The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

School of Computer Science

Enhancing Decision Support for Solutions of Packing Problem in Additive Manufacturing: Features, Datasets and Experimental Studies

Luiz Jonatã Pires de Araújo

Thesis submitted to the University of Nottingham
for the degree of Doctor of Philosophy

June 2019

Abstract

Additive manufacturing (AM) encompasses a set of technological advancements that enable objects to be produced in an incremental layer-by-layer material deposition process. The advantages of such techniques include a more flexible production chain and the capacity to manufacture highly customised products. The manufacturing process takes place within an enclosed build container, referred to as a 'build volume', which should be fully utilised to achieve more efficient production times and reduce costs. This requirement is at the core of cutting and packing problems, which are well-known combinatorial problems that have been algorithmically addressed by the operations research community. This study devotes particular attention to the understanding of three-dimensional irregular packing (3DIP) problems, i.e., the task of arranging arbitrary three-dimensional geometries. It is motivated by the necessity for more precise and well-informed terminology and categorisation criteria in this problem domain. The thesis also investigates the properties of existing 3DIP algorithms and the performance patterns with respect to build volume utilisation and the feature space. These topics have been scarcely addressed in the literature due to the amount of available data and relevant features on this problem domain. The primary objective of this work is to contribute to more efficient AM processes by assessing how volume utilisation can be maximised within the machine at every build. First, the research assists in the characterisation of 3DIP problems by introducing new measurements for assessing part complexity. Experiment results demonstrate that such metrics are suitable for describing entrant geometric features in non-convex three-dimensional objects. Second, this study extends the existing taxonomy for cutting and packing and provides the most significant benchmark for 3DIP in the literature, which is aligned with the challenging requirements observed in the AM environment. Third, it evaluates some of the most commonly used packing approaches based on the deepest bottom left with fill heuristic. Lastly, this thesis presents one of the first reported applications of algorithm selection to 3DIP problems, mapping the problem instance features, including the newly proposed ones, to the best packing algorithm. The results confirm the potential of the algorithm selection approach to deliver increased build volume utilisation in AM processes.

Contents

List of Figures

List of Tables

List of Abbreviations

3DBP	Three-dimensional Bin Packing
3DIK	Three-dimensional Irregular Knapsack
3DIP	Three-dimensional Irregular Packing
3DSP	Three-dimensional Strip Packing
3DISP	Three-dimensional Irregular Strip Packing
3MF	3D Manufacturing Format
AM	Additive Manufacturing
AMF	Additive Manufacturing File Format
ATV	Automated Transfer Vehicle
BL	Bottom Left
BVU	Build Volume Utilisation
C&P	Cutting and Packing
CART	Classification and Regression Trees
CP	Crossover Probability
CSP	Constraint Satisfaction Problem
CV	Cross-validation
DBL	Deepest Bottom Left
DBLF	Deepest Bottom Left with Fill
DBLFD	Deepest Bottom Left with Fill Decreasing
FFD	First Fit Decreasing
GA	Genetic Algorithm
GARP	Genetic Algorithm for part packing in rapid prototyping
GLS	Guided Local Search
GP	Genetic Programming
KNN	K-nearest Neighbours
LS	Laser Sintering
MBB	Minimum Bounding Box

MCV	Mean Connectivity Value
MIP	Mixed Integer Programming
ML	Machine Learning
MLP	Multi-Layer Perceptron
MP	Mutation Probability
NASA	National Aeronautics and Space Administration
NFL	No Free Lunch
NFP	No-Fit-Polygon
NI	Not informed
O1	Order 1 crossover
OR	Operations Research
RFECV	Recursive Feature Elimination with Repeated Cross-validation
RKCV	Repeated K-fold Cross-validation
ROC AUC	Computed area under the receiver operating characteristic curve
RSD	Relative Standard Deviation
SA	Simulated Annealing
SAHC	Steepest Ascending Hill Climbing with Random Restart
SAT	Propositional Satisfiability
SD	Standard Deviation
SLA	Stereolithography
SR	Spies Ratio
STL	Stereolithography
TS	Tabu Search
WB	Wall-building
XML	Extensible Markup Language

Chapter 1

Introduction

Additive manufacturing (AM), also known as 3D printing, is an umbrella term for a set of technologies that facilitate the production of highly complex parts in an incremental layer-by-layer material deposition process. In most AM technology variants, this process takes place within the machine in an enclosed build container, referred to as a 'build volume'. This differs from conventional production, which is mostly based on subtracting (where raw material is iteratively removed), moulding (casting soft material within a mould), and forming processes (forces are applied to shape the material) (Hague et al., 2004; Gibson et al., 2014). The benefits of AM include a more flexible supply chain that quickly connects the mechanical design stage to the production (Cotteleer and Joyce, 2014; Achillas et al., 2017) and the ability to compose customisable parts (Tuck et al., 2008).

AM was initially developed in the 1980s for prototyping physical models to provide insights into a product before committing to final manufacturing processes (Jacobs, 1992). However, in recent years, AM has been adopted for the manufacture of end-use products in areas such as aerospace and the production of industrial, automotive and medical equipment. AM has significant potential for adoption in applications that are characterised by high per-unit revenues and that involve high degrees of geometric complexity or customisation (Tuck et al., 2008).

As a parallel manufacturing process, AM enables the simultaneous production of different geometric parts in a single build volume (Ruffo and Hague, 2007). This gives rise to a build volume packing problem during the machine setup process, which requires a computational solution (Nyaluke et al., 1996; Hur et al., 2001). Baumers et al. (2013, 2017a) have demonstrated that solving this problem effectively during AM execution significantly affects the manufacturing cost. The process requirement to fully utilise the available build volume can be mapped to cutting and packing (C&P) problems, which are well-known combinatorial optimisation problems discussed by the operational research (OR) community (Dyckhoff, 1990; Wäscher et al., 2007).

In the field of OR, considerable research has been conducted on C&P algorithms. Three-dimensional irregular packing (3DIP) problems are combinatorial optimisation problems in which a set of arbitrary volumetric items must be placed into given containers, or build volumes, in such a way that the total empty space (between the items) is minimised (Wäscher et al., 2007). Such optimisation problems are classified as NP-hard problems (Garey and Johnson, 1979). One factor that increases the inherent difficulty of such problems is the presence of non-overlapping constraints between the geometries involved.

A wide variety of packing approaches have been discussed in the literature, and choosing the most suitable technique often depends on the technology employed, the material used and the production constraints (Ikonen et al., 1997; Hur et al., 2001; Canellidis et al., 2006). Some algorithms, for example, address the free arrangement of geometries within the build volume, which may result in configurations in which one item is located on top of other items. Such methods are suitable for relatively unconstrained techniques such as laser sintering (LS), for which support structures are not required (Gibson et al., 2014). On the other hand, other technology variants such as resin vat and metallic powder bed processes, require adjustments to allow for support structures.

Another factor to consider when selecting an appropriate packing algorithm are the characteristics of the problem to be addressed. This is explored in the algorithm selection problem described by Rice (1975), which aims to investigate relationships between problem features and algorithmic performance. Algorithm selection has been successfully applied to several combinatorial optimisation problems, including scheduling (Beck and Freuder, 2004; Burke et al., 2006; Smith-Miles, 2009) and the travelling salesman problem (Pihera and Musliu, 2014; Kotthoff, 2014), both of which have been explored by the Automated Scheduling Optimisation and Planning (ASAP) research group at the University of Nottingham. However, far too little has been published on the application of algorithm selection to 3DIP problems for reasons that include:

- There are a limited number of numeric features for characterising 3DIP problems that can capture elements such as shape complexity and demand variation in the context of AM applications;
- Few representative datasets currently exist, i.e., few problem instances are available for benchmark packing procedures. This occurs because such instances are often selected or designed according to the characteristics of the introduced algorithms and not the other way around (Jaeggi et al., 2008; Egeblad et al., 2009);
- The computational costs of processing, or labelling, a single 3DIP instance are considerably high. As a result, no significant amount of data on algorithm performance has been

published, which inhibits an unbiased analysis of the prospective relationship between problem features and algorithms in the 3DIP-domain.

A solution to the aforementioned issues would support the development of approaches to practical packing problems and contribute reducing production costs and time when manufacturing large demands (Baumers et al., 2013, 2017a). Additionally, addressing those points would pave the way for research on the application of complementary techniques such as concurrent optimisation and data science to 3DIP, particularly in the AM environment.

1.1 Aims and scope

The main purpose of this thesis is to enable an increased understanding of the main characteristics of 3DIP problems by introducing descriptive features and a taxonomy that can help researchers and AM practitioners to correctly address such problems using suitable techniques. Also, this study demonstrates the implementation of an automatic algorithm selection pipeline for solving complex 3DIP problems aligned with challenging AM requirements, which appears for the first time in the literature. In summary, this research contributes to the following research directions:

- Expanding on knowledge related to 3DIP problems within the AM sector by:
  - introducing and analysing features that can support 3D printing by quantifying degrees of part complexity;
  - proposing a more informed and precise taxonomy for C&P problems;
  - generating a new comprehensive dataset for 3DIP that contains industrial-like parts, which will be a useful tool for better benchmarking of 3DIP solutions.
- Enabling the identification of the most suitable algorithmic approaches for 3DIP problems and how such methods relate to corresponding problem features through:
  - experimental analysis of the practicality of three of most commonly used 3DIP approaches;
  - contributing what, to the best of the authors' knowledge, is the first report in the literature of an algorithm selection pipeline for 3DIP problems. Thus, the study provides recommendations regarding classifiers, configuration settings and relevant features that should not be omitted in future reports in this domain.

Achieving the above requires an interdisciplinary approach involving a wide range of techniques within the context of OR and Data Sciences including:

- implementation of optimisation algorithms;
- application of mathematical and statistical models;
- preprocessing and classifying data using a machine learning library;
- AM technologies (e.g. the representation and parsing of geometries using non-structured data).

1.2 Contributions of this thesis

This thesis introduces new approaches and features for quantifying shape complexity of three-dimensional geometric models that are suitable for AM processes. It also provides a practical method for calculating the Spies ratio, which is one of the currently used metrics in this context. The motivation for carrying out such a study is to generate and analyse data to support well-informed decision systems capable of quickly identifying the best packing approaches for certain classes of problems within the current C&P taxonomy. The current categorisation for such problems, however, does not sufficiently address the variants for objectives and constraints observed in AM. Another complication factor is the sometimes ambiguous terminology employed by researches in this domain. This work proposes a new taxonomy for cutting and packing, which is aligned but not restricted to AM requirements. Also, a more comprehensive dataset was made publicly available to facilitate and encourage more research in this field.

To date, most of the existing body of research on 3DIP within the AM sector has employed variants of the deepest bottom-left heuristic. However, there has been only limited discussion on the effects that different algorithm parameters and degrees of freedom for part rotation have on performance. This research provides valuable insights into the most common packing approaches, especially regarding the integration of genetic algorithms, which has been the dominant solution technique.

Past research has introduced new algorithmic solutions which arguably perform better for certain classes of packing problems. For example, wall-building heuristics have been said to result in higher volume utilisation for the demands of many parts with few different specifications. The identification of the most suitable algorithm for a new instance is at the core of algorithm selection problems and has found fertile soil for several combinatorial optimisation problems. However, this technique has not been applied to 3DIP and is first reported in the literature in this thesis. Experiments were conducted to support a series of recommendations for machine learning classifiers, parameter settings and essential features. As a result, the generation of the most extensive dataset comparing the performance of two of the most popular 3DIP approaches.

1.3 Publications and presentations

During the course of researching this thesis, the following papers have been published:

- Chapter 2
  - Araújo, Luiz JP, Ender Özcan, Jason Atkin, Martin Baumers, Christopher Tuck, and Richard JM Hague. "Toward better build volume packing in additive manufacturing: classification of existing problems and benchmarks." Proceedings of the Solid Freeform Fabrication Symposium (2015): 401-410.
- Chapter 3
  - Araújo, Luiz JP, Ender Özcan, Jason AD Atkin, and Martin Baumers. "A part complexity measurement method supporting 3D Printing." In NIP & Digital Fabrication Conference, vol. 2016, no. 1, pp. 329-334. Society for Imaging Science and Technology, 2016.
  - Araújo, Luiz JP, Ender Özcan, Jason AD Atkin, and Martin Baumers. "Analysis of irregular three-dimensional packing problems in additive manufacturing: a new taxonomy and dataset." International Journal of Production Research (2018): 1-15.

In addition to the above published papers, the following have been submitted or their preparation for submission is near conclusion:

- Chapter 4
  - Araújo, Luiz JP, Panesar, Ajit, Ender Özcan, Jason AD Atkin, and Martin Baumers, and Ashcroft, Ian. 'An experimental analysis of Deepest Bottom Left with Fill packing methods for Additive Manufacturing'. Submitted to *International Journal of Production Research*.
- Chapter 5
  - Araújo, Luiz JP, Ender Özcan, Jason AD Atkin, and Martin Baumers. 'An experimental investigation of a machine-learning-based algorithm selection for three-dimensional irregular packing'. In preparation.

The conducted work has been presented at the following events and universities:

- 26th Annual International Solid Freeform Fabrication (SFF) Symposium - An Additive Manufacturing Conference, University of Texas, Austin, United States of America (08/10/2015)

- 32nd International Conference on Digital Printing Technologies (NIP) - Printing for Fabrication, University of Manchester, United Kingdom (14/09/2016)

- 'Perspectives on the utilisation of machine learning for three-dimensional irregular packing', presentation at Innopolis University, Russia (29/05/2018)

1.4 Structure of the thesis

The structure of this thesis is summarised in Figure 1.1 to help the reader follow how this interdisciplinary work is aligned with the research directions discussed in Section 1.1. Chapter 2 reviews the literature concerning part complexity features, taxonomies for C&P, the characteristics of commonly used packing approaches, the algorithm selection problem and a summary of some classification models in machine learning. Chapter 3 contributes to the domain knowledge of 3DIP by introducing new part complexity features appropriate for AM applications. Moreover, it improves the existing taxonomies for C&P, which supports a precise categorisation of 3DIP problems and literature. Finally, a comprehensive 3DIP dataset aligned to the realistic characteristics of AM applications is made publicly available. Such a resource is timely, as it can contribute to development of algorithmic solutions for challenging 3DIP instances. Chapter 4 presents an experimental investigation of the usefulness and the pitfalls of deepest bottom left algorithms, which have been at the core of most of the packing approaches applied in AM. Chapter 5 demonstrates the first implementation of machine-learning-based algorithm selection to 3DIP problems. The application of the entire pipeline provides insights on essential aspects, including the efficiency of classification models, parameter settings and the relative importance of 3DIP features through different feature selection methods. The final chapter summarises the contributions of this thesis and presents directions for future research.

	Research directions			
	Enhancing the domain knowledge		Increasing understand of 3DIP solutions	
	Part complexity features	C&P taxonomies and data	Popular 3DIP approaches	Algorithm selection for 3DIP
Background and related work (Chapter 2)	Section 2.2 and 2.3	Section 2.4 and 2.6	Section 2.5	Section 2.7 and 2.8
Enhancing the domain knowledge (Chapter 3)	Section 3.1	Section 3.2		
Analysis of the DBLF algorithm (Chapter 4)			Sections 4.1, 4.2 and 4.3	
Algorithm selection for 3DIP (Chapter 5)				Sections 5.1, 5.2 and 5.3

Figure 1.1: Research directions related to 3DIP problems approached in this work.

Chapter 2

Background and related work

This section presents the literature on the following topics, which are essential for understanding three-dimensional packing problems: the most common occurrences of 3DIP problems in the industry, three-dimensional geometry features, alternatives for categorising C&P problems and available benchmarks. It is equally important to be aware of the main algorithmic approaches in this domain and their strengths and weaknesses, which are mostly observed through experimentation. Finally, it surveys work on the application of algorithm selection to combinatorial optimisation problems and on several machine learning classifiers commonly utilised to identify patterns between search space and algorithm performance.

2.1 Occurrences of 3DIP problems in the industry

Three-dimensional irregular packing problems arise in several industrial applications and vary widely with regard to constraints and objectives. This thesis focuses on the occurrence of such problems, solutions to them and their relevant features in the context of AM processes. Over the past two decades, several studies have stressed that the application of 3DIP algorithms to maximise build volume utilisation during operations has significant potential for improving production time and costs (Baumers et al., 2016a). Such computation methods are particularly useful for AM technologies, such as LS, that allow a relatively unconstrained layout configuration for the parts within the machine and do not require supporting structures (Gibson et al., 2014).

Another example of an AM technology that benefits from the application of C&P algorithms to achieve increased build volume utilisation is stereolithography (SLA). Two distinguishing characteristics of SLA are the necessity for supporting structures - as the powder alone cannot hold in place the upwardly deposited layers - and the existence of restrictions on the superposition of parts (Jacobs, 1992). Packing solutions used in these applications aim to arrange parts in placement configurations with horizontal projections that maximise the effectively used surface area of the build platform (Canellidis et al., 2006, 2013), while also minimising the build volume height, which affects the estimated production time (Baumers et al., 2013).

The systematic application of 3DIP algorithms to industrial environments began in the 1980s, at the same time as the birth of AM (Hull, 1986; Udy et al., 1988). However, the first studies focussed on the design layout of engineering components to address the need for more complex mechanical assemblies and small circuits (Szykman and Cagan, 1995; Kolli et al., 1996; Cagan et al., 2002). Packing algorithms used in this domain often have multiple objectives, including optimising the centre of gravity and aligning the objects (Wong and Hernandez, 2012; Sherwani, 2012). Functional constraints connected to a specific technology, such as policies on the minimum distance between any pair of components, cannot be ignored (Grignon and Fadel, 2004).

3DIP problems commonly occur in computational simulations of granular particle movement, a field of study with diverse applications in the physical and chemical and engineering industries (Rémond and Gallias, 2004). Most of these studies consider the arrangement of simple geometries such as spheres and rectangles (Song et al., 2006; Al-Raoush and Alsaleh, 2007; Fraige et al., 2008), although recent methods have been developed to handle irregular-shaped particles (Jia and Williams, 2001; Lee et al., 2003). Nevertheless, the latter techniques suffer from performance degradation due to computationally expensive operations for detecting overlaps between complex surfaces (Lee et al., 2009).

The arrangement of objects into larger containers, which can themselves be arbitrarily shaped, is an essential requirement in most transportation systems (Fasano, 2004; Eisenbrand et al., 2005). For example, the Automated Transfer Vehicle (ATV) is a set of standards provided by the National Aeronautics and Space Administration (NASA) that support the cargo transportation to the International Space Station. Such standards define norms about the number of items, the accommodation rules and the balancing constraints for each launch and for each compartment of the ATV carrier (Fasano, 2003). In the OR literature, such problems are commonly referred to as *three-dimensional non-convex domain loading* problems (Boccia et al., 2011).

2.2 Representations of rigid three-dimensional solids

The mostly commonly used file format for describing three-dimensional solids in the layered manufacturing industry is STL, which is a facet-based (mesh) representation that approximates surfaces and solid 3D geometries (Wan, 1999). Alternative formats have also been introduced in response to recent advances in manufacturing, such as additive manufacturing file format (AMF) and 3D manufacturing format (3MF). Specifications regarding the material, colour, texture and other physical properties of the parts are provided in these formats (Rengier et al., 2010; Ambrosi and Pumera, 2016).

In addition to the file format, geometry representation plays an essential role in the design and performance of the packing approach. For example, the mathematical model proposed by

Udy et al. (1988) represents each object as the volume defined by precisely twelve edges. As a result of this constraint, the approach is limited to simple shapes. Another example is the algorithm implemented by Skiena (1997) for validating the geometric feasibility of the positions of two given geometries. Such an algorithm was shown to have a complexity of $O(n^3m^3)$, where n and m are the respective numbers of edges, which means that its performance degrades quickly as the number of faces and edges increases.

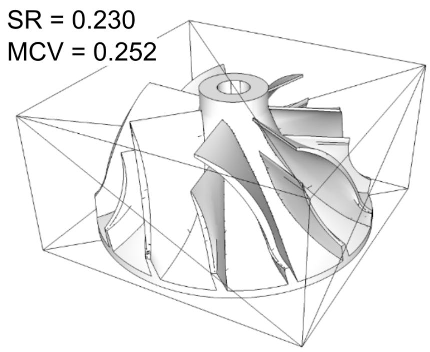
Geometries have also been represented by raster models which discretise a three-dimensional space and enumerate the set of points that belong to its interior. This approach was adopted by Cagan et al. (1998), whoce packing algorithm used octrees to decompose the models and enable efficient intersection testing. The discretisation of complex parts into a rectangular three-dimensional grid results in a geometric representation in which each object is denoted by a collection of voxels (unit values in a three-dimensional grid) (Min, 2004; Tiwari et al., 2010; Zhang and Zhou, 2012). Despite the fact that a certain degree of information and precision is lost, raster representations allow a quicker feasibility check during the packing process.

