Calibration of Drucker-Prager-Cap Model Parameters Using Particle Swarm Optimization Method for Powder Compaction Process

By

Sanaz Davarpanah

A thesis submitted to the Department
of Computer Science in conformity
with the requirements for the degree of
Master of Science

Bishop's University

Canada

June 2023

Abstract

In this dissertation, a novel method for powder compaction process optimizations using the particle swarm optimization method is proposed. In order to optimize the design and functionality of powder compaction machines, it is essential to increase the accuracy of powder compaction simulations. Due to the extremely nonlinear and multi-objective character of the issue, accurate modelling of this process necessitates the calibration of intricate constitutive models.

To create an effective and accurate calibration procedure, the suggested method includes computer science based techniques such as particle swarm optimization algorithm, finite element modeling, and inverse optimization approach. The optimal parameters of the most widely used powder compaction model (Drucker-Prager-Cap) are sought using the particle swarm optimization method in order to reduce the difference between the simulated and experimental compaction curves.

Through a series of numerical experiments and a case study of a methodology powder, the proposed method is evaluated. The outcome demonstrates that, in comparison to the traditional calibration techniques, the suggested strategy can greatly increase the accuracy of powder compaction simulations. In addition, the suggested method is computationally effective and simple to include into a computer programmer, making it appropriate for controlling and monitoring powder compaction processes in real-time.

This study emphasizes the value of computer science in coming up with creative solutions to complex engineering challenges as well as the possibilities of combining various computer science techniques to increase the effectiveness and accuracy of manufacturing processes. Other manufacturing procedures that require for the calibration of complex constitutive models can use the proposed methodology.

Key word: Inverse Optimization, Particle Swarm Optimization, Finite Element Method, Material Model Calibration

Acknowledgments

I would like to take this opportunity to express my deep appreciation and gratitude to everyone who has supported me during the course of my thesis.

First, I would like to extend my endless thanks to Dr. Madjid Allili, my supervisor, for his tremendous encouragement and unfailing dedication. Not only has he taught me a great deal technically but also, I have found his other lessons especially in time management as an engineer equally important. His invaluable and feedback were crucial is shaping my research and improved the quality of my work and I would like to express my most gratitude for his support throughout these two years.

I am also grateful to Dr. Hugh Scott for his crucial support and kindness during my Master's degree. His guidance and belief in me helped me achieve my goals, and I am deeply thankful for his unwavering support.

Additionally, I am grateful to Bishop's University's Computer Science department for giving me the opportunity to pursue a master's degree. I'd like to express my heartfelt gratitude to all the professors in the Department of Computer Science, especially Dr. Mohammed Ayoub Alaoui Mhamdi and Dr. Layachi Bentabet for their guidance, support, and invaluable insights during my academic path.

Hard periods make good friends as remarkable. Thanks to Nasir, Thomas, Aditya and Moinuddin for putting up with me in every single minute, without them surviving to the end of this way was not possible, you are more than a friend to me.

At last, I would like to dedicate this thesis to my beloved family. To my mother, I am grateful for her unwavering support and patience throughout my academic journey. And to my late father, who passed away during my Master's degree, I am deeply grateful for his love and unwavering support that has shaped me into the person I am today. Although he could not witness this significant milestone, his memory will continue to inspire me in all my future endeavors. I feel so lucky to have you as my parents. Your unconditional love and willingness to do anything for your children is beyond remarkable. I am so thankful to both of you.

Thank you all for your support, encouragement, and guidance, I could not have done this without you.

Table of Contents

List of Tables

List of Figures

List of Notations

α: Transition surface radian

β: Angle of friction are

c_1: Weight of personal best

c_2: Weight of global best

d: Material cohesion

d_1: Optimization variable (first coefficient of material cohesion)

d_2: Optimization variable (second coefficient of material cohesion)

E: Young's modulus

E_1: Optimization variable (first coefficient of Young's modulus)

E_2: Optimization variable (second coefficient of Young's modulus)

ε_0: Initial yield surface position

ε_v^{pl}: Volumetric plastic strain

$f(x)$: Objective function

F_s: Shear failure line

F_c: Cap surface

F_t: Transition surface

$g_i(x)$: Inequality constraints

$h_j(x)$: Equality constraints

K: Flow stress ratio

p: Hydrostatic pressure stress

p_a: Evolution parameter

p_b: Hydrostatic pressure yield stress

P_i^k: Best previous position of the swarm at iteration k

P_g^k: Best global position of the swarm at iteration k

P_1: Optimization variable (first coefficient of hydrostatic pressure yield stress)

P_1: Optimization variable (Second coefficient of hydrostatic pressure yield stress)

q: Mises equivalent stress

R: Cap eccentricity

r_1: Random numbers distributed uniformly in [0, 1]

r_2: Random numbers distributed uniformly in [0, 1]

ρ: Current relative density

ρ_0: Initial relative density

σ_Y: Yield strength

V_i^k: Current position of the ith particle at iteration k

v: Poisson's ratio

ω: Inertia weight

$\{x_i\}$: Feasible space

x^*: Global optimal solution

X_i^k: Velocity of the ith particle at iteration k

Chapter 1: Introduction

1.1 Introduction

Predicting and controlling the behavior of systems that directly or indirectly affect human life is one of the most important topics that are studied in various sciences. In this regard, a wide range of subjects such as weather forecasting, public transportation, the final price of a product and even reducing deaths caused by the coronavirus can be studied. In the case of physical phenomena, we need a material model that shows the relationships between the input variables and the desired output in the form of a mathematical relationship. The main difficulty in this regard is determining the coefficients in material models which sometimes requires spending a lot of time and money to conduct numerous experimental tests.

The application of Optimization algorithms to the determination of material coefficients has emerged as an important topic of study in computer science. This is due to the fact that optimization problems are increasingly seen as urgently needed solutions in a variety of 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 modelling the behaviour of materials in industrial processes. As a result, optimization algorithms are becoming increasingly crucial in the advancement of materials science and engineering. The main goal of this research is 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 modelling the industrial powder compaction process. This work requires providing experimental data for training the optimization algorithm, finite element modeling of the process in a commercial software, coding the PSO in Python, establishing a connection between the algorithm and the finite element software, and finally validating the results. In this chapter, the general concept of each of the above items is explained.

1.2 Fabrication of industrial parts using powder compaction method

Powder compaction is one of the most common methods of producing industrial parts, especially those with complex geometries or high porosity. The outstanding features of powder compaction, such as low waste of raw materials, high speed in large production rate, the ability to control the physical properties of the parts, the ability to produce composite parts, and the high safety of the method have caused an increasing demand for powder compaction in diverse industries. Figure 1.1 shows various industrial applications of the powder compaction process. The main attraction

of powder compaction comes from its ability to directly transform the powdered material into the net-shape part.

Figure 1.1 Industrial applications of the powder compaction process

The main limitation of the powder compaction is the complexity of the modeling process in this method. So far, several material models have been presented to express the relationship between the effective parameters in the process, among which the DPC has been more popular. Although the existence of a large number of coefficients, whose accurate determination requires multiple powder compaction tests, has caused these coefficients to be extracted so far only for a limited number of commonly used powders [1].

To achieve the desired density and strength of the final product, we may simulate and optimize the powder compaction process using the optimal DPC model parameters. While powder compaction is frequently quicker and more affordable than other fabrication methods, this could result in a more effective manufacturing process. Additionally, we can reduce material waste and flaws and improve the accuracy and dependability of the production process by accurately predicting the behaviors of the metal powder during compaction. In conclusion, by enabling us to precisely simulate and tune the behaviors of materials throughout production processes, the combination of materials science and computer science might result in more precise and efficient manufacturing processes.

1.3 An introduction to mathematical optimization

Mathematical optimization of a system is the process of minimizing or maximizing a function called the objective function, which is actually a criterion of system performance [2]. So, the optimization process ultimately leads to the improvement of the performance and efficiency of the system. With this definition, optimization is considered an inseparable part of nature and human life such that nowadays many scientific fields such as economics, engineering, basic sciences, natural resources, and even some trends of social sciences, use mathematical optimization methods to solve their problems. In general, an optimization problem is mathematically defined as follows:

$$\begin{aligned} &\text{Minimize} \quad f(x) \\ &\text{Subject to} \quad g_i(x) \leq 0 \quad i = 1,2,\cdots,n \\ &\qquad\qquad\ \ h_j(x) = 0 \ \ j = 1,2,\cdots,m \end{aligned} \tag{1.1}$$

In optimization problem 1.1, $f(x)$ is objective function, $h_j(x)$ are equality constraints, $g_i(x)$ are inequality constraints and set of answers $\{x \in R \mid g_i(x) \leq 0; h_j(x) \leq 0\}$ is the feasible space of the problem. On the other hand, x^* is considered the global optimal solution if we have $f(x^*) \leq f(x)$. Also, a maximization problem for the objective function $f(x)$ is equivalent to minimizing -$f(x)$. Figure 1.2 shows a schematic representation of searching the possible space for the global optimal solution in a simple mathematical optimization problem.

Classical optimization methods are divided into two categories: direct search methods and gradient-based methods [3]. Direct search methods only use the function $f(x)$ and constraints $h_j(x)$ and $g_j(x)$ for the search algorithm and do not need any information about the gradient (derivatives) of the objective function (Figure 1.3a). These methods usually have a slow movement and require a lot of functional calculations for convergence, nevertheless, they can be used in problems including discontinuous or differentiable objective functions.

On the other hand, gradient-based methods use the first or second-order derivative of the objective

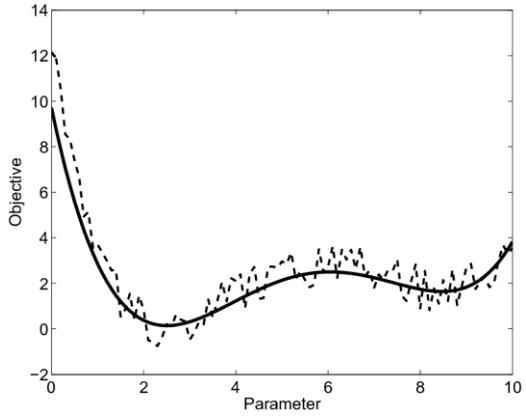
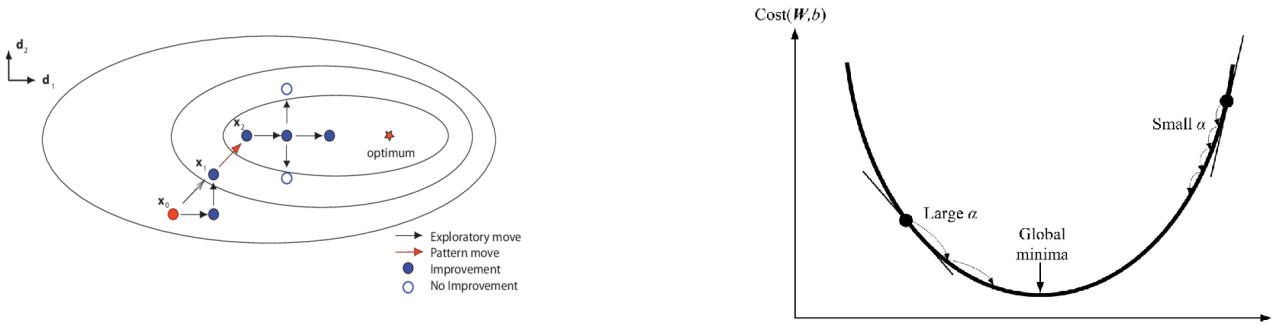


Figure 1.2 Schematic representation of a single-value optimization problem [4]



(a) (b)

Figure 1.3 Schematic representation of: a) direct search method, b) gradient-based method [5]

function and constraints for their searching algorithm (Figure 13b). Unlike direct methods, these methods have fast convergence around the optimal solution, but still, they are not effective in problems with non-differentiable functions or discontinuous possible space. The common problems that exist in the classical search methods are as follows:

1. Convergence to the optimal solution depends on the selected initial solution.

2. Some search algorithms tend to converge to the local optimal solution instead of global one.

3. An algorithm that is efficient in solving one optimization problem may not be efficient in solving another problem.

4. Some algorithms are not efficient when dealing with optimization problems with discrete feasible space.

1.3.1 An introduction to Evolutionary Algorithm:

Creatures in nature must adapt themselves to the environment in a way to remain in the cycle of evolution. Therefore, creatures that have the best characteristics are preserved and reproduce in competition with other ones whilst those with weaker characteristics have disappeared. This theory, which was proposed by the ancient Greeks under the title of “survival of the fittest”, is the basis of another group of optimization methods that are inspired by the principles governing nature [6]. These methods which are called “evolutionary algorithms” are random methods based on the initial population that solve optimization problems by the random combinations of change factors (crossover, mutation, and selection). Evolutionary algorithms, unlike classical methods, instead of using only one solution, use a population of solutions in the search process and try to reach the optimal solution by using their operators. Also, evolutionary algorithms have the ability to combine with other numerical methods which leads to obtaining multiple optimal solutions in some specific optimization problems. In addition to all the mentioned cases, the main advantage of the evolutionary algorithms is that they do not need the derivability or even the continuity of the objective function as well as constraints of the optimization problem in their search process. Figure 1.4 shows the flowchart of evolutionary searching algorithms in an optimization problem.

The goal of optimization is to find the optimum solution to a given issue given certain limitations. Optimization is used by all professions to discover the best answer to their problems. Engineers' first and greatest goal is also optimization. As a result, especially in future engineering applications, optimization will be an essential component of the product.

Optimization may be found anywhere. Optimization is the most crucial component of the application whether creating a new device, a new artificial intelligence technique, a big data application, or a deep learning network. Optimization is required to build a device with the smallest possible footprint while consuming the least amount of energy, to train a network, and to minimise the difference between the desired and actual output values.

Because of the challenges of classical optimization methods, scientists began to look for a simpler solution to their problems in the 1960s. The advancement of computers has simplified scientists' efforts, and whole new problem-solving techniques are being researched. These heuristic information-based strategies were derivative-free, simple to apply, and reduced solution time.

1.3.2 Genetic Algorithm

Holland invented the genetic algorithm (GA) [8]. The evolutionary concept has been used to solve optimisation challenges. Instead of generating only one solution, the algorithm employs a set of solutions known as population. Each answer is referred to as unique. Such algorithms could be run with several processors in this manner. Following GA, simulated annealing [9] was widely regarded as the second algorithm, which was inspired by the annealing process of physical materials. Particles travel randomly in high temperatures to explore the solution space. While the temperature falls, particles attempt to form a flawless crystalline structure using only local movements.

1.3.3 Particle Swarm Optimization

The second population-based technique influenced by animals is particle swarm optimisation (PSO). Since James Kennedy (a social psychologist) and Russell C. Eberhart simulated the foraging behaviours of birds and fish, they have used this simulation to solve an optimisation problem and published their idea in a conference in 1995 [10] for the optimisation of continuous nonlinear functions. The PSO population and each member of the group are called “swarm” and “particle” respectively also PSO algorithm is divided into two parts: velocity and coordinates for each particle. In a solution space, each particle has a coordinate and an initial velocity. The particles converge towards the optimal solution coordinates as the programme advances. PSO requires less memory and has no operator because it is straightforward to implement. PSO is a quick algorithm because of its simplicity. Since the original version of PSO was released, many versions have been developed, each using a different set of operators. Figure 1.5 shows a schematic view of the searching process for the population of particles to find the global optimal solution in the PSO optimization algorithm. According to the figure 1.5, each particle is randomly generated in the search space at the beginning of the PSO process. The PSO group has a group memory so that each particle always remembers the best position it has obtained and also the best position the group has found and based on a combination of these two parameters, it moves in the space with its orientation factor (velocity). Therefore, social intelligence is the best interpretation that can be proposed about PSO [11].

In the earliest stage of PSO, the velocity was estimated using a basic formula that multiplied stochastic variables with current velocity, personal best, and local best values. The present particle

updates not just its previous best velocity, but also the global best. Using stochastic variables, the whole probability was allocated between local and global best.

Shi and Eberhart introduced an inertia weight in the next version to control the velocity in 1998 [12]. The inertia weight balances the algorithm's ability to search locally and globally. The inertia weight determines the rate at which former velocity contributes to current velocity. Researchers made many contributions to the idea of inertia weight. Various researchers proposed linearly, exponentially, or randomly decreasing or adapting inertia weight. [13].

Clerc and Kenedy [14,15] proposed a new parameter called constriction factor in the next edition of PSO. In the investigations on the stability and convergence of PSO, the constriction factor (K) was added. Clerc claims that the use of a constriction factor ensured the PSO's convergence. Shi and Eberhart [16] reported a comparison of inertia weight and constriction factor.

PSO has been used for solving nearly all engineering and science challenges. PSO has been used to tackle problems in Electrical Engineering, Computer Sciences, Industrial Engineering, Biomedical Engineering, Mechanical Engineering, and Robotics. PSO is used in Electrical Engineering to solve the power distribution problem [17]. Economic dispatch is another well-studied subject in Electrical Engineering [18, 19]. Face localization [20], edge detection [21], imaging segmentation [22], and image denoising are all examples of how particles are dispersed in the solution space in computer science. Particles have initially positions as well as velocities. Using swarm intelligence, they arrive at the best option.

