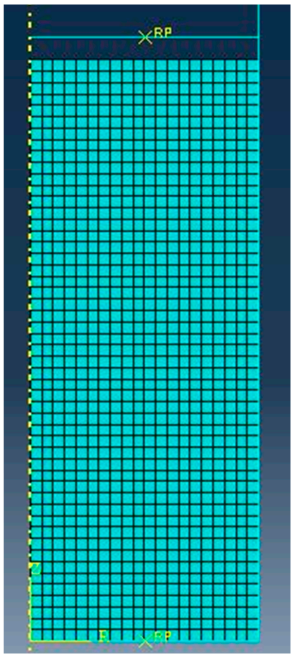


Figure 2. The meshing of the model in the mesh module.

Table 2. Characteristics of the selected elements in the mesh module.

Elements Properties	Powder	Die	Punch
Element Library	Standard	Standard	Standard
Element Family	Discrete Rigid	Discrete Rigid	Discrete Rigid
Geometric Order	Linear	Linear	Linear
Element Type	CAX4R	CAX4R	CAX4R
Element Number	1020	80	20

Formulation of the DPC Material Model in Abaqus

After meshing the model in the property module, we must select a material model that will control the mechanical behavior of the material during the process. Each material model has a set of constants called "material properties" in Abaqus. So, we must specify the model's material properties in the property module. However, each material has many properties, and according to the type of problem we are facing, we must only provide the required material properties for the selected model. For example, suppose we intend to model a situation of stress analysis (which is related to the field of solid mechanics). In that case, we must enter material properties such as Young's modulus and Poisson's coefficient, and there is no need to enter the coefficient of thermal conductivity.

As previously discussed, DPC is the most widely used material model in modeling the powder compaction process-based FEM. The material properties of the DPC model are different in each FE software according to its formulation. Here, the formulation of the DPC model implemented in Abaqus and the related material properties are explained.

The DPC material model includes three parts of the yield surfaces, according to Figure 3. The shear failure line (F_s) indicates the failure of the material under the shear stress, the cap surface (F_c) provides an inelastic hardening mechanism to represent plastic compaction and controls volume dilatancy when the material yields shearing, and the transition surface (F_t) provides a smooth transition between two previous surfaces (see Figure 3). Equations (1) to (3) show the formulation of the three yield surfaces in the DPC model implemented in Abaqus:

$$F_s = q - p\,tan\beta - d = 0 \tag{1}$$

$$F_c = \sqrt{(p - p_a)^2 + \left[\frac{R\,q}{1 + \alpha - \alpha/cos\beta}\right]^2} = 0 \tag{2}$$

$$F_t = \sqrt{(p - p_a)^2 + \left[q - \left(1 - \frac{\alpha}{cos\beta}\right)(d + p_a\,tan\beta)\right]^2} - \alpha(d + p_a\,tan\beta) = 0 \tag{3}$$

where q and p are the Mises equivalent stress and the hydrostatic pressure stress, respectively. Young's modulus (E) and Poisson's ratio (v) are two material properties that determine the elastic behavior of powder during the unloading stage (i.e., removing the punch pressure from the powder). Material cohesion (d) and angle of friction (β) are two material properties that define the shear failure line. To determine the cap surface, four parameters, including cap eccentricity (R), initial yield surface position (ε_0), the transition surface radian (α), and the flow stress ratio (K), must be determined. Also, p_a is an evolution parameter representing the volumetric plastic strain-driven hardening/softening and is given as [2]:

$$p_a = \frac{p_b - Rd}{(1 + R\,tan\beta)} \tag{4}$$

where p_b is the hydrostatic pressure yield stress that defines the position of the cap and is generally expressed as a function of volumetric plastic strain ε_v^{pl} as [2]:

$$p_b = f\left(\varepsilon_v^{pl}\right) \tag{5}$$

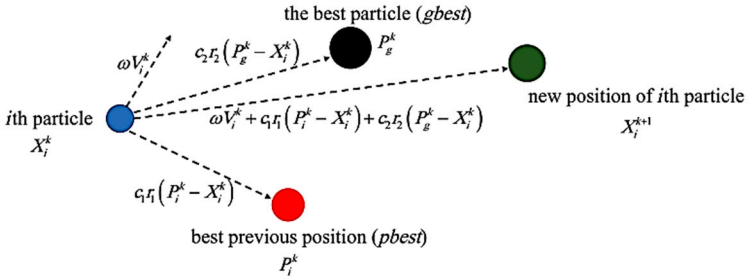


Figure 3. The schematics of updating the position of the ith particle in the kth iteration in the PSO optimization [49].

One can see the FEM simulation of the powder compaction process using the DPC material model in Abaqus software 2020 requires the determination of ten material properties (E, v, d, β, R, ε_0, α, K, p_b, ε_v^{pl}). As explained earlier, the experimental data provided in [2] were used as the material properties for the FE simulation of the compaction of ASCI powder using DPC.

2.3. Scripting PSO Optimization Algorithm in Python

PSO is one of the most essential intelligent optimization algorithms widely used in various scientific fields. Despite its simplicity, PSO can find the optimal solution. PSO is sometimes classified as an evolutionary algorithm because the modifier mechanism repeats itself and introduces a new population based on the information-sharing process [49]. In another classification, PSO is considered one of the "swarm intelligence" optimization algorithms. These algorithms search for the optimal solution with the collective cooperation of members and use a mechanism called "self-organization", which controls the individual and social search process in each iteration [49]. Wherever the information flow and self-organization exist, swarm intelligence will emerge to provide optimal conditions for the group. PSO was inspired by the behavior of a group of fish when they faced the threat of a hunter [49]. Through their swarm intelligence, they are divided into several groups (swarms) when a hunter attacks and then gather together again after the danger has vanished. In this algorithm, the two factors of information exchange (the swarm becoming aware of the hunter's position) and self-organization (the rules that determine the direction and velocity of the swarms' movement) play the leading role [49].