Complex objects have also been represented as the result of successive union and intersection operations between regular objects (Fasano, 2004; Stoyan et al., 2004, 2005). Mathematical tools for automatically describing geometries in this fashion have been successfully applied to two-dimensional problems (Stoyan and Ponomarenko, 1977; Bennell et al., 2010). However, the extension of such techniques to three-dimensional packing and their feasibility in terms of computer efficiency have not been systematically addressed (Egeblad, 2008). A drawback of this approach for geometry representation is the requirement for a pre-processing stage that parses each intended part to a data structure describing the set of regular objects, their position and rotation and the sequence of union and intersection operations (Scheithauer et al., 2005).

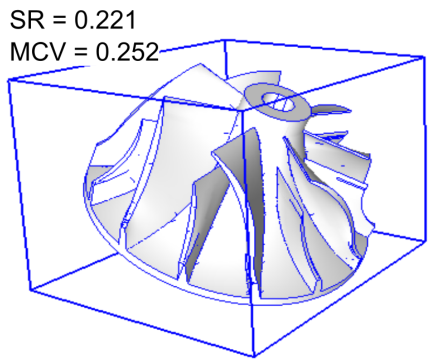
2.3 Features of three-dimensional geometries

It is reasonable to affirm that the complexity of the requested parts is among the aspects that affect packing efficiency. Quantifying the complexity of three-dimensional geometries, therefore, provides interesting indications that can support the choice of packing approach. Nevertheless, measuring shape complexity is a challenging and sometimes subjective task as it depends on the particular problem domain (Psarra and Grajewski, 2001). For example, in an architectural context, complexity is connected to quantifying the accessibility between space segments or areas (Hillier and Hanson, 1984; Hiller, 1996) and the characteristics of the perimeter (Rowe and Banham, 1977). From a computer graphics perspective, by contrast, complexity is related to the proportion of the surface area that is visible (Benedikt, 1979).

One approach for quantifying shape complexity is the Spies ratio (SR) (Spies, 1957), which is the ratio between the volume of the three-dimensional geometry and the volume of its minimal constraining primitive (e.g. cylinder or cuboid). The intuition behind the SR is the quantification of the unutilised space when an object is reduced to its envelope. The efficiency of packing algorithms that reduce parts to their respective bounding boxes is affected by the alignment between these entities, which also results in different values for SR as illustrated in Figure 2.1.



(a) Spies ratio equal to 23%



(b) Spies ratio equal to 15%

Figure 2.1: Illustration of how the alignment affects the value of the Spies ratio (Spies, 1957).

Some of the available shape complexity measurements collect attributes of the file used to share the geometries (Valentan et al., 2011). Such metrics are mostly based on the STL (an abbreviation of 'stereolithography') format (Wan, 1999), as the additional information in newer formats (e.g. elasticity and colour) has little or no effect on the layout configuration of the part within the build chamber. The relationship between attributes in the STL model and optimal manufacturing in rapid prototyping was first investigated by Valentan et al. (2006, 2008), who compared a series of metrics to understand shape complexity. These features are shown as follows:

- The number of faces (triangles)
- The ratio between the volume of the part and the number of faces
- The ratio between the volume of the part and the surface area
- The ratio between the volume of the bounding box of the part in its original orientation (i.e., no rotation applied) and the volume of the part

Alternative complexity measurements were also generated through non-linear combinations of basic data in the STL model. Valentan et al. (2011) compared the previous measurements (Valentan et al., 2006, 2008) with a new indicator obtained by equation 2.1:

$$\frac{\text{surface area} * \text{number of facets}}{\text{volume of the bounding box of the part in its original rotation}} \tag{2.1}$$

One advantage of using easily accessible data in the STL model for assessing geometry complexity is the avoidance of additional processing effort to compute feature values. However, Valentan et al. (2006, 2008, 2011) have admitted that such data is unable to capture or measure degrees of non-convexity and concavity, which are essential characteristics for assessing manufacturing quality. Another disadvantage of most of these metrics is their dependence on the original orientation of the model, which is often provided by a human operator.

Psarra and Grajewski (2001) introduced a complexity measurement to quantify the presence of concavities in two-dimensional perimeters, which is useful in an architectural context. This metric assesses the 'connectivity' of each pixel p laying on the perimeter P. Two perimeter pixels are said to be connected if the line between these pixels is entirely within the polygon defined by the perimeter. The mean connectivity value of a polygon is the mean of the connectivity values of all the pixels on P. The connectivity value of a pixel (CV_p) and the mean connectivity value of a polygon (MCV) are given in equations 2.2 and 2.3, respectively.

$$CV_p = \frac{\text{number of perimeter pixels that are connected to } p}{|P|} \tag{2.2}$$

$$MCV = \sum_{p \in P} \frac{CV_p}{|P|} \tag{2.3}$$

Another complexity measure is the v-value, also introduced by Psarra and Grajewski (2001), which captures the variation of CV among the perimeter locations. An advantage of using both the MCV and the v-value measurements is that they can detect which areas of the perimeter contribute to higher complexity.

2.4 Taxonomies for cutting and packing problems

The first classification of C&P problems was presented by Dyckhoff (1990), who proposed representing problems as four-tuples in the form $\alpha/\beta/\gamma/\delta$ to capture elements relevant for C&P. In Dyckhoff's notation, problems are categorised according to their dimensionality (α), the kind of assignment or requirements regarding the selection of items and containers (β), the quantity and shape of the containers (γ), and the multiplicity of types of required objects (δ). The values for each category in the four-tuple notation are listed with their original terminology as follows.

- Dimensionality (α)
  - One-dimensional (1)

  - Two-dimensional (2)
  - Three-dimensional (3)
  - N-dimensional with N > 3 (N)
- Kind of assignment (β)
  - All objects and a selection of items (B): a subset of items that optimises an objective function is selected and then distributed among available large objects
  - A selection of objects and all items (V): all items have to be assigned to a proper selection of large objects
- Assortment of large objects (γ)
  - One object (O)
  - Identical figure (I): all large objects have the same dimensions
  - Different figures (D): large objects have different dimensions
- Assortment of small items (δ)
  - Few items of different figures (types or models) (F)
  - Many items of many different figures (M)
  - Many items of relatively few different (non-congruent) figures (R)
  - Congruent figures (C)

Despite its importance for the study of C&P, the taxonomy introduced by Dyckhoff (1990) has several weaknesses (Wäscher et al., 2007). First, it 'does not necessarily result in homogeneous problem categories' and contains a 'partially inconsistent' categorisation Wäscher et al. (2007, p.1112). In addition, Wäscher et al. noticed that Dyckhoff's coding scheme was 'not self-explanatory from the view point of an international (English-speaking) community or researchers' Wäscher et al. (2007, p.1111). For example, the kind of assignment defining the selection of a subset of items is denoted by the symbol B, shorthand for the German word 'Beladeproblem', while the second assignment is assigned as 'V' after the word 'Verladeproblem'. Wäscher et al. (2007) addressed the identified issues by introducing refined terminology and categorisation criteria, which are summarised as follows.

- Dimensionality
  - One-dimensional

  - Two-dimensional
  - Three-dimensional

- Kind of assignment
  - Output (value) maximisation: a subset of the small items of maximal value (usually the volume) has to be assigned
  - Input (value) minimisation: all small items are assigned

- Assortment of small items
  - Identical small items
  - Weakly heterogeneous assortment: the demand for each item type (items of identical shape and size) is relatively large
  - Strongly heterogeneous assortment: only few elements are of identical shape and size; the demand for each item type is equal to one (Wäscher et al., 2007, p.1115)

- Assortment of large objects
  - One large object
    * All dimensions fixed
    * One or more variable dimensions
  - Several large objects
    * Identical large objects
    * Weakly heterogeneous assortment
    * Strongly heterogeneous assortment

Wäscher et al. (2007) have also identified three main categories of packing problems that are frequently found in the literature and that receive considerable attention in this thesis: (i) the *knapsack problem*, which pertains to the arrangement of a subset of small items that maximise the total value within a larger object with limited capacity; (ii) the *open dimension* or *strip packing* problem, which addresses the minimisation of an open dimension (e.g. height) of a single large object in such a way that it accommodates the entire demand for small parts; and (iii) the *bin packing* problem, which addresses the minimisation of the number of large objects (or bins) necessary to accommodate the entire demand for small objects.

Wäscher et al.'s taxonomy was a significant improvement of the previous typology and is the most widely accepted in the C&P literature. However, as in Dyckhoff's classification, some of the categories still depend upon on ambiguous linguistic terms such as 'few' or 'many' to describe the variation of both small and large objects. Moreover, Wäscher et al. did not offer an encoding scheme which could aid researchers to quickly represent objectives, constraints and other characteristics of a particular problem. Such an encoding would be timely, as it would enable researchers to quickly and easily access literature on a particular category of C&P problem.

The taxonomies proposed by Dyckhoff and Wäscher et al. have contributed to the standardisation of the C&P terminology, which is helpful considering the fact that many researchers approaching such problems belong to different fields and often refer to similar problems using different denominations. Nevertheless, there are still inconsistencies in the terminology used in this area. For example, Egeblad (2008) has noted that while the term 'knapsack packing' is commonly utilised in the OR literature, 'container loading' and 'multi-pallet loading' have been systematically employed to describe the same problem in work on industrial applications.

Jia et al. (2007) have observed that the geometric characteristics of the parts are among the aspects that affect the packing density obtained by packing algorithms. In this regard, Dyckhoff's (1990) typology separate parts according to their shape, being either regular (forms that can be described by a few parameters such as rectangles or spheres) or irregular. Wäscher et al. (2007) discriminated between items according to the variation in demand, this either *weakly* or *strongly heterogeneous*. A more detailed classification for two- and three-dimensional rectangular geometries is provided by Pisinger and Egeblad (2006), who categorised parts into five groups:

- Flat: the width and height of a rectangle is larger than its depth
- Long: the depth is larger than both the width and the height
- Cube: the three axes are of equal dimensions
- Uniform: the largest dimension is no more than 200% of the smallest
- Diverse: the largest dimension is up to 50 times larger the smallest

It can be noted that according to the above classification criteria objects can belong to more than one category. Another drawback is that aspects connected to irregular packing problems, such as convexity, shape complexity and multiplicity of demands, are ignored.

2.5 Algorithmic approaches to 3DIP problems

C&P problems are multidisciplinary optimisation problems that have been extensively addressed by studies in the area of OR since the 1960s (Art, 1966). From a computer science perspective, Fowler et al. (1980) and Fowler et al. (1981) have demonstrated that C&P problems belong to the class of NP-hard problems (Garey and Johnson, 1979), which means that their solution time scales up at least as fast as a exponential rate as the size of the input increases (Woeginger, 2003). One aspect that adds a layer of complication is the presence of a large number of geometrically complex parts. Hence most of the 3DIP algorithms use heuristics or rules of thumb to guide the search for satisfactory solutions in acceptable time (Hochbaum, 1996). A brief overview of general strategies for solving 2D Irregular Packing problems can provide insights into how 3DIP can be addressed, even though the latter are considerably more complex. Bennell and Oliveira (2009) have classified 2D packing approaches according to the performed task and representation for solutions, which are summarised as follows:

- Build partial solutions: methods which incrementally construct the solution, piece by piece
  - Placement rules: heuristics that define how parts are placed, such as policies for orthogonal translation
    - Bottom Left: fixed translation sequence (Art, 1966)
    - Floating placement: the position where a piece is attached is determined by a heuristic procedure
  - Pieces sequencing: methods which aim to identify the best order in which parts should be packed
    - Random
    - Fixed rule (e.g. pieces with the largest dimension on the x-axis are placed first (Art, 1966))
    - Dynamic selection
- Improve complete solutions: procedures which interactively perturb the initial solution until termination criteria are satisfied
  - Layout representations: solutions are represented by sequences of pieces to be processed by a placement rule
  - Physical layout: the layout solution is explicit, i.e., it contains information on the location and orientation for each piece

Packing algorithms often combine different techniques from the categories shown above (Bennell and Oliveira, 2009). For example, a particular procedure may improve a complete solution represented as a sequence of parts while also using a placement rule (e.g. the bottom-left heuristic) to process the sequence (Egeblad et al., 2007). The following sections use Bennell and OliveiraâĂŹs criteria and categorise packing algorithms according to the contribution and the main component of the particular study.

2.5.1 Exact methods

The spectrum of approaches for addressing packing problems includes exact methods for local or near-global optimisation, including mathematical programming and geometric calculus (Scott and Kilgour, 1969; Visscher and Bolsterli, 1972). These methods have been used to arrange similar geometry primitives like spheres or cubes within rectangular spaces. However, as some authors note, relatively few studies have applied exact methods to 3DIP problems (Stoyan et al., 2005; Chernov et al., 2010; Paul and Anand, 2012). One reason for this is that 'three-dimensional packing problems can be problematic to conventional gradient-based optimisation methods due to discontinuities and/or severe nonlinearities in their objective function spaces' (Szykman and Cagan, 1995, page 309). The following topics are noteworthy in the existing methods that describe 3DIP problems as mathematical models (Udy et al., 1988; Scheithauer et al., 2005; Stoyan et al., 2004, 2005; Chernov et al., 2010): (i) the geometry representation, (ii) the detection of geometric feasibility, (iii) the optimisation objective and (iv) the limitations of this approach. These are discussed in the following paragraphs.

As mentioned in Section 2.2, geometry representation affects the efficiency of packing approaches, particularly given the number of variables and constraints in these problems. Most of the literature on this type of approach addresses convex primitives such as spheres or cylinders (Scheithauer et al., 2005), convex mesh objects (Stoyan et al., 2005), or the decomposition of non-convex objects into convex ones (Stoyan et al., 2004; Chernov et al., 2010). A common aspect of these methods is the existence of a considerable number of inequalities that increase in accordance with the number of parts (n), including the components that form each part decomposition (see Section 2.2). For example, the number of inequalities in the models described by Stoyan et al. (2004) and Chernov et al. (2010) is larger than $n!$, which makes these models unfeasible for problems of 'practical interest' (Stoyan et al., 2004, page 11).

The second aspect is the validation of geometric intersection, which is undoubtedly the most computationally expensive component used in processing three-dimensional objects (Scheithauer et al., 2005). One example of a mathematical tool for intersection testing is the no-fit polygon (NFP), which constructs a space around a polygon in within which a second geometry cannot be placed without resulting in overlap (Bennell and Song, 2008; Bennell et al., 2010). Figure 2.2 illustrates the construction of the NFP area (defined by the thick border) of the two-dimensional object P in relation to Q, i.e., P does not overlap with Q, provided that its reference point is not within the NFP.

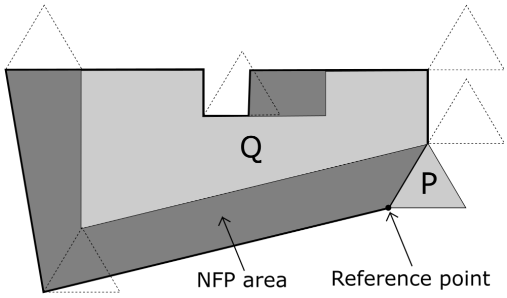


Figure 2.2: Example of the no-fit polygon (NFP) of a 2D object *P* in relation to *Q*.

Another mathematical tool for verifying geometric feasibility is the Φ-function, which returns the distance between two objects (in this context, called 'Φ-objects'). The signal of the value returned by the Φ-function indicates whether the two objects overlap, have tangent boundaries after being translated, or result in feasible placement (Stoyan and Yaskov, 1983; Stoyan et al., 2004, 2005). Other methods for overlap detection model geometries as the intersection of a fixed number of hyperplanes (Udy et al., 1988).

The main disadvantage of using the aforementioned functions to test for geometric feasibility is the computational cost (Bennell and Song, 2008). Dowsland et al. (2002) have studied the computational cost required for NFP calculations and have found that "several hours are quoted for problems of around 500 (two-dimensional) pieces" Dowsland et al. (2002, page 373). Despite the reduced amount of time required to construct Φ-functions compared to NFP (Stoyan et al., 2005), this function is still unfeasible for problem instances with more than five parts (Stoyan et al., 2004).

A third aspect to consider when addressing 3DIP problems with exact methods is the objective function. Several methods following this approach aim to minimise the volume height required to accommodate the entire demand for parts (Udy et al., 1988; Stoyan et al., 2004, 2005; Scheithauer et al., 2005; Chernov et al., 2010). Another recurring similarity is the adoption of decision variables that determine the values and directions for translation vectors (one per part), which are applied simultaneously with the current placement configuration (Stoyan et al., 2004, 2005; Scheithauer et al., 2005; Chernov et al., 2010; Paul and Anand, 2012).

Another disadvantage of using exact methods for 3DIP problems is the extreme complexity of these models, due to a high number of variables and constraints (Stoyan et al., 2004; Chernov et al., 2010). Moreover, additional restrictions related to part rotation are still problematic (Stoyan et al., 2004, 2005) and, in some models, considered an 'open problem' (Scheithauer et al., 2005). Another limitation is the need for promising starting positions, which can pose a barrier to finding effective local optimal solutions. For example, Udy et al. (1988) used an initial configuration provided by a human expert, whereas this process is automated in other methods by incorporating a placement heuristic (Stoyan et al., 2004, 2005; Chernov et al., 2010).

The major drawback of exact methods is that they cannot handle large number of geometries (considering non-convex objects as compositions of convex primitives) (Fasano, 2004). For example, the mathematical models proposed by Stoyan et al. “do not permit to obtain a proved optimal solution (...) for problems of medium size (number of parts greater than five) in acceptable time” (Stoyan et al., 2005, page 10). In another work, the same authors have suggested that such a method is recommended provided that the number of parts is ‘less than 250 and if the polytopes (geometries) are sufficiently different’ (Stoyan et al., 2004, page 25). Finally, the number of parts also considerably affects the number of operations and the processing costs. In the case of Φ-functions, for example, the number of computations grows in a degree-two polynomial rate function for each pair of objects (Scheithauer et al., 2005).

2.5.2 Packing approaches which build partial solutions

This section reviews the main approaches that incrementally construct solutions, including placement policies, notably the deepest bottom left (DBL) heuristic, search algorithms combined to a placement policy, and wall-building approaches.

Placement policies

A placement policy is a simple strategy which describes how the parts are accommodated, one at a time, within the available space. Most 3DIP algorithms use variants of the placement policy known as the bottom-left (BL) heuristic (Ikonen and Biles, 1997; Canellidis et al., 2006; Gogate and Pande, 2008; Canellidis et al., 2010, 2013). BL was first introduced by Art (1966) for solving the problem of arranging two-dimensional irregular polygons within a rectangular sheet with the smallest width possible. Such a procedure arranges a sequence of parts at the bottom-left-most available position within the available space. After being inserted, each part is then orthogonally slid towards the origin when no overlap is detected.

Dowsland et al. (2002) divide the variants of the BL heuristic into two classes: with hole filling and without hole filling (traditional BL). Although the former method results in more efficient use of the available space, it also leads to a significantly long processing time (Dowsland et al., 2002). BL variants can also be categorised according to the number of dimensions contained in the problem. While the majority of published work addresses two-dimensional problems, relatively few studies have extended the BL, notably the deepest bottom left with fill (DBLF) method proposed by Karabulut and Murat (2004), to the three-dimensional domain. DBLF interactively translates objects, one at a time, towards the origin while it maintains a list of internal spaces to be filled using later geometries.

Methods employing the DBLF suffer from a high computational runtime required for verifying geometric feasibility during the placement of each part (Burke et al., 2004). Real-world parts produced by AM means often involve complex entrant features and require expensive validations (Milenkovic and Schmidl, 2001; Dowsland et al., 2002; Dean et al., 2006; Bennell and Song, 2008). The integration of functions for testing geometric feasibility such as the no-fit polygon and Φ-functions, discussed in Section 2.5.1, require expensive computational processing before packing (Dowsland et al., 2002).

The volume density achieved by BL-based methods, including DBLF for three-dimensional problems, depends on the sequence and orientations in which parts are processed, among other aspects (Ikonen and Biles, 1997; Bennell and Oliveira, 2009). In the original implementation of the BL (Art, 1966), parts are sorted in increasing order of width (dimension on the x-axis), increasing value of area and the decreasing wasted area on the left of the part when packed. The sorting criteria area selected to tackle the objective function to be addressed. For example, the First-Fit Decreasing Height algorithm packs parts in decreasing order of heights (Koulamas, 1998).

Baumers et al. (2013) presented an example of DBLF algorithm applied to AM to investigate models for predicting energy consumption and production costs in laser sintering processes. Their packing algorithm packed items in decreasing order of priority, with the largest objects processed first. This method yielded promising results regarding the maximisation of the utilised surface area of the build platform and helped the authors to demonstrate that the full use of available machine capacity has an impact on process efficiency. Another use of DBLF algorithm for AM was demonstrated by Wu et al. (2014), who employed one such method to examine how certain packing configurations affect surface quality, the volume of supporting structures, and build volume utilisation (Li and Zhang, 2013). Wu et al. analysed the product quality and the expected manufacturing costs of the generated set of solutions to support robust decisions through Pareto-based heuristics.

Search algorithms integrated to a placement policy

The sequence of parts and their orientations in which they are arranged within the build volume play an important a role in packing efficiency (Ikonen and Biles, 1997; Bennell and Oliveira, 2009). Such configurations are often contained in a data structure referred to as a 'packing plan' which is inserted into a placement heuristic (Ikonen et al., 1996; Ikonen and Biles, 1997; Gogate and Pande, 2008). Identifying which packing plan leads to maximum volume utilisation is a challenging task due to the large search space that results from the many available permutations, which grows at a factorial rate, and the exponential number of rotations (Velleman, 2006).