Although swarm intelligence has improved optimisation algorithms, there is no single technique that is successful in all types of optimisation issues. As a result, efforts to imitate animal behaviours and swarm intelligence will continue. Simultaneously, hybrid algorithm development will continue until the best combination of algorithms is discovered.

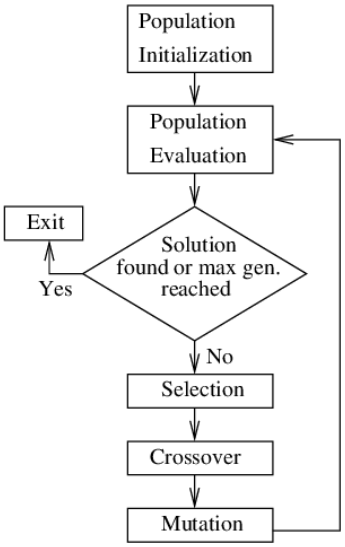


Figure 1.4 Flowchart of evolutionary searching algorithms [23]

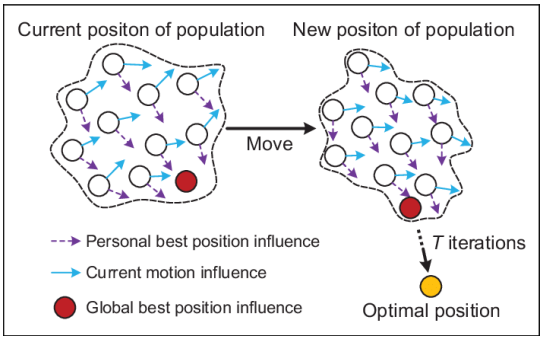


Figure 1.5 schematics of searching for the optimal solution in the PSO optimization algorithm [24]

In recent years, the use of computer simulations in materials fabrication processes has grown in significance since it enables a more in-depth comprehension of material behaviors and can improve production procedures. It can be challenging to accurately predict and simulate the behaviors of materials under complex stress states. Using robust constitutive models, such the DPC, which can accurately depict the complicated behaviors of materials under a variety of stresses, is one method for enhancing simulation accuracy. These models, however, frequently need for parameters that are challenging to determine through experimentation.

This problem can be solved by calibrating the model parameters with optimizations algorithms like PSO, which enables a more realistic portrayal of the material behaviors. The DPC model and PSO can be combined to produce a simulation tool that is more precise and effective for studying material behaviors and optimizing production procedures.

In this research, mechanics of materials and computer science were combined to provide a method for predicting the behaviors of materials under stress that is more effective and precise. This technique has the potential to increase the accuracy of simulations used in a variety of industries, including biomedical engineering, aerospace, and the automobile industry.

1.3.2 Inverse optimization

In an inverse optimization problem, it is assumed that the solution x^* is given as an input so that it may or may not be the optimal point. Now we have to determine the coefficients of the objective function so that x^* is optimized under the new values of these parameters. Finding these parameters is an optimization model itself, which is called an "inverse model". In fact, the aim of solving an inverse optimization problem is to convert a possible solution into an optimal one with the least possible changes in the coefficients of the objective function. Inverse optimization is an emerging subject. Its application in fields such as economics, management, and industry increase daily [25]. This has caused the improvement of inverse optimization and extension of its application to be of great interest to researchers.

Figure 1.6 shows a typical flowchart of an inverse optimization problem. As can be seen in most inverse optimization problems, we have an experimental data set that is considered as the possible

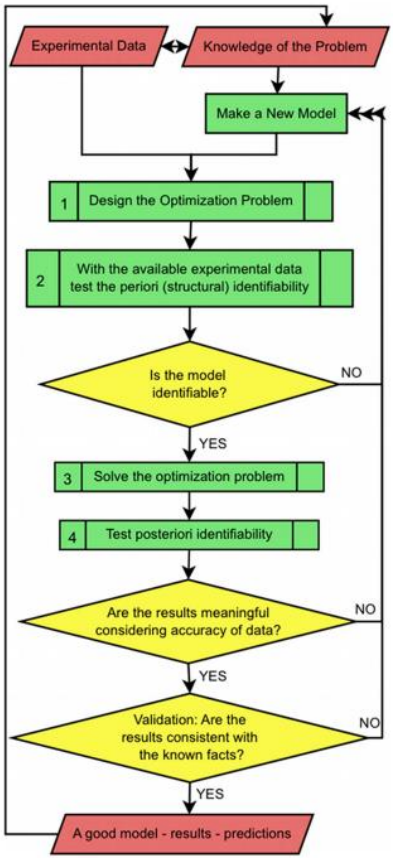


Figure 1.6 A typical flowchart of an inverse optimization problem [26]

solution x^*. In the next step, using the knowledge and information we have about the nature of the problem, we consider a model (objective function) which is the same as the initial population. We now have an optimization problem with the purpose of determining the parameters of the proposed model. Then, based on the experimental data, we check the proposed model. If the model was solvable, we will solve it and then check it again. In the following, if the optimization results (model coefficients) are physically meaningful, their agreement with the experimental results is checked and the proposed model is considered as the optimal model otherwise, a new model is proposed, and this process continues until the optimal model is reached.

1.4 An introduction to Finite Element Method

The finite element method (FEM) is a numerical method for solving problems in the fields of engineering, mathematics, and physics. This method is used in issues such as forming of industrial parts, joints, metal and concrete structures, heat transfer, fluid mechanics, mass transfer, and electromagnetic fields. To solve such problems through analytical methods, it is necessary to obtain the solution to several boundary value problems for partial differential equations. Physical phenomena have governing equations that describe their conditions. These equations are mostly nonlinear and complex due to the conditions that exist in the model and its surrounding environment. In these cases, the usual analytical methods that are presented to solve the differential equations governing the problem are no longer applicable [27]. Therefore, other methods are used to solve these equations, which are called numerical methods. FEM is one of these numerical methods that can be used to numerically solve complicated differential equations.

According to Figure 1.7, to simplify the problem of solving the governing differential equations in this method, the whole region in which the differential equations must be solved is divided into smaller components during a process called “discretization”, and each component is called an “element”. From the name of this method, it can be inferred that the number of elements may be large, but it is not infinite and can be counted [28]. In the next step, according to the nature of the problem, simple equations that represent the behavior of these finite elements are set next to each other in a system of algebraic equations and generate the general form of the main problem. It can then be said that the practical application of the FEM is to convert the differential equations governing the main problem into ordinary

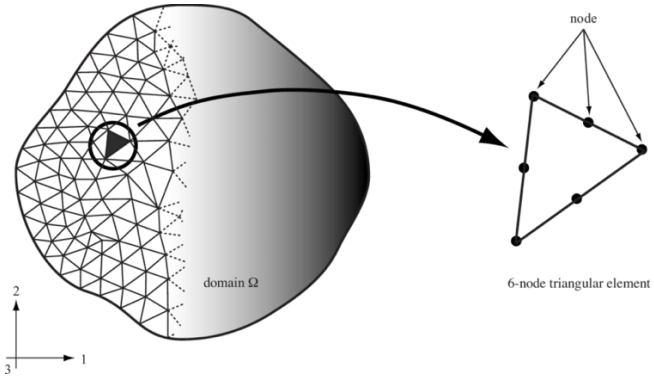


Figure 1.7 Discretization process in FEM [28]

differential equations governing the elements, which can be solved by numerical methods. FEM is especially useful in problems involving complex geometry (such as vehicles), variable domain (such as metal forming processes), and when high accuracy is not required everywhere in the domain (such as automobile crash tests). The analysis of a problem by the FEM includes five general parts which are:

Pre-processing: It consists of discretization of the problem domain into finite elements. At this stage, a complex geometry representing a continuous domain is divided into simpler geometric shapes called elements.

Constitutive model: In this step, we assume a mathematical relation that describes the behavior elements during the process and satisfies the boundary conditions at the same time. Such a relationship is called a constitutive model in the case of engineering problems.

Assembly: In this step, after obtaining the governing equations for elements, the set of governing equations for the entire domain of the problem is determined by assembling the governing equations of elements.

Solving equations: In this step, we use all kinds of existing numerical methods such as finite difference, Euler, and Runge-Kutta to obtain the unknown physical parameters by solving the set of differential equations governing the assembled elements.

Post-processing: in this stage, after obtaining the unknowns of the problem, we determine the secondary parameters (e.g., tension, pressure, velocity, etc.) that are considered. Of course, in

simple cases such as systems including springs or trusses, the behavior of the component can be defined directly and there is no need to consider the governing differential equation.

1.5 Combining FEM and inverse optimization for modeling engineering problems

One of the new topics in engineering sciences that has recently attracted the attention of researchers in different fields is the integration of FEM and inverse optimization for modeling different phenomena. Figure 1.9 shows an example of these efforts. According to the figure, in this method, first a simple and low-cost test is performed on the material and the result is considered as the experimental data. Then a mathematical model that describes the behavior of the material during the process is considered and the parameters in the model are assumed as inputs (set of answers). Based on the available knowledge and information, the process is modeled and analyzed using the FEM. In the following, the output of FEM analysis is compared with experimental data. Usually, the difference between the output of FEM analysis and experimental data is defined as an objective function and minimized using an optimization algorithm. If the difference between these two is greater than a certain limit, another population of initial answers is produced, and this process continues until reaching a reasonable agreement.

1.5.1 The relationship between the method used in research and computer science

The field of computer science studies computers and computing systems and its main focus is on the theory, design, development, and application of software systems to solve challenges and problems in various fields. According to this definition, computer science is used in a wide range of issues such as statistics, economics, management, transportation, health, engineering, and other fields related to human life [29]. The main subject of this research is determining the material coefficients in the Drucker-Prager-Cap (DPC) model using the combination of FEM and inverse optimization method which ultimately leads to the modeling of the metal powder compaction. For example, computational simulations and models are likely to be used to calibrate the model parameters and optimize the powder compaction process. Computer science has played a vital role in creating and implementing the computing parts of this research.

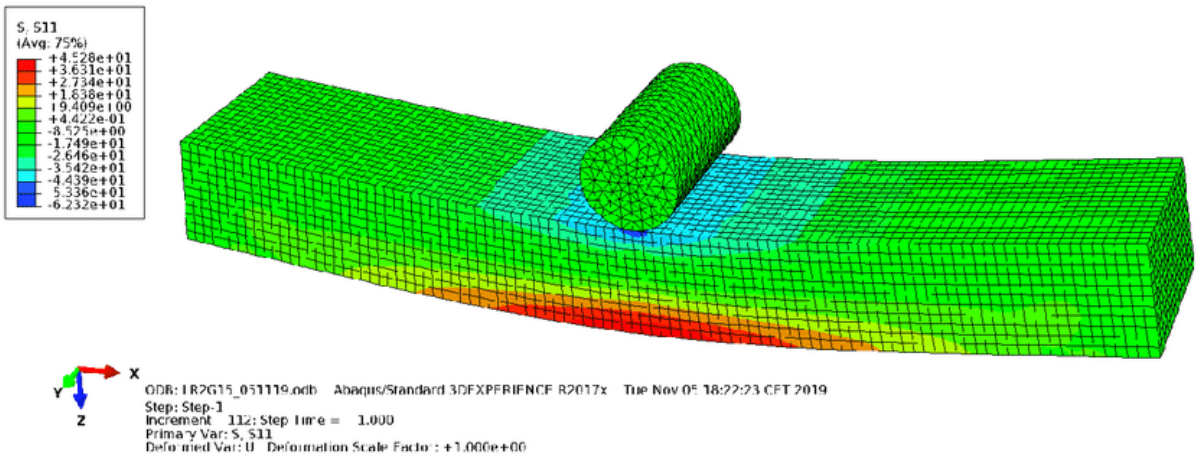


Figure 1.8 Stress field resulted from FEM modeling of three-point bending test of a beam [30]

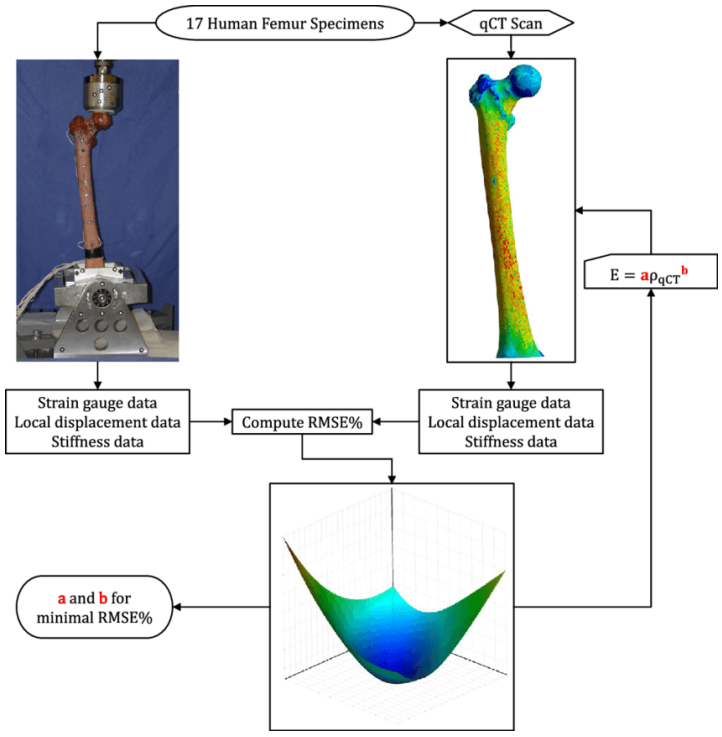


Figure 1.9 Combining FEM and inverse optimization method for modelling bone under axial loading [31]

Numerical techniques and algorithms, which are fundamental ideas in computer science, would be used in these simulations. Due to the computational tools and techniques involved, calibration of DPC model parameters using PSO technique for powder compaction process is an interesting applications of Computer science approaches.

1.5.1.1 FEM modeling of the powder compaction process

As explained earlier, the FEM is considered as a very appropriate solution in cases where the modeling of the desired process is complicated for various reasons such that the differential equations governing the process cannot be solved using common computational methods. Today, the principles governing this method have been implemented and made available to users in the form of FE commercial software such as Abaqus, Ansys, Comsol, Ls-Dyna, Nastran, etc. One of the phases of this research is related to the FEM model of the powder compaction process in Abaqus software. On the other hand, one of the main fields of computer science is dedicated to the use of software sets for inventing, improving, and developing industrial processes. Improving the powder compaction process by means of its modeling in FE software can be considered as one of the applications of computer science.

1.5.1.2 Inverse optimization of the DPC model to determine material coefficients

As explained earlier, the optimization of the DPC material model coefficients for simulating powder compaction process requires the use of an optimal solution search algorithm. In the present research, the PSO algorithm is used for this purpose. Implementation of the PSO optimization algorithm to determine the coefficients in the DPC model requires programming. We know that programming is one of the subsets of computer science [32]. Therefore, it can be said that the implementation of the principles governing the PSO optimization algorithm in the form of command codes in a programming environment is another example of the relationship between the research topic and the field of computer science. On the other hand, PSO is a metaheuristic optimization technique used in computer science to repeatedly update a swarm of candidate solutions in order to discover the optimal answer to a problem. In this instance, the DPC model parameters are calibrated using the PSO method to produce a more accurate view of the powder compaction process.

1.5.1.3 Creating a connection between the optimization algorithm and FE software

In the classical optimization methods of engineering problems, first, according to the nature of the problem, the effective parameters (e.g., length, pressure, velocity, energy, etc.) are identified and one or more output parameters (e.g., the weight and deformation of the structure, consumed energy, efficiency, etc.) are determined as the objective function. Then, based on the effective parameters and the objective function, an input-output data set is produced using experiments, in

which the value of the objective function is reported for different values of the effective parameters. In the following, an optimization algorithm is used to determine the value of the effective parameters to optimize the objective function. Considering that the necessary experiments to determine the coefficients of the material in the DPC model require costly and time-consuming experiments, using the experimental method to determine DPC coefficients is not cost and time effective in any way.

The innovation presented in this research, which makes it unnecessary to perform these experiments, is that the optimization algorithm is developed in a way that directly communicates with the FE software and controls its process. In other words, first, a set of model coefficients is generated according to the constraints defined for them by the optimization algorithm. This set is defined as DPC material model coefficients for the FE software and the software solves the problem according to the input coefficients and then reports the desired output (force-displacement diagram). Next, the optimization algorithm calls the diagram from the FE software and compares it with the objective function (experimental force-displacement diagram) and produces a new set of coefficients according to their difference value as well as its previous learning. According to the explanations provided, the method used in this research creates a direct and two-way connection between the FE software and the optimization algorithm. However, the DPC model's parameters, which are challenging to obtain experimentally, impact how accurate the simulation will be. Computer science can help with this. In order to ensure that the simulated results strongly match the experimental results, we can apply optimizations methods like PSO to determine the ideal values for these parameters.