In the PSO algorithm, each member of the swarm is called a "particle", and the position of each particle in the feasible space is considered one of the potential solutions to the optimization problem. Particles start moving in the possible space to search for the optimal solution, so each particle takes advantage of previous experiences of itself and the swarm. This collective search process ends when the condition set for the minimization of the objective function is satisfied. As can be seen in Figure 3, in each iteration, the previous position of the particle is updated according to the following equations [49]:

$$V_i^{k+1} = \left[\omega V_i^k\right] + \left[c_1 r_1\left(P_i^k - X_i^k\right)\right] + \left[c_2 r_2\left(P_g^k - X_i^k\right)\right] \quad (6)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (7)$$

where subscript k denotes the number of the current iteration, V_i^k and X_i^k are the current position and the velocity of the ith particle, P_i^k and P_g^k are the best previous position and of the ith particle (called personal best) and the best global position of the swarm at iteration k (called global best), r_1 and r_2 are random numbers distributed uniformly in [0, 1], c_1 and c_2 are weights of personal best and global best, and ω is inertia weight (a positive constant that controls the weight of the previous velocity on the current one).

The first term in Equation (6) (ωV_i^k) searches for new solutions and finds the regions with potentially the best solutions. The parameter ω is essential for balancing the global search. It makes the particle move in the same direction and with the same velocity. When

higher values are set for ω, it is known as exploration, while when the lower values are set, it is known as exploitation [49]. The second and third terms explore the previous solutions and find the best solution for a given region. The second term ($c_1 r_1 \left(P_i^k - X_i^k \right)$ represents the effect of the personal experience of each particle. So, it makes the next position of the particle better than the current. The third term ($c_2 r_2 \left(P_g^k - X_i^k \right)$)) represents the effect of neighbors' social experiences and causes the particle to follow the best neighbors' directions.

Here, we propose a customized Python-based framework to develop an efficient and robust technique for calibrating the parameters of the DPC material model and simulating the compaction process of any desired powder. To accomplish the objective, we only need the powder's force-displacement curve, which is easily obtained through a routine uniaxial die compression test. Then, we use this experimental data and the corresponding force-displacement curve obtained from the FE simulation as the inputs in the PSO optimization algorithm (see Figure 4) scripted in Python.

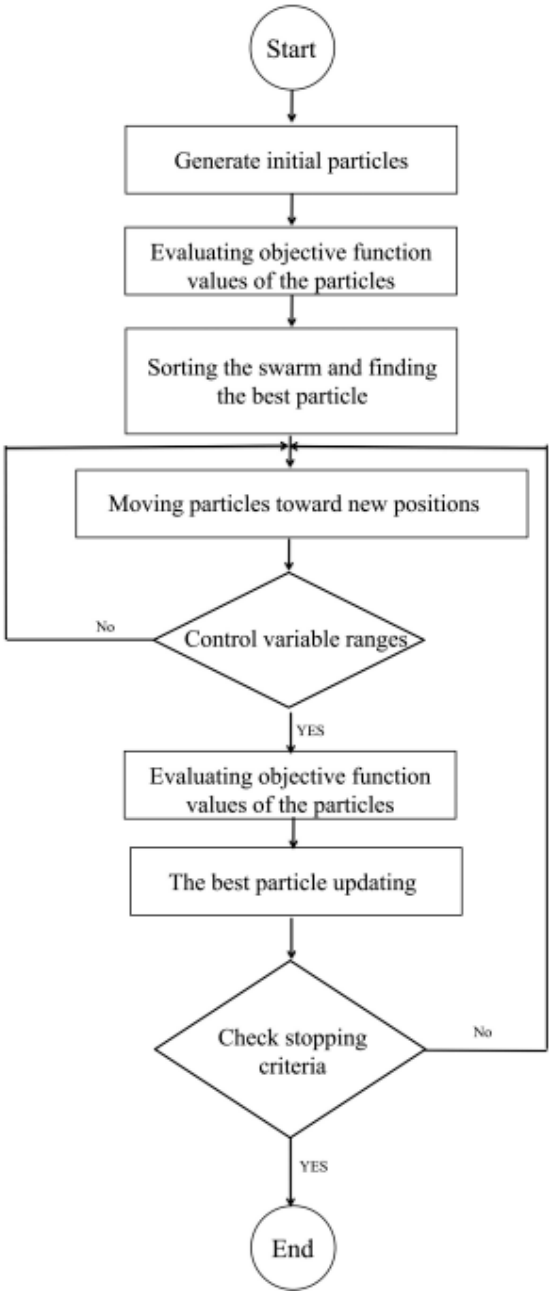


Figure 4. The flowchart of the searching process in the PSO optimization [49].

In the method employed in the current research, the initial population is the DPC model parameters. To avoid the complexity of the optimization problem, we should only consider the parameters that significantly affect the powder's mechanical behavior during the compaction. To evaluate the effect of DPC model parameters on powder behavior, a few studies have been carried out based on the FE simulation of metal, pharmaceutical, and ceramic powders [2]. In these studies, through a process called "sensitivity analysis", the values of a parameter are changed in each simulation run to evaluate its effect on the powder response.