Much of the literature on 3DIP applied to AM combines BL-based heuristics with search algorithms, particularly metaheuristics. These are high-level search algorithms that return satisfactory solutions in reasonable runtime. One of the first examples of such an approach was undertaken by Ikonen et al. (1996) to investigate the advantages of implementing genetic algorithms for part packing in rapid prototyping machines (GARP) (Ikonen et al., 1996, 1997; Ikonen and Biles, 1997; Ikonen et al., 1998). Genetic algorithms (GA) are population-based meta-heuristics inspired by the natural process of reproduction, involving the survival of the fittest individuals in a population, which evolves over several generations (John, 1992). To solve packing problems, each individual encodes a packing plan while its fitness function measures the resulting volume density and penalises overlaps between the geometries (Ikonen et al., 1997, 1998). Several methods have been published that extend the GARP approach (Ikonen et al., 1998). For example, Hur et al. incorporated a pre-processing stage to locally optimise the height of each part individually before packing "for achieving the minimum build height [to] reduce build time" (Hur et al., 2001, page 237). Gogate and Pande (2008) adopted a similar approach for maximising the weighted sum of the build volume height, metrics for the staircase effect, the volume of support structures and the total surface area in contact with the support structures (Alexander et al., 1998; Lan et al., 1997; Byun and Lee, 2006). Other methods have implemented discretised geometric representations (e.g. voxelised objects or octrees) to enable evaluation of more candidate solutions (Tiwari et al., 2010; Ravindran, 2003).

The studies mentioned above focus on 3DIP applied to LS technology, which allows higher degrees of freedom for placement layouts. Another AM technology that uses packing algorithms is Stereolithography (SLA), as illustrated in work conducted by Canellidis et al. (2006, 2010, 2013). These GARP methods reduce three-dimensional packing problems to the two-dimensional domain and integrate a pre-processing stage to maximise the utilisation of the printing platform and minimise the volume of supporting structures, which are attributes particularly relevant in SLA manufacturing. A summary of GARP approaches is presented in Table 2.1.

Another example of a meta-heuristic used to improve placement policies is Tabu Search (TS), which has largely been used for solving regular one- and two-dimensional packing problems (Lodi et al., 2004; Gendreau et al., 2008; Crainic et al., 2009). Tabu Search uses a local search procedure to explore the solution space beyond local optimum by discouraging the testing of previously-visited solutions (Glover, 1986). This meta-heuristic has been noted to be a more aggressive local optimum search strategy than GA, as it moves the best available solution at each iteration, devoting significant computational effort to exploring regions where solutions are good (Glover, 1989). However, its performance is highly sensitive to parameter settings, and it is "not unusual to see the performance of a procedure dramatically improve after changing the value of one or two key parameters" (Gendreau and Potvin, 2010, page 41). As a result, parameter tuning such a metaheuristic for 3DIP problems within AM can be costly.

Reference	Problem	# of runs	Elitism	Replacement strategy	Population	Crossover	PC	Selection scheme	Mutation	PM	Rotation
(Ikonen et al., 1996)	Min. overlap	NI	NI	Generational	NI	O1[c]	1.0	Ranking	Swap[c]	0.005 - 0.05	45
(Ikonen et al., 1997)	3DISP and min. overlap	5	NI	Generational	50	O1[c]	0.9	Roulette wheel	Swap[c]	0.01 - 0.2	45
(Ikonen et al., 1998)	3DISP and min. overlap	1	NI	Generational	100	O1[c]	0.8	Roulette wheel	Swap[c]	0.07 - 0.13	45
(Hur et al., 2001)	3DISP	NI1	NI1	Generational	NI1	O1	NI1	NI1	Swap	NI1	NI1
(Ravindran, 2003)	3DBP2	NI1	NI1	Generational	75	O1	0.7	Ranking	Non-uniform (angles)	0.12	90
(Canellidis et al., 2006)	2D Knapsack	NI1	NI1	NI1	50	SJX	NI1	Roulette wheel	Swap; rotate parts in 90	0.75	90
(Gogate and Pande, 2008)	3DISP	1	Yes	Generational	3N	Single point	0.75	Roulette wheel	Comple-mentary	0.2	45
(Tiwari et al., 2010)	3D Knapsack	1	Yes	Steady-State	N	Order 1; Single point	1.0	Roulette wheel	Swap; bit-flipping	$1/N$	90
(Canellidis et al., 2013)	2D Knapsack	NI1	NI1	Generational with weak replacement	50	SJX	1.0	Roulette wheel	Swap	0.0125	90

1 Not informed 2 Three-dimensional Bin Packing (3DBP) [c]Order 1 crossover (Falkenauer and Bouffouix, 1991) [d] (Bagchi et al., 1991)

Table 2.1: Reported characteristics of the GA implementations for AM in the 3DIP literature.

Vanek et al. (2014) have employed TS to solve 3DIP problems within manufacturing applications. This method first divides the input objects into smaller parts or segments. These segments are processed in sequence by a placement heuristic known as *height field based packing*, or Tetris packing (Sander et al., 2003), after the game Tetris. Tabu search is then used to find the sequence of segments that results in the optimised objective function, considering the minimisation of support material and a bounding box volume that accommodates all the segments and number of segments. The main drawback of this method is that the segments must be manually assembled to form the whole parts in the post-processing stages.

The application of metaheuristics including TS and GA shown in this section differ from those in the next section in the sense that this approach interactively improves the directives given to the DBLF heuristic, in the form of packing plans, instead of perturbing partial solutions (e.g. by translation and rotation).

2.5.3 Packing approaches which improve partial solutions

The 3DIP approaches presented in this section have similarities with *perturbative heuristics*, which are optimisation algorithms that interactively improve complete solutions via local moves (or perturbations) towards global optima (Szykman and Cagan, 1995). In the context of packing problems, these algorithms commonly represent current solutions as the explicit definition of the position and rotation of each part within the container, which may result in geometric overlap (Cagan et al., 1998). A recurring characteristic of these methods is that the mechanism for perturbing layout configurations is implemented by translating, rotating or swapping parts in the current complete solution.

Szykman and Cagan (1995, 1997) employed simulated annealing to optimise a randomly generated initial configuration applied to three-dimensional component packing. Simulated annealing is an optimisation method introduced by Kirkpatrick et al. (1983) and inspired in the physical annealing process used in metallurgy. In this process, metal is submitted to controlled heating and cooling until the particles in the material reach an equilibrium state (Černý, 1985). The SA algorithm follows this physical analogy by continuously accepting improved configurations, occasionally also accepting (with a gradually decreasing probability) a candidate solution with a worse measure for the objective function. Cagan et al. (1998) extended the SA-based methodology presented in previous work (Szykman and Cagan, 1995, 1997) to arrange more complex geometries. This addressed the problems of costly and time-consuming collision detection by representing each geometry as an octree (Carlbom et al., 1985), a voxel-like hierarchical data structure that allows quick geometric feasibility verification. The method was validated with instances comprising of up to seven arbitrarily shaped components within a non-rectangular container. Further literature has implementing more elaborate move operators (Zhang et al., 2002) and alternative strategies for coping with the high computational cost for collision detection (Li et al., 2007, 2010).

Another example of the use of metaheuristics to progressively improve complete 3DIP solutions is the guided local search (GLS) method, which is a penalty-based search optimisation algorithm that improves exploration by escaping local minima (Voudouris et al., 2010). Guided local search algorithms implement an evaluation function, which incorporates a penalty value that changes dynamically according to the characteristics of the latest local optimum to direct the search towards unexplored regions (Glover and Kochenberger, 2003). This method has been systematically employed by Egeblad et al. to solve 3DIP problems (Egeblad et al., 2007; Egeblad, 2009; Egeblad et al., 2009, 2010). Guided local search algorithms interactively improve an initial placement configuration obtained by the DBLF heuristic with regard to geometry overlap and build volume height, which is a feature that affects volume density and manufacturing time. A special component of these algorithms is an efficient procedure for perturbing a part in the current solution through orthogonal translations to minimise the resulting overlap (Egeblad, 2009). Empirical results have demonstrated an improved packing density using GLS over earlier 3DIP approaches (Ikonen et al., 1997; Stoyan et al., 2005).

A significant amount of work has been conducted on 3DIP algorithms that interactively perturb convex particles and mimic the physical shaking of non-overlapping particles (Lee et al., 2009). After generating a random feasible initial state, these algorithms perform successive perturbation to the parts, one at a time. Each part is translated towards a random direction, which is biased to the bottom of the container to simulate the downward movement generated by gravitational force. Similar 3DIP approaches have been applied to particle simulation (Jia and Williams, 2001; Rémond and Gallias, 2004). Interestingly, Lee et al. (2009) and Lutters et al. (2012) have demonstrated that in these methods larger particles tend to rise to the top in a phenomenon called the 'Brazil nut' effect (Rosato et al., 1987; Cleary and Sawley, 2002; Fraige et al., 2008).

The task of packing Tetris-like items into non-rectangular containers has been addressed by Fasano (Fasano, 2003; Fasano, Giorgio and Novelli, 2003; Fasano, 2004, 2008). To address this task, each part is modelled as a combination of smaller rectangles that have a fixed relative distance between them. A perturbative heuristic is then employed to interactively include, swap or remove items from the current subset of parts. After the perturbation, each subset is given to a mixed integer programming model. Unlike earlier models for three-dimensional regular packing (Chen et al., 1995; Fasano, 1999; Padberg, 2000; Fischetti and Luzzi, 2003), Fasano's formulation includes constraints to model practical aspects such as fixed position, orientation of the items and static balancing.

2.6 Available datasets for 3DIP

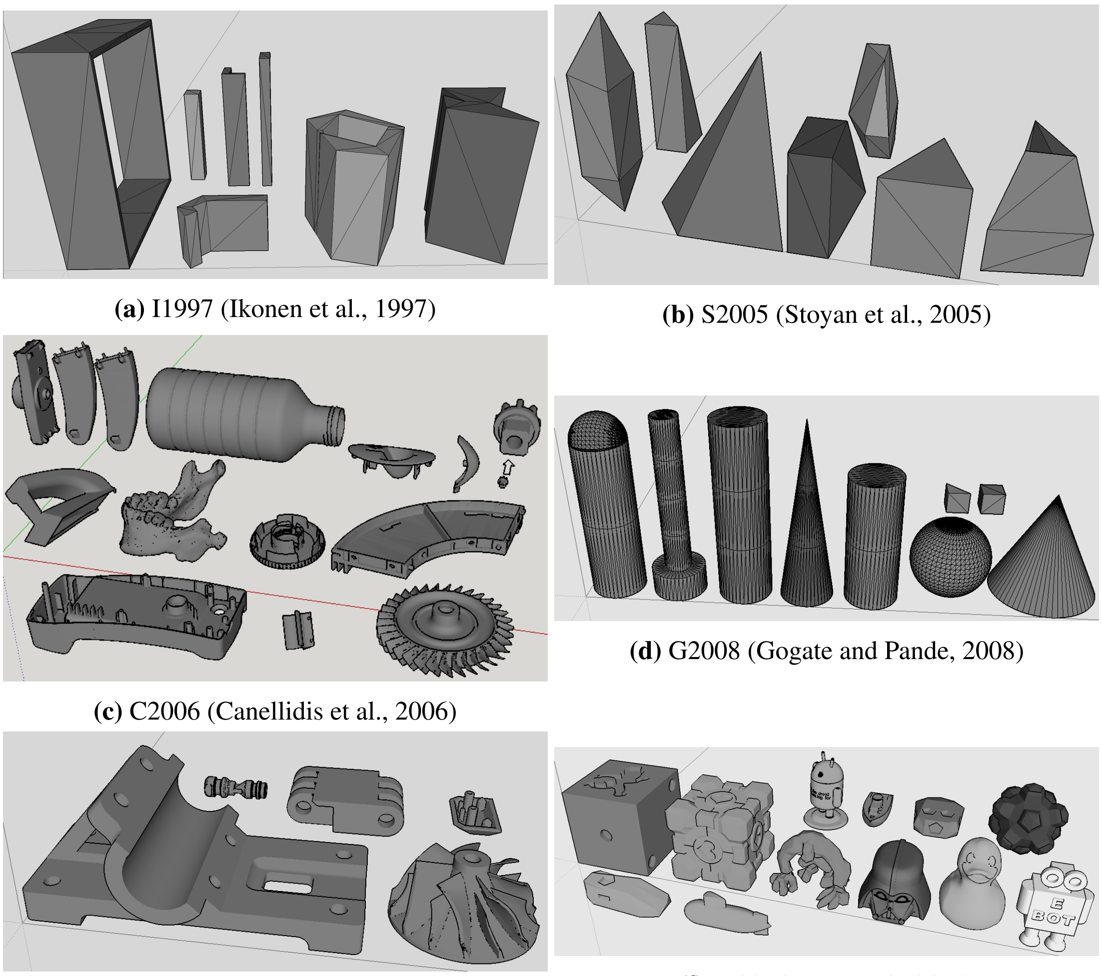
In the context of algorithm implementation, benchmark datasets are defined to reflect common problems and assess the usefulness of methods. Compared to the number of datasets for regular packing problems, there is a limited number of datasets available for 3DIP problems aligned with realistic AM features (Egeblad, 2009; López-Camacho et al., 2014). These are presented in Table 2.2. Most of the 3DIP datasets shown consist of only a few parts, 13 at most, even though AM processes often require the manufacturing of large numbers of items in a single build operation (Hopkinson and Dickens, 2003). Furthermore, these datasets present similar attributes with regard to quantities of instances, STL models, and facets of parts and contain little or no information on the complexity of such geometries (Araújo et al., 2018).

Table 2.2: Datasets for 3DIP problems.

Reference	Short-hand[a]	AM	# of instances	# of STL models	# of parts	# of facets
Ikonen et al. (1997)	I1997	LS[b]	1	7	7	12 - 52
Stoyan et al. (2005)	S2005	n/a	3	7	7 - 25	4 - 18
Canellidis et al. (2006)	C2006	SLA[c]	1	7	14	620 - 447,578
Gogate and Pande (2008)	G2008	LS	1	9	9	8 - 2,600
Canellidis et al. (2010)	C2010	LS	1	13	13	828 - 100,948
Baumers et al. (2013)	B2013	LS	8	5	5 - 100	12,712 - 107,806
Wu et al. (2014)	W2014	LS	1	12	12	124 - 33,520
Araújo et al. (2018)	A2018	LS	2,343	350	6 - 2,000	4 - 447,578

[a] Shorthand is assigned to each benchmark to aid the reader throughout this thesis.
[b] Laser Sintering [c] Stereolithography

A general benchmark for applying of 3DIP within AM should incorporate shapes with features that realistically reflect AM product geometry. As shown in Figure 2.3, this realism is lacking in several benchmarks encountered in the literature (Ikonen et al., 1997; Stoyan et al., 2005; Gogate and Pande, 2008). This is a significant drawback of these benchmarks, as the ability to producing complex objects is regarded as a prime motivation for adopting AM (Hague et al., 2003). The reasons for the use of visually simple representations of the composition of parts in the available datasets are related to the high computational costs required by packing algorithms. Several authors have noted that solutions often have long runtimes when handling 3D geometries due to the complexity of validating geometric feasibility (Scheithauer et al., 2005; Lee et al., 2009; Bennell et al., 2010). However, algorithms that implement strategies for shape simplification, such as reducing parts to their respective bounding boxes instead of the original definitions, overcome difficulties associated with collision detection of non-convex parts (Baumers et al., 2013; Canellidis et al., 2006; Wu et al., 2014).



(a) I1997 (Ikonen et al., 1997)

(b) S2005 (Stoyan et al., 2005)

(c) C2006 (Canellidis et al., 2006)

(d) G2008 (Gogate and Pande, 2008)

(e) B2013 (Baumers et al., 2013)

(f) W2014 (Wu et al., 2014)

Figure 2.3: Available benchmarks for 3D AM build volume packing.

Another aspect that 3DIP benchmarks should address is the possibility of different quantity requirements for each of the shapes, which limits the usefulness of the typology in Wäscher et al. (2007). This aspect is contained in the above definition of demand. Typically, in AM processes, especially those using LS, individual builds include large numbers of identical items to maximise process efficiency (Hopkinson and Dickens, 2003). This means that studies focusing on the insertion of single units of each geometry into the available build space (Hur et al., 2001) may not be reflective of real-world applications. This aspect is addressed in other studies (Stoyan et al., 2005; Baumers et al., 2013) which model variation in demanded quantity, as illustrated in Table 2.3.

Table 2.3: Characteristics of the demands in (Stoyan et al., 2005) and (Baumers et al., 2013)

Benchmark	Profile	Number of models	Quantity of available parts	Demand	
				Mean	Standard deviation
Stoyan et al. (2005)	A	7	7	1	0
	B	7	12	1.71	0.88
	C	7	25	3.57	1.29
Baumers et al. (2013)	A	5	∞	∞	n/a
	B	5	25	5	0
	C	5	15	3	0
	D	5	5	1	0
	E	1	∞	∞	n/a
	F	2	∞	∞	n/a
	G	2	∞	∞	n/a
	H	1	1	1	0

Table 2.3 shows that the mean and the standard deviation for the demand in each profile fall within a narrow range of values, provided that the infinite availability of parts is accepted even though this is not the case in the realistic scenarios. Other 3DIP benchmarks (Ikonen et al., 1997; Canellidis et al., 2006; Gogate and Pande, 2008; Wu et al., 2014) are equally limited with regard low demand variation, being characterised by the request for a single part for each model. Such lack of variation in demand requirements becomes evident when we categorise all the instances in the existing 3DIP benchmarks, being labelled as ‘few items assortments’ by Dyckhoff (1990) or ‘strongly heterogeneous’ according to Wäscher et al. (2007).

It is also noteworthy that only a few benchmarks have been applied more than once in studies on 3D packing. Table 2.4 presents the subset of studies which employed more than one of the datasets mentioned above. It is apparent from Table 2.4 that the lack of an extensive amount of data comparing different packing methods is one of the challenges for determining the ‘best’ algorithm for a particular input. Therefore, this study attempts to produce valuable resources to support decision-making during the development of 3DIP solutions and shine new light on the primary attributes of 3DIP problems in practical circumstances.

Table 2.4: Example of studies on 3DIP applied to AM and relevant shared datasets

Reference	I1997	S2005	G2008	Own
Dickinson and Knopf (1998)	✓			
Egeblad et al. (2009)	✓	✓		✓
Wu et al. (2014)			✓	✓

When addressing hard combinatorial optimisation problems such as C&P, researchers face the challenge of identifying the best algorithm. For this task, several aspects must be considered, which include (but are not limited to) shape complexity, demand variation, availability of parts, characteristics of the available build volume number. Therefore, it is desirable that test datasets contain a large number of cases representing a diversity of scenarios so that they can better support algorithms that tackle unrepresentative data and lead to unbiased results. Moreover, adding meaningful domain features can improve understanding of the problem and support more efficient data science techniques in this domain (Smith-Miles, 2009; Kotthoff, 2016).

2.7 Algorithm selection for combinatorial optimisation problems

Solutions to combinatorial optimisation problems commonly adopt one of the following strategies: improvement of the state-of-art algorithms or customisations that aim to address realistic constraints and objectives related to a particular problem domain (Kotthoff, 2016; Araújo et al., 2018). One limitation of these approaches is duplicated work for solving instances for which existing methods can obtain similar or better performance. Furthermore, newly proposed algorithms are likely to have lower performance for certain classes of instances due to different assumptions about the problem (Kotthoff, 2016; Wolpert and Macready, 1997).

Combining various algorithms into a framework to identify the most suitable algorithm for a new instance is at the core of the algorithm selection problem, which was first described by Rice (1976). Implementing an algorithm selection workflow begins with the extraction of the problem features, followed by prediction of which algorithm in a portfolio is likely to result in maximum performance. An essential part of such a computational tool is learning the functions that match feature space to algorithm performance, which is mostly based on past observation. Generating such mapping functions has been described for as a challenging or, as Rice puts it in his seminal work, 'nebulous', task (Rice, 1976; Muñoz et al., 2013). Preliminary work on algorithm selection used hand-crafted decision rules provided by experts and statistical models. Methods based on such mechanisms, notably expert systems, gained enormous popularity in the 1980s and 1990s (Jackson, 1998). By contrast, recent developments in algorithm selection employ data science techniques capable of automatically processing and analysing large amounts of data (Leyton-Brown et al., 2002). Moreover, most of the recent literature is dedicated to offline algorithm selection, i.e., the identification of the best algorithm occurs before solving the new instance (Kotthoff, 2016). Kotthoff has also observed that a clear reduction in the use of online methods, due to the inherent complexity and computational cost of this category of applications.