1.6 Research necessity

Powder compaction is one of the methods of producing industrial products which has been widely used in various industries due to its unique advantages. Fabrication of car parts using the powder metallurgy process as well as the production of various pills and tablets in pharmaceuticals are the main applications of this method in the industry. At the same time, the modeling of this process is very difficult and complicated, which prevents the prediction of the properties of the parts resulting from compaction. This results from the fact that the accurate determination of the coefficients in the material models used to simulate the powder compaction process requires time-consuming and expensive experiments. Overcoming this challenge and providing new methods that determine the

coefficients in material models without the need to perform numerous and difficult experiments have recently become an important issue in the field of powder compaction and have attracted the attention of many researchers. In continuation of this process, this research seeks to provide a new solution to overcome this limitation and introduce a robust method to the determination of DPC as the most common material model used to simulate the powder compaction process.

1.7 Research objectives

To overcome the problem of experimentally determining the coefficients in this model, this research aims to provide a new solution for the accurate determination of material coefficients in the DPC model by using the combination of FEM and PSO inverse optimization. This requires simulating the powder compaction process in a FE software, implementing the PSO optimization algorithm in a programming environment, and creating a two-way communication between the two, all of which are done in this research.

1.8 Thesis structure

This research is presented in five chapters. In the first chapter, the basics and concepts related to the research topic are explained and introductory descriptions regarding the powder compaction method, mathematical optimization process, PSO optimization algorithm, inverse optimization method, and FEM are provided. In the second chapter, the background of research related to the subject is presented based on a specific category and the most recent research related to the subject is cited. The third chapter explains the methods used in the research, which includes how to define the objective function, how to FEM modeling of the powder compaction process, how to implement the PSO algorithm in the Python programming environment, and how to establish a connection between the optimization algorithm and the FE model. In the fourth chapter, the most important results of the research are presented, and the details related to these results are discussed in detail. Finally, in the fifth chapter, a summary of the most important findings of the research are presented and suggestions for the continuation of this topic in the form of future research is presented.

Chapter 2:
Literature Review

2.1 Introduction

In computer science, optimizing describes the method of choosing the best component from a list of available options. Finding the smallest or largest number in the set, for example, is one criterion that is used to base the selection. Computer scientists use a variety of methods to modify software systems, including programmer, code, and software optimizations. In data science, optimized data analytics are utilized to assist human decision-making. The objective of design and optimization can be to reduce production costs or improve manufacturing effectiveness. An optimizations algorithm is a process that involves comparing different solutions until the best or most satisfactory option is found.

2.2 Optimization in computer sciences researches

Optimization in computer science the process of maximizing the use of time, space, and other resources in a computer system, programmer, or process. It may involve finding innovative ways to use resources more efficiently or enhancing the speed, effectiveness, or quality of a system or process. A powerful optimization algorithm that has been extensively applied in numerous computer science domains is PSO. To increase PSO's effectiveness and performance, numerous researchers have looked into various PSO factors. In this overview of the literature, we'll talk about a few recent research that looked into the PSO optimization parameters in computer science.

The social behaviour of fish schools or flocks of birds served as the basis for the metaheuristic optimisation method known as PSO. In several areas of computer science, including machine learning [33, 34], data mining [35], image processing [36], and optimisation issues, PSO has been extensively used. Here, we will talk about some recent computer science researches on PSO:

1. Machine Learning: Many machine learning applications, such as feature selection, clustering, classification, and regression, have made use of PSO. The authors of a recent study suggested a hybrid PSO and artificial neural network (ANN) model for predicting the stock market index [37, 38]. The PSO algorithm was applied to optimise the ANN parameters, and the outcomes were superior to typical ANN models.

2. Data Mining: PSO has been used to perform a variety of data mining tasks, including association rule mining, outlier detection, and dimensionality reduction. The authors of a recent work developed a unique PSO-based algorithm for feature selection in high-dimensional data [39]. The method was able to identify relevant features and increase the model's classification accuracy.

3. Image Processing: PSO has been used in image segmentation, registration, and enhancement, which is the act of splitting an image into several sections based on similarities. Registration is the act of aligning two or more photos; enhancement is the process of improving the quality of an image; and denoising is the process of removing noise from an image. The authors of a recent study proposed an FPGA implementation of a PSO Based RGB-Y Filter [40]. FPGA-based RGB to Y (Luma) conversion is critical in image denoising, video processing, and computer vision applications, and this approach was able to eliminate noise from images while keeping their edges and details.

4. Optimization Problems: PSO has been widely employed in the solution of different optimisation issues, such as the travelling salesman problem, vehicle routing problem, and scheduling problem. In a recent study, the authors suggested a hybrid algorithm for feature selection based on PSO and genetic algorithm (GA) [41]. One of the most difficult tasks in machine learning and data mining is developing a basic model with a few features in order to reduce the computational complexity of the algorithms involved in categorization. According to a recent research, feature collection is essential in the classification process in order to minimise computing time, which reduces data size and boosts the precision and effectiveness of specific machine learning operations. The method can find optimal solutions for several objectives at the same time.

According to the review, PSO has demonstrated excellent results in handling complicated optimisation problems, and researchers are constantly investigating new applications and enhancing the performance of PSO algorithms. Also, PSO has been widely used in various fields of computer science and has shown promising results in solving complex optimization problems. The review, however, emphasises the need for additional research on the scalability, resilience, and efficiency of PSO algorithms in large-scale and real-world applications.

With the use of the PSO method for optimization process, my thesis proposed a novel way of calibrating the DPC model parameters. Pharmaceutical, ceramic, and metallurgical industries all use powder compaction as a manufacturing technique. For the purpose of optimizing the design

and use of powder compaction machines, accurate modelling of this process is essential. Yet, because of the process' complexity and absence of a systematic method, calibrating the model's parameters is a difficult undertaking.

In conclusion, PSO optimisation parameters have received a great deal of attention in computer science. Some researchers have even suggested improved PSO algorithms for a variety of applications, such as inverse kinematics and inverse design issues. To achieve better optimisation outcomes, parameter tuning is crucial since the optimal values of these parameters may change depending on the application. PSO has been demonstrated to be a useful tool for inverse optimisation problems in computer science. Inverse optimisation is a developing subject that seeks to tackle inverse problems using optimisation techniques. The swarm intelligence inspired suggested PSO method is a potent optimization method that has been successfully applied to several optimizations issues. In this study, the DPC model parameters are optimized using the PSO technique to minimize the difference between the simulated and experimental compaction densities. The proposed method is suited for real-time monitoring and control of powder compaction processes since it is computationally effective and simple to apply in a computer programmer. This study emphasizes the value of computer science in creating novel solutions for challenging manufacturing processes as well as the possibility of optimizations strategies to enhance powder compaction's effectiveness and quality. Computer science techniques played a vital role in the success of this study. 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 findings of this study can serve as a foundation for further research and advancements in this field.

2.3 Modeling the powder compaction process

The research conducted in the field of modeling the powder compaction process can be classified based on two general approaches including phenomenological and micromechanical. The phenomenological approach considers the powder as a continuous medium and thus uses the constitutive equations governed continuum mechanics. Another approach in powder compaction modeling is the micromechanical approach. In the micromechanical approach, the powder environment in the mold is considered as a set of discrete particles. Most of the researchers who

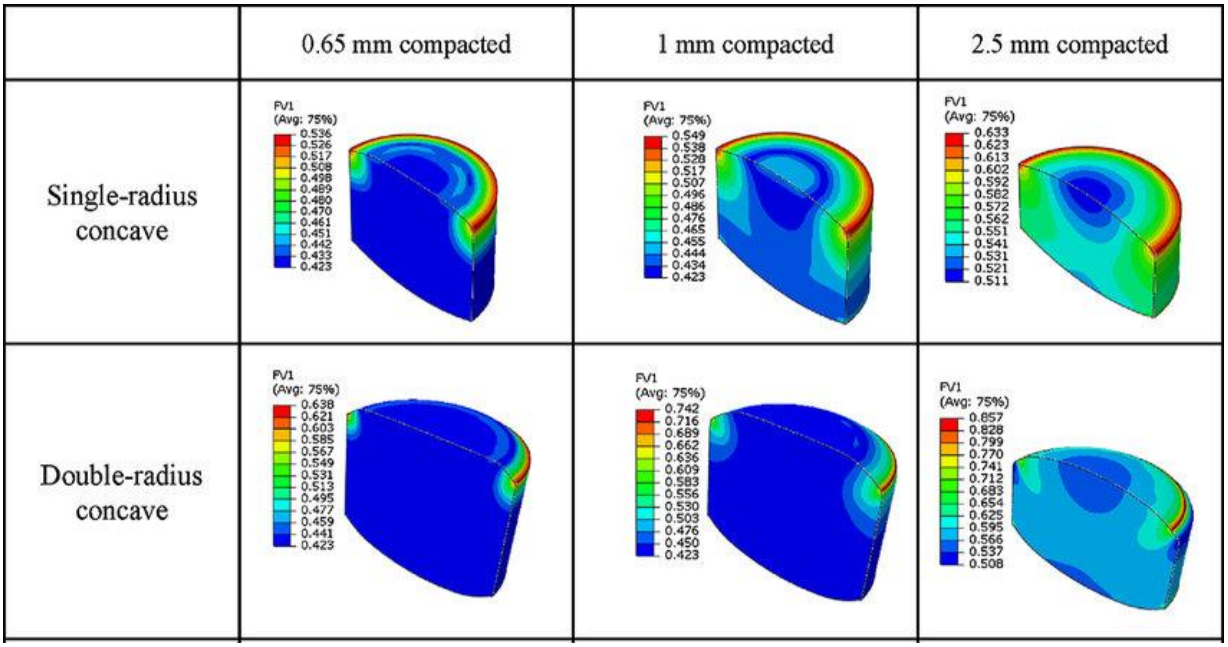
have used the micromechanical approach to model the powder compaction have considered the powder particles as spheres without deformability (Discrete Element Method-DEM). Recently, the Multi-Particle Finite Element Method (MPFEM) has been introduced as a promising micromechanical-based modeling method for modeling materials that are composed of small deformable particles. Unlike the DEM, In the MPFEM each particle is considered as a deformable material, and as a result, it will be possible to model the forming processes of particulate materials using this method [42]. Figure 2.1 shows the difference in the modeling of the powder compaction process in phenomenological and micromechanical approaches.

2.3.1 DPC phenomenological model

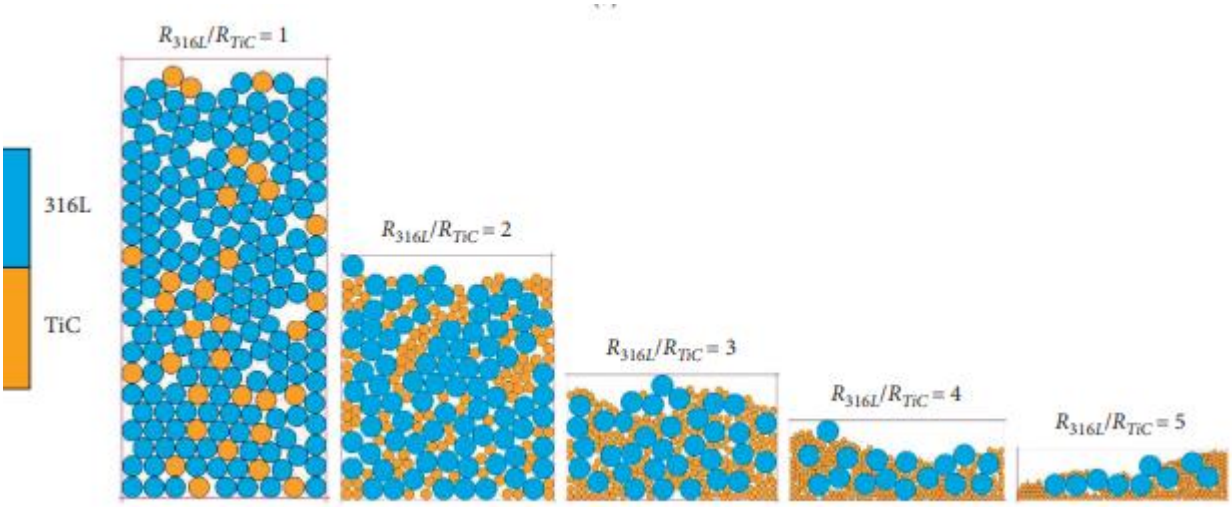
Among the various phenomenological models that have been presented by researchers to simulate the powder compaction process, the DPC model has received much more attention than the others, especially in recent years. The reason for the widespread acceptance of the DPC is that the effects of the three 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 [43]. This causes the results predicted by this model to be in good agreement with those obtained by experiments. The initial version of the DPC model, which was introduced in 1952 by Drucker and Prager, was a stress-dependent yield criterion for predicting the plastic yield of soil [44]. 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. To overcome this limitation, DiMaggio and Sandler modified the Drucker-Prager model taking into account the powder strain-hardening effect by adding a cap yield surface instead of assuming a perfect plastic material for the powder [45]. This 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.

2.3.2 Application of FEM method in modeling of powder compaction

As explained in the first chapter, to model a process, FEM discretizes the material into a finite number of elements, which is called “meshing”. If a phenomenological model is used to simulate



a

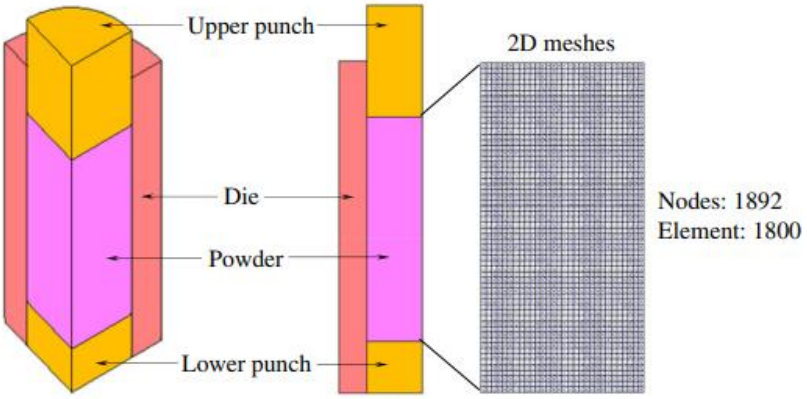


b

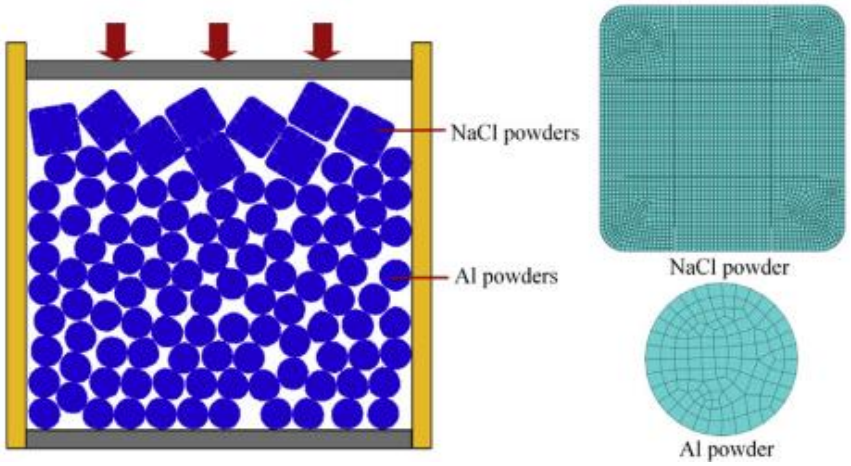
Figure 2.1 Two common approaches in powder compaction process: a) phenomenological approach [46], b) micromechanical approach [47].

powder compaction, the whole material must be meshed (Figure 2.2 a), and if a micromechanical model is used, each of the powder particles must be meshed independently (Figure 2.2 b).

The second step in the FEM modeling of the powder compaction process is to use a constitutive model (material model) that expresses the physical behavior of the powder during compaction as a mathematical relationship [48]. Due to the unique advantages of the DPC material model, most



A



B

Figure 2.2 Meshing process in FEM modeling of powder compaction: a) phenomenological approach [49], b) micromechanical approach [50].

of the research related to FEM modeling of powder compaction has been conducted through the implementation of this model for different powders. Wang [51] tried to understand the mechanical behavior of powders during the compaction process and perform calculations to control the design and manufacture of the required dies by combining a material model with FEM. In fact, Wang implemented the FEM approach based on a time-dependent material model and reported that the results obtained by the proposed method were in good agreement with the experimental results.