By performing a sensitivity analysis on the DPC parameters, Majzoobi and Jannesari [35] reported that Young's modulus ($0.01 < E < 120$), material cohesion ($0.01 < d < 50$), and hydrostatic pressure yield stress ($0.1 < P_b < 250$) are strongly dependent on the density. Therefore, all three parameters are fitted by an exponential function, as suggested by Zhou et al. [2]. Sensitivity analysis also indicated that Poisson's ratio ($0.01 < v < 0.3$), angle of friction ($69 < \beta < 73$), and cap eccentricity ($0.1 < R < 0.8$) did not have much effect on the force-displacement response of the powder. Therefore, the experimental values reported by Zhou et al. [2] are considered for these parameters. Additionally, the initial yield surface position ($\varepsilon_0 = 0$), the transition surface radian ($\alpha = 0.02$), and the flow stress ratio ($K = 1$) are set to their default values in Abaqus because the sensitivity analysis determined that these parameters do not have much effect on the response of powder particles during the compaction [2].

Table 3 lists DPC model parameters as optimization variables and their fitted relations. As you can see, we need nine unknown coefficients whose optimal values should be determined to calibrate the DPC model. Now that the variables of the optimization problem are defined, an initial population of them must be created in the feasible space to start the search process. The initial position of the generated particles is determined through the values assigned to the decision parameters. These values should be assigned in such a way as to make sure that the generated particles cover the entire feasible space both in terms of number and position. In this research, the values of the decision variables have been determined based on this significant point. In the following, we will explain the calibration process of the DPC model parameters using PSO optimization algorithm step by step in detail:

Table 3. Defining the DPC model parameters as the PSO optimization variables.

Parameter	Value	Optimization Variables and Constraints
E [GPa]	$E = E_1 \times exp(E_2 \times \rho)$	$0 < E_1 < 1000$ and $0 < E_2 < 1000$
v	$v = 0.031 \times exp(1.73 \times \rho)$	-
d [MPa]	$d = d_1 \times exp(d_2 \times \rho)$	$0 < d_1 < 1000$ and $0 < d_2 < 1000$
β	$\beta = 71.3$	-
R	$v = 0.281 \times exp(0.64 \times \rho)$	-
ε_0	0	-
α	0.01	-
K	1	-
P_b [MPa]	$P_b = P_1 \times exp(P_2 \times \rho)$	$0 < P_1 < 1000$ and $0 < P_2 < 1000$

- Step 1: We start the optimization process by importing the required modules to the code.
- Step 2: We acquire and assign the current directory to the class's directory.
- Step 3: We import three classes, each of which is called for a specific purpose during the code's execution. For example, the updater class updates DPC parameters in each iteration.
- Step 4: We create a folder named "Results" to save the outputs and check its existence.
- Step 5: In this part of the calibration process, first, a specified matrix (including the optimization variables E_1, E_2, d_1, d_2, P_1, P_2 that the PSO must update) and the experimental data of the force-displacement curve are defined as the global variables.

Then, the function f(x) is determined based on the global variable matrix, and its components are specified.

- Step 6: As explained earlier, one of Abaqus's capabilities in the job module is to write an input file from the generated FE model. The input file is a text file in which all the details of the FE model are written so that the model can be created and solved directly without additional operations by importing it into Abaqus. Here, we use this feature of the input file for our purpose so that Python reads the input file line by line and writes a new input file based on it while the desired changes can be applied wherever necessary. For this purpose, the input file named "Validation" is first called. Then, it is opened, and its content is read line by line.

To manipulate the optimization variables (E, d, P_b), the process of reading the contents of the input file continues until we reach the line that creates the DPC material parameters. As soon as this section, which is titled by a specific header in the input file, is reached, Python generates new coefficients using the PSO algorithm and writes them in the new input file. For example, in the case of the cap hardening parameter (P_b), when the code reaches the line titled "*Cap Hardening*", the first and last lines related to P_b are specified. Then, its values are updated from the "*Inp_Updater*" function considering coefficients P_1 and P_2. Such a process is also used to update the values of Young's modulus (E) and material cohesion (d) parameters. At the final stage of the process, we update the elastic parameters (E, v), the cap plasticity parameters (d, β, R, α, ε_0, K), and the cap hardening parameter (P_b). Now we write a new input file named "Trial". The input file contains the updated values of the optimization variables (E, d, P_b). Here, Python links Abaqus to run the new input file, and two seconds are considered for the CPU delay.

- Step 7: When Abaqus completes running the updated input file in ith iteration, it generates an output file called "*odb*" file containing the obtained results. First, a copy of the odb file is made, and the file's content is read line by line. Then, the code saves the desired outputs (e.g., the force and displacement of the punch during the compaction) into an odb file named "Trial_i".
- Step 8: Failure to generate the output file means an error occurred during the running of the model by *Abaqus*. So, here, we have to check whether an error occurred. With this objective, we consider a specific time. If the output file is not generated after this time, it means there is an error in the model. In that case, the code neglects the current iteration and goes to the next one.
- Step 9: In this part of the code, the data in the "Trial_i" odb file is read and saved in a CSV file. Then, the file is opened, and its content, including the time intervals and their corresponding forces and displacements, is read.
- Step 10: Here, we call the force-displacement curve extracted from the punch. In that phase, these values are compared with the corresponding experimental values at specific points, and the objective function is defined as follows:

$$OBJ = \sum_{i=1}^{N} \left(y_i^{opt} - y_i^{exp} \right)^2 \tag{8}$$

where y_i^{opt} and y_i^{exp} are the corresponding optimized and experimental values, respectively, and N is the number of data points. By defining the objective function, the error value of each iteration can be considered as follows:

$$error = \sqrt{\frac{OBJ}{N}} \tag{9}$$

- Step 11: First, an initial population of particles is randomly generated. Then, each particle is randomly assigned a velocity and a position to determine the secondary position of the particles according to Equation (6). Then, the objective function is calculated for each particle to determine the values of the personal and global best for

the particles. Next, suppose the particle's new position is improved compared to its previous position; its velocity and position are updated according to Equations (6) and (7), respectively, and simultaneously. In that case, they are compared with the range of the feasible space.