Although algorithm selection techniques date back to the 1970s (Rice, 1976), they have only recently gained increasing popularity, mainly as a result of due solver competitions for propositional satisfiability (SAT) and constraint satisfaction problems (CSP) (Kotthoff, 2016). Xu et al. (2008) introduced a framework called SATZilla, which was used to select the most suitable SAT solver from a portfolio for a given input problem. A similar approach was adopted for CSP by O'Mahony et al. (2008), for which they won the 2008 Constraint Solver Competition. For a detailed survey of algorithm selection approaches, we refer to Kotthoff (2014). Table 2.5 presents examples of selected studies that employ algorithm selection to solve combinatorial optimisation problems. The learning techniques employed and the category, i.e. whether the training took place online or offline, are also displayed.

Table 2.5: Algorithm selection on combinatorial optimisation problems in the literature.

Problem	Reference	Learning models	Prediction
Planning	De La Rosa et al. (2008); De La Rosa et al. (2011)	Decision tree	online
	Domshlak et al. (2010)	Naive Bayes classifier	online
	Garbajosa et al. (2014)	Classifier ensemble	online
Scheduling	Terashima-Marín et al. (1999)	Genetic algorithms	offline
	Petrovic and Qu (2002)	Case-based reasoning	offline
	Burke et al. (2003b)	Reinforcement learning	online
	Beck and Freuder (2004)	Hand-crafted rules	offline
	Cicirello and Smith (2005)	Reinforcement learning	online
	Burke et al. (2006)	K-nearest neighbour	offline
	Smith-Miles (2009)	Decision tree, neural networks, self-organizing maps	offline
	Burke et al. (2012)	Reinforcement learning	offline
Travelling salesman problem	Kanda et al. (2010, 2011)	K-nearest neighbour, decision tree, SVM, Naive Bayes	offline
	Smith-Miles and van Hemert (2011)	Decision tree, neural network and self-organizing map	offline
	Pihera and Musliu (2014)	ML classifiers	offline
	Kotthoff et al. (2015)	Classification, regression, pairwise regression	offline
Vehicle routing problem	Caseau et al. (1999)	Genetic algorithms	offline
2D Bin Packing	Burke et al. (2010)	Genetic programming	online
	López-Camacho et al. (2014)	K-Nearest neighbour	online

Burke et al. (2010) presented one of the first studies on the application of learning techniques for C&P, which aimed to minimise the length of a two-dimensional sheet to accommodate the entire demand of rectangular polygons. The authors used genetic programming (GP) to automatically generate evaluation functions that ranked pieces and vertices and indicated how packing should proceed. Genetic programming is a computational optimisation method for creating and evolving programs or functions using a process based on the biological evolution of individuals, similar to GA. One of the main elements that distinguishes GP from GA is the search space, which in GP consists of programs that run on solutions instead of assessing directly on solutions as in GA. A similar approach was developed by Pillay (2012) for solving the one-dimensional bin packing problem.

Another application algorithm selection for C&P was proposed by López-Camacho et al. (2014) to address one and two-dimensional bin packing problems. Such a method runs a GA that uses a set of 23 problem features to select one of six existing packing heuristics (Bennell and Oliveira, 2009). This technique belongs to a class of search methods referred to as hyper-heuristics, which are algorithms that search through a space of heuristics to select those that are most useful for generating solutions (Burke et al., 2003a, 2013). However, the use of hyper-heuristic methods has been restricted to one and two-dimensional packing Burke et al. (2009, 2010).

To date, the limited number of algorithm selection techniques published (including predictive statistical models and hyper-heuristics) apply only to regular packing. Some factors contribute to the lack of studies on the benefits of such computational methods for solving 3DIP problems, including the challenges for fulfilling prerequisites of learning models (Kotsiantis et al., 2006; Smith-Miles, 2009; Kotthoff, 2016):

- the consideration of a limited number of features that appropriately characterise the 3DIP-domain;
- the scarce availability of data representing non-similar instances and real-world scenarios; and
- the intensive computational effort required to labelling a significant number of 3DIP instances.

Interestingly, arguments in favour of using particular packing algorithms for addressing specific problem features are mostly supported by intuitions and display little or no statistical rigour, although there are a few exceptions (Allen et al., 2009). For example, some studies that employ wall-building (WB) algorithms have affirmed that such methods are suitable for problems involving several parts with few part types (Abdou and Elmasry, 1999; Pisinger, 2002; Araújo and Pinheiro, 2010; Bortfeldt and Gehring, 1998; Gehring H., 1997). The use of data science techniques for algorithm selection has been of great value in numerous fields, such as bioinformatics (Daly, 2009), medical diagnosis (Davatzikos et al., 2005; Tourassi et al., 2012), and image recognition (Bartlett et al., 2005). Because these computational tools are not restricted to a particular problem domain, it is expected that their application to 3DIP problems should lead to general performance improvements and an enhanced understanding of this problem domain.

2.8 Classification models in Machine Learning

For algorithm selection, it is essential to correctly predict which predictive statistical model will result in the best performance for unseen instances (Rice, 1975). Machine learning (ML) techniques are often employed for this task. Machine learning is a field within artificial intelligence that uses algorithms and models that generalise from existing (i.e., training) data to new settings or unseen observations (i.e., testing data) in the domain (Kotsiantis et al., 2006).

Machine learning techniques can generally be separated into three categories, depending on the characteristics of the data used during the learning process, i.e., training phase (Marsland, 2011). *Supervised learning* methods use previously labelled training data, i.e., observations that contain the correct outcome (Kotsiantis, 2007); *unsupervised learning* uses models that learn from unlabelled data (Murty et al., 2000); *reinforcement learning* entails interactively assessing the predicted outcomes obtained from unlabelled instances and tuning the parameters of the learnt model (Sutton and Barto, 2018).

An important aspect of ML is the learning task. Some algorithms are trained to predict a continuous-valued output (regression problems), while some are trained to predict a discrete number of values (classification problems) (Marsland, 2011). However, these categories are not mutually exclusive as classification algorithms can be utilised to predict a continuous value in the form of the probability of an instance belonging to one class label, while regression algorithms can be used to predict a value that falls within ranges of values (Criminisi et al., 2011). This section focuses on supervised ML methods for classification that have been used for algorithm selection applied to several combinatorial optimisation problems, including 2D packing (see Table 2.5).

2.8.1 Decision tree

Decision trees are classification and regression models that represent decision rules as hierarchical data structures (trees) that recursively partition the data (Murthy, 1998). These data structures are comprised of decision nodes, which include the root (topmost decision node) and a set of leaf nodes that correspond to classification decisions (Leo et al., 1984). At each node, the algorithm partitions the dataset by applying a threshold-based splitting rule on a feature (Shai and Shai, 2014). Figure 2.4 presents an example of a decision tree for determining a suitable packing algorithm for certain 3DIP features. In this example, if the number of parts is 4 and the mean demand is 2.7, then the DBLF heuristic should be used for packing.

At each node, one feature and a threshold value are selected to optimise a particular *impurity metric* (Murthy, 1998). The most commonly used impurity metrics in binary decision trees are

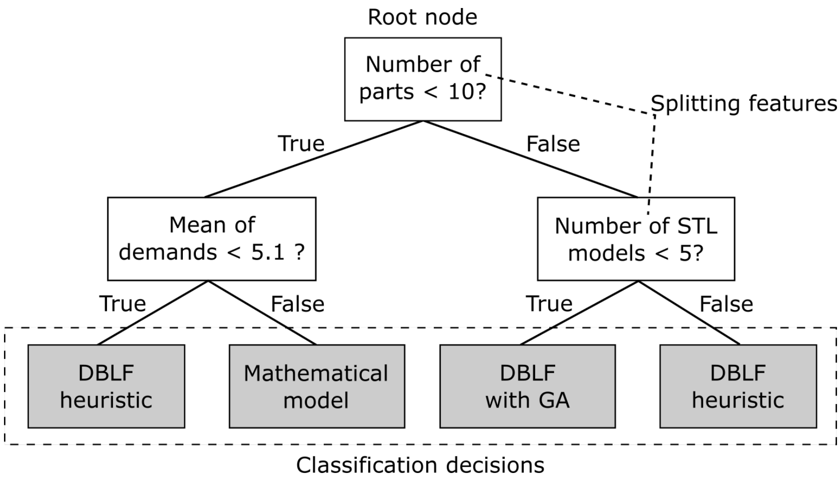


Figure 2.4: Example of a decision tree for determining the recommended packing algorithm (decision) given 3DIP features.

as follows.

- *Gini index* indicates the probability of misclassification. It quantifies how mixed the classes are in the two resultant branches (Leo et al., 1984) and ranges from 0 (representing perfect separation where all the samples belonging to the same category) to 0.5 (a split that results in fifty-fifty classification).

- *Entropy* attempts to maximise the mutual information in the decision tree and ranges from 0 (where all samples of a node belong to the same class) to 1.0 (representing a uniform class distribution) (Hunt et al., 1966). The entropy E of a split is given in equation 2.4, where p_i is the probability of class i in the training data.

$$E = \sum_i -p_i \log_2 p_i \tag{2.4}$$

It is also worth mentioning that such an algorithm results in a reasonable time complexity for decision trees containing features with discrete values. In this case, the complexity is $O(m * n^2)$, where m is the size of the training set and n is the number of features (Luckert and Schaefer-Kehnert, 2016). A characteristic of decision trees is that they tend to overfit based on the training data, leading to poor generalisability. One strategy to address this issue is to limit the depth of the tree, a technique referred to as *pruning*.

2.8.2 Random forest

A random forest (RF) is a classifier that generates an ensemble of decision trees, each of which casts a vote for the predicted output label (Breiman, 2001). To achieve diversity, each tree in the RF is trained using a random sample from the original data (known as a bootstrap sample) (Breiman, 1996). Then, determination of the splitting rule for each node of the decision tree is limited to a randomly selected subset of features (Marsland, 2011). This approach is adopted to add a degree of randomness to the learning process and to mitigate the overfitting effect that tends to occur with decision trees (Zadrozny and Elkan, 2001). Figure 2.5 illustrates the training of an RF that produces three decision trees.

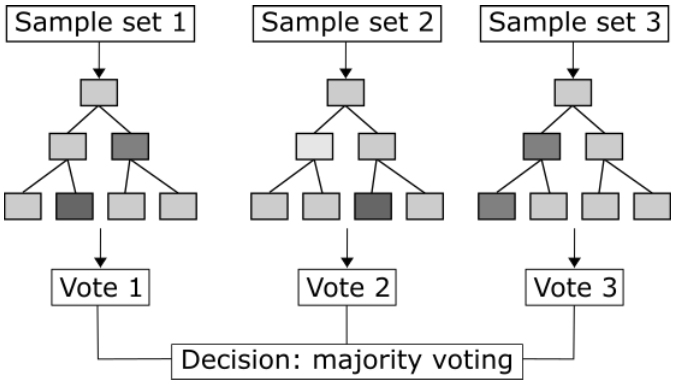


Figure 2.5: Training of an RF with three decision trees (Breiman, 2001).

2.8.3 Naive Bayes

Naive Bayes methods are statistical classification methods that uses Bayes' theorem to predict the probability that a certain observation belongs to a particular class (Murphy, 2012). Naive Bayes classifiers are said to be 'naive' in the sense that they assume independence between the predictors, i.e., class-conditionally independent features (Leung, 2007). This classifier calculates the probability $P(X|C)$ that an instance X with features $x_1, x_2, ..., x_n$ belongs to class C, as follows:

$$P(X|C) = \prod_{i=1}^{n} P(x_i|C) \tag{2.5}$$

The probability $P(x_i|C)$ can easily be estimated from the training set. Finally, the NB classifier assigns to X the class label that maximises equation 2.5. Naive Bayes classifiers are simple to implement and lead to competitive accuracy scores if the conditional independence assumption holds. These classifiers are often recommended for processing data streaming and large datasets as they requires less training time than other classifiers (Langley et al., 1992; Smola and Vishwanathan, 2008).

2.8.4 K-nearest neighbours

K-nearest neighbours (KNN) algorithms classify unlabelled instances using their nearest neighbours in the training set. This approach requires the following set of parameters: (i) the distance function to measure the similarity between observations, (i) the number of neighbours [k] to be considered, (iii) a weighting function to quantify the relevance of found neighbours, and (iv) an evaluation method of how to identify the label given the set of neighbours (e.g. majority vote) (Marsland, 2011). These elements are illustrated in Figure 2.6.

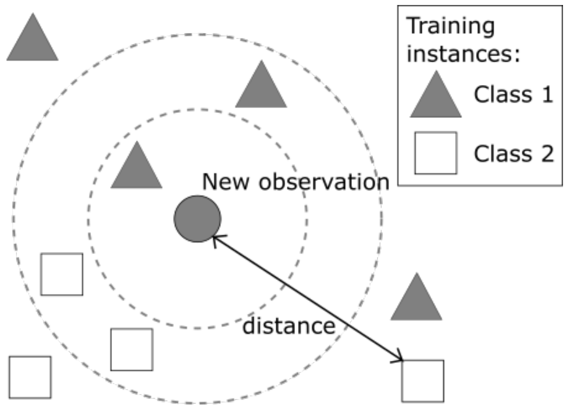


Figure 2.6: Main elements in KNN algorithms

KNN is one the first and most commonly used ML methods for pattern classification due to its straightforward implementation and competitive results (Cover and Hart, 1967). However, using this method for large datasets with an extensive number of features requires expensive computation, often involving dimensionality reduction prior to training (Andoni and Indyk, 2006).

2.8.5 Logistic regression

Logistic regression (LR) is a method that uses a statistical model to describe the relationship between a dichotomous dependent variable, for which only two possible outcomes are possible, and a set of independent variables (Cox, 1958; Kleinbaum and Klein, 2010). This method generally uses the logistic function (Pearl and Reed, 1920) to identify a fitted linear combination of the independent variables and output the corresponding probability of a sample belonging to one of the output classes (Walker and Duncan, 1967). An important element in LR is the use of a loss function to measure the fit between the mathematical model and the actual data. The parameters of the model are adjusted using optimisation techniques, such as the gradient descent algorithm (Rumelhart et al., 1986), to minimise the loss function. In many of these models, the loss function is described as maximum likelihood (Albert and Anderson, 1984). Despite their time efficiency, LR has several disadvantages, including a tendency to overfit the model, a high sensitivity to outliers and the requirement for an appropriate estimation of the sample size (Peduzzi et al., 1996).

2.9 Summary

This chapter presents relevant work on 3DIP problems in the AM sector and approaches that have been reported in the literature. Its purpose is to provide the reader with an understanding of the context of the present study. First, the chapter provided an overview of the existing variables and measurements for characterising three-dimensional geometries that are relevant for AM. Concerning the available benchmark data on 3DIP problems, there is an apparent need for more observations and studies on the 3DIP-domain within AM to support better decision-making. The first research direction investigated in this work aims to fill the gap in provided resources regarding tools and data to promote the development of algorithms for applications in this field.

The review of existing taxonomies demonstrates developments in understanding of the possible objectives and constraints in C&P. These classification criteria are essential to aid researchers and practitioners to appropriately identify the problem under investigation and devise solution methods. However, this chapter shows that there is a need for more explicit terminology and classification criteria. This requirement is satisfied by the introduced taxonomy yielding unambiguous categorisation criteria and a terminology which is familiar to AM practitioners.

Most state-of-art approaches for solving 3DIP problems combine search algorithms with variants of the BL heuristic (Art, 1966). The performance of such a strategy depends on the sequence, orientation of the parts, and parameters of the search algorithm. However, there has been little consideration of the choice of algorithm parameters and how these choices affect performance with regard to volume utilisation. Furthermore, empirical results on algorithm selection for 3DIP problems are scarce in the literature (Allen et al., 2009). This study includes the experimental analysis of the main algorithmic approaches for 3DIP reported in the literature.

Chapter 3

Introducing features, a taxonomy and a dataset for 3DIP in AM

Three-dimensional irregular packing (3DIP) problems are combinatorial optimisation problems, in which a set of arbitrary volumetric items must be placed into containers, or build volumes, in such a way that the total empty space is minimised (Wäscher et al., 2007). These optimisation problems are classified as NP-hard problems (Garey and Johnson, 1979). In addition to overcoming the computational challenge inherent in 3DIP problems, a number of ancillary issues must be addressed when proposing practical solutions for AM. These include, but are not limited to, the following:

- Identification of the characteristics and features of 3DIP problems which are the most pertinent for the construction and tailoring of solution algorithms in this domain.
- An identification of the problem class based on the existing taxonomies from the scientific literature and the technological requirements of a particular AM system. This will normally require relating the problem under investigation to precedents from the OR literature, which proposes methods for efficient solutions.

These issues are recurring and, if not met adequately, are liable to lead to misidentification of problems or the choice of a poorly performing algorithm for a particular instance. This section introduces new features of 3DIP problems that are relevant for AM applications and extends the existing general C&P taxonomies (Dyckhoff, 1990; Wäscher et al., 2007). To support the development of packing solutions and enable performance comparisons, this chapter also assembles a new and more realistic 3DIP benchmark with an enriched set of features.

3.1 Part complexity measurement methods to support 3D printing

Previous research on the evaluation of part complexity, sometimes referred to as shape complexity, has been applied to areas other than 3D printing, such as architecture (Psarra and Grajewski,

2001) and design (Turner and Penn, 1999). In this section, a shape complexity used in architectural applications is extended to support the complexity measurement within 3D printing processes. By discussing the proposed complexity measure in the context of those measures that have already been identified in the literature, the authors expect to lay the groundwork for future research into irregular packing.

The proposed metrics for measuring the characteristics of given 3D parts are a 3D version of MCV and v-value and a practical approach for calculating SR. The MCV measure, mentioned in Section 2.3, assesses two-dimensional geometries by evaluating the connectivity between *areas* on their perimeters (Psarra and Grajewski, 2001). This research extends (Psarra and Grajewski, 2001) this measure to a 3D situation, by considering the way in which the voxelisation can affect the areas that are connected by the algorithm and the effects of the granularity of the voxelisation upon the value of the metric. An optimisation method for calculating the minimum bounding box of a part is also proposed to support the calculation of the Spies Ratio (SR) (Spies, 1957).

3.1.1 Calculation of MCV and v-value

The first step of the MCV calculation is to rasterise the geometric model of the part into a 3D voxel grid with an appropriate resolution, considering performance and execution time. This process is called voxelisation (Min, 2004). In contrast to the original approach (Psarra and Grajewski, 2001), which calculates the connectivity between areas on the perimeter, this technique calculates the connectivity between voxels on the surface.

Following the discretisation process, the connectivity value (CV) of each voxel belonging to the surface of the part is calculated. This value is the percentage of other surface voxels that can be connected to the one in question by a Bresenham's line (Bresenham, 1965), such that all voxels on that line belong to the interior of the shape. A voxel with a low CV has poor interconnectivity to other portions of the part. The MCV of a part is then formed by calculating the mean of the CV values of all the surface voxels. For fully convex parts, the MCV is equal to 1. Entrant features reduce the MCV value, so the more concave the shape, the closer the MCV will be to 0.

The resolution of the voxelisation has a significant effect upon performance, in terms of both runtime and accuracy. To visualise the differences in this regard, Figure 3.1 illustrates an example of part used by Baumers et al. (2013), which was voxelised using a grid of 20, 100 and 200 parts, respectively. It was found that low resolution led to fast but imprecise MCV calculations, as might be expected from the visualisation. However, high resolution resulted in computationally expensive processing. For example, tests conducted on a 3.40GHz Intel PC with 8.00 GB RAM showed that resolutions above 100 voxels resulted in a runtime of over 5 minutes per part for MCV calculation. The calculations of the CV and MCV metrics for

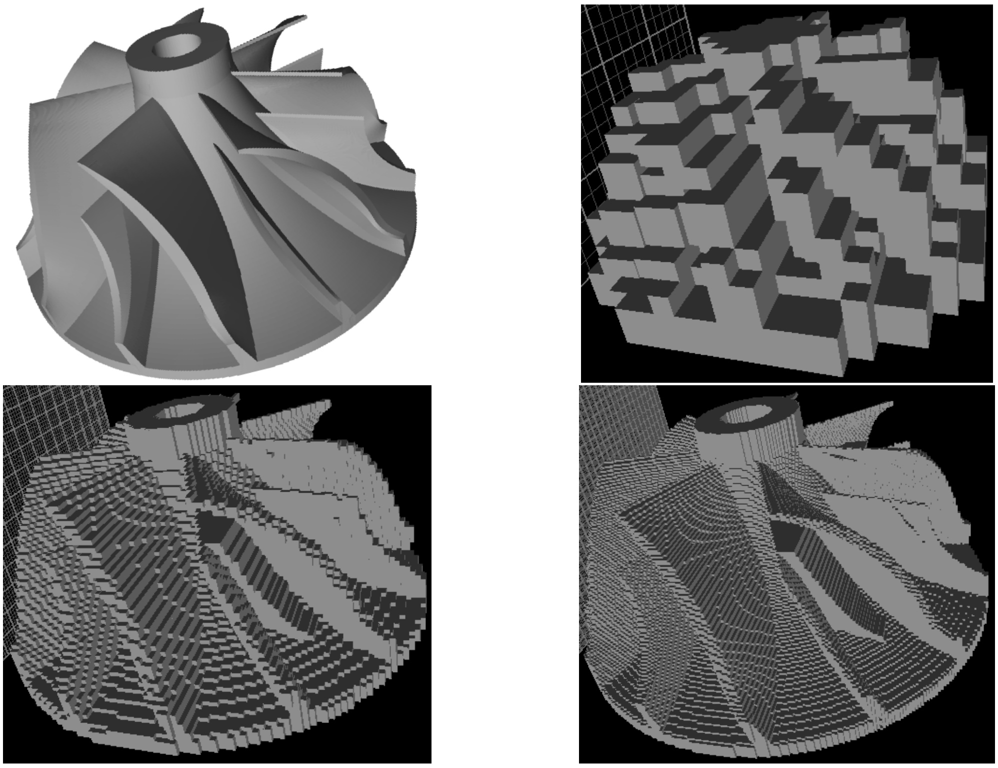


Figure 3.1: An example from (Baumers et al., 2013) rasterised in resolutions of 20, 100 and 200 voxels.

voxel pairs and parts, respectively, enable the extension of the auxiliary metric v-value, which describes the variation of connectivity through the surface of the part. This measure is obtained by adding together the standard deviation of the CV values of all the voxels on the surface of the part.