Shin et al. [52] determined the parameters of the DPC model by considering the barreling of the sample with the help of simulation. Based on their results, the stress path of the barreled sample during the triaxial test is dependent on the dimensions of the sample. Their proposed solution was

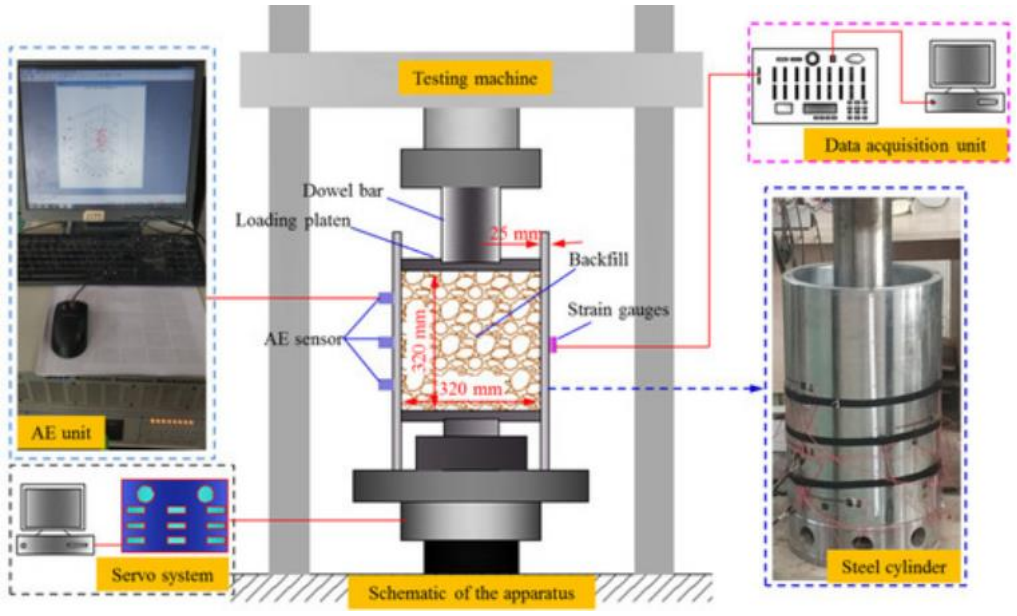
to correct the inelastic volumetric strain-pressure relation obtained from the triaxial compression test at high pressures.

2.3.3 Optimization-based calibration of the material coefficients in the DPC model

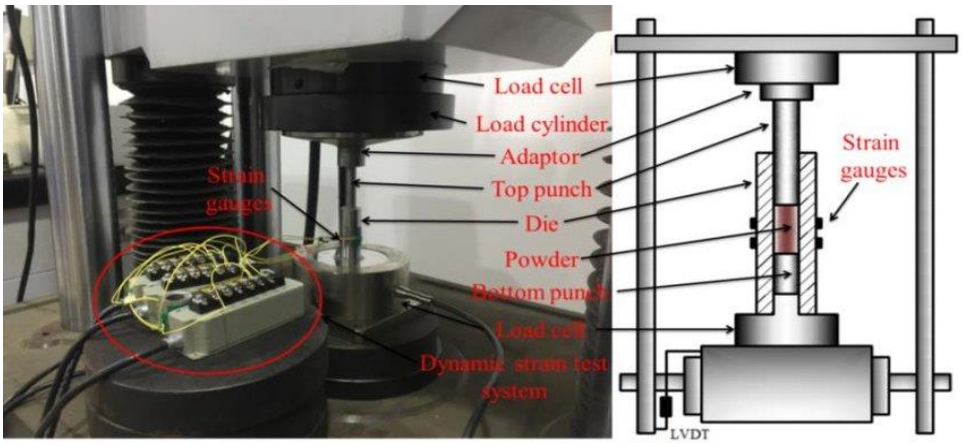
The experimental methods of determining the coefficients used in the DPC material model include triaxial and uniaxial compression tests. The triaxial compaction test (Figure 2.3 a) requires using of a specific die equipped with several types of sensors as well as an acquisition system for real-time data recording, which makes model calibration a complicated and difficult matter [53]. On the other hand, calibration of the DPC model using the uniaxial compression test requires special equipment called instrumented die (Figure 2.3 b) as well as performing numerous experiments for different levels of density and recording the data measured by the sensors at the same time. Also, this method requires performing several types of post-compaction tests on the ejected specimens, which makes its use time-consuming and expensive [55].

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 [56-57] developed a method to determine material parameters for the DPC model by a combination of uniaxial compression test, simulation, and optimization methods for metal powders. The PSO was initially noticed by mathematicians and computer scientists, and after some modifications by them, it attracted the attention of researchers from various scientific fields, so that recently it has been used in very wide and diverse areas such as estimation of the total number of confirmed COVID-19 cases [58] and computer science [59].

The reason why the PSO is widely accepted by researchers is the fact that 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), which makes it simple and efficient [60]. These features have caused overcoming the limitations of the PSO algorithm and introducing new modified versions of it become one of the attractive topics for researchers in different fields [61]. As an example, Zhang et al. [62] proposed a modified version of the PSO to enhance the performance of the algorithm in multi-objective optimization problems. They replaced the global learning strategy with a dynamic neighborhood-based one to enhance the diversity of the particles in the feasible space. Also, they used a competitive mechanism between particles to avoid getting



A



B

Figure 2.3 Experimental methods for determination of material coefficients in the DPC model: a) triaxial compaction test [63], b) uniaxial compaction test [64]

trapped in local optimal. In another study, Li et al. [65] proposed a multi-population cooperative particle swarm optimization algorithm. They employed a dynamic segment-based mean learning strategy to construct learning exemplars, achieves 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.

Among the material models presented 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 effective parameters and the accuracy of the predicted results, so that today it has 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. 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. In this regard, Hrairi et al. [66] studied the FE simulation of die compaction of metal powders using the DPC model implemented in Abaqus software along with an inverse optimization algorithm to calibrate the coefficients. They defined the difference of the density distribution data between the prediction of the FE simulation and the experiment as the objective function for the optimization of 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 maximum absolute error of 2.3% between densities. Majzoobi and Jannesari [67] used this method to calibrate the coefficients of DPC model for aluminum, iron and copper powders. They used the powder force-displacement curve obtained from the compaction test as the experimental data. Also, they performed the numerical simulation of powder compaction using Ls-Dyna FE software. In their research, the difference between numerical and experimental force-displacement curve was defined 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 use of this method is not suitable for copper powder. Figure 2.4 shows a schematic of the combined experimental/numerical/optimization method for calibrating DPC coefficients.

Buljak et al. [85] 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 determine 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, while, at the

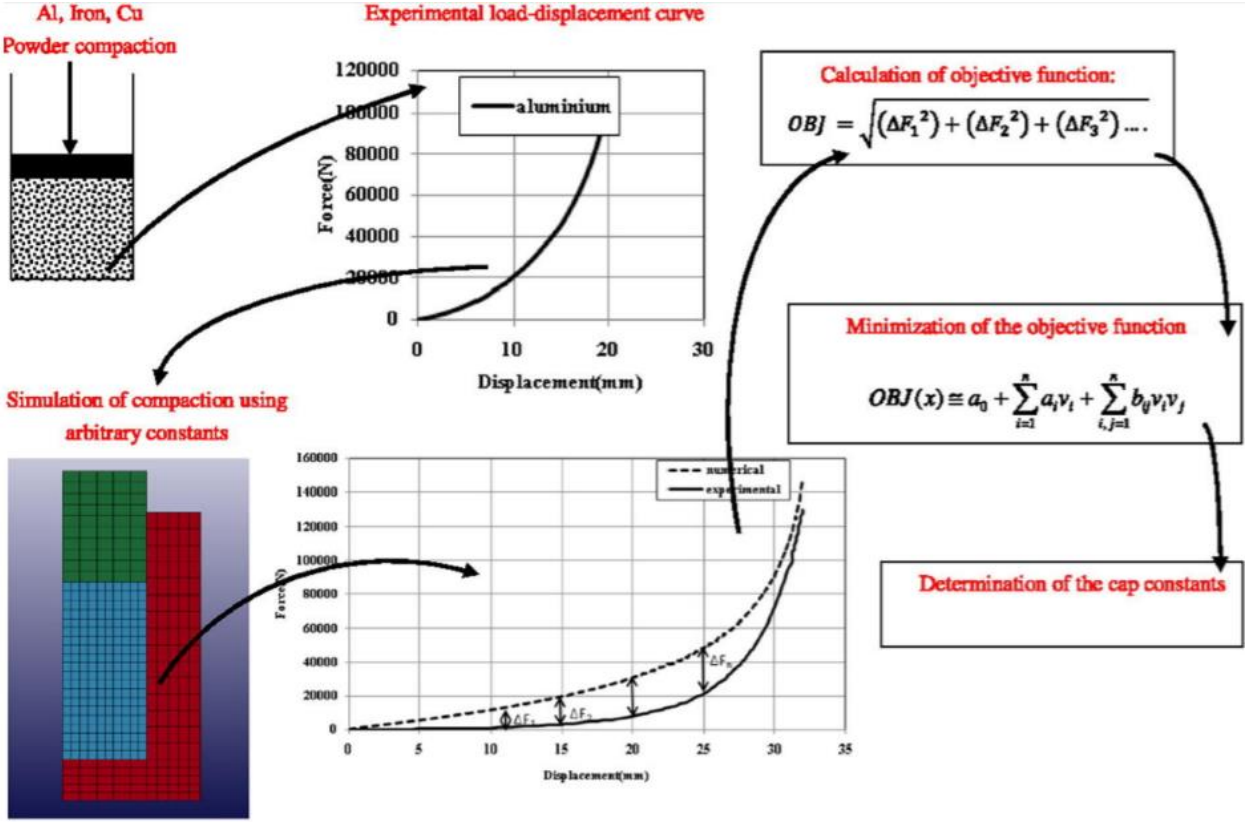


Figure 2.4 Schematics of calibration of the DPC coefficients using experimental/numerical/optimization method [67]

same time, it provides data for a wider range of relative densities [68]. Zhou et al. [69] proposed accuracy an integrated method of modeling using the inverse optimization to simulate the compaction process of metal powders. They mentioned avoiding numerous tests as well as high accuracy among the 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 considered density-dependent to achieve sufficient.

Chapter 3:
Materials and Methods

3.1 Introduction

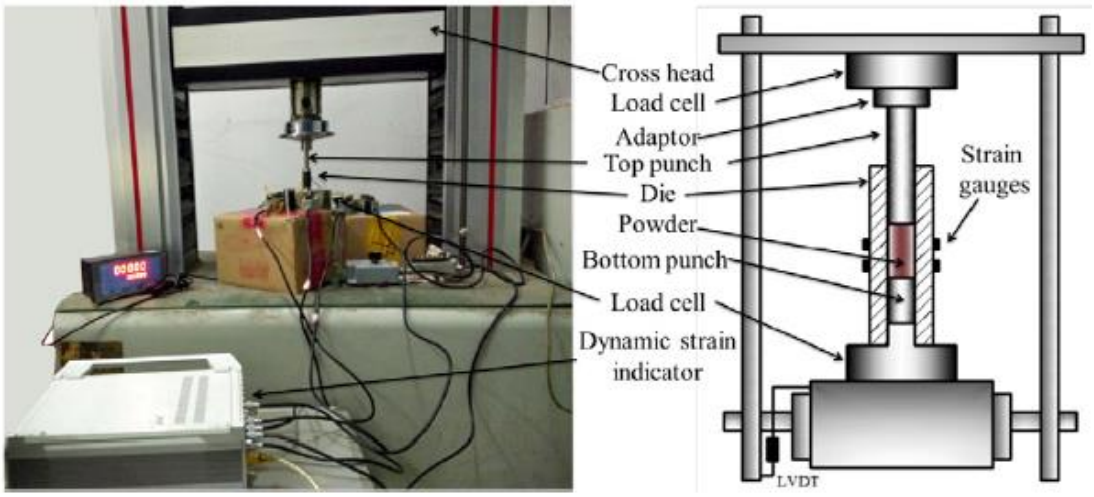
The primary goal of this thesis is the calibration of the coefficients of the DPC material model. Due to the fact that in the DPC material model, most of the characteristics of powder particles such as adhesion, displacement, deformation, friction, hardening, and volumetric plastic strain are taken into 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 a lot of attention due to avoidance of the experimental determination of coefficients of material models. In this chapter, the details of this method are fully explained.

3.2 Experimental Data

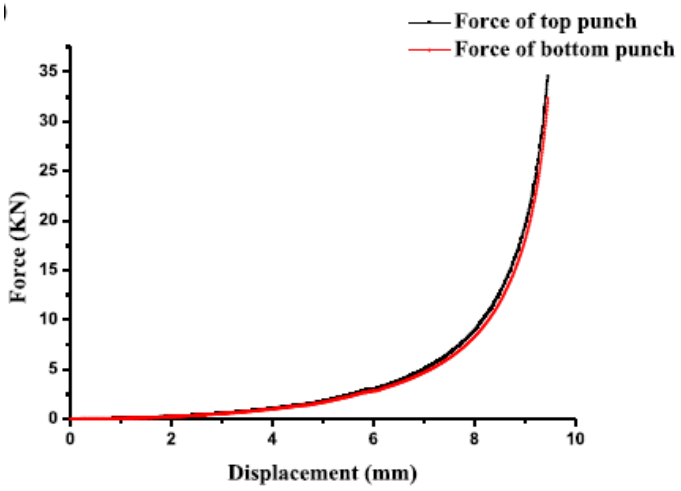
In this research, we are trying to determine the coefficients of the DPC model using the proposed method for a mixed metal powder. The selection of the powder has been done according to its importance and application in the production of industrial parts based on powder metallurgy route. For this purpose, Ag57.6-Cu22.4-Sn10-In10 (ACSI) has been selected as the mixed metal powder. The force-displacement curve of ASCI powder has been measured experimentally by Zhou et al. [55] and is used in this research. They measured this curve from the uniaxial compression test into an instrumented cylindrical die as shown in figure 3.1. As one can see, the compaction curve of a powder can be easily measured by performing a simple uniaxial compression test, while experimental determination of DPC material constants requires a large number of compression tests for different densities so it is not time and cost-effective at all. 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.

3.3 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, FEM has gained a special place in scientific and industrial centers as a powerful modeling tool. FEM is taught as a prerequisite in many engineering courses, and its mastery is considered one of the most important skills for engineers. Today, the FEM has been implemented in many commercial software in which users can simulate their desired problem



a



b

Figure 3.1 Experimental measurement of the force-displacement curve for ASCI powder [70]: a) instrumented die under uniaxial compaction test, b) force-displacement data of the powder compact.

in different scientific fields. Student FE software package Abaqus is a powerful, yet simple, method for investigating the structural modeling of powder compaction processes. Accessibility and ease of use make Abaqus a suitable software for our purpose. As explained earlier, FE simulation of a process has different steps. In the next sections, the FE simulation performed by Zhou et al [55] will be repeated and the compaction curve obtained from the developed FE model is compared with the experimental data to check the validity of the FE modeling in the present

study. After that, the quantified features are transferred to a customized Python-based modeling framework.

3.3.1 Part module

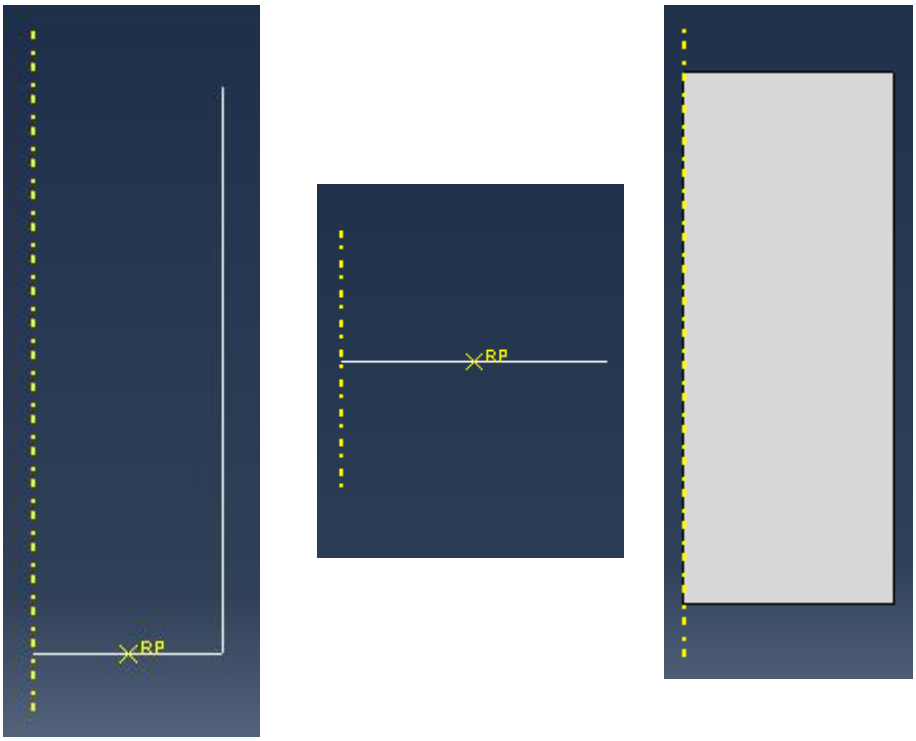
The first step in the FEM simulation of physical processes is the geometric modeling of the parts that make up the model in the part module. The FE model of the powder compaction process usually consists of three parts including the die, punch, and powder. Figure 3.2 shows the geometric modeling of these three parts in Abaqus. A part can be modelled as deformable or discrete rigid. Also, the axisymmetric model means that the model has symmetry both in terms of geometry and loading, so in this case, the model can be sketched in 2D with respect to a symmetry line. Table 3.1 lists the type of modeling of the three parts as well as their dimensions. Also, in Abaqus, each part of the discrete rigid type is introduced with a reference point, which is used to assign different properties and boundary conditions to the part.