- Step 12: At the end of the optimization code, the best position and the best error for the group of particles are determined. Based on them, the optimization loop produces particles with a new position.

3. Results and Discussion

3.1. Validation Analysis of Proposed FE Model

After preparing the FE model of the ASCI powder compaction process according to the steps explained in Section 2, we compare the results of our proposed FE model with the experimental data reported by Zhou et al. [2] to ensure the model's accuracy. Figure 5 compares FE and experimental force-displacement curves. As can be seen, the curve obtained from the FE simulation has accurately predicted the experimental data, which validates the provided FE model of ASCI powder.

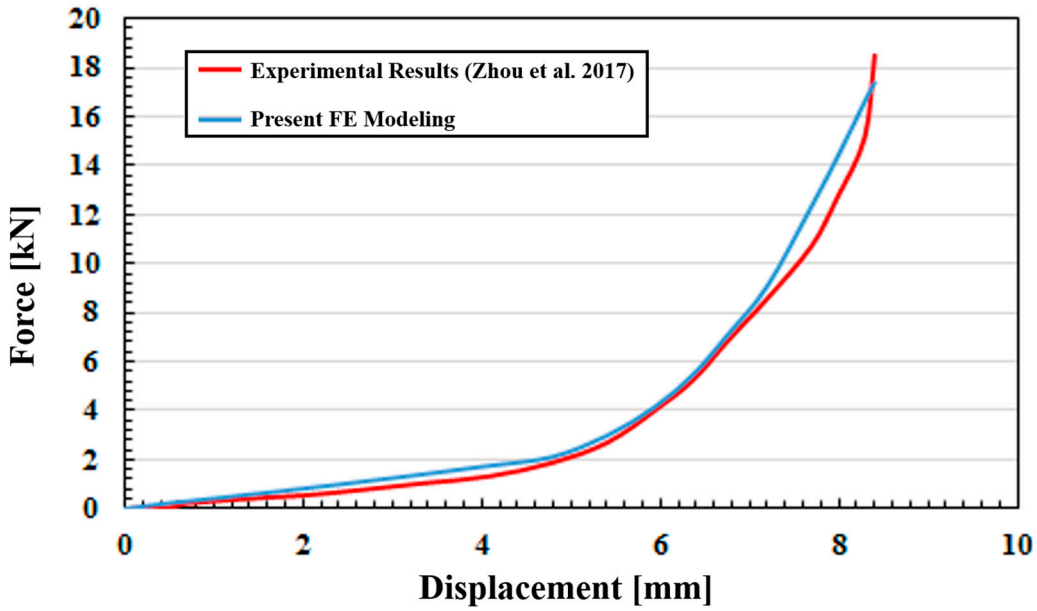


Figure 5. The comparison analysis between the results of the present FEM model and experimental results [2] as validation analysis.

3.2. Inverse Optimization Analysis

Here, the results of the calibration of DPC model parameters by the inverse optimization method are compared with the experimental results. These parameters (i.e., Young's modulus (E), material cohesion (d), and the hydrostatic pressure yields stress (p_b)) are the same optimization variables introduced in the previous section.

Table 4 presents the optimized coefficients. By placing these coefficients in Equations (10) to (12), it is possible to plot the curves of E, d, and p_b, respectively.

$$E = E_1 \times \exp(E_2 \times \rho) \tag{10}$$

$$d = d_1 \times \exp(d_2 \times \rho) \tag{11}$$

$$P_b = P_1 \times \exp(P_2 \times \rho) \tag{12}$$

Table 4. The optimized values of the coefficients.

Coefficient	E_1	E_2	d_1	d_2	P_1	P_2
Optimized value	16.71	8.78	0.91×10^{-4}	13.08	0.53	7.51

On the other hand, the results listed in Table 4 prove that the method applied in the research at hand has succeeded in providing specific values as optimal coefficients. In the next part, the curve of each parameter is plotted in terms of density and compared with the experimental results to evaluate the accuracy of the proposed calibration method. It should be noted that to determine the error of the values predicted by the optimization method, the Root-Mean-Square Error (RMSE) is calculated according to Equation (9).

Figure 6 compares the variations of Young's modulus (E) according to the values obtained from the optimization with those obtained by the experimental method. The optimization curve is plotted by placing the optimized coefficients (E_1 and E_2) in Equation (10). As can be seen, the curve resulting from placing the optimal coefficients concerning 10 has been able to predict the experimental curve with acceptable accuracy (RMSE = 1.95). As previously explained, the parameter E of the DPC model affects the powder's mechanical behavior during unloading, which will be examined at the end of this section after the other parameters are applied.