3.1.2 Using local search to calculate the Spies ratio

Certain non-trivial characteristics of a part, such as the volume of the minimum bounding box (MBB), the maximum and minimum dimension of any of the axes, the projected area on the building platform and the volume of supporting structures, depend upon the part's orientation. MBB can be obtained the product of the height, width and depth of the part after being rotated. This study presents a practical approach for determining the orientation of parts in order to calculate the SR (Spies, 1957). The objective is to determine the rotation of the original object about the x, y and z-axes that minimises the MBB without modifying its structure.

This study uses the steepest ascent hill climbing with random restart (SAHC) (Sean, 2009), which is a greedy local search metaheuristic that can often eventually reach global optimum for problems with relatively small search space (Selman and Gomes, 2006). SAHC starts with a random solution, determines a neighbourhood around the solution (the set of solutions that can be reached with a single change) and iteratively changes the current solution to the best one in its current neighbourhood. Each solution in the SAHC implementation is represented by an array containing the rotation on the x, y and z-axes. The neighbourhood of a solution corresponds to a small variation of the angles ($\pm 1^{\circ}$) on all the axes. The SAHC which is used is presented in Algorithm 1.

Algorithm 1: Pseudo-code of the steepest ascent hill climbing with random restart

```
1  currentSolution = randomSolution();
2  bestSolution = currentSolution;
3  while not termination criterion do
4      bestNeighbour = NULL;
5      for each solution sol in the neighbourhood of (currentSolution) do
6          if bestNeighbour == NULL or evaluate(sol) > evaluate(bestNeighbour) then
7              bestNeighbour = sol;
8          end
9      end
10     if evaluate(bestNeighbour) > evaluate(currentSolution) then
11         currentSolution = bestNeighbour ;
12         if evaluate(bestNeighbour) > evaluate(bestSolution) then
13             bestSolution = bestNeighbour;
14         end
15     end
16     else
17         currentSolution = randomSolution();
18     end
19 end
20 return bestSolution;
```

The starting random solution (line 1) is a tuple (r_x, r_y, r_z) with the rotation around the x (r_x), y (r_x) and z-axes (r_x), which range between 0° and 360°. The neighbourhood of a solution (line 6) is the set of tuples that can be obtained by either adding or subtracting 1° to at least one element of the original tuple. Evaluating a candidate solution (lines 7, 11 an 12) returns $-1 * MBB$ (as a minimisation objective function), where MBB is the minimum bounding box of the part after being rotated using r_x,r_y and r_z. The algorithm terminates when it exceeds a previously defined number of tested solutions.

3.1.3 Experimental design

The following datasets, which have all been used in previous 3D printing publications and have been named according to their first author's last initial and the years in which they appear, were evaluated: I1997 from Ikonen et al. (1997); S2005 from Stoyan et al. (2005); C2006 from Canellidis et al. (2006); G2008 from Gogate and Pande (2008); B2013 from Baumers et al. (2013); and W2014 from Wu et al. (2014). It can be observed in Figure 3.2 that the objects range from simple convex polygons (e.g. Stoyan2005) to elaborate parts with holes and entrances (e.g. Baumers2013).

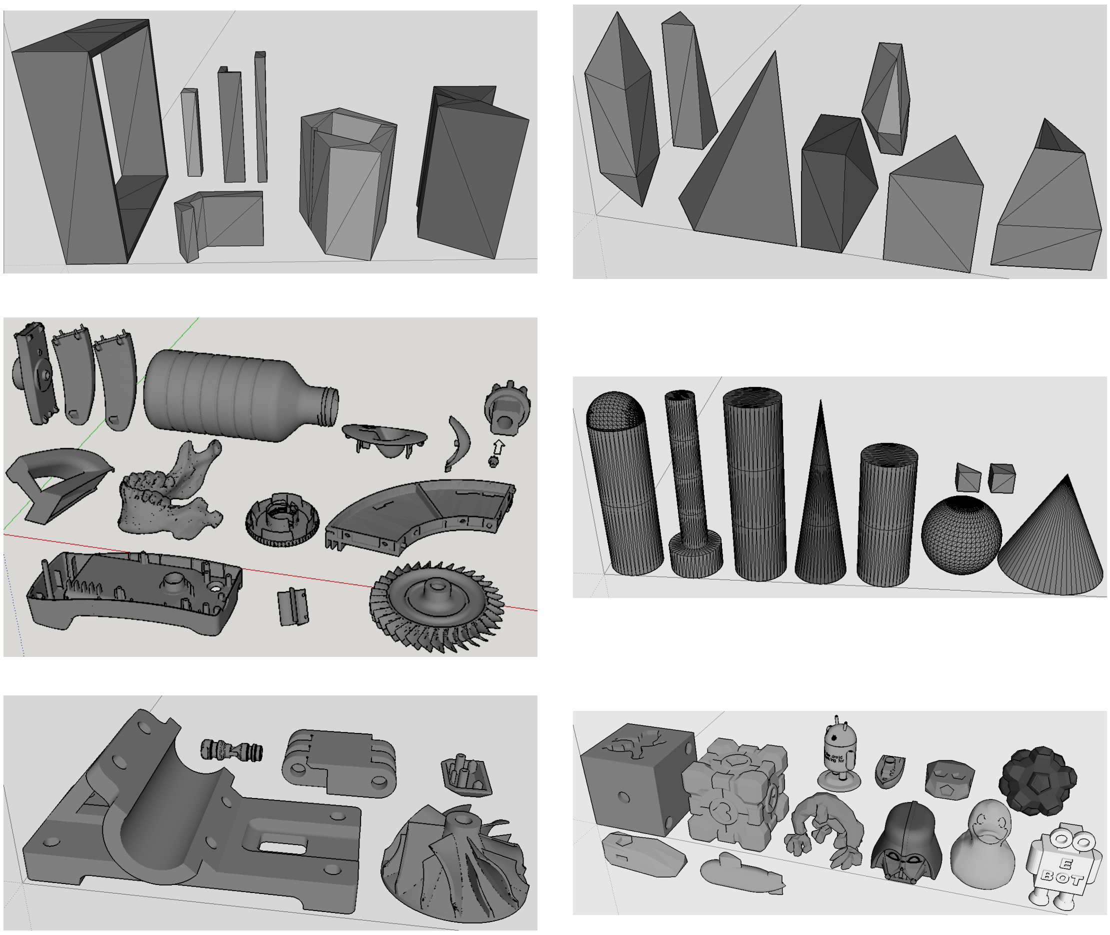
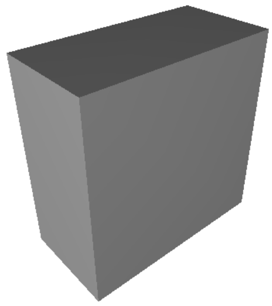


Figure 3.2: Datasets I1997 (Ikonen et al., 1997), S2005 (Stoyan et al., 2005), C2006 (Canellidis et al., 2006), G2008 (Gogate and Pande, 2008), B2013 (Baumers et al., 2013) and W2014 (Wu et al., 2014).

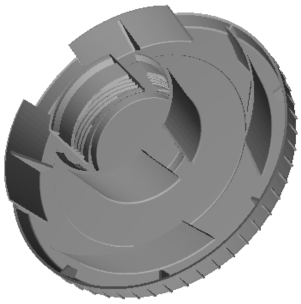
Table 3.1 shows the calculated values of the MCV, the v-value, the SR (Spies, 1957), the metric proposed by Valentan et al. (2011), the volume and dimensions of the minimum bounding box achieved by the SAHC, the number of facets, the surface area and the volume of the parts. The SAHC used in these experiments was terminated after it had visited a certain number of candidate solutions and returning the best solution at that time. Preliminary experiments conducted on a 2.16 GHz 64-bit Pentium with 8 GB of memory showed that up to 10,000 evaluations can be performed within five minutes, which is a reasonable amount of time for part assessment according to interviewed AM specialists.

3.1.4 Results and discussion

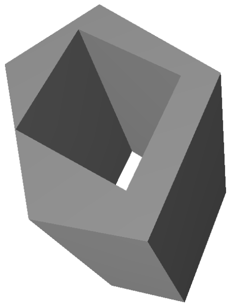
The results allow for an improved understanding of the correlations between different part complexity metrics. For example, Figures 3.3a and 3.3b illustrate the relation between the MCV and the v-value of a simple and complex part. The simple convex part has a high MCV and low v-value, while the complex part has a low MCV and a comparatively high v-value. The other two pairs of figures (Figure 3.3c and 3.3d, Figure 3.3e and 3.3e) show parts with the same MCV and different v-values. The part in Figure 3.3c has a lower v-value than the part in Figure 3.3d, which means a smaller variation of connectivity over the surface of the part and, consequently, lower complexity. The same applies to the parts shown in Figures 3.3e and 3.3f, which illustrates how the MCV and v-value metrics interact to indicate a higher degree of part complexity.



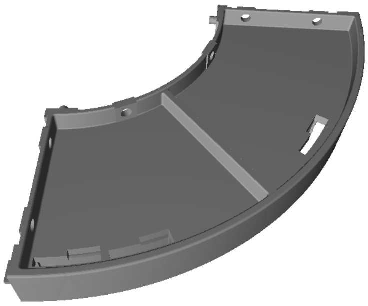
(a) I1997-1-1; MCV = 1.00; V-Value = 0.00



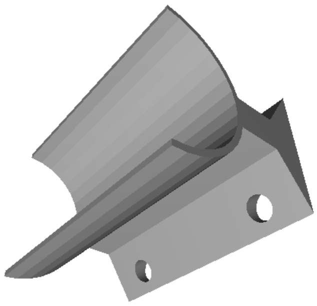
(b) C2006-10. MCV = 0.22; V-Value = 0.20



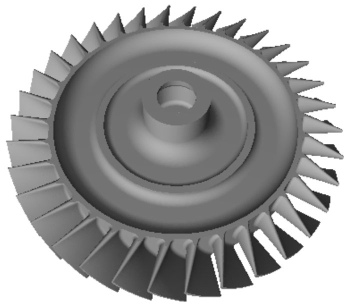
(c) I1997-c2-4. **MCV = 0.32**; V-Value = 0.08



(d) C2006-11. **MCV = 0.32**; V-Value = 0.23



(e) C2006-13. **MCV = 0.32**; V-value = 0.09



(f) C2006-14. **MCV = 0.32**; V-value = 0.20

Figure 3.3: Comparison between a part with high MCV and low v-value to another with low MCV and comparatively high v-value; and comparisons between parts sharing the same MCV but different v-values.

The second observation from the data in Table 3.1 concerns the relationship between the

Part	MCV	V-value	SR	Valentan	Vol. MBB	Facets	Surface	Volume
I1997-1-1	1.00	0.00	1.00	48.00	4.00	12	16.00	4.00
I1997-2-1	1.00	0.00	1.00	172.00	0.18	12	2.58	0.18
I1997-2-2	0.75	0.10	0.56	148.35	5.19	24	17.81	2.88
I1997-2-3	0.69	0.18	0.38	405.04	0.77	28	4.28	0.30
I1997-2-4	0.36	0.08	0.39	489.01	5.73	52	20.88	2.22
I1997-2-5	0.94	0.15	0.52	366.56	0.31	20	2.96	0.16
I1997-2-6	0.74	0.15	0.42	411.10	0.58	20	5.00	0.24
I1997-2-7	0.28	0.06	0.12	1061.21	10.50	48	27.71	1.25
S2005-1	1.00	0.00	0.73	14.93	240.00	14	187.68	176.00
S2005-2	1.00	0.01	0.17	10.33	429.63	4	192.88	74.67
S2005-3	1.00	0.00	0.83	14.55	144.00	10	174.59	120.00
S2005-4	1.00	0.00	0.65	21.83	192.00	16	170.11	124.67
S2005-5	1.00	0.00	0.42	23.44	320.00	18	173.59	133.33
S2005-6	1.00	0.00	0.50	10.42	294.00	8	191.40	147.00
S2005-7	0.86	0.16	0.30	23.18	513.71	16	220.92	152.50
C2006-1	0.19	0.08	0.14	18269.28	202730.78	17718	29440.74	28552.36
C2006-2	0.43	0.11	0.17	10431.58	118113.17	10394	20624.27	20549.98
C2006-3	0.43	0.11	0.17	10407.18	117966.54	10370	20624.30	20550.63
C2006-4	0.56	0.11	0.02	297226.88	1945215.98	75432	131365.20	33338.64
C2006-5	0.15	0.08	0.09	15633.79	131076.30	12498	14709.57	11759.15
C2006-6	0.18	0.06	0.07	13990.58	27808.22	6552	4427.49	2073.46
C2006-7	0.46	0.10	0.37	1733.93	1474.44	1420	659.05	539.73
C2006-8	0.20	0.08	0.08	10245.34	345621.17	9698	29844.34	28249.97
C2006-9	0.38	0.12	0.10	31920.49	666950.26	100948	20355.35	64373.44
C2006-10	0.22	0.20	0.07	148936.88	82525.65	50910	16474.17	5631.24
C2006-11	0.36	0.23	0.19	8323.79	486592.01	14353	52233.99	90068.93
C2006-12	0.22	0.11	0.08	32060.56	640822.91	30004	57384.79	53703.78
C2006-13	0.31	0.09	0.23	799.04	9953.23	620	2958.71	2295.75
C2006-14	0.31	0.20	0.16	223892.84	490003.69	447578	40348.61	80659.79
G2008-1	1.00	0.00	0.75	2454.55	27.91	1088	47.05	20.85
G2008-2	0.66	0.16	0.28	2101.98	27.92	524	31.36	7.82
G2008-3	1.00	0.00	0.26	866.84	27.62	250	25.32	7.30
G2008-4	1.00	0.00	0.30	286.61	55.86	118	40.62	16.72
G2008-5	1.00	0.00	1.00	72.00	1.00	12	6.00	1.00
G2008-6	1.00	0.00	0.78	760.72	27.92	332	50.20	21.91
G2008-7	1.00	0.00	0.78	596.59	19.94	248	37.65	15.65
G2008-8	1.00	0.00	0.52	5215.85	26.79	2600	28.19	14.05
G2008-9	1.00	0.00	0.50	70.63	1.00	8	4.41	0.50
B2013-1	0.45	0.17	0.19	3806.47	504085.52	12712	28939.36	96645.28
B2013-2	0.33	0.15	0.54	7768.05	30971.08	16000	8056.92	16594.99
B2013-3	0.35	0.20	0.23	33761.81	7565.57	26892	2216.72	1765.66
B2013-4	0.44	0.18	0.25	60785.56	81645.58	107806	11625.39	20618.17
B2013-5	0.41	0.09	0.44	57649.74	3015.40	59294	1277.93	1314.38
W2014-1	0.46	0.13	0.18	48919.81	11064.83	33520	2845.49	1949.74
W2014-2	0.81	0.13	0.49	8162.00	13647.05	22776	2374.04	6624.74
W2014-3	0.78	0.15	0.94	861.41	27000.00	3555	6148.68	25375.45
W2014-4	0.91	0.15	0.41	4642.77	2876.30	7350	753.48	1192.83
W2014-5	0.76	0.11	0.37	332.62	18124.40	706	3188.20	6767.17
W2014-6	0.70	0.15	0.38	3582.17	17636.75	10960	2183.87	6681.77
W2014-7	0.71	0.11	0.76	1255.94	26994.03	4730	5420.57	20414.48
W2014-8	0.86	0.12	0.64	76.37	2814.30	124	1116.93	1813.62
W2014-9	0.20	0.11	0.15	833.71	14272.39	892	1979.49	2117.90
W2014-10	0.87	0.13	0.66	74.43	4724.65	144	1612.59	3119.68
W2014-11	0.61	0.21	0.45	1973.88	5685.25	3172	1584.00	2545.48
W2014-12	0.83	0.18	0.40	5474.86	3016.10	6966	951.77	1211.00

Table 3.1: Table 3: Features of the parts from datasets I1997 (Ikonen et al., 1997), S2005 (Stoyan et al., 2005), C2006 (Canellidis et al., 2006), G2008 (Gogate and Pande, 2008), B2013 (Baumers et al., 2013) and W2014 (Wu et al., 2014).

MCV, the v-value, the additional part complexity measurement techniques and the basic characteristics of parts. The correlations between each pair of features are presented in Table 3.2.

	MCV	V-value	SR	Valentan	VMBB	Facets	Surface	Volume
MCV	n/a	-0.42	**0.75**	-0.26	-0.27	-0.25	-0.38	-0.41
V-value	-0.42	n/a	-0.36	0.28	0.21	0.32	0.24	0.40
SR	**0.75**	-0.36	n/a	-0.34	-0.39	-0.25	-0.44	-0.37
Valentan	-0.26	0.28	-0.34	n/a	**0.71**	**0.70**	**0.72**	0.33
VMBB	-0.27	0.21	-0.39	**0.71**	n/a	0.38	**0.93**	0.64
Facets	-0.25	0.32	-0.25	**0.70**	0.38	n/a	0.36	0.52
Surface	-0.38	0.24	-0.44	**0.72**	**0.93**	0.36	n/a	0.64
Volume	-0.41	0.40	-0.37	0.33	0.64	0.52	0.64	n/a

(*) Strong correlations ($\geq$0.7) are shown in bold.

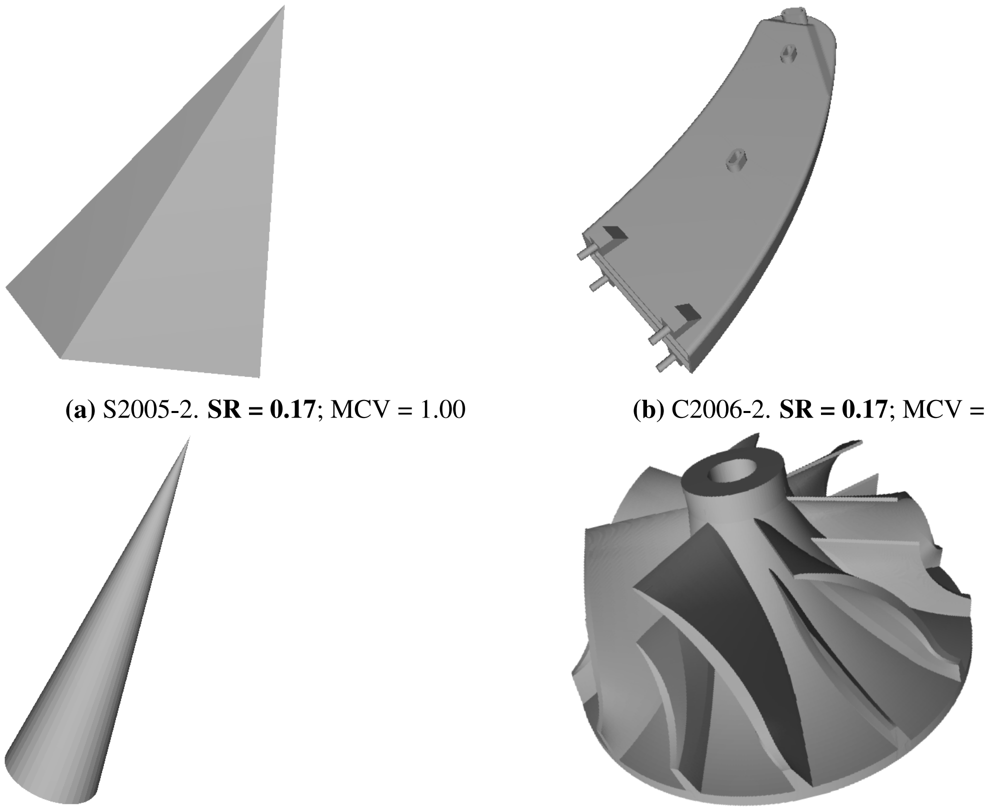
Table 3.2: Correlations between pairs of part features.