3.3.2 Assembly module

After the geometric modeling in the part module, we must determine how to arrange the parts in the assembly module. For this purpose, we first create an instance from each part so that from now on all operations are performed on the instances. Figure 3.3 shows how the three parts are placed next together in the assembly module. Also, by enabling the dependent check box, any changes on the instances will be applied to the corresponding part.

3.3.3 Step module

After the assembly of the instances, we need to specify the type of analysis in the step module. Abaqus has two solvers for solving structural problems. These solvers are Abaqus/Standard and Abaqus/Explicit. Each of these solvers has its own steps. When you select the desired step in the step module, you are in fact selecting the solver. The main difference between these two solvers is that the Abaqus/Standard solver uses implicit methods to solve problems while the Abaqus/Explicit solver uses explicit methods. Figure 3.4 shows the step manager window in Abaqus. As can be seen, the process starts with a default initial step and continues with step-1 of the general static type which is used for quasi-static problems. Also, the settings related to the incrimination of the problem are shown in the edit step window. After receiving this information from the user, Abaqus divides the whole problem into small increments and starts solving the problem in each increment.



a b c

Figure 3.2 Geometric modeling of three parts of the powder compaction process: a) die, b) punch, c) powder

Table 3.1 Geometric modeling of the punch, die and powder in part module

Part	Modeling Space	Type	ASCI Powder [55]	
			Diameter (mm)	Height (mm)
Punch	Axisymmetric	Discrete Rigid	10	16.02
Die	Axisymmetric	Discrete Rigid	10	17
Powder	Axisymmetric	Deformable	10	15.82

Another setting that is determined in the step module is the output variables that we intend to analyze at the end of the simulation. Since in this research we are looking to compare the force-displacement curve obtained from FEM simulation with the experimental data, in this part we call the force and displacement for the reference point defined on the punch according to Figure 3.5.

3.3.4 Interaction module

In this module, the type and properties of the contacts between the surfaces of the model are defined. The powder compaction FE model includes 2 interactions which are shown in figure 3.6.

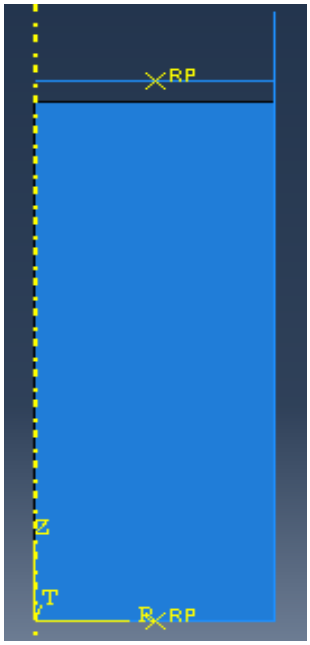


Figure 3.3 Arrangement of three parts in the assembly module

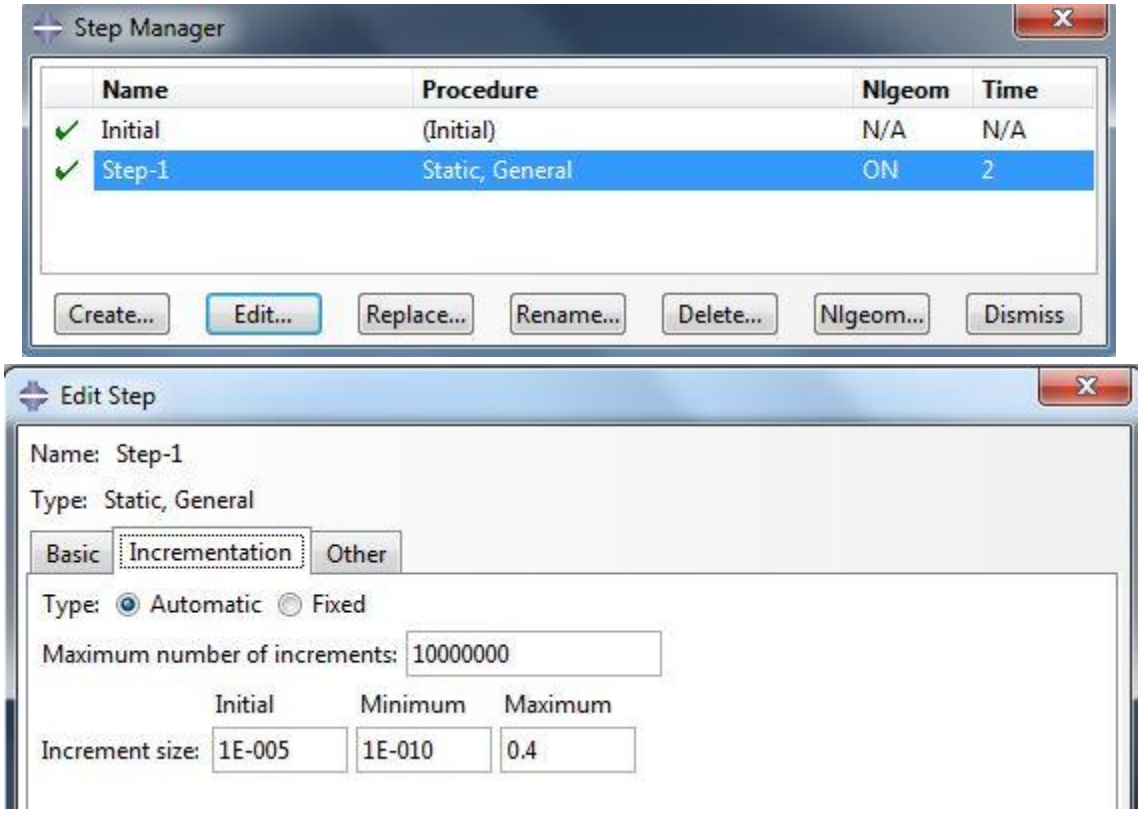


Figure 3.4 The step module settings

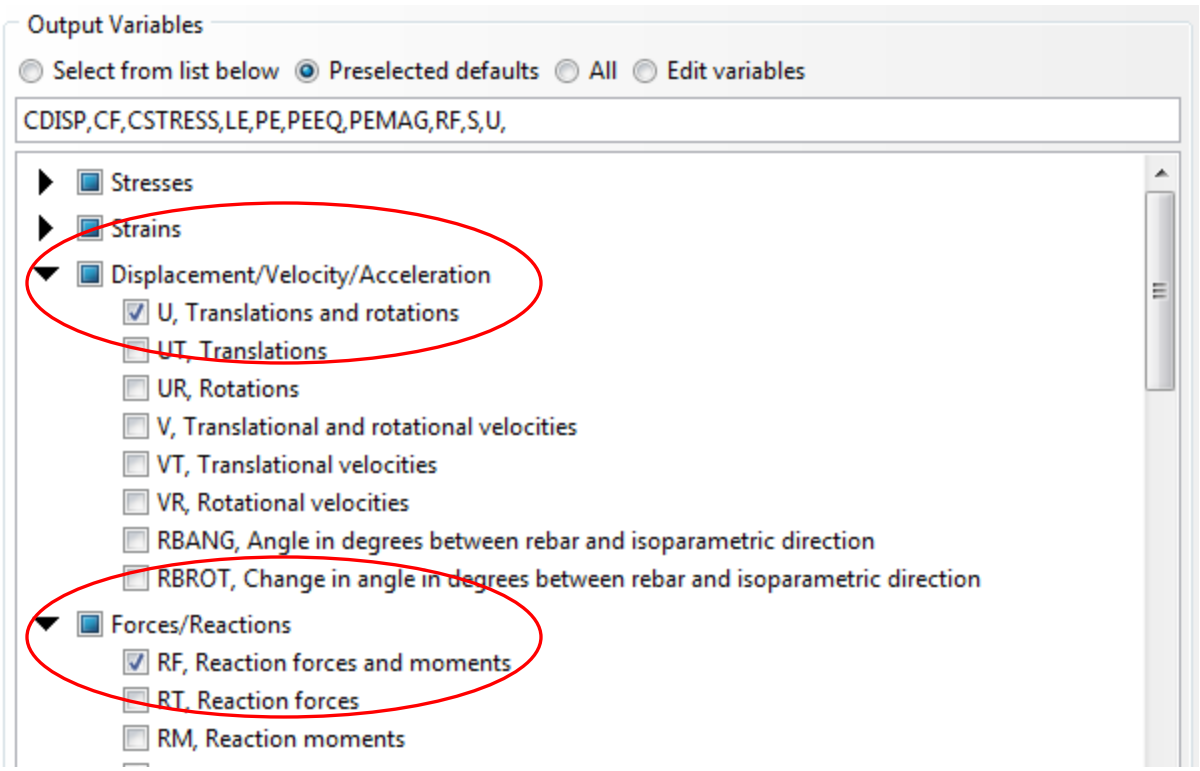


Figure 3.5 Selection of force and displacement of the punch as output parameters in the step module

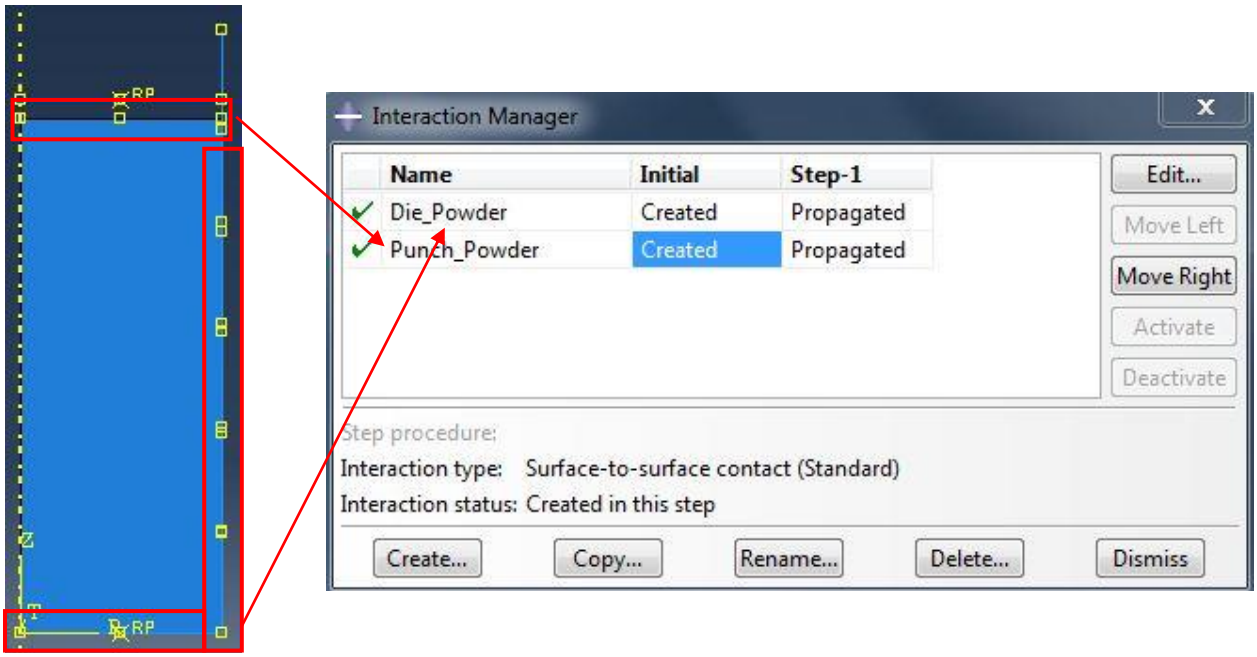


Figure 3.6 Defining the contacts between the surfaces of the model in the interaction module

As can be seen, these two interactions are created in the initial step and propagated in the stap-1. Both interactions are surface-to-surface contacts with tangential behavior which means two surfaces have a relative motion (sliding) with respect to each other. The friction between the surfaces is also defined using the penalty formulation [70] with a friction coefficient of 0.8.

3.3.5 Load module

The next step in the FE simulation of the powder compaction process is to apply boundary conditions to the model, which is done in the load module. Figure 3.7 shows the powder compaction model after completing load module. In this 2D model, the die must remain fixed during compaction, and therefore all its degrees of freedom are closed. In other words, the displacement in the x and y directions as well as the rotation around the z axis all are equal to zero. Also, the punch is only allowed to move vertically (in the y direction). 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.

3.3.6 Mesh module

After performing all the previous steps in the FE simulation, the model should be discretized into small elements by the meshing process, which is done in the mesh module. Figure 3.8 and Table 3.2 show the meshed model and the details of the meshing for the powders. Abaqus has a library of different elements that in the analysis of FE problems, the user must choose the appropriate type of element according to the nature of the problem. As can be seen, CAX4R (4-node, axisymmetric solid element) is used for the powder and RAX2 (2-node, axisymmetric rigid element) is used for both the die and punch.

An important point in meshing the FE model is that the more the number of elements, the more the time to solve the problem. Therefore, the number of elements generated in the meshing process should be such that while obtaining an accurate solution, the time to solve the problem is also reasonable. This is done through a process called mesh sensitivity analysis, in which the number of elements in the problem is gradually increased and the degree of convergence to the final solution is checked each time [71]. The optimal number of elements is obtained when increasing more elements does not have much effect on the convergence of the results. Since the mesh sensitivity analysis for ASCI powder has been done and the optimal number of elements is reported by Zhou [70], repeating this process is avoided here.

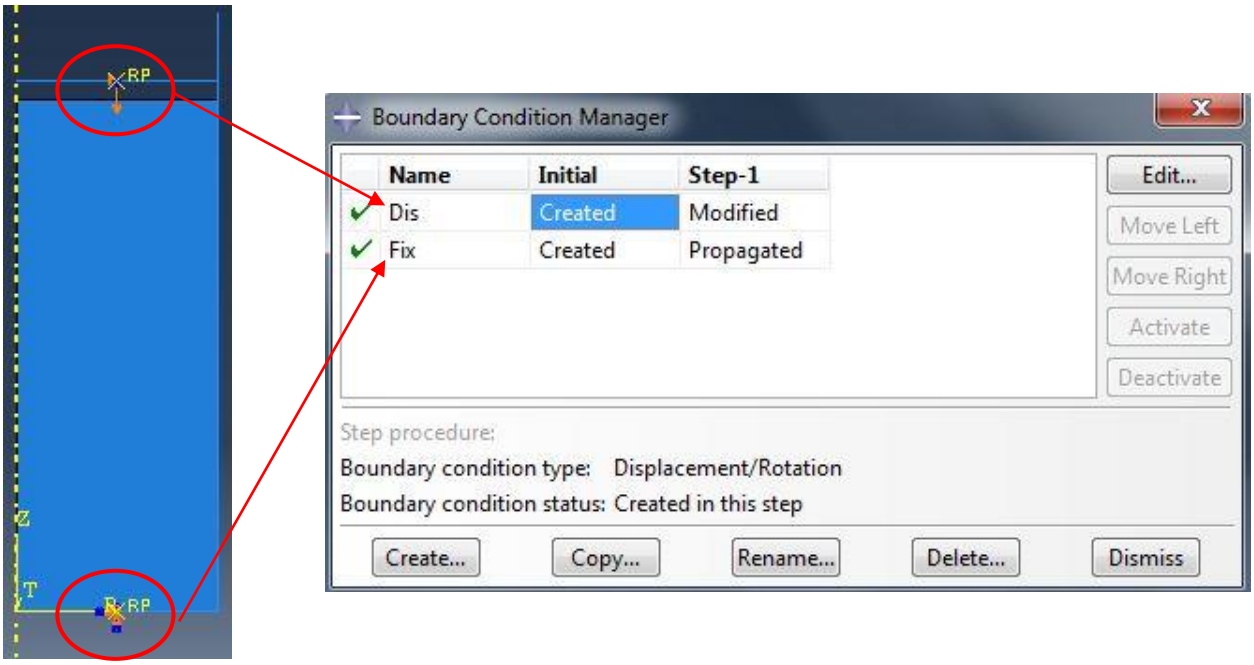


Figure 3.7 Defining the boundary conditions of the model in the load module

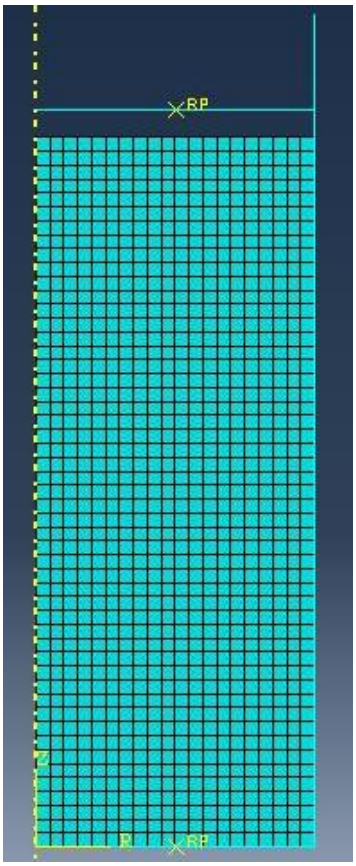


Figure 3.8 Meshing of the model in the mesh module

Table 3.2 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
Element Type	CAX4R	RAX2	RAX2
Element Number	1020	80	20

3.3.7 Property module

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, which are called "material properties" in Abaqus. So, in the property module, we actually have to specify the material properties of the model. The important thing is that Abaqus does not have a library of materials and their properties, and the definition of the material is completely up to the user. For example, if our part is made of aluminum, we cannot choose aluminum in Abaqus because there is no predefined material. Therefore, we must define a material and enter the properties of aluminum for it and then assign it to the part. However, each material has many properties according to the type of problem we are facing, we must only provide the required material properties of the selected model.