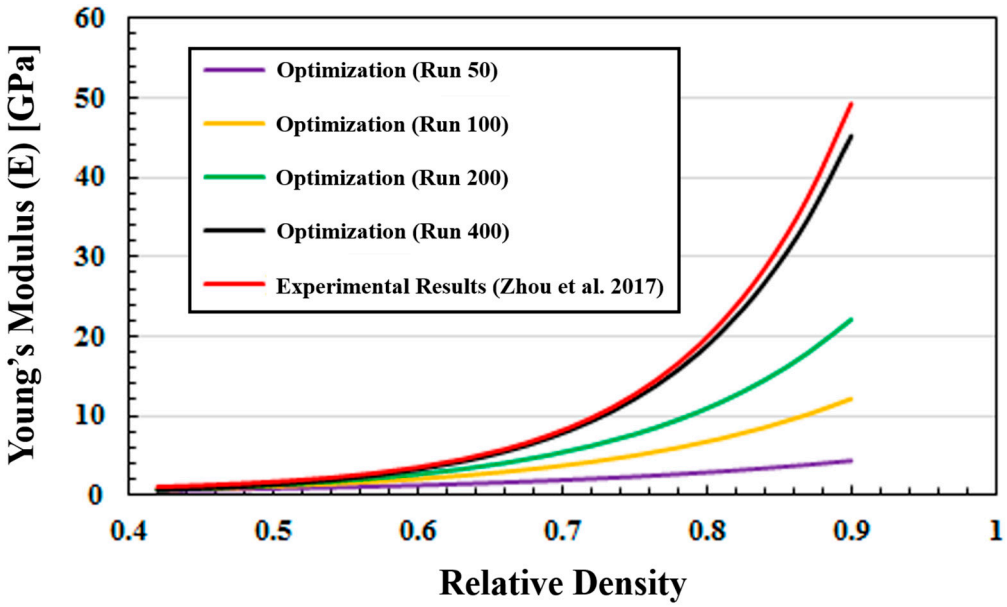


Figure 6. Comparison between the optimization and experimental results [2] for Young's modulus (E).

Figure 7 shows the variations of the optimized values of the material cohesion parameter (d) compared to the experimental values. To plot the optimization curve, the optimized coefficients d_1 and d_2 are placed in Equation (11). It can be seen again that the calibration of this parameter using the inverse optimization method has succeeded in predicting the experimental values with acceptable accuracy (RMSE = 0.12). As discussed earlier, material cohesion is a parameter that determines the q-axis intercept of the shear yield line in the DPC model. As a result, it affects the mechanical behavior of the powder during compaction. The effect of this parameter is also evaluated at the end of the section.

Figure 8 compares the variations of the hydrostatic pressure yield stress (P_b) in terms of the volumetric plastic strain for the optimized values and those obtained experimentally. The optimization curve is obtained by placing the optimal coefficients (P_1 and P_2) in Equation (12). In a similar process with parameters E and d, the optimization of P_b was also successful so that PSO has been able to predict parameter P_b with appropriate accuracy (RMSE = 324.64). P_b is the most crucial parameter of the DPC model, which shows the

powder's hardening behavior during the compaction process. The parameter's impact on the powder's behaviour is evaluated at the section's conclusion.

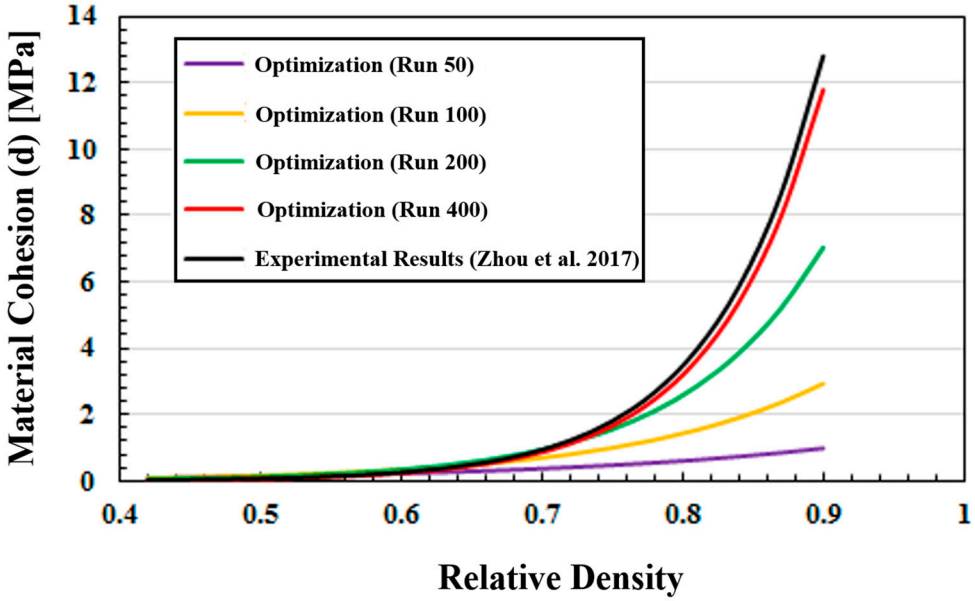


Figure 7. Comparison between the optimization and experimental results [2] for the material cohesion (d).

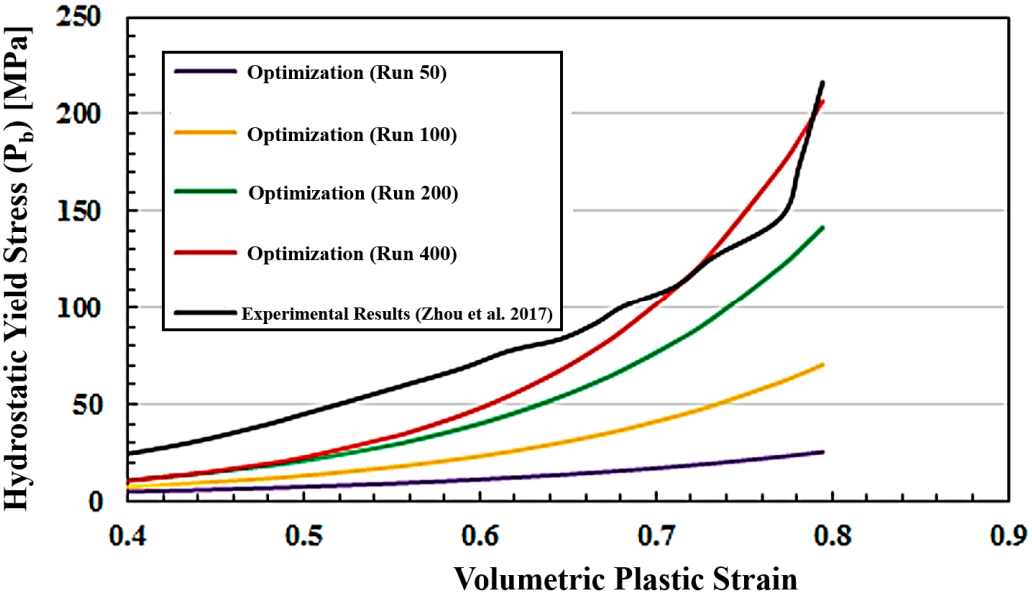


Figure 8. Comparison between the optimization and experimental results [2] for hydrostatic pressure yield stress (P_b).