It can be observed that the MCV measure presents a low negative correlation to the metric used in Valentan et al. (2011) and a low negative correlation to various more basic metrics, which show moderate with high positive correlations to each other. The correlation values between the MCV and all the other features is also reported in Table 3.2. The lack of a high (positive or negative) correlation with another metric, except possibly the SR, indicates that the use of the MCV with this set of features does not result in data redundancy. Despite the comparatively high correlation between the SR and the MCV, the MCV appears to be more successful in expressing non-convexity. For illustrative purposes, Figure 3.4 shows two pairs of parts with similar values of SR but different MCV values. The parts in Figures 3.4a and 3.4b have a relatively low ratio between the volume and the MBB but the low mean of connectivity on the surface and shape convexity is better captured by the MCV, which is 1 for the convex part. The same applies to the second pair of parts, shown in Figures 3.4c and 3.4d.

These results show that the MCV and the extended v-value work well in capturing higher complexity and irregularity. Additionally, the results indicate that the combined use of the MCV along with existing metrics does not result in data redundancy. This, therefore, provides new information which can be of potential advantage for heuristic and hyper-heuristic based packing methods, since the information on the parts and state can be used to improve packing decisions. It is expected that such improvements would further enhance the efficiency and cost performance of 3D Printing technologies.

3.2 Analysis of irregular three-dimensional packing problems in additive manufacturing: A new taxonomy and dataset

This section introduces an extended taxonomy that addresses the limitations of existing typologies (Dyckhoff, 1990; Wäscher et al., 2007). This extension is formulated based on a review



(a) S2005-2. **SR = 0.17**; MCV = 1.00 **(b)** C2006-2. **SR = 0.17**; MCV = 0.43

(c) G2008-3. **SR = 0.26**; MCV = 1.00 **(d)** B2013-4. **SR = 0.25**; MCV = 0.44

Figure 3.4: Comparison between parts with the same SR but different MCV values.

of the categorisation criteria for packing problems and the classifications of 3DIP problem instances encountered in AM. It also introduces a new dataset, which covers a wide variety of features and contains instances that more closely simulate real-world scenarios than existing datasets. We discuss the outcomes of our analysis and identify the links between our analysis and AM technology operations.

3.2.1 An alternative taxonomy for packing problems

To address the limitations of the existing taxonomies, this section presents a taxonomy for 3DIP problems that is comprised of a terminology and categorisation criteria for problems and instances, which can be used to realistically describe 3DIP problems found in AM. A taxonomy is a useful resource for researchers and practitioners since ambiguous terminology and imprecise classification criteria can lead to miscommunication. The terms *nesting* and *packing*, for example, have sometimes been used interchangeably, although nesting is normally associated with a pattern in which some larger objects envelope one or more smaller objects. To eliminate this ambiguity, we introduce a consistent terminology (illustrated by the example in Figure 3.5), which is in accordance with the usual nomenclature employed in the AM community:

- **Model**: A comprehensive definition of an object such that the construction of multiple objects of that type is possible by referring only to the definition. In an AM context, a model is usually representation as a three-dimensional mesh in the STL format (or similar) (Gardan, 2016). From a digital model in this format, it is possible to calculate the volume, surface area and the number of triangles, or *faces*, of the object.

- **Part**: A *part* or *item* is a single physical instantiation of a model. Extant taxonomies (Dyckhoff, 1990; Wäscher et al., 2007) used the term 'small item' instead.

- **Multiplicity**: Quantities of parts to be produced per *model*.

- **Demand profile**: Additional information describing a client's requirements regarding parts, such as the due date or profit values.

- **Container**: The geometry and dimensions of the volume available to accommodate the requested *parts*. In AM, this corresponds to the specifications of the available space within the AM system and can be associated with build speed and cost parameters.

- **Build**: A group of *parts* produced simultaneously by an AM system at a particular time in a single operation. The assignment of a *part* to a particular build is constrained by volume capacity and geometrical non-overlapping conditions.

- **Time horizon**: Estimated length of time necessary to complete production, which can be continuous or discretised into equal time intervals.

- **Problem instance**: Refers to a single input problem and includes information on one or more containers or parts, multiplicity, and demand profile. Sometimes, it also includes a planning horizon, which requires the specification of the due dates for the requested parts.

- **Dataset**: A set of problem instances for assessing and comparing solutions.

3.2.2 Categorisation of three-dimensional irregular packing problems

The proposed categorisation criteria and notation refine the most recent taxonomy (Wäscher et al., 2007) and capture 3DIP problems using four-tuples $D|C|B|A$. Although there are similarities with the $\alpha/\beta/\gamma/\delta$ format proposed by Dyckhoff (1990), this formulation is intended to address the aforementioned issues with the previous taxonomies, with the addition of including further information regarding AM-related objectives and features of the problem. Table 3.3 summarises the encoding scheme described in this section.

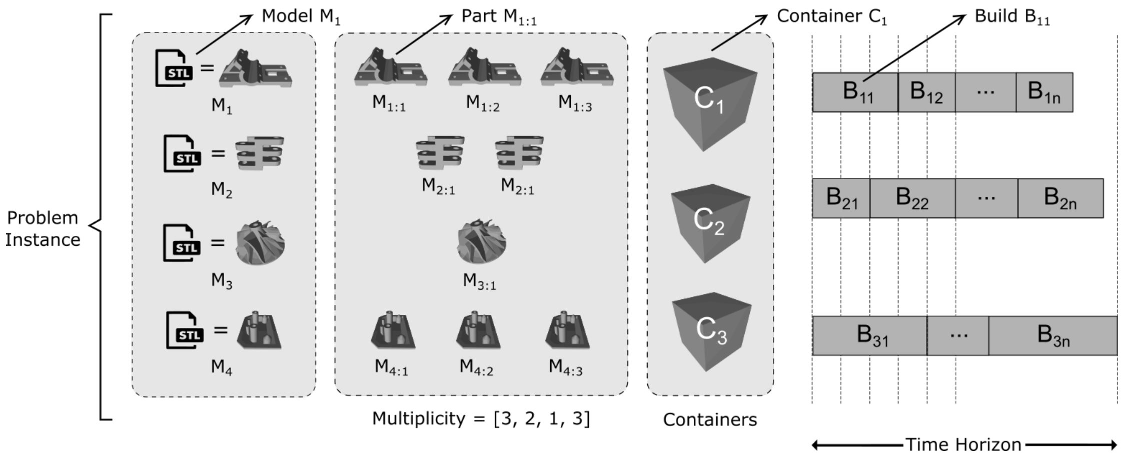


Figure 3.5: Terminology for 3DIP integrating terms used in the context of AM.

Table 3.3: Summary of the four-tuple $D|C|B|A$to describe 3DIP problems.

Element	Description	Values
D	Dimensionality of the problem	*1*, *2*, *2+1* or *3*
C	Criteria for optimisation	*Ou*, *Si*, *Cp* or *Tp*
B	Build volume types	*Of*, *Oo*, *I* or *H*
A	Attributes/features of the assortment of parts	*ll*, *hl*, *lh*, *hh*

The first element in the four-tuple (D) denotes the dimensionality of the problem, which is usually three-dimensional ('*3*') due to the nature of packing problems in AM. Some methods (Canellidis et al., 2006, 2009), however, reduce the input problem to a two-dimensional instance where the XY-projection of items must be arranged on the build platform or substrate. Dyckhoff (1990) used the notation *2+1* to refer to the dimensionality of this kind of problem; in AM operations this is applicable to technology variants that require sacrificial support structures, such as material jetting, resin vat processes and material extrusion.

The optimisation criterion (C) identifies one of the following objectives: output maximisation (*Ou*), single input minimisation (*Si*), time-minimisation parallel production (Tp) or cost-minimisation parallel production (Cp). *Output maximisation* (*Ou*) describes the objective of packing the subset of items that maximises the total volume of packed parts. This optimisation criterion applies to 3DIP variants involving a single build volume, an unspecified time horizon and a demand for parts that is higher than the container capacity. *Single input minimisation* (*Si*) describes problems that address a single hypothetical container with one or more variable-length dimensions, where the objective is to find a configuration for packing items that minimises the container's variable dimensions. In practical cases, this problem occurs when it is known that the entire set of parts can be accommodated within the container (Wäscher et al., 2007), which is not always the case. The optimisation criteria of *cost-minimisation parallel production* (Cp) and *time-minimisation parallel production* (Tp) apply to more complex scenarios where multiple builds are necessary to manufacture the parts required, which is defined by demand. For problems where the containers have the same estimated building cost, Cp is equivalent to the minimisation of the number of builds, i.e., the classic *Bin Packing Problem* (Wäscher et al., 2007). For the *time-minimisation parallel production* criterion, the due time and completion time of parts is more important (Tavakkoli-Moghaddam et al., 2005). This criterion involves the minimisation of one or more of the following variables: the *makespan* (time necessary to manufacture the entire demand), the total tardiness, and the earliness. Tardiness and earliness are computed by calculating the difference between the completion time and due time of required parts. The solution to this type of problem requires a combination of scheduling and 3DIP techniques. To date, few studies have addressed this class of problems (Lawler et al., 1993; Iori and Martello, 2010).

The third element of the $D|C|B|A$ notation (B) is a descriptor of the attributes of the container or containers in the problem. It is proposed that this specification should correspond to one of the following: one container with fixed dimensions (Of), one container with open Z-height or variable-length dimensions (Oo), multiple identical containers (I) or multiple heterogeneous containers (H). It is also assumed that the standard shape of the large object is rectangular, if it is two-dimensional, or cuboid, if it is three-dimensional, as is usual in build volume specifications.

The last element (A) is optional when specifying 3DIP problems, since it characterises the instance rather than the problem class, and adds information that might support the choice of a suitable solution method. It is proposed that Avaries according to the demand variation and average part complexity, assuming one of the following four permutations: low demand variation + low mean complexity (ll), high demand variation + low mean complexity (hl), low demand variation + high mean complexity (lh), high demand variation + high mean complexity (hh), or simply A to describe the presence of multiple instances with different characteristics or unknown feature values.

This study also proposes that the issue of subjective evaluations of 'low' and 'high' be addressed by determining the characteristics of the problem instance (A) using the relative standard deviation (RSD) of the multiplicities, which is a measure of the ratio of the standard deviation to the mean of demand quantities. Low demand variations are here defined as those that have RSD values equal to or less than 0.5; high variations have RSD values of greater than 0.5. Another factor to be considered is complexity, which is an inherently multi-faceted notion with many conceivable metrics based on interpretations (Gell-Mann, 2002). The Spies ratio (SR) is a traditional complexity measure used in manufacturing, which is obtained by dividing the volume of the geometric part by the volume of the primitive that contains it (Spies, 1957). Section 3.1.2 showed a practical computation-based method to calculate the SR by dividing the volume of the part by the volume of its minimum bounding box, as optimised using a local search algorithm. This complexity metric ranges between approximately 0 for complex parts and 1 for entirely convex primitive geometries.

The mean connectivity value (MCV) is a complexity metric that has been applied in empirical work in the field of AM (Araújo et al., 2016; Baumers et al., 2016b). Originally proposed to capture complexity in architecture (Psarra and Grajewski, 2001), this metric can support AM processes by analysing geometries according to the frequency and regularity with which concavities and entry features occur. For specifying part assignment, a lower complexity measurement exhibits an MCV less than or equal to 0.5, while a high complexity measurement has an MCV in the interval (0.5,1]. Table 3.4 presents the values for A based on demand variation and mean complexity and the conventional approaches that can be used to solve each category.

3.2.3 A new dataset for 3DIP problems in AM

The new dataset[1], which we will refer to as A2018, was created to provide an extensive number of packing problems that better reflect realistic scenarios in AM. In this dataset, each instance is provided in two formats: extensible markup language (XML) files and serialised Java objects.

[1]http://www.cs.nott.ac.uk/~psxlja/dataset

Table 3.4: Features of the assortment of parts (A) based on relative standard deviation (RSD) of demands and mean connectivity value (MCV).

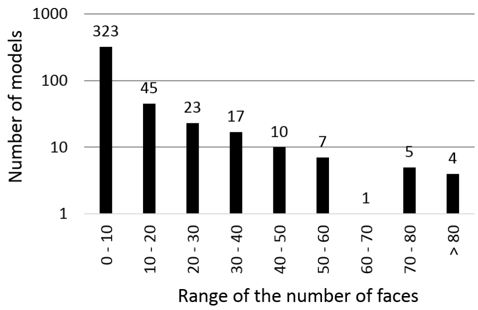
A	Demand variation	Mean complexity	Common solution
Low demand variation + low mean complexity (ll)	RSD $\in$ [0,0.5]	MCV $\in$ [0,0.5]	Iterative application of placement heuristics such as Deepest Bottom Left (Baumers et al., 2013).
Low demand variation + high mean complexity (lh)	RSD $\in$ [0,0.5]	MCV $\in$ (0.5,1]	Parts can be simplified into regular (often rectangular) geometries and then solved by exact methods (Stoyan et al., 2005).
High demand variation + low mean complexity (hl)	RSD > 0.5	MCV $\in$ [0,0.5]	This type of problem has not been systematically addressed in the literature.
High demand variation + high mean complexity (hh)	RSD > 0.5	MCV $\in$ (0.5,1]	Reduction of the parts to their minimum bounding boxes and application of wall-building heuristics (Canellidis et al., 2006).

Furthermore, each instance contains the description of the container(s), set of models and demands, parts and respective due dates. Together, the representation of the input problems and the achieved configuration of parts as XML files and in Java source code provides a valuable tool for use in the early stages of developing problem solutions, as it contains a much larger number of real-world geometries than are available in existing datasets. The parts in the proposed dataset are analysed with respect to the features presented in Section 3.2.1. Figure 3.6 shows histograms of the number of faces and the MCV for the 435 models in A2018.

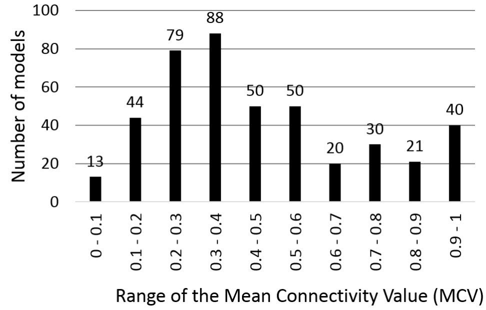
Figure 3.6a shows that dataset A2018 contains more models than any other dataset, which are also characterised by a significant number of facets (more than 10,000). All value ranges for the number of faces and the MCV are presented in Figure 3.6b. Existing datasets have a limited number of instances, each containing a small number of parts (typically < 20). The new dataset, by contrast, meets many of the challenges of solving 3DIP problems, since the models they contain possess a wide range of all characteristics as illustrated in Figure 3.6c and Figure 3.6d. Moreover, the 2,343 instances in A2018 cover the entire range of values for mean and relative standard deviation of both MCV and SR as presented in Figures 3.6e and 3.6f.

3.2.4 Illustration of the use of the dataset for assessing packing approaches

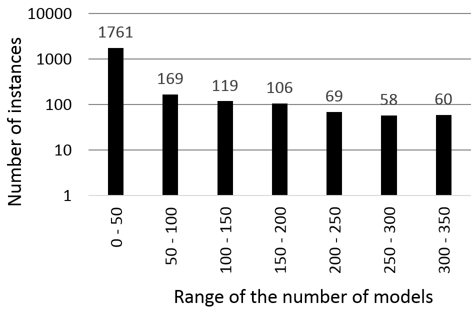
This section illustrates the usefulness of the new dataset (A2018) in the assessment of various 3DIP techniques by performing a computational experiment using a subset of 13 randomly selected instances, as detailed in Table 3.5. These are solved by employing two well-known packing heuristics. The first method is the DBLF (Karabulut and Murat, 2004). The algorithm employed in the experiment used the native orientation of parts; no additional rotations were applied to the geometries. The second technique is a wall-building method (WB) (Pisinger, 2002), which



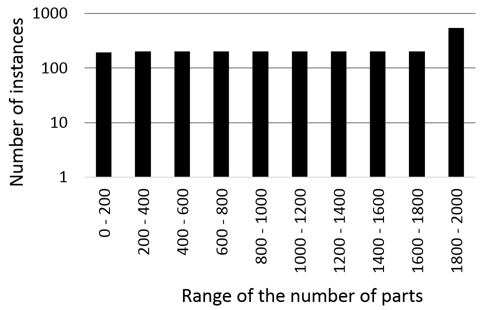
(a) Number of models in relation to number of faces.



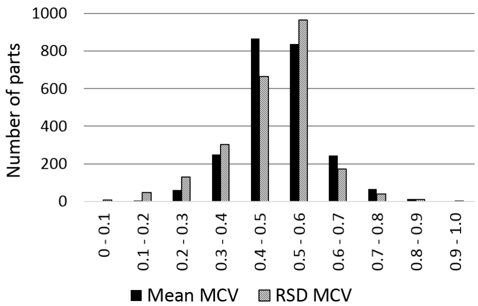
(b) Number of models in relation to MCV.



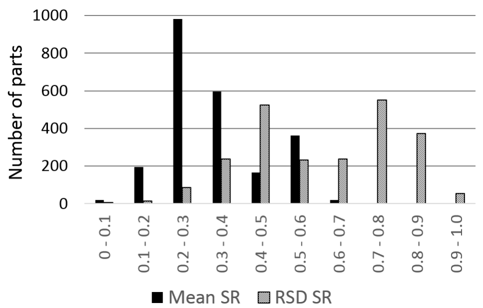
(c) Number of instances in relation to number of models.



(d) Number of instances in relation to number of parts.



(e) Number of parts per mean and RSD of MCV.



(f) Number of parts per mean and RSD of SR.

Figure 3.6: Histograms of features for models and instances in the new dataset.

Table 3.5: A sample of 13 instances from A2018 and their respective features.

Instance	*A*	# of parts	# of models	RSD of demands	Mean MCV	RSD of MCV	Mean SR	RSD of SR	RSD of volumes	Mean # of faces
A2018:0004	*ll*	9	8	0.294	0.421	0.344	0.248	0.867	1.040	2794.9
A2018:0007	*ll*	12	9	0.354	0.471	0.517	0.289	0.815	1.025	2892.3
A2018:0008	*ll*	13	5	0.308	0.487	0.528	0.312	0.774	0.902	8117.4
A2018:0009	*ll*	14	4	0.319	0.316	0.286	0.253	0.895	0.872	4724.3
A2018:0010	*lh*	15	14	0.240	0.527	0.339	0.372	0.537	0.521	1004.7
A2018:0015	*hh*	20	8	0.600	0.597	0.561	0.486	0.380	0.898	4231.6
A2018:0020	*lh*	25	8	0.492	0.590	0.344	0.337	0.656	0.792	9902.5
A2018:0021	*lh*	26	14	0.493	0.511	0.436	0.393	0.633	0.922	2681.2
A2018:0023	*lh*	28	8	0.319	0.519	0.531	0.239	0.972	1.291	6823.2
A2018:0026	*hl*	31	13	0.582	0.438	0.539	0.221	0.687	0.853	6791.3
A2018:0028	*hh*	33	8	0.599	0.76	0.298	0.356	0.820	1.759	7510.2
A2018:0046	*hh*	51	25	0.643	0.523	0.522	0.376	0.626	0.957	6267.2
A2018:0051	*hl*	56	38	0.533	0.421	0.583	0.299	0.650	0.997	4794.8

arranges the bounding boxes of geometries by rotating them orthogonally into horizontal and vertical layers until the container is filled (Araújo and Pinheiro, 2010).

For this experiment, each instance was solved using two approaches which address two of the most recurrent AM problems. The first problem relates to the AM technology variant stereolithography (Jacobs, 1992) and is denoted as $2{+}1|Ou|Of|A$ according to the classification presented in section 3.2.2, amounting to a two-dimensional cutting problem of the vertical projection of parts with the objective of maximising the used area over the horizontal building platform of one single machine. The second problem concerns the operation of the AM technology variant LS (Gibson et al., 2014) and is represented by the notation $3|Ou|Of|A$. This allows for the unconstrained positioning of parts in three-dimensional space and aims to maximise the volume utilisation within a single machine with fixed proportions. The first instance (A2018:0004) was solved by a DBLF algorithm according to the constraints and objectives of $2{+}1|Ou|Of|ll$ problems. Table 3.6 presents the build volume utilisation (BVU) and running time (in seconds) obtained by employing the DBLF and WB algorithms for the selected instances and problems. Figure 3.7 illustrates the graphical visualisation output obtained from the different packing algorithms for the same instances and problems and allows to observe, for example, how DBLF can arrange parts within 'open regions' of larger objects.

At this point, it is appropriate to investigate possible correlations between the features of instances, the key performance indicators of the algorithms and the characteristics of the packing problem addressed. Table 3.7 shows the correlations with the performance indicators BVU and running time of DBLF and WB.