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. In the next section, the formulation of the DPC model implemented in Abaqus and the related material properties are explained.

3.3.7.1 Formulation of the DPC material model in Abaqus

The DPC model includes three parts of the yield surfaces according to figure 3.9. 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 3.10). Equations (3.1) to (3.3) show the formulation of the three yield surfaces in the DPC model implemented in Abaqus.

$$F_s = q - p\,tan\beta - d = 0 \tag{3.1}$$

$$F_c = \sqrt{(p - p_a)^2 + \left[\frac{R\,q}{1 + \alpha - \alpha/cos\beta}\right]^2} = 0 \tag{3.2}$$

$$F_t = \sqrt{(p - p_a)^2 + \left[q - (1 - \frac{\alpha}{cos\,\beta})(d + p_a\ tan\,\beta)\right]^2} - \alpha(d + p_a\ tan\,\beta) = 0 \tag{3.3}$$

where q and p are the Mises equivalent stress and the hydrostatic pressure stress, respectively.

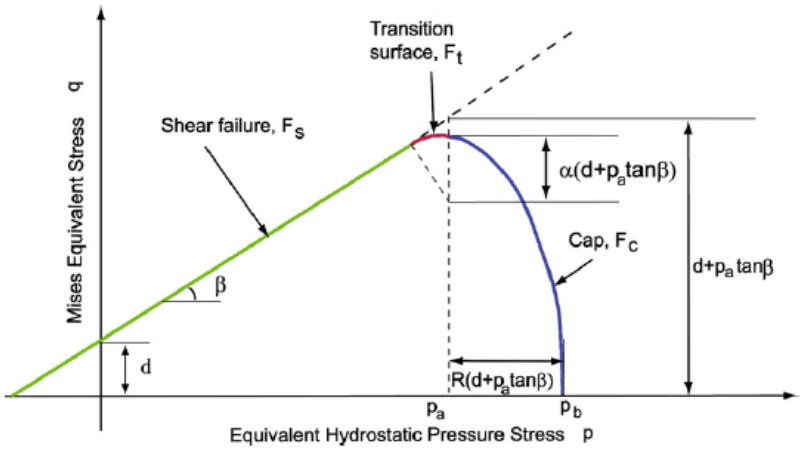


Figure 3.9 Three yield surfaces of the DPC material model implemented in Abaqus

Young's modulus (E) and poisson's ratio (v) are two material properties that determine the elastic behavior of powder during the un-loading 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 define 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 [55]:

$$p_a = \frac{p_b - Rd}{(1 + R\ tan\beta)} \tag{3.4}$$

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 [55]:

$$p_b = f\left(\varepsilon_v^{pl}\right) \tag{3.5}$$

Figure 3.10 shows the material properties windows related to the elastic, cap plasticity, and cap hardening in the property module. One can see the FEM simulation of the powder compaction process using the DPC material model in Abaqus software requires the determination of ten material properties ($E, v, d, \beta, R, \varepsilon_0, \alpha, K, p_b, \varepsilon_v^{pl}$). As explained earlier, for FE simulation of the compaction of ASCI powder using DPC, the experimental data provided in [55] has been used as the material properties. It should be noted that in computer science, the physical concept of parameters affecting the behavior of the system is not very important and we only seek to identify these parameters and consider the most important of them as optimization variables.

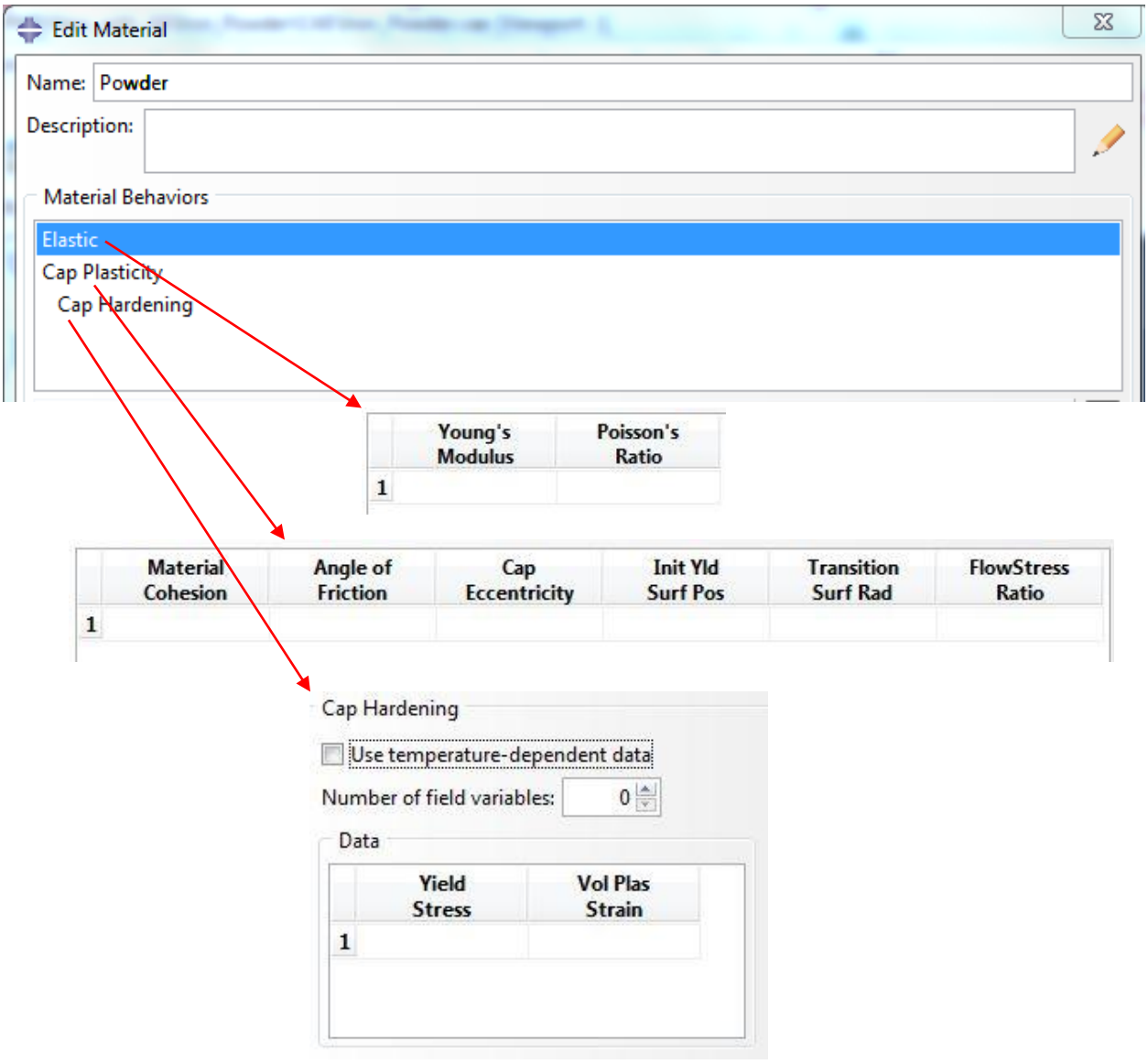


Figure 3.10 Elastic, cap plasticity, and cap hardening material properties in the property module

3.3.8 Job module

The last step in the FE simulation in Abaqus is completed in the Job module. This module is responsible for solving the problem and processing it in Abaqus software. In fact, before this module, we have completely specified the problem for the Abaqus software while in the job module, we are going to solve the problem. The job module also provides the possibility of creating an input file as well as calling a subroutine file for the user. The input file is a text file in which all the specifications related to the FE model are written, so it can be processed by the software as a complete model. Also, subroutine files enable the user to introduce some settings to the software through that are not possible in modules. For example, in the powder compaction process, the density of the powder is changing at every moment of the process, and on the other hand, the material properties are also dependent on the density (their values changes with the density). For this purpose, in this research, a USDFLD subroutine has been written in Fortran compiler, in which the property module is defined as the field variable, and then its value is updated in each solving increment, according to the following equation:

$$\rho = \rho_0 \exp\left(\varepsilon_v^{pl}\right) \tag{3.6}$$

where ρ is the current relative density, and ρ_0 is the initial relative density of filling powders in the die [55]. It should be noted that before writing USDFLD subroutine that updates material properties in each increment based on the field variable value, Abaqus software needs to be linked with Visual studio and Fortran compiler software.

3.3.9 Validation of the FE model

After preparing the FE model of the ASCI powder compaction process according to the previously explained steps, in order to ensure the accuracy of the model, we compare the results of our FE simulation result with the experimental data reported by Zhou [55]. Figure 3.11 compares FE and experimental force-displacement curves. The curve obtained from the FE simulation has accurately predicted the experimental data which validates provided FE model of ASCI powder. This shows that the finite element modeling of the powder compaction process has been done correctly. Therefore, we can use it in the PSO algorithm to calibrate DPC parameters.

3.4 PSO Optimization Algorithm

PSO is one of the most important intelligent optimization algorithms and has been widely used in

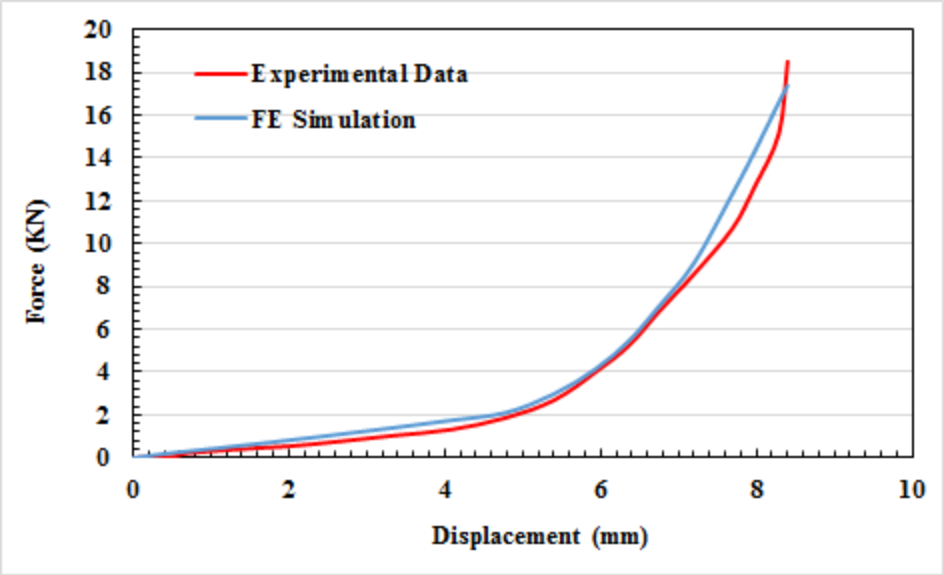


Figure 3.11 Force-displacement of the ASCI powder

a wide range of scientific fields. Despite its simplicity, PSO has a great ability to find the optimal solution. PSO is sometimes classified as an evolutionary algorithm because the modifier mechanism is repeating itself and introduces a new population based on the information sharing process [72]. In another classification, PSO is considered as one of the "swarm intelligence" optimization algorithms. These algorithms search for the optimal solution with collective cooperation of members and use a mechanism called "self-organization" which controls the individual and social search process in each iteration [73]. 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 group of fishes when they faced with the threat of a hunter. Through their swarm intelligence, they are divided into several groups (swarms) when a hunter attacks and then gather together again after the danger is vanished. In this algorithm, the two factors of information exchange (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 main role [73].

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 as one of the potential solutions to the optimization problem. Particles start moving in the feasible space to search for the optimal solution, so that each particle in its movement 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 3.12, in each iteration, the previous position of particle is updated according to the following equations [74]:

$$V_i^{k+1} = [\omega V_i^k] + [c_1 r_1 (P_i^k - X_i^k)] + [c_2 r_2 (P_g^k - X_i^k)] \quad (3.7)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (3.8)$$

where subscript k denotes the number of the current iteration, V_i^k and X_i^k are the velocity and the current position of the i[th] particle, P_i^k and P_g^k are the best previous position and of the i[th] 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 which controls the weight of the previous velocity on the current one). It is worth mentioning that P_i^k and P_g^k are calculated iteratively by the code, using the error obtained in each iteration, which will be further explained. The first term in equation 3.7 (ωV_i^k) searches new solutions and find the regions with potentially the best solutions. The parameter ω is important for balancing the global search. It makes the

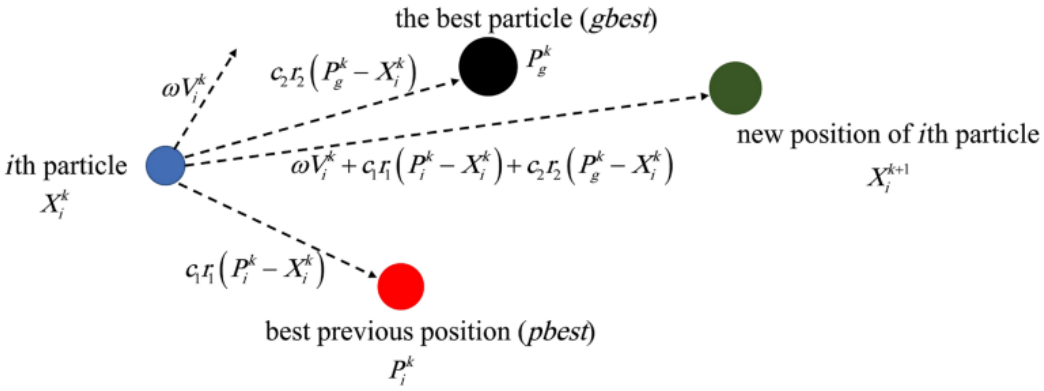


Figure 3.12 The schematics of updating the position of ith particle in kth iteration in the PSO optimization [60]

particle move in the same direction and with the same velocity. When higher values are set for ω, it known as exploration while when the lower values are set it known as exploitation [91]. The second and third terms explore the previous solutions and find the best solution of a given region. The second term ($c_1 r_1 (P_i^k - X_i^k)$) represents the effect of personal experience of each particle. So, it makes the next position of the particle better than the current. The third term ($c_2 r_2 (P_g^k - X_i^k)$) represents the effect of social experience of neighbors and makes the particle to follow the best neighbors directions.

3.5 Calibration of the DPC parameters using inverse optimization analysis

In this section, we will 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. For this purpose, we only need the force-displacement curve of the powder which is easily obtained through a routine uniaxial die compression test. Then, we use this experimental data along with the corresponding force-displacement curve obtained from the FE simulation as the inputs in the PSO optimization algorithm (see figure 3.13) scripted in Python.

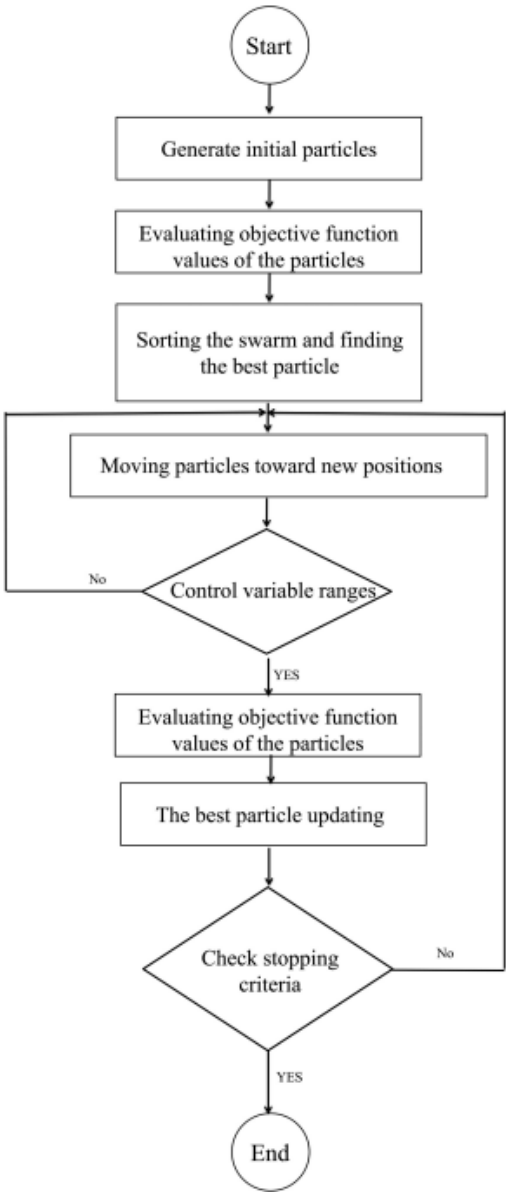


Figure 3.13 The flowchart of searching process in the PSO optimization [91]

3.5.1 Sensitivity analysis

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

By performing a sensitivity analysis on the DPC parameters, Majzoobi [52] reported that Young's modulus ($0.01 < E < 120$), material cohesion ($0.01 < d < 50$), and hydrostatic pressure yield stress ($0.1 < P_b < 250$) is strongly dependent on the density. Therefore, all three parameters are fitted by an exponential function as suggested by Zhou [55]. 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 are considered for these parameters [55]. 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 [55]. Table 3.3 lists DPC model parameters as optimization variables along with their fitted relations. As you can see, to calibrate the DPC model, we need 9 unknown coefficients whose optimal values should be determined. 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 randomly 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 important point.