According to the sensitivity analysis, in the case of parameters that do not affect the mechanical behavior of the powder, their default values or the values reported in other references have been used. Finally, Table 5 presents the final values of the DPC model parameters for ASCI powder. It should be noted that in Table 5, E, d, p_b parameters are calibrated using the method applied in the present research.

Table 5. DPC model parameters of ASCI powder used in FE simulation.

E (GPa)	v	d (MPa)	β	R	ε_0	α	K	ρ	p_b (MPa)	ε_v^{pl}
0.67	0.068	0.02	71.3	0.381	0	0.02	1	0.42	0.53	0.00
0.87	0.070	0.03	71.3	0.385	0	0.02	1	0.45	7.39	0.35
1.13	0.073	0.05	71.3	0.389	0	0.02	1	0.48	12.42	0.42
1.47	0.076	0.07	71.3	0.393	0	0.02	1	0.51	18.20	0.47
1.91	0.080	0.11	71.3	0.398	0	0.02	1	0.54	23.28	0.50
2.49	0.083	0.16	71.3	0.403	0	0.02	1	0.57	34.12	0.55
3.24	0.087	0.23	71.3	0.409	0	0.02	1	0.6	35.06	0.56
4.22	0.092	0.34	71.3	0.416	0	0.02	1	0.63	44.22	0.59
5.49	0.096	0.51	71.3	0.424	0	0.02	1	0.66	54.27	0.62
7.15	0.102	0.76	71.3	0.432	0	0.02	1	0.69	67.52	0.65
9.30	0.107	1.12	71.3	0.441	0	0.02	1	0.72	78.47	0.67
12.10	0.114	1.66	71.3	0.452	0	0.02	1	0.75	88.73	0.68
15.75	0.121	2.45	71.3	0.463	0	0.02	1	0.78	110.39	0.71
20.49	0.128	3.63	71.3	0.476	0	0.02	1	0.81	130.05	0.73
26.67	0.136	5.38	71.3	0.490	0	0.02	1	0.84	173.23	0.77
34.70	0.145	7.96	71.3	0.506	0	0.02	1	0.87	188.02	0.78
45.16	0.155	11.79	71.3	0.524	0	0.02	1	0.9	206.88	0.79

4. Conclusions

According to the novel method employed in this research for modeling the powder compaction process, the following conclusions can be drawn:

- The simulation of the compaction process for ASCI powder was successfully carried out using Abaqus^{TM}. Then, the difference between the force-displacement curve obtained from the simulation and the experimental data was defined as the objective function of an optimization problem.
- The PSO optimization algorithm was coded in Python, and the link between the optimization code and Abaqus was successfully established.
- The DPC model parameters are considered the optimization variables in each iteration; these variables are generated by the PSO and are considered updated material properties in Abaqus. Then, the software solves the FE model, and the error is calculated. According to the error value, PSO approaches the optimal solution based on previous experiences of individual particles.
- The results showed that the proposed method in this research has been very successful in calibrating the DPC model so that three parameters of Young's modulus, material cohesion, and hydrostatic pressure yield stress are obtained, respectively, with RMSE 1.95, 0.12, and 324.64 compared to their experimental values.

Finally, computer science techniques played a vital role in the present work's success. The utilization of Abaqus and the PSO optimization algorithm coded in Python allowed for the efficient calibration of the DPC model parameters, ultimately improving the accuracy of the powder compaction process simulation. The present study's findings can serve as a foundation for further research and advancements in the field of the applications of optimization methods in powder compaction. The present work has several shortcomings, such as the Drucker-Prager-Cap model's inability to fully capture the intricacies of powder compaction and the Particle Swarm Optimization technique's processing requirements, which may restrict its practical usefulness. Additionally, even though our method was verified on the particular alloy Ag57.6-Cu22.4-Sn10-In10, more research is necessary to see whether it can be applied to other materials. Future research opportunities include combining machine learning with conventional numerical techniques to improve parameter calibration, more experimental validations on different powders to support our conclusions, and investigating real-world industrial applications of our approach to gauge its efficacy.