The use of correlation coefficients provides useful insights regarding the relationship between the variables in the problem. For example, the calculations show a strong correlation

Table 3.6: Build volume utilisation (BVU) and running time for two packing approaches: the deepest bottom left with fill (DBLF) (Karabulut and Murat, 2004) and a wall-building based method (WB) (Araújo and Pinheiro, 2010)

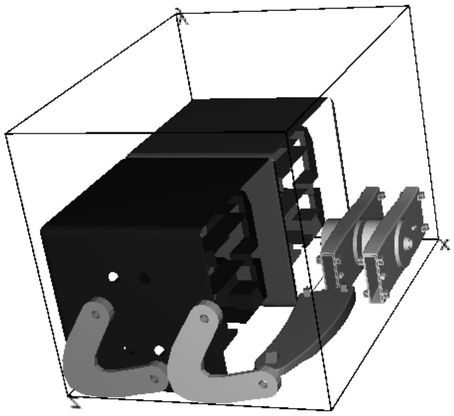
	$2{+}1\|Ou\|Of\|A$				$3\|Ou\|Of\|A$			
	DBLF		WB		DBLF		WB	
Instance	BVU	Time (s)	BVU	Time (s)	BVU	Time (s)	BVU	Time (s)
A2018:0004	0.159	154.4	0.166	0.5	0.181	707.8	0.166	3.9
A2018:0007	0.073	19.6	0.150	1.4	0.108	190.9	0.150	14.6
A2018:0008	0.062	1226.2	0.086	0.4	0.062	19878.6	0.079	1.9
A2018:0009	0.125	994	0.185	0.5	0.125	2317.4	0.148	0.2
A2018:0010	0.127	155	0.199	3.2	0.173	1786.2	0.194	2.5
A2018:0015	0.123	826.8	0.237	2.0	0.228	2041	0.237	87.6
A2018:0020	0.065	19743.4	0.116	1.0	0.065	217498.8	0.116	1.4
A2018:0021	0.037	1004.7	0.515	3.7	0.037	3595.3	0.515	105.8
A2018:0023	0.132	11670.4	0.499	1.0	0.140	36818.7	0.502	1200[a]
A2018:0026	0.199	2306.8	0.239	1.2	0.214	37915.7	0.270	1200[a]
A2018:0028	0.379	15990.6	0.396	2.5	0.396	99380.9	0.396	1200[a]
A2018:0046	0.426	18905.4	0.547	1.4	0.484	151196.7	0.619	1200[a]
A2018:0051	0.347	5203	0.328	1.7	0.370	13290.4	0.360	1200[a]

[a]Time limit for the WB.

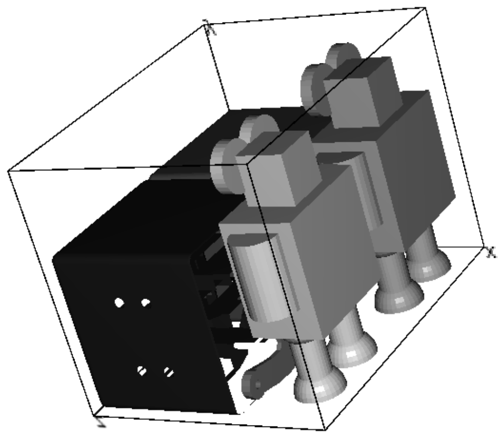
Table 3.7: Correlations between instance features, key performance indicators of packing algorithms and characteristics of the problems.

	DBLF				WB			
	$2{+}1\|Ou\|Of\|A$		$3\|Ou\|Of\|A$		$2{+}1\|Ou\|Of\|A$		$3\|Ou\|Of\|A$	
	BVU	Time	BVU	Time	BVU	Time	BVU	Time
# of parts	**0.78**	0.56	**0.74**	0.4	**0.64**	0.18	**0.70**	**0.81**
# of models	**0.62**	0.16	**0.61**	0.07	0.42	0.25	0.49	0.53
RSD of demand	0.59	0.48	**0.62**	0.47	0.45	0.24	0.51	0.55
Mean MCV	0.24	0.55	0.28	0.48	0.26	0.45	0.25	0.21
RSD of MCV	0.09	-0.12	0.16	-0.19	0.21	-0.1	0.27	0.37
Mean SR	0.04	0.1	0.18	0.13	0.19	**0.61**	0.19	-0.22
RSD of SR	-0.03	0.08	-0.18	-0.06	0.01	0.52	-0.04	0.15
RSD of volume	0.46	0.41	0.39	0.14	0.42	-0.02	0.38	0.53
Mean # of faces	0.17	**0.69**	0.09	**0.72**	-0.02	-0.47	0	0.36

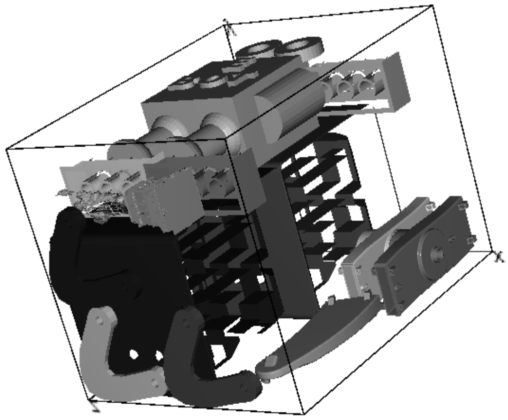
Moderate and strong correlations (greater than 0.5) are shown in bold.



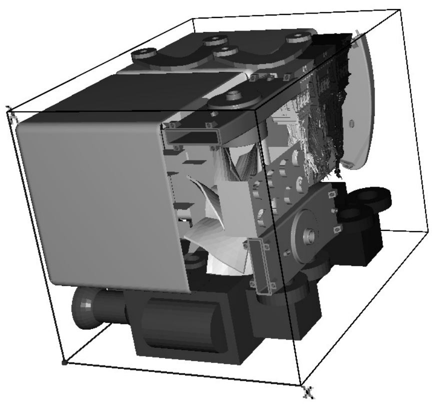
(a) Solution achieved by DBLF for A2018:0046 under $2{+}1|Ou|Of|hh$



(b) Solution achieved by WB for A2018:0046 under $2{+}1|Ou|Of|hh$



(c) Solution achieved by DBLF for A2018:0046 under $3|Ou|Of|hh$



(d) Solution achieved by WB for A2018:0046 under $3|Ou|Of|hh$

Figure 3.7: Examples of solutions found by the DBLF and WB algorithms in the experiment.

between the running time of DBLF and the mean number of faces in the problem, regardless of whether the problem is $2+1|Ou|Of|A$ or $3|Ou|Of|A$. This simple observation reinforces the argument that the collision detection between faces of irregular parts contributes to the poor performance of DBLF. Other insights obtained from the experiment are that there is a strong correlation between the BVU and the number of parts for different models and that the variation of demand is high regardless of the approach used for packing. The results and conclusions drawn here have been generated for illustration only. They are presented for the purpose of demonstrating the potential uses of the dataset for observing patterns between the features of the instances, the characteristics of the problem and the solution method.

3.2.5 Discussion

As previously identified, the quality of packing is important for managing costs. Hence, the selection of a packing approach affects the production performance frontier of AM (Schmenner and Swink, 1998). In order to choose which approach to use, it is important to identify which characteristics of the problem are important and which solution methods have been successfully applied to such problems in the past. By proposing the extended taxonomy for 3DIP problems in AM, including a consistent terminology, a comprehensive notation and refined categorisation criteria, this work supports the development of theory on the efficient operation of highly flexible digital manufacturing technologies. As a central research resource in both operations management and OR, formal theory of this kind consists of propositions that are not limited to particular contexts but are generally applicable (Glaser and Strauss, 1967). By updating the existing taxonomy for 3DIP, we provide AM researchers with the vocabulary to apply existing 3DIP theory to the emerging field of AM in order to determine features of relevance. Our updated taxonomy can also enable OR and algorithm experts to identify potential applications for their research.

The increased capacity to flexibly manufacture complex and diverse geometries makes it necessary to manage complexity in manufacturing control structures. This study aims to eliminate some of the subjectivity introduced through the adoption of linguistic criteria, such as those proposed by Dyckhoff (1990) and Wäscher et al. (2007), to categorise parts. Instead, this approach is based on the evaluation of measurable features of digital geometric models and the information systems that support manufacturing. This is timely since digital manufacturing processes, such as AM, are based on virtual representations of products and processes, thereby cutting across the engineering and operations functions (Slansky, 2008). Alongside the popularisation of the notion of 'Industrie 4.0' (Kagermann et al., 2011), the ubiquity and cost-effectiveness of information processing, and, with it, networked production facilities drawing on real-time information, means that reactive and flexible processes are becoming central in manufacturing

(Schwab, 2017). An example of such an application is the use of hyper-heuristics to automatically generate and evolve packing heuristics for three-dimensional regular packing problems (Allen et al., 2009). It is well-known that heuristic optimisation techniques often deliver varied performances, depending on the instances to which they are applied. This phenomenon is observed in our experiments. For example, for the $3|Ou|Of|A$ problem, DBLF performs better than WB in two instances, A2018:4 and A2018:51, whereas the situation is reversed in the all other instances.

There is a growing number of studies on the use of data science techniques to map instance features to the best performing algorithm to solve problems (Ross, 2005; Kotthoff, 2014). Using the features of unseen instances and the discovered mapping, the best performing heuristic can be selected automatically to solve the given problem. Another strand of ongoing research which may be of major benefit for solving this type of problem is on the use of hyper-heuristics to mix and control multiple heuristics to exploit their relative strengths (Burke et al., 2013). Machine learning, which uses algorithms designed to learn from data in order to make predictions, is another methodology that can be applied to 3DIP. Machine learning has been adopted in flexible manufacturing technologies for a variety of purposes, ranging from specifying predictive maintenance scheduling, the generation of demand forecasts, to manufacturing process monitoring and optimisation (Joseph et al., 2014).

Chapter 4

An experimental analysis of Deepest Bottom Left with Fill packing methods for Additive Manufacturing

The DBLF algorithm has been the basis for most of the approaches that address 3DIP problems in the AM sector (Karabulut and Murat, 2004; Canellidis et al., 2006; Gogate and Pande, 2008; Canellidis et al., 2009, 2013; Araújo et al., 2018). However, there has been no comprehensive investigation of the effects that different degrees of freedom for rotation have on the performance of the DBLF heuristic, which can be assessed by observing the resulting algorithm runtime and volume density. This chapter investigates the usefulness of the DBLF algorithm by: (i) discussing the main DBLF-based approaches and their strengths and weaknesses for use in AM; (ii) investigating trade-offs between computational runtime and packing efficiency (z-height); and (iii) demonstrating through the experiments the effects that variations in the constraints and input parameters for DBLF have on packing performance.

4.1 DBLF-based approaches: Brute force search, DBLF Decreasing and Genetic Algorithms

This section discusses the uses of three selected strategies for DBLF and their advantages and disadvantages. Presented in increasing order of number of occurrences in the literature, these strategies are (i) brute force search, (ii) deepest bottom left with fill decreasing (DBLFD), and (iii) genetic algorithm with DBLF. Each approach is discussed in detail in the following sections.

4.1.1 Brute force search

Brute force algorithms exhaustively search through all possible candidates to find globally optimum solutions. Therefore, their algorithmic computational cost can be gauged by assessing the size of the search space (Schaeffer et al., 1993). In the context of DBLF algorithms, the cardinality of the search space is the number of possible packing plans for the given input, which depends on the number of parts and allowable orientations per part. For instance, for the problem of packing a set of 'n' parts where part orientations arising from angle increments of 'θ' are allowed, the attainable 'b' angular states about the x, y and z-axes that give rise to a part orientation are captured by Equation 4.1. For example, 90° increments result in 4 possible angular states $(0°, 90°, 180°, 270°)$.

$$b = \begin{cases} 1, & \text{if } \theta = 0 \\ \lfloor \frac{360}{\theta} \rfloor, & \text{otherwise} \end{cases} \tag{4.1}$$

The cardinality of the solution space can, therefore, be estimated by equation 4.2, which reflects the runtime of the brute force algorithm based on n and b.

$$f(n,b) = n! \times b^{3n} \tag{4.2}$$

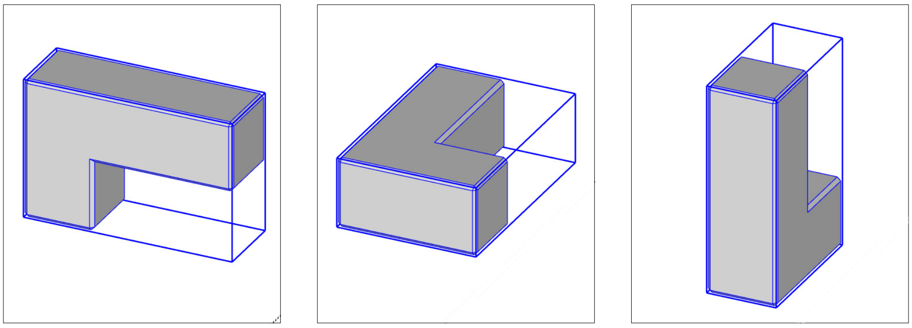
The brute-force complete search method yields the best packing sequence with the minimum volume height after investigating each candidate in the search space. For inputs of size n and for a constant b, it requires $O(n!\ b^n)$ time, which is not practical even for very small values of b. Hence, solving 3DIP problems using brute-force search is not and amenable strategy for problems containing a large number of demanded parts as in most of the real-world instances.

4.1.2 DBLF Decreasing

Deepest bottom left with fill decreasing is a strategy to manage the large search space of packing plans by sorting the parts in decreasing order of a particular numeric feature. It extends the standard DBLF by adding a preliminary task in which parts are sorted in decreasing order of volume; the sequence of parts is then processed using DBLF. Another particularity of the implemented approach is that three different orientations per part, which are shown in Figure 4.1, are tested while the standard DBLF uses only the native (i.e. original) orientation. This mechanism mitigates the effect of poor native orientations achieved during the part design and explores a more extensive area of the search space (see equation 4.2).

4.1.3 A Genetic Algorithm combined with DBLF

Ikonen et al. (1998) used the acronym GARP to refer to methods that integrate genetic algorithms (GAs) with the DBLF heuristic for rapid prototyping, which has been one of the dominant strategies for applying 3DIP in AM (Ikonen et al., 1996, 1997, 1998; Hur et al., 2001; Canellidis et al., 2006, 2009, 2013; Araújo et al., 2015). GAs are evolutionary algorithms (population-based metaheuristics) that mimic the process of evolving a population throughout the selection



(a) Original model and first possible orientation. **(b)** Second possible orientation for the given part. **(c)** Third possible orientation for the part.

Figure 4.1: Three orientations per part, as in the implemented DBLFD.

and combination of the fittest individuals (John, 1992).

The characteristics of an individual are represented by a data structure called a chromosome, which consists of an array of values or genes. In GARP, a chromosome encodes a sequence and sometimes (depending on the solution design) the orientation in which parts are processed by the DBLF heuristic, to create a data structure also known as 'packing plan' (Gogate and Pande, 2008). The assessment of the quality of an individual (fitness) is measured by the resulting volume density, which is reflected by factors such as the z-height of the build volume constructed by DBLF given the packing plan. The fitness has its maximum when the parts are perfectly assembled with no gaps between them, resulting in a build volume with minimum z-height.

Parameter tuning is highly recommended when applying metaheuristics to any problem domain, although this has not been done for 3D packing studies in AM (Grefenstette, 1986). Approaches using GA within this context have explored only a restricted range of parameter values, which have been selected in an arbitrary manner or using limited rationalisation (see Table 2.1). The GA components that have been used in such studies, which are also employed in the experiments shown in section 4.2, are explained below.

Chromosome: A chromosome is an array of genes with a length (L) that is equal to the number of parts (n). Each gene is a 4-tuple comprised of the id of the part and the rotations around the x, y and z-axis (r_x, r_y and r_z respectively), as illustrated in Figure 4.2.

Crossover: Crossover is responsible for transmitting genetic characteristics from good individuals (parents) to the next generation (John, 1992). In GARP, order-1 crossover is often used since this operator is suitable for combining individuals that are represented as non-binary chromosomes and is commonly used for permutations (Poon and Carter, 1995). An aspect to consider is the probability that crossover is applied and the strategy for selecting parents. Based

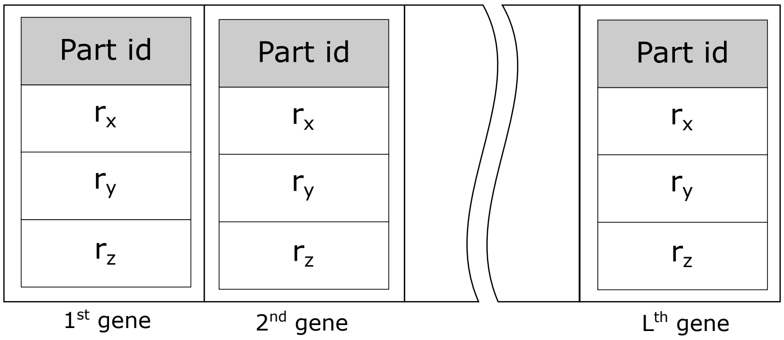


Figure 4.2: Chromosome in the current GA implementation.

on the parameters used in reported GARP solutions and suggestions in the literature, the following values are used for crossover probability (CP): 0.5, 0.75 and 1.0 (Srinivas and Patnaik, 1994).

Selection: Individuals with higher fitness values have higher probability of being select to generate new individuals. Three parent selection schemes appear in GARP solutions and are used in section 4.2 (Goldberg and Deb, 1991):

- Roulette wheel or fitness proportionate selection: the probability of selecting an individual is proportionate to the fitness
- Ranking selection: sorts a population with size p in decreasing order of fitness and then applies proportional selection (similar to roulette wheel described above) based on the 'new' fitnesses of the individuals with values $1/2$, $1/3$, ..., $1/(p+1)$; particularly useful for distinguishing individuals with similar fitness values
- Tournament: selects the best individual from a randomly selected group of individuals, typically comprised of 2 or 4 elements

Mutation: Mutation adds genetic diversity to the population by changing the value of each gene with mutation probability (MP) (John, 1992). Three different schemes have been used in GARP:

- Insert (Sivanandam and Deepa, 2007): the gene is moved to a random position in the chromosome
- Creep (Sivanandam and Deepa, 2007): different values are assigned to the r_x, r_y and r_z of the gene

- Insert and creep: the gene values for r_x, r_y and r_z are changed, and then the gene is randomly moved to a different position in the chromosome. This scheme is used in the experiments shown in section 4.2.

Concerning MP, two commonly used values are tested: $1/L$ and 0.01.

Fitness function: The fitness function maps how well the individual meets the problem objective. In this work, this function focuses on solutions to the three-dimensional strip packing (3DSP) problem, which aims for minimum build volume height. The fitness function is calculated by the equation 4.3.

$$f = \frac{H^*}{H + PEN} \tag{4.3}$$

where, H^* corresponds to the optimal minimum height, H is the resulting build volume height, and PEN is the penalty value added to prioritise solutions in which all the parts are packed. In real-world instances, H^* is often unknown and can be replaced by a strictly positive constant, and PEN is needed in cases where some part orientations result in x-y cross-sections that exceed the bounds of the container. In this implementation, PEN is set to 10 times the optimal value if at least one part is not packed, and 0 otherwise. This results in a fitness value (f) of between 0 and 1. Good packing configurations attain values closer to 1, while poor configurations approach 0. For example, where all the parts can be perfectly assembled with no loss of space, then $H = H^*$ and $PEN = 0$, resulting in fitness value of 1.0.

Replacement strategy: This is the strategy to replace individuals of the current population by the offspring. Two replacement strategies are used the experiments presented in section 4.2:

- Generational replacement with elitism: (i) copies the best two individuals of the current population to the pool of individuals which will constitute the population of the next generation, (ii) successively combines individuals, and (iii) copying the offspring to next generation until its population is complete

- Steady state: instead of generating a new population, the best two individuals among the two selected parents and the offspring are copied back to the population

Termination criteria: These criteria limit the computational costs of GA. The GA method implemented in this study stops and returns the best individual (solution) found at a point when at least one of the following conditions is satisfied:

- the best individual is above the acceptable threshold, i.e., its fitness is within the interval $[0.99, 1]$ (Safe et al., 2004)

- the algorithm ends if there is no improvement of the best individual after 1,400 (calculated after preliminary tests) consecutive evaluations

Determining appropriate termination criteria depends on the problem domain and intended search length (Safe et al., 2004). Preliminary tests conducted by the authors showed that solutions with a fitness within the interval $[0.9, 1]$ have satisfactory results. Regarding the second criterion, no further improvement to the best individual was observed after approximately 1,400 consecutive evaluations (about 35 minutes runtime) for population up to 200 in size. This is a reasonable amount considering that the runtime for one single evaluation is at most 1.5 seconds.

Parameter sets: A parameter set is comprised of the definition of the population size, CP, the parent selection scheme, MP and the replacement strategy. Table 4.1 summarises the parameters and values that have been tested in the existing GARP solutions and that are employed in the experiments shown in section 4.2.

Table 4.1: Summary of the parameter values used in the existing GARP methods.