3.5.2 Scripting PSO optimization algorithm in Python

To calibrate DPC model parameters using optimization, we first need to script the PSO algorithm in Python. In the following, we will explain the PSO optimization code step by step in detail.

Table 3.3 Defining the DPC model parameters as the PSO optimization variables

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

Step 1: we start the optimization process by importing required modules to the code.

Step 2: we get the current directory and assign it to the classes directory.

Step 3: we import the required classes each of which is called for a specific purpose during the execution of the code. For example, an updater class updates DPC parameters in each iteration.

Step 4: we create a folder named "Results" to save the outputs and its existence is checked.

Step 5: In this part of the code, first, a default matrix (including the optimization variables E_1, E_2, d_1, d_2, P_1, P_2 that must be updated by the PSO) and the experimental data of the force-displacement curve are defined as the global variables. Then, the function $f(x)$ is defined based on the global variable (default matrix) and its components are specified.

Step 6: As explained earlier, one of the capabilities of Abaqus 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 by importing it into Abaqus, the model can be created and solved directly without the need for additional operations. Here we use this feature of the input file for our purpose in such a way 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, first

the input file named “Validation” is called. Then, it is opened, and its content is read line by line (line 60).

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 reaching this section which is titled by a specific header in the input file, Python generates new coefficients using the PSO algorithm and writes them in the new input file. For example, in the case of cap hardening parameter (P_b), when the code reaches to the line titled “Cap Hardening”, the first and last lines related to P_b are specified and then its values are updated from the updater class 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 end of this part of the code, 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” calling the required class. This 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 of the updated input file in ith iteration, it generates an output file called “odb” file containing the obtained results. Here, first a copy of odb file is made and the content of this file is read line by line. Then, the code save 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 or not. For this purpose, we consider a certain 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 file with the format of csv. Then, this file is opened and its content including the time intervals and their corresponding forces and displacements are read.

Step 10: Here, we call the force-displacement curve extracted from the punch. At this step, these values are compared with the corresponding experimental values at certain points and the objective function is defined as follows (see figure 3.14):

$$OBJ = \sum_{i=1}^{N} (y_i^{opt} - y_i^{exp})^2 \tag{3.9}$$

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:

$$error = \sqrt{\frac{OBJ}{N}} \tag{3.10}$$

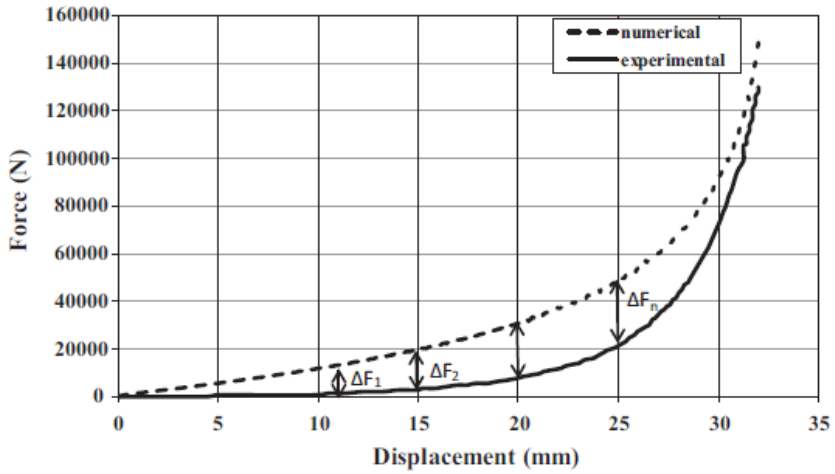


Figure 3.14 Comparing the corresponding experimental and numerical values at certain points on the force-displacement curve to define the objective function [81]

Step 12:

First, an initial population of particles is randomly generated and then each particle is randomly assigned a velocity and a position to determine the secondary position of the particles according to equation 3.7. Then, the objective function is calculated for each particle to determine the values of the personal and global best for the particles. Next, if the new position of the particle is improved compared to its previous position, its velocity and position are update according to the equations 3.7 and 3.8 respectively and at the same time, they are compared with the range of the feasible space.

Step 13: At the end of the optimization code, the best position and the best error for the group of particles are determined and based on them, the optimization loop produces particles with a new position.

3.6 Summery and the workflow of the project

Here as it is illustrated schematically in Fig. 3.15, the workflow of the computational tool contains three major sections for getting and processing data: the structural modeling part with sufficient design parameters like initial density, weight of punch, its speed and etc. to implement a precise structural model of powder compaction process. The remaining two parts of the computational tool, material modeling (DPC material model) and tuning sections (PSO part), execute in parallel to generate the best representative model of material properties (calibration of DPC material model parameters) of powder particles.

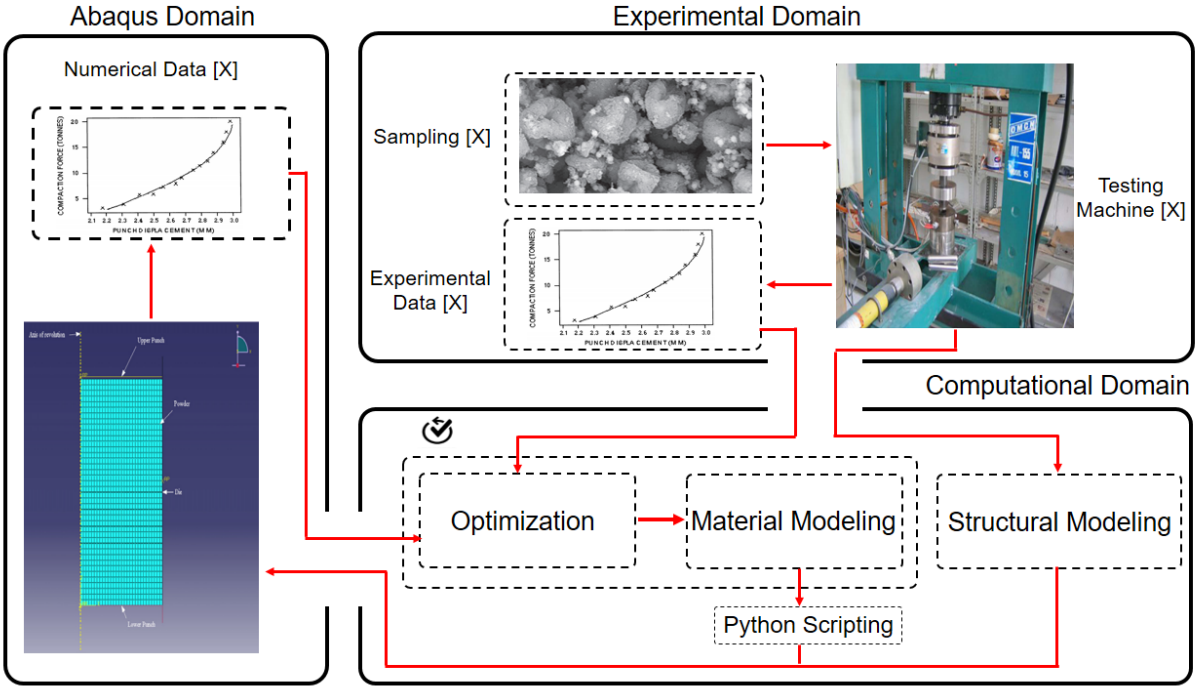


Fig. 3.15 Workflow of the project

Chapter 4:
Results and Discussion

4.1 Introduction

In this chapter, the results of the method described in the previous chapter are fully presented and discussed. In other words, the results of the calibration of DPC model parameters by the inverse optimization method are presented and compared with the experimental results provided by Zhou et al [55]. These parameters (i.e. Young's modulus (E), material cohesion (d), and the hydrostatic pressure yield stress (p_b)) are actually the same optimization variables that were introduced in the previous chapter.

4.2 Optimized DPC model parameters

Table 4.1 presents the optimized coefficients. By placing these coefficients in relations 4.1 to 4.3, it is possible to plot the curves of E, d, and p_b respectively.

Table 4.1 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

$$E = E_1 * \exp(E_2 * \rho) \quad (4.1)$$

$$d = d_1 * \exp(d_2 * \rho) \quad (4.2)$$

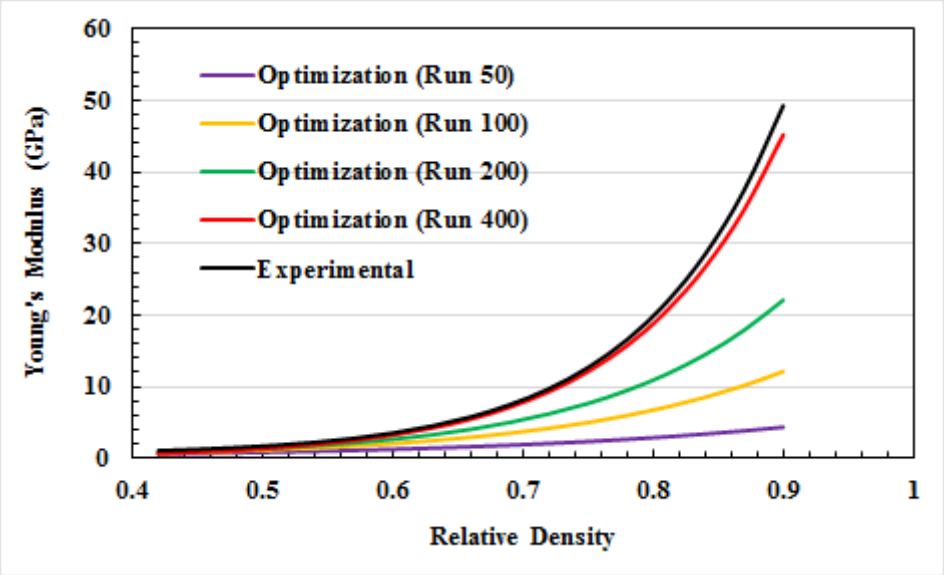
$$P_b = P_1 * \exp(P_2 * \rho) \quad (4.3)$$

On the other hand, the results presented in Table 4.1 prove that the method presented in this research has succeeded in providing certain 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 used, which is calculated according to equation 3-10.

4.2.1 Young's modulus

Figure 4.1 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 4.1. As can be seen, the

Figure 4.1 Comparison between the optimization and experimental results for E



accuracy of the Young's modulus curve can be improved significantly by increasing the number of iterations. At 400th iteration, Young's modulus curve resulting from placing the optimal coefficients (E_1=16.71, E_2=8.78) in relation 4.1 has been able to predict the experimental curve with good accuracy (RMSE=1.95).

4.2.2 Material cohesion

Figure 4.2 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=0.91×10^{-4}, d_2=13.08) are placed in equation 4.2. It can be seen again that the calibration of this parameter using the inverse optimization method has succeeded in predicting the experimental values with an excellent accuracy (RMSE=0.12). According to the explanations presented in chapter 3, material cohesion is a parameter that determines the q-axis intercept of the shear yield

line in the DPC model, and as a result, it affects the mechanical behavior of the powder during compaction. It can be seen again that the increase in the number of iterations of the optimization loops has increased the accuracy of the material cohesion values predicted by the PSO algorithm.

4.2.3 Hydrostatic pressure yield stress

Figure 4.3 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=0.53 and P_2=0.53 in equation

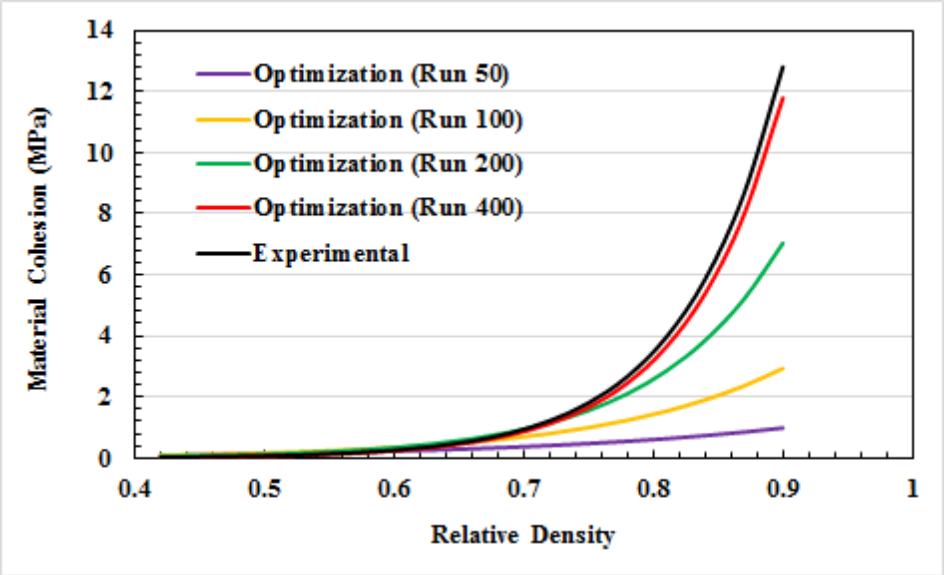


Figure 4.2 Comparison between the optimization and experimental results for d

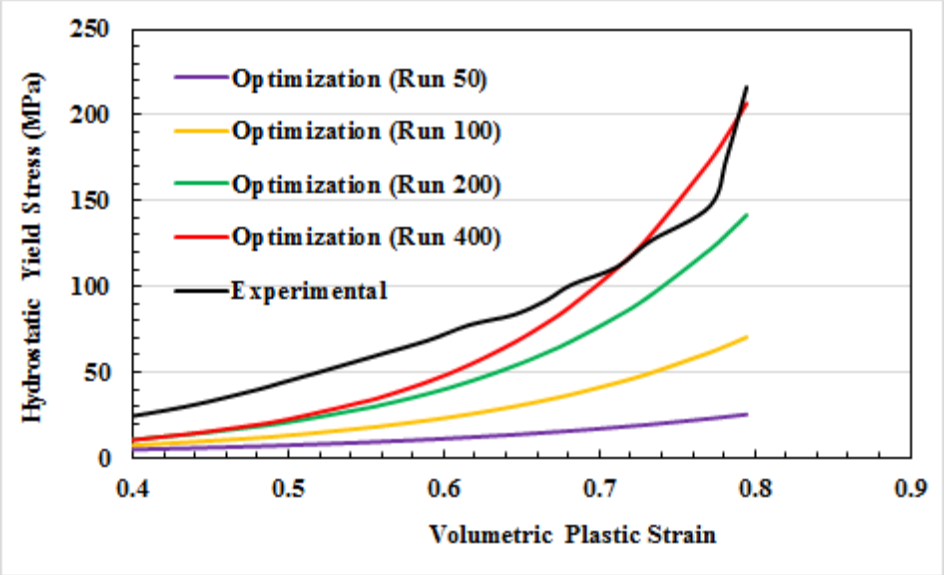


Figure 4.3 Comparison between the optimization and experimental results for P_b

4.3. 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 important parameter of the DPC model, which shows the hardening behavior of the powder during the compaction process. In a process similar to E and d, it can be seen that increasing the number of iterations has led to an increase in the accuracy of the hydrostatic pressure yield stress.

4.2.4 Other parameters

According to the sensitivity analysis (see section 3.5.1), in the case of parameters that do not have much effect on the mechanical behavior of the powder, their default values or the values reported in other references have been used. Finally, Table 4.2 presents the final values of the DPC model parameters for ASCI powder. It should be noted that in Table 4.2, E, d and p_b parameters are calibrated using the method presented in this research.

Table 4.2 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

Chapter 5: Conclusion and Suggestions

5.1 Introduction

The great importance of the powder compaction process as a promising method in the powder metallurgy, pharmaceutical, detergents, cosmetics, and ceramic industries, on the one hand, and the difficulty of experimental determination of DPC model parameters, as the most widely used material model in the FE simulation of the process, on the other hand, have caused that the calibration of the model using more convenient methods is considered as a challenge for the researchers. In this study, the calibration of DPC model parameters for ASCI alloy powder, which has recently been widely used in various industries due to its excellent properties, was considered. In other words, computer science (development of a Python-based code to establish the connection between the PSO algorithm and the finite element model of the industrial powder compaction process) and mechanics of materials (the DPC material model that captures the mechanical behavior of powder particles during the compaction) are combined in the research.