Author Contributions: Conceptualization, S.D. and M.A.; methodology, S.D.; software, S.D.; validation, S.D., M.A. and S.S.M.A.; formal analysis, S.D.; investigation, S.D.; resources, M.A.; data curation, S.S.M.A.; writing—original draft preparation, S.D. and S.S.M.A.; writing—review and editing, S.D. and S.S.M.A.; visualization, S.D.; supervision, M.A.; project administration, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hartmann, S.; Bier, W. High-order time integration applied to metal powder plasticity. *Int. J. Plast.* **2008**, *24*, 17–54. [CrossRef]
2. Zhou, M.; Huang, S.; Hu, J.; Lei, Y.; Xiao, Y.; Li, B.; Yan, S.; Zou, F. A density-dependent modified Drucker-Prager Cap model for die compaction of Ag57.6-Cu22.4-Sn10-In10 mixed metal powders. *Powder Technol.* **2017**, *305*, 183–196. [CrossRef]
3. Kim, K.; Cho, J. A densification model for mixed metal powder under cold compaction. *Int. J. Mech. Sci.* **2001**, *43*, 2929–2946. [CrossRef]
4. Nwose, E.N.; Pei, C.; Wu, C.-Y. Modelling die filling with charged particles using DEM/CFD. *Particuology* **2012**, *10*, 229–235. [CrossRef]
5. Denny, P. Compaction equations: A comparison of the Heckel and Kawakita equations. *Powder Technol.* **2002**, *127*, 162–172. [CrossRef]
6. Desai, P.M.; Liew, C.V.; Heng, P.W.S. Review of disintegrants and the disintegration phenomena. *J. Pharm. Sci.* **2016**, *105*, 2545–2555. [CrossRef]
7. Dobrzański, L.A.; Dobrzański, L.B.; Dobrzańska-Danikiewicz, A.D. Overview of conventional technologies using the powders of metals, their alloys and ceramics in Industry 4.0 stage. *J. Achiev. Mater. Manuf. Eng.* **2020**, *98*, 56–85. [CrossRef]
8. Babaei, H.; Mostofi, T.M.; Alitavoli, M.; Namazi, N.; Rahmanpoor, A. Dynamic compaction of cold die Aluminum powders. *Géoméch. Eng.* **2016**, *10*, 109–124. [CrossRef]
9. Roveri, M.; Goidanich, S.; Dotelli, G.; Toniolo, L. Semi-empirical models to describe the absorption of liquid water in natural stones employed in built heritage before and after the application of water repellent treatments. *Constr. Build. Mater.* **2020**, *241*, 117918. [CrossRef]
10. Zhou, J.; Zhu, C.; Zhang, W.; Ai, W.; Zhang, X.; Liu, K. Experimental and 3D MPFEM simulation study on the green density of Ti–6Al–4V powder compact during uniaxial high velocity compaction. *J. Alloys Compd.* **2019**, *817*, 153226. [CrossRef]
11. Krok, A.; Wu, C.Y. Finite element modeling of powder compaction. In *Engineering Crystallography: From Molecule to Crystal to Functional Form*; Springer: Dordrecht, The Netherlands, 2017; pp. 451–462.
12. Drucker, D.C.; Prager, W. Soil mechanics and plastic analysis or limit design. *Q. Appl. Math.* **1952**, *10*, 157–165. [CrossRef]
13. DiMaggio, F.L.; Sandler, I.S. Material model for granular soils. *J. Eng. Mech. Div.* **1971**, *97*, 935–950. [CrossRef]
14. Hrairi, M.; Chtourou, H.; Gakwaya, A.; Guillot, M. Modeling the powder compaction process using the finite element method and inverse optimization. *Int. J. Adv. Manuf. Technol.* **2011**, *56*, 631–647. [CrossRef]
15. Berdychowski, M.; Górecki, J.; Biszczanik, A.; Wałęsa, K. Numerical simulation of dry ice compaction process: Com-parison of drucker-prager/cap and cam clay models with experimental results. *Materials* **2022**, *15*, 5771. [CrossRef] [PubMed]
16. Zhou, R.; Zhang, L.-H.; He, B.-Y.; Liu, Y.-H. Numerical simulation of residual stress field in green power metallurgy compacts by modified Drucker–Prager Cap model. *Trans. Nonferrous Met. Soc. China* **2013**, *23*, 2374–2382. [CrossRef]
17. Wagle, G.S.; Engel, R.S.; Bollina, R.; German, R.M. Statistical analysis of modified drucker-prager cap model parameters for application to modeling die compaction. *Adv. Powder Metall. Part. Mater.* **2003**, 4–24. Available online: https://www.cavs.msstate.edu/publications/docs/2003/07/2003-27.pdf (accessed on 29 October 2024).
18. Li, R.; Liu, W.; Li, J.; Liu, J. Inverse Identification of Drucker–Prager Cap Model for Ti-6Al-4V Powder Compaction Considering the Shear Stress State. *Metals* **2023**, *13*, 1837. [CrossRef]
19. Jung, A.; Diebels, S. Yield surfaces for solid foams: A review on experimental characterization and modeling. *GAMM-Mitteilungen* **2018**, *41*, e201800002. [CrossRef]
20. Jin, W.; Klinger, J.L.; Westover, T.L.; Huang, H. A density dependent Drucker-Prager/Cap model for ring shear simulation of ground loblolly pine. *Powder Technol.* **2020**, *368*, 45–58. [CrossRef]
21. Lu, C. *Determination of Cap Model Parameters Using Numerical Optimization Method for Powder Compaction*; Marquette University: Milwaukee, WI, USA, 2010.
22. Wagle, G.S. Die Compaction Simulation: Simplifying the Application of a Complex Constitutive Model Using Numerical and Physical Experiments. Ph.D. Thesis, The Pennsylvania State University, University Park, PA, USA, 2007.
23. Eberhart, R.; Kennedy, J. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
24. Pervaiz, S.; Ul-Qayyum, Z.; Bangyal, W.H.; Gao, L.; Ahmad, J. [Retracted] A Systematic Literature Review on Particle Swarm Optimization Techniques for Medical Diseases Detection. *Comput. Math. Methods Med.* **2021**, *2021*, 5990999. [CrossRef]