In this chapter, the most important results obtained from the calibration of DPC model parameters using the PSO optimization algorithm are expressed, and in the following, suggestions are presented for the continuation of this subject in the form of future researches.

5.2 Conclusion

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

1- The simulation of the compaction process for ASCI powder was successfully carried out using Abaqus ™ and then the difference between the force-displacement curve obtained from the simulation and the experimental data was defined as the objective function in an optimization problem.
2- The PSO optimization algorithm was coded in Python and the link between the optimization code and Abaqus was successfully established.
3- The DPC model parameters are considered as the optimization variables and in each iteration, these variables are generated by the PSO and are considered as updated material properties in Abaqus. Then the FE model is solved by the software and the error is

calculated. According to the error value, PSO approaches the optimal solution based on previous experiences of individual particles as well as the.

4- 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 success of this study. 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 findings of this study can serve as a foundation for further research and advancements in this field.

5.3 Suggestions

Considering the successfulness of the method proposed in this research which leads to a reduction in the cost and time required to calibrate the DPC parameters as well as modeling of the powder compaction process with acceptable accuracy, the following suggestions are presented as potential subjects for future researches:

1- Application of the proposed method for other types of powders such as cosmetic, ceramic, detergent, and pharmaceutical powders in order to overcome the limitations of their compaction modeling.

2- Utilizing other optimization algorithms for calibration of DPC parameters and comparing their results with the PSO algorithm.

3- Using of modern intelligent modeling methods such as machine learning and artificial intelligence to model the powder compaction process based on the framework proposed in this study.

References

[1] Shin, H., Kim, J.B., Kim, S.J. and Rhee, K.Y., 2015. A simulation-based determination of cap parameters of the modified Drucker–Prager cap model by considering specimen barreling during conventional triaxial testing. *Computational Materials Science*, *100*, pp.31-38.

[2] Diwekar, U.M., 2020. *Introduction to applied optimization* (Vol. 22). Springer Nature.

[3] Deb, K., 2001. Nonlinear goal programming using multi-objective genetic algorithms. *Journal of the Operational Research Society*, *52*(3), pp.291-302.

[4] Bock, H.G. and Schulz, V., 2008. Mathematical Aspects of CFD-based Optimization. In *Optimization and Computational Fluid Dynamics* (pp. 61-78). Springer, Berlin, Heidelberg.

[5] Do Koo, Y., An, Y.J., Kim, C.H. and Na, M.G., 2019. Nuclear reactor vessel water level prediction during severe accidents using deep neural networks. *nuclear Engineering and Technology*, *51*(3), pp.723-730.

[6] Van Den Bergh, F., 2007. *An analysis of particle swarm optimizers* (Doctoral dissertation, University of Pretoria).

[7] Ali, M.M. and Kaelo, P., 2008. Improved particle swarm algorithms for global optimization. *Applied mathematics and computation*, *196*(2), pp.578-593.

[8] Holland H.Adaptation in natural and artificial systems. Second edition (first edition, 1975) ed. Cambridge, MA: MIT Press; 1975/1992

[9] Kirkpatrik S, Gelatt C, Vecchi M. Optimization by simulated annealing. Science. 1983; 220(4598):671-680 Retrieved from http://www.jstor.org/stable/1690046)

[10] Kennedy J, Ebrehart R. Particle Swarm Optimization. 1995. Pp. 1942-1948

[11] Brits, R., Engelbrecht, A.P. and van den Bergh, F., 2007. Locating multiple optima using particle swarm optimization. *Applied Mathematics and Computation*, *189*(2), pp.1859-1883.

[12] Shi Y, Eberhart R. A modified Particle swarm Optimizer. In: IEEE International conference on Evolutionary Computation Proceedings. 1998.pp.69-73

[13] C.Paper, I. Technology, and T. Kharagpur. Inertia weight strategies in particle swarm inertia weight strategies in particle swarm. No. May 2014, 2011

[14] Clerc M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. In: Proceedings of the 1999 ICEC. Washington, DC.1999.pp 1951-1957

[15] Clerc M, Kennedy J. The particle swarm- Explosion, stability, and convergence in a multidimensional complex space. In: IEEE Transaction on Evolutionary Computation. Feb 2002; 6(1):58-73. DOI: 10.1109/4235.985692

[16] [8] Eberhart RC, Shi Y. Comparing inertia weight and constriction factors in particle swarm optimization. In: proceedings of the 2000 Congress on Evolutionary computation.CEC00(Cat.No.00TH8512.La Jolla, CA.2000;1:84-88. DOI:10.1109/CEC.2000.870279

[17] GAO S, Wang H, Wang C, Gu S, Xu H, Ma H. Reactive power optimization of low voltage distribution network based on improved particle swarm optimization. In: Proceedings of the 2017 20th International Conference on Electrical Machines and systems (ICEMS). Sydney, NSW.2017.pp.1-5

[18] Abbas G, Gu J, Farooq U, Asad MU, EI-Hawary M. Solution of an economic dispatch problem through particle swarm optimization: A detailed survey-part I. In: IEEE Access. 2017;5:15105-15141

[19] Abbas G, Gu J, Frooq U, Reza A, Asad MU, EI – Hawary ME. Solution of an economic dispatch problem through particle swarm optimization: A detailed survey- Part II. In: IEEE Access. 2017;5:24426-24445

[20] Jois S, Ramesh R, Kullkarni AC. Face localization using skin colour and maximal entropy based particle swarm optimization for facial recognition. In: Proceedings of the 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON). Mathura, India. 2017.pp.156-161

[21] Chaudhary R, Patel A, Kumar S, Tomar S. Edge detection using particle swarm optimization technique. In: Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA). Greater Nooida, India. 2017.pp.363-367

[22] [14] Mozaffari MH, Lee WS. Convergent heterogeneous particle swarm optimisation algorithm for multilevel image thresholding segmentation. In: IET Image Processing. 2017; 11(8): 605-619

[23] Lukac, M. and Krylov, G., 2017, May. Study of GPU acceleration in genetic algorithms for quantum circuit synthesis. In *2017 IEEE 47th International Symposium on Multiple-Valued Logic (ISMVL)* (pp. 213-218). IEEE.

[24] Chen, Z., Li, X., Zhu, Z., Zhao, Z., Wang, L., Jiang, S. and Rong, Y., 2020. The optimization of accuracy and efficiency for multistage precision grinding process with an improved particle swarm optimization algorithm. *International Journal of Advanced Robotic Systems*, *17*(1), p.1729881419893508.

[25] Chan, T.C., Mahmood, R. and Zhu, I.Y., 2021. Inverse optimization: Theory and applications. *arXiv preprint arXiv:2109.03920*.

[26] Javaheri, N., Dries, R. and Kaandorp, J., 2014. Understanding the sub-cellular dynamics of silicon transportation and synthesis in diatoms using population-level data and computational optimization. *PLoS computational biology*, *10*(6), p.e1003687.

[27] Reddy, J.N., 2019. *Introduction to the finite element method*. McGraw-Hill Education.

[28] Bobet, A., Fakhimi, A., Johnson, S., Morris, J., Tonon, F. and Yeung, M.R., 2009. Numerical models in discontinuous media: review of advances for rock mechanics applications. *Journal of geotechnical and geoenvironmental engineering*, *135*(11), pp.1547-1561.

[29] Dodig-Crnkovic, G., 2002, April. Scientific methods in computer science. In *Proceedings of the Conference for the Promotion of Research in IT at New Universities and at University Colleges in Sweden, Skövde, Suecia* (pp. 126-130).

[30] Irez, A.B., Bayraktar, E. and Miskioglu, I., 2020. Fracture toughness analysis of epoxy-recycled rubber-based composite reinforced with graphene nanoplatelets for structural applications in automotive and aeronautics. *Polymers*, *12*(2), p.448.

[31] Eberle, S., Göttlinger, M. and Augat, P., 2013. Individual density–elasticity relationships improve accuracy of subject-specific finite element models of human femurs. *Journal of biomechanics*, *46*(13), pp.2152-2157.

[32] Zelle, J.M., 2004. *Python programming: an introduction to computer science*. Franklin, Beedle & Associates, Inc.

[33] Gambella, C., Ghaddar, B. and Naoum-Sawaya, J., 2021. Optimization problems for machine learning: A survey. *European Journal of Operational Research*, *290*(3), pp.807-828.

[34] Sun, S., Cao, Z., Zhu, H. and Zhao, J., 2019. A survey of optimization methods from a machine learning perspective. *IEEE transactions on cybernetics*, *50*(8), pp.3668-3681.

[35] Durán, O., Rodriguez, N. and Consalter, L.A., 2010. Collaborative particle swarm optimization with a data mining technique for manufacturing cell design. *Expert Systems with Applications*, *37*(2), pp.1563-1567.

[36] Omran, M.G., 2006. *Particle swarm optimization methods for pattern recognition and image processing* (Doctoral dissertation, University of Pretoria).

[37] Jin, C. and Jin, S.W., 2015. Prediction approach of software fault-proneness based on hybrid artificial neural network and quantum particle swarm optimization. *Applied Soft Computing*, *35*, pp.717-725.

[38] Ji, Y., Liew, A.W.C. and Yang, L., 2021. A novel improved particle swarm optimization with long-short term memory hybrid model for stock indices forecast. *IEEE Access*, *9*, pp.23660-23671.

[39] Tran, B., Zhang, M. and Xue, B., 2016, July. A PSO based hybrid feature selection algorithm for high-dimensional classification. In 2016 IEEE congress on evolutionary computation (CEC) (pp. 3801-3808). IEEE.

[40] Sambamurthy, N. and Kamaraju, M., 2020. FPGA Implementation of PSO Based RGB-Y Filter. International Journal, 9(4).

[41] Yu, X., Aouari, A., Mansour, R.F. and Su, S., 2021. A hybrid algorithm based on PSO and GA for feature selection. *Journal of Cybersecurity*, *3*(2), p.117.

[42] Zhou, J., Zhu, C., Zhang, W., Ai, W., Zhang, X. and Liu, K., 2020. Experimental and 3D MPFEM simulation study on the green density of Ti–6Al–4V powder compact during uniaxial high velocity compaction. *Journal of Alloys and Compounds*, *817*, p.153226.

[43] Krok, A. and Wu, C.Y., 2017. Finite Element Modeling of Powder Compaction. In *Engineering Crystallography: From Molecule to Crystal to Functional Form* (pp. 451-462). Springer, Dordrecht.

[44] Drucker, D.C. and Prager, W., 1952. Soil mechanics and plastic analysis or limit design. *Quarterly of applied mathematics*, *10*(2), pp.157-165.

[45] DiMaggio, F.L. and Sandler, I.S., 1971. Material model for granular soils. *Journal of the Engineering mechanics Division*, *97*(3), pp.935-950.

[46] Furukawa, R., Chen, Y., Horiguchi, A., Takagaki, K., Nishi, J., Konishi, A., Shirakawa, Y., Sugimoto, M. and Narisawa, S., 2015. Numerical evaluation of the capping tendency of microcrystalline cellulose tablets during a diametrical compression test. *International Journal of Pharmaceutics*, *493*(1-2), pp.182-191.

[47] Wang, D., An, X., Han, P., Fu, H., Yang, X. and Zou, Q., 2020. Particulate scale numerical investigation on the compaction of TiC-316L composite powders. *Mathematical Problems in Engineering*, *2020*.

[48] Cocks, A.C., 2001. Constitutive modelling of powder compaction and sintering. *Progress in materials science*, *46*(3-4), pp.201-229.

[49] An, X., Zhang, Y., Zhang, Y. and Yang, S., 2015. Finite element modeling on the compaction of copper powder under different conditions. *Metallurgical and Materials Transactions A*, *46*(8), pp.3744-3752.

[50] Feng, Y., Mei, D. and Wang, Y., 2019. Cohesive zone method based multi particle finite element simulation of compaction densification process of Al and NaCl laminar composite powders. *Journal of Physics and Chemistry of Solids*, *134*, pp.35-42.

[51] Wang, W., 1999. *Numerical modeling of compaction of particulate systems* (Doctoral dissertation, State University System of Florida).

[52] H. Shin, J.-B. Kim, S.-J. Kim, and K. Y. Rhee, "A simulation-based determination of cap parameters of the modified Drucker–Prager cap model by considering specimen barreling during conventional triaxial testing," Computational Materials Science, vol. 100, pp. 31-38, 2015.

[53] Jung, A. and Diebels, S., 2018. Yield surfaces for solid foams: A review on experimental characterization and modeling. *GAMM-Mitteilungen*, *41*(2), p.e201800002.

[54] Jin, W., Klinger, J.L., Westover, T.L. and Huang, H., 2020. A density dependent Drucker-Prager/Cap model for ring shear simulation of ground loblolly pine. *Powder technology*, *368*, pp.45-58.

[55] Zhou, M., Huang, S., Hu, J., Lei, Y., Xiao, Y., Li, B., Yan, S. and Zou, F., 2017. A density-dependent modified Drucker-Prager Cap model for die compaction of Ag57. 6-Cu22. 4-Sn10-In10 mixed metal powders. Powder Technology, 305, pp.183-196.

[56] C. Lu, "Determination of cap model parameters using numerical optimization method for powder compaction," PHD, Marquette University, 2010.

[57] G. S. Wagle, "Die compaction simulation: Simplifying the application of a complex constitutive model using numerical and physical experiments," Doctor of Philosophy, The Pennsylvania State University, 2006.

[58] Ceylan, Z., 2021. Short-term prediction of COVID-19 spread using grey rolling model optimized by particle swarm optimization. *Applied soft computing*, *109*, p.107592.

[59] Gontara, S., Boufaied, A. and Korbaa, O., 2019, October. Fault localization algorithm in computer networks based on the boolean particle swarm optimization. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)* (pp. 4347-4352). IEEE.

[60] Banks, A., Vincent, J. and Anyakoha, C., 2007. A review of particle swarm optimization. Part I: background and development. *Natural Computing*, *6*(4), pp.467-484.

[61] Shami, T.M., El-Saleh, A.A., Alswaitti, M., Al-Tashi, Q., Summakieh, M.A. and Mirjalili, S., 2022. Particle swarm optimization: A comprehensive survey. *IEEE Access*.

[62] Zhang, X., Liu, H. and Tu, L., 2020. A modified particle swarm optimization for multimodal multi-objective optimization. *Engineering Applications of Artificial Intelligence*, *95*, p.103905.

[63] Le, Z.H., Yu, Q.L., Pu, J.Y., Cao, Y.S. and Liu, K., 2022. A Numerical Model for the Compressive Behavior of Granular Backfill Based on Experimental Data and Application in Surface Subsidence. *Metals*, *12*(2), p.202.

[64] Zou, F., Huang, S., Zhou, M., Lei, Y., Yan, S., Zhang, J. and Wang, B., 2019. Constitutive model and compaction equation for aluminum alloy powder during compaction. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, *13*(1), pp.JAMDSM0015-JAMDSM0015.

[65] Li, W., Meng, X., Huang, Y. and Fu, Z.H., 2020. Multipopulation cooperative particle swarm optimization with a mixed mutation strategy. *Information Sciences*, *529*, pp.179-196.

[66] Hrairi, M., Chtourou, H., Gakwaya, A. and Guillot, M., 2011. Modeling the powder compaction process using the finite element method and inverse optimization. *The International Journal of Advanced Manufacturing Technology*, *56*(5), pp.631-647.

[67] Majzoobi, G.H. and Jannesari, S., 2015. Determination of the constants of cap model for compaction of three metal powders. *Advanced Powder Technology*, *26*(3), pp.928-934.

[68] Buljak, V., Baivier-Romero, S. and Kallel, A., 2021. Calibration of Drucker–Prager Cap Constitutive Model for Ceramic Powder Compaction through Inverse Analysis. *Materials*, *14*(14), p.4044.

[69] Zhou, R., Yang, L., Liu, Z.W. and Liu, B.F., 2020. Modeling the powder compaction process by an integrated simulation and inverse optimization method. *Materials Today Communications*, *25*, p.101475.

[70] Shi, J., Chopp, D., Lua, J., Sukumar, N. and Belytschko, T., 2010. Abaqus implementation of extended finite element method using a level set representation for three-dimensional fatigue crack growth and life predictions. Engineering Fracture Mechanics, 77(14), pp.2840-2863.

[71] Abdulsalam, H., 2021, April. Mesh Sensitivity Assessment on 2D and 3D Elastic Finite Element Analysis on a Compact Tension Specimen Geometry Using Abaqus/CAE Software. In IOP Conference Series: Earth and Environmental Science (Vol. 730, No. 1, p. 012032). IOP Publishing.

[72] Chen, K., Xue, B., Zhang, M. and Zhou, F., 2021. Evolutionary Multitasking for Feature Selection in High-Dimensional Classification via Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, *26*(3), pp.446-460.

[73] Ali, Z.A., Han, Z. and Masood, R.J., 2021. Collective motion and self-organization of a swarm of UAVs: A cluster-based architecture. *Sensors*, *21*(11), p.3820.

[74] Mashayekhi, M., Harati, M. and Estekanchi, H.E., 2019. Development of an alternative PSO-based algorithm for simulation of endurance time excitation functions. *Engineering Reports*, *1*(3), p.e12048.