25. Pace, F.; Santilano, A.; Godio, A. A review of geophysical modeling based on particle swarm optimization. *Surv. Geophys.* **2021**, *42*, 505–549. [CrossRef] [PubMed]
26. Ceylan, Z. Short-term prediction of COVID-19 spread using grey rolling model optimized by particle swarm optimiza-tion. *Appl. Soft Comput.* **2021**, *109*, 107592. [CrossRef] [PubMed]
27. Gontara, S.; Boufaied, A.; Korbaa, O. Fault localization algorithm in computer networks based on the boolean particle swarm optimization. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 6–9 October 2019.
28. Banks, A.; Vincent, J.; Anyakoha, C. A review of particle swarm optimization. Part I: Background and development. *Nat. Comput.* **2007**, *6*, 467–484. [CrossRef]
29. Shami, T.M.; El-Saleh, A.A.; Alswaitti, M.; Al-Tashi, Q.; Summakieh, M.A.; Mirjalili, S. Particle swarm optimization: A comprehensive survey. *IEEE Access* **2022**, *10*, 10031–10061. [CrossRef]
30. Zhang, X.; Liu, H.; Tu, L. A modified particle swarm optimization for multimodal multi-objective optimization. *Eng. Appl. Artif. Intell.* **2020**, *95*, 103905. [CrossRef]
31. Li, W.; Meng, X.; Huang, Y.; Fu, Z.-H. Multipopulation cooperative particle swarm optimization with a mixed mutation strategy. *Inf. Sci.* **2020**, *529*, 179–196. [CrossRef]
32. Alitavoli, M.; Khaleghi, E.; Babaei, H.; Mostofi, T.M.; Namazi, N. Modeling and prediction of metallic powder behavior in explosive compaction process by using genetic programming method based on dimensionless numbers. *Proc. Inst. Mech. Eng. Part E J. Process. Mech. Eng.* **2018**, *233*, 195–201. [CrossRef]
33. Dec, R.T.; Zavaliangos, A.; Cunningham, J.C. Comparison of various modeling methods for analysis of powder compaction in roller press. *Powder Technol.* **2003**, *130*, 265–271. [CrossRef]
34. LaMarche, K.; Buckley, D.; Hartley, R.; Qian, F.; Badawy, S. Assessing materials' tablet compaction properties using the Drucker–Prager Cap model. *Powder Technol.* **2014**, *267*, 208–220. [CrossRef]
35. Majzoobi, G.; Jannesari, S. Determination of the constants of cap model for compaction of three metal powders. *Adv. Powder Technol.* **2015**, *26*, 928–934. [CrossRef]
36. Atrian, A.; Majzoobi, G.H.; Markert, B.; Nourbakhsh, S.H. A novel approach to calibrate the Drucker–Prager Cap model for Al7075 powder. *Arch. Appl. Mech.* **2018**, *88*, 1859–1876. [CrossRef]
37. Buljak, V.; Baivier-Romero, S.; Kallel, A. Calibration of Drucker–Prager cap constitutive model for ceramic powder compaction through inverse analysis. *Materials* **2021**, *14*, 4044. [CrossRef] [PubMed]
38. Zhou, R.; Yang, L.; Liu, Z.-W.; Liu, B.-F. Modeling the powder compaction process by an integrated simulation and inverse optimization method. *Mater. Today Commun.* **2020**, *25*, 101475. [CrossRef]
39. Wu, P.; Gao, L.; Zou, D.; Li, S. An improved particle swarm optimization algorithm for reliability problems. *ISA Trans.* **2011**, *50*, 71–81. [CrossRef] [PubMed]
40. Yang, I.-T.; Hsieh, Y.-H. Reliability-based design optimization with cooperation between support vector machine and particle swarm optimization. *Eng. Comput.* **2012**, *29*, 151–163. [CrossRef]
41. Perez, R.E.; Behdinan, K. Particle swarm optimization in structural design. *Swarm Intell. Focus Ant Part. Swarm Optim.* **2007**, *532*, 373–394. [CrossRef]
42. Li, Z.; Tian, G.; Cheng, G.; Liu, H.; Cheng, Z. An integrated cultural particle swarm algorithm for multi-objective reliability-based design optimization. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2013**, *228*, 1185–1196. [CrossRef]
43. Tseng, K.Y.; Zhang, C.B.; Wu, C.Y. An enhanced binary particle swarm optimization for structural topology optimization. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2010**, *224*, 2271–2287. [CrossRef]
44. Jalili, S.; Hosseinzadeh, Y.; Kaveh, A. Chaotic biogeography algorithm for size and shape optimization of truss structures with frequency constraints. *Period. Polytech. Civ. Eng.* **2014**, *58*, 397–422. [CrossRef]
45. Kaveh, A.; Hamedani, K.B.; Kamalinejad, M. Set theoretical variants of optimization algorithms for system reliability-based design of truss structures. *Period. Polytech. Civ. Eng.* **2021**, *65*, 717–729. [CrossRef]
46. Shi, J.; Chopp, D.; Lua, J.; Sukumar, N.; Belytschko, T. Abaqus implementation of extended finite element method using a level set representation for three-dimensional fatigue crack growth and life predictions. *Eng. Fract. Mech.* **2010**, *77*, 2840–2863. [CrossRef]
47. Chen, K.; Xue, B.; Zhang, M.; Zhou, F. Evolutionary multitasking for feature selection in high-dimensional classification via particle swarm optimization. *IEEE Trans. Evol. Comput.* **2021**, *26*, 446–460. [CrossRef]
48. Ali, M.M.; Kaelo, P. Improved particle swarm algorithms for global optimization. *Appl. Math. Comput.* **2008**, *196*, 578–593. [CrossRef]
49. Mashayekhi, M.; Harati, M.; Estekanchi, H.E. Development of an alternative PSO-based algorithm for simulation of endurance time excitation functions. *Eng. Rep.* **2019**, *1*, e12048. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.