Parameter	Values
Size of population (p)	100, 200
Crossover probability (CP)	0.5, 0.75 and 1.0
Mutation probability (MP)	$1/L$ [a], 0.01
Parent selection scheme	Roulette wheel, ranking, tournament-2 [b] and tournament-4 [c]
Replacement strategy	Generational with elitism, steady state

[a] L: Length of chromosome; [b] Tournament of 2 individuals; [c] Tournament of 4 individuals

4.2 Experimental design

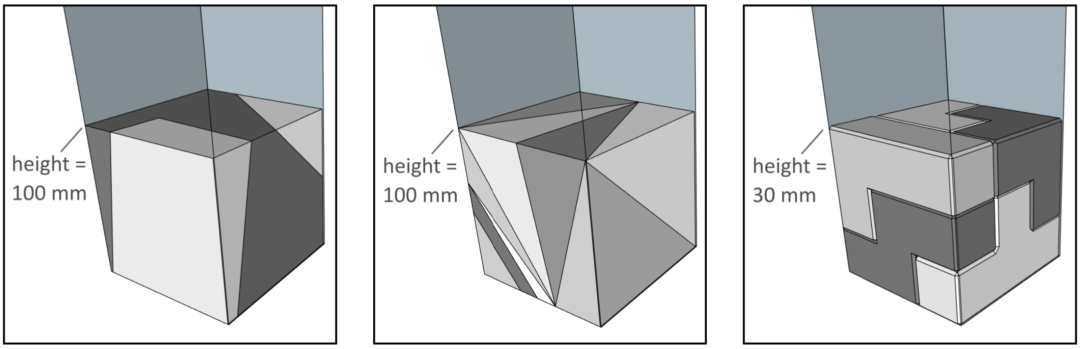
This section proceeds from the description of the three DBLF approaches in Section 4.1 and focuses on a series of experiments to compare their algorithmic performances. Also presented here are the benchmark problem instances and their properties, evaluation methods and targeted observations.

4.2.1 Problem instances used in the experiments

Three problem instances were generated to investigate the packing performance of DBLF-based approaches since the 3DIP dataset, shown in section 3.2, was not available at the time. The first two instances, Cutcube1 with four parts and Cutcube2 with 11 parts, are generated by slicing a three-dimensional cube using bounded regions of non-orthogonal planes. The third instance is a Somacube (Peter-Orth, 1985) comprised of seven polycube parts. Figure 4.3 illustrates the problem instances arranged in their best packing solution, which is known.

The instances[1] shown in Figure 4.3 are comprised of relatively few parts: 4, 11 and 7 for the Cutcube1, Cutcube2 and Somacube, respectively. The parts considered here pose a greater challenge from a packing perspective than the bounding box approximations. While they do not

[1] Available at `http://www.cs.nott.ac.uk/~psxlja/dblf`



(a) Cutcube1 (4 parts) **(b)** Cutcube2 (11 parts) **(c)** Somacube (7 parts)

Figure 4.3: The number of parts and optimal packing configuration for the three instances.

represent AM parts designed for industrial applications, they approximate the levels of complexity found in reality. Using instances with intermediate levels of complexity and known optimal packing arrangements is useful for comparing the effectiveness of DBLF-based approaches for packing AM parts in reasonable runtime. Underlying this approach, of course, is the acknowledgement that obtaining the optimal packing for a random set of AM parts, as pursued in some studies, is an unfathomable task. Therefore, having known optimal packing configurations is a distinct advantage for benchmarking (Szykman and Cagan, 1995). The experiments for the present study also use the original STL models for the parts to prevent the loss of information that would result from using alternative primitive representations.

4.2.2 Experiments

In this section, we describe our experiments comparing the three DBLF approaches shown in section 4.1 and the effects that different algorithm parameters have on performance regarding volume utilisation and runtime.

Estimation of the runtime for brute force search. For each instance-θ, randomly selected packing plans were processed by the DBLF heuristic to calculate their average runtime. A sample size of 35 was used, except in the case of instance Cutcube1 with no part rotation which has search space comprises only 24 elements, to enable statistical analysis on the results (Royall, 1986).

Parameter tuning for the GA approach. Studies presenting GARP techniques often fail to analyse how different values for GA parameters affect packing performance. Moreover, limited rationalisation is given regarding the choice of these parameters, which is unfortunate since parameter tuning is an essential task while applying heuristics to any problem domain (Smit and Eiben, 2009). In this experiment, 96 parameter sets were generated from combinations of the

GA parameters shown in Table 4.1. For each parameter set, GA was executed 35 times to ascertain the average runtime and obtained build volume height when solving the Cutcube2 instance with angle increment of 90°. As a result, the parameter set that maximises the average fitness withing reasonable runtime is selected for further testing.

Comparing the GA approach to DBLFD. This experiment analyses the relationship between different degrees of freedom for part rotation and packing efficiency. Using the parameter set selected previously, the GA is executed 35 times to solve each instance and angle increment, which we will refer to in the following as an instance-θ pair. The average runtime and build volume height are calculated and compared to the deterministic result obtained from the DBLFD method.

4.3 Results and discussion

4.3.1 Testing for brute force search

As explained above, 50 packing plans were randomly generated for each instance-θ, except in the case of Cutcube1-0 as its search space is comprised of 24 elements. Table 4.2 shows the mean runtime per instance and part, as well as the number in non-convex parts in each instance.

Table 4.2: Estimated runtime for the brute force per instance-angle increment.

Instance	Number of parts	Number of non-convex parts	Mean time (seconds)	
			Per instance	Per part
Cutcube1	4	1	0.031	0.008
Cutcube2	11	0	0.058	0.005
Somacube	7	7	1.445	0.206

The results confirm the intuition that the runtime per instance depends on the quantity and complexity of the parts processed. The runtime to solve Cutcube1 is lower than Cutcube2 due to a lower number of parts. On the other hand, the mean runtime per part of Cutcube1 is higher, mainly because of the presence of one non-convex part. The Somacube instance, which has the most non-convex parts, had the highest mean runtime for both instance and parts. The mean runtime per instance and equation 4.2 enables an estimation of the time necessary to investigate each element in the search space.

Brute force search often requires an unreasonable amount of time for most of instances-θ, especially those with a large number of parts or higher degrees of freedom regarding part rotation. For example, processing every candidate solution in the search space of packing plans for Cutcube2-90 would require approximately $1.62 * 10^{17}$ years, according to Equation 4.2. This reinforces the argument that approximate methods capable of giving results quickly and with reduced computational cost are preferable to the brute force algorithm when addressing real-world packing instances. More interesting, however, is the observation that instances comprised of higher mean shape complexity (in terms of the number of faces) can require higher runtime than instances with more parts. For example, the average runtime for processing the Somacube instance (see Table 4.2) is higher than the average runtime for solving Cutcube2, regardless the angle increment[2]. These findings demonstrate the importance of adopting simpler boundary representations for implementing DBLF approaches (Stroud, 2006).

4.3.2 Parameter tuning for GA-based approach

The mean runtime and build volume height are used to compare the 96 different sets of parameter configurations generated from combinations of GA parameter values as shown in Table 4.1. First, each attribute was analysed separately with respect to its effects on the performance (see Table 4.3).

Table 4.3: Comparing the mean and standard deviation of GA parameters separately.

Parameter	Value	Runtime		Fitness	
		Average (s)	SD	Average	SD
Crossover probability	0.5	55.9	19.3	**0.2309**	0.1296
	0.75	84.2	30.2	**0.2673**	0.1227
	1	111.84	39.2	**0.3010**	0.1059
Mutation probability	0.011	84.07	37.9	0.2628	0.1245
	1/L	83.8	38.5	0.2699	0.1218
Population size	100	81.1	37.2	0.2627	0.1244
	200	86.8	39.0	0.2700	0.1219
Selection scheme	Ranking	86.1	40.1	0.2637	0.1238
	Roulette wheel	81.9	36.7	0.2651	0.1243
	Tournament 2	84.0	37.7	0.2715	0.1213
	Tournament4	83.7	38.2	0.2652	0.1234
Replacement strategy	Generational	87.5	40.5	0.2726	0.1211
	Steady state	80.3	35.4	0.2602	0.1249

It can be observed from Table 4.3 that GA parameters do not strongly affect the mean runtime and mean build volume height, except for CP. Increasing the value of CP from 0.5 to 1.0 results in a noticeable improvement ($> 10\%$) in build volume z-height at a modest increase in computational cost. It is of note that generational GA with elitism produced marginally better results compared to steady state implementation. Figure 4.4 shows that build volume height decreased as the time spent on the search increased, which was expected. The first cluster of parameter sets with mean times of less than 65 seconds is entirely comprised of configurations

[2]Data available at `https://github.com/ljonata/ExperimentalAnalysisDBLF`

Table 4.4: Comparing three selected parameter sets (33, 35 and 47).

Parameter set	Crossover probability	Mutation probability	Population	Selection scheme	Replacement strategy	Mean fitness
33	1	$1/L$ (0.090)	200	Tournament-2	Generational with elitism	0.2426
35	1	0.011	200	Tournament-2	Generational with elitism	0.2860
47	1	0.011	200	Tournament-4	Generational with elitism	0.2880

with CP equal to 0.5, while the group of parameters between 65 and 105 seconds is predominantly comprised of configurations with CP equal to 0.75 (33 out of 42 parameter sets). The graph stresses the two parameter sets that resulted in highest mean fitness: 35 and 47. Table 4.4 presents the attributes and the mean fitness obtained by the parameter sets 35, 47 and 33, which is formed by the best individual parameters extracted from Table 4.3.

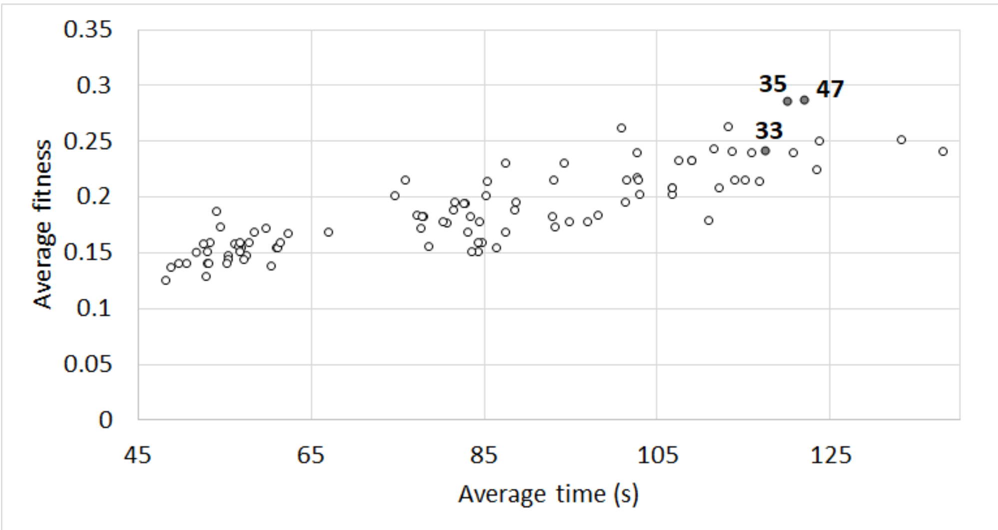


Figure 4.4: Scatter graph summarising the mean fitness and runtime for each parameter set.

The three parameter configurations (sets) shown in Table 4.4 share the same values for the probability of crossover, size of population and replacement strategy. The parameter configuration identified by the ID 35 was chosen for further experiments, since the tournament-2 selection method yielded better mean results than the tournament-4 method (see Table 4.3).

4.3.3 Comparing DBLFD and GA-based packing methods

The mean fitness and mean runtime are reported in Figure 4.5, together with the result from DBLFD. The data obtained from this experiment reinforces the need to adopt simpler primitive representations when aiming to minimise computational effort.

Although the Somacube instance is comprised of fewer parts than Cutcube2 (7 and 11, respectively), the higher complexity, i.e., non-convexity of parts, resulted in considerable higher

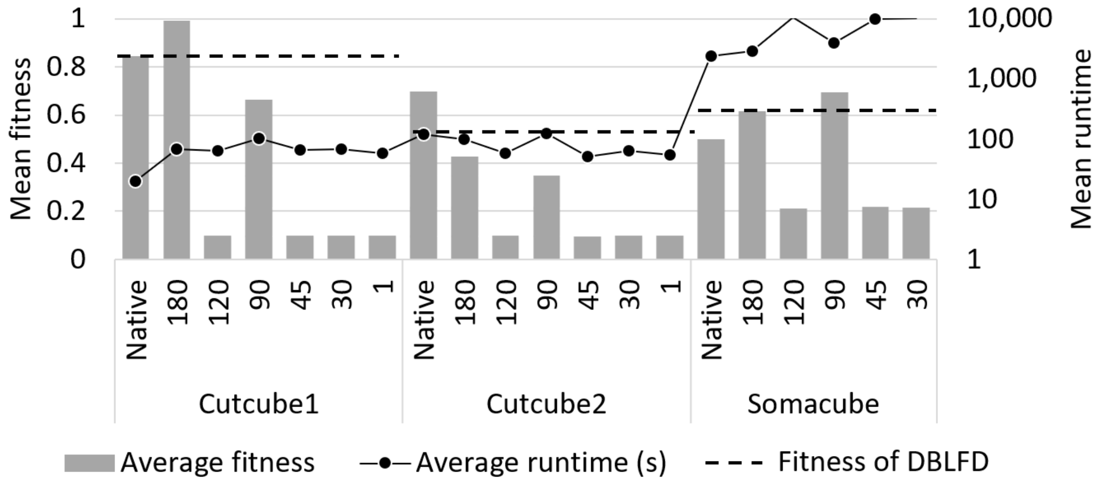


Figure 4.5: Mean fitness and runtime obtained for each instance-θ

runtime. While the mean runtimes for all Cutcube1-θ and Cutcube2-θ are less than 140 seconds, all the Somacube-θ configurations have mean runtime over 2,000 seconds. Interestingly, for most of the instance-$theta$ pairs (except for Cutcube2-0, Somacube-180 and Somacube-90), the DBLFD heuristic produced more competitive results than the GA approach. This finding suggests that the former method is a viable alternative, as it simpler to implement and has a more reasonable runtime than the popular GA approach with angle increments less than 90°.

The effects of part orientation on mean fitness and the importance of 'good' initial orientation of the parts can also be observed in Figure 4.5. For configurations with non-orthogonal rotation ($\theta = 120°$ or $\theta < 90°$), initial populations were comprised of several packing plans that resulted in non-packable parts. This was due to horizontal projection of the parts exceeding the bounds of the container. This observation indicates that local search methods should be run before proceeding with the packing process to ensure that each part fits within the container (Canellidis et al., 2006, 2009, 2013).

4.3.4 Discussion on packing for real-world problems

This study provides evidence that shape complexity affects computational runtime to a greater extent than the number of parts. As shown in the first experiment, instances with a smaller number of parts exhibiting higher degrees of non-convexity require greater computational effort than those with higher quantity requirements but simple geometries. For example, the mean runtime for solving a Somacube instance is greater than in Cutcube2, despite its fewer number of parts (see Table 4.2). This implies that high-resolution voxelised representations (Min, 2004), which bear a resemblance the non-convex polycubes contained in the Somacube instance, are likely to result in higher runtimes compared to convex hull envelopes. Therefore, the results underline the importance of the adoption of simple boundary representations in the early stages of packing solution development. This is of special significance in AM, since the adoption of AM is often justified by the necessity to manage highly complex or non-convex product geometries (Baumers et al., 2017b).

Another observation that can be made from these experiments concerns the limitations of DBLF in obtaining configurations with no waste of space between objects (when such configurations exist) due to the fixed order in which geometric operations (translation and rotation) are performed. For example, the use of brute force search for solving Cutcube1-0 fails in achieving a 'no-waste' configuration despite the small number of parts, which have been designed to be perfectly assembled. As shown in Table 4.5, the best build volume height achieved by DBLF (depicted in Figure 4.6) was 118.32mm. This demonstrates a need for placement heuristics that are more flexible regarding the identification, translation and rotation of the processing part. Advantageous strategies would be to temporarily allow infeasible states and to adopt more efficient methods of detecting eventual geometry overlaps.

Table 4.5: Results of DBLF after processing the 24 possible packing plans for Cutcube1-0.

Best height (mm)	Worst height (mm)	Mean height (mm)	The standard deviation of height
118.32	190.31	144.03	23.86

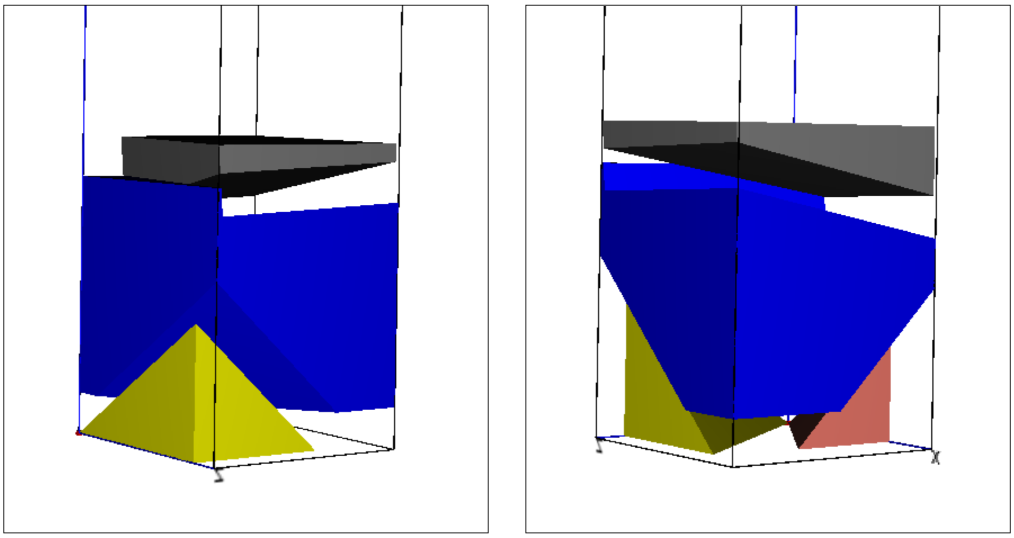


Figure 4.6: The best out of 24 possible packing plans of Cutcube1-0 with no rotation allowed.

The above experiments and the comparison of the performance from GA an DBLFD can be used to stress the necessity to optimise the orientation of each part individually before running the packing algorithm. Incorporating this task into the process is likely to prevent both parts remaining unpacked. It would also prevent part rotations that lead to technically problematic solutions due to large horizontal sections, which are avoided in practice (to avoid part deformation and curling). Therefore, it would be practical to integrate such an orientation determination step within the design software that generates the final to-be-manufactured model instead delegated the task to a human machine operator. The results discussed in this section have a number of additional implications for AM practice:

- Simplifying part representations can lead to quicker determination of satisfactory build configurations. This is essential in some practical situations, such as when cost estimates or price quotations are required instantaneously.
- Similarly to other combinatorial optimisation problems, parameter tuning can heavily influence the entire functioning and performance regarding runtime and volume utilisation of packing approaches. However, most of GARP techniques in the literature omit this critical preprocessing stage and use arbitrary values for parameters and rotation. The incorporation of such task into the AM workflow can, therefore, improve the packing outcome and thereby reduce manufacturing costs in AM (Ruffo and Hague, 2007; Baumers et al., 2017a).
- As shown in the experiment comparing GA to DBLFD, higher degrees of freedom for part rotation does not necessarily result in a gain of performance concerning volume density for packing algorithms. Instead, it occurs in additional computational effort due to the exponential increase of the search space. The use of orthogonal rotation, therefore, seems a reasonable approach for this domain.

Products that have been designed for AM are particularly likely to feature high levels of complexity (Hague et al., 2003), which, together with orientation in build, affects manufacturing parameters, such as surface roughness, the area of contact with the build platform and supporting structures. Therefore, pre-processing of the parts in the digital design environment would allow the simulation and prevention of losses associated with orientation-related issues during the manufacturing step (Peko et al., 2018).

Chapter 5

An algorithm selection approach to three-dimensional irregular packing

Techniques that identify matching patterns between the feature space and algorithm performance are at the core of the algorithm selection problem introduced by Rice (1976). Over the last decade, algorithm selection methods have become increasingly popular for solving a range of complex optimisation problems, including propositional satisfiability, scheduling and the travelling salesperson problem (Monostori, 2003). However, to the best of the author's knowledge, the viability of an algorithm selection approach for solving 3DIP problems has not been reported in the literature. This chapter focuses on the use of machine-learning-based algorithm selection for 3DIP problems that have the objective of promoting efficiency in single-build manufacturing (Kotsiantis et al., 2006). An empirical analysis of the results using several learning models, parameter sets and data transformations is also presented. The algorithm selection solution tailored for 3DIP provides valuable insights to support the development of more effective algorithms suitable for practical problems aligned with AM requirements (Xu et al., 2008; Mittelmann, 2015).

5.1 An algorithm selection pipeline to 3DIP

This section presents an algorithm selection pipeline applied to realistic 3DIP problems for the first time in literature. This pipeline uses the 3DIP dataset introduced in chapter 3 and two of the most commonly used approaches for solving 3DIP problems: a wall-building-based algorithm and the deepest bottom-left decreasing heuristic seen in chapter 4.

Firstly, the 3DIP instances are solved using the aforementioned methods and labelled with the algorithm that delivers higher volume utilisation. This constitutes an extended dataset to be publicly available to the community. The data is then preprocessed to transform raw data into a clean data that enables machine learning models to obtain better results. It also prevents that numeric features with high magnitude values dominate others in the objective function.

After, different predictive models and parameter sets are compared and used to evaluate the most relevant features in this domain. The results allow valuable insights regarding the packing approaches and confirm the usefulness of the introduced features and methods in this study. The methodology is summarised in Figure 5.1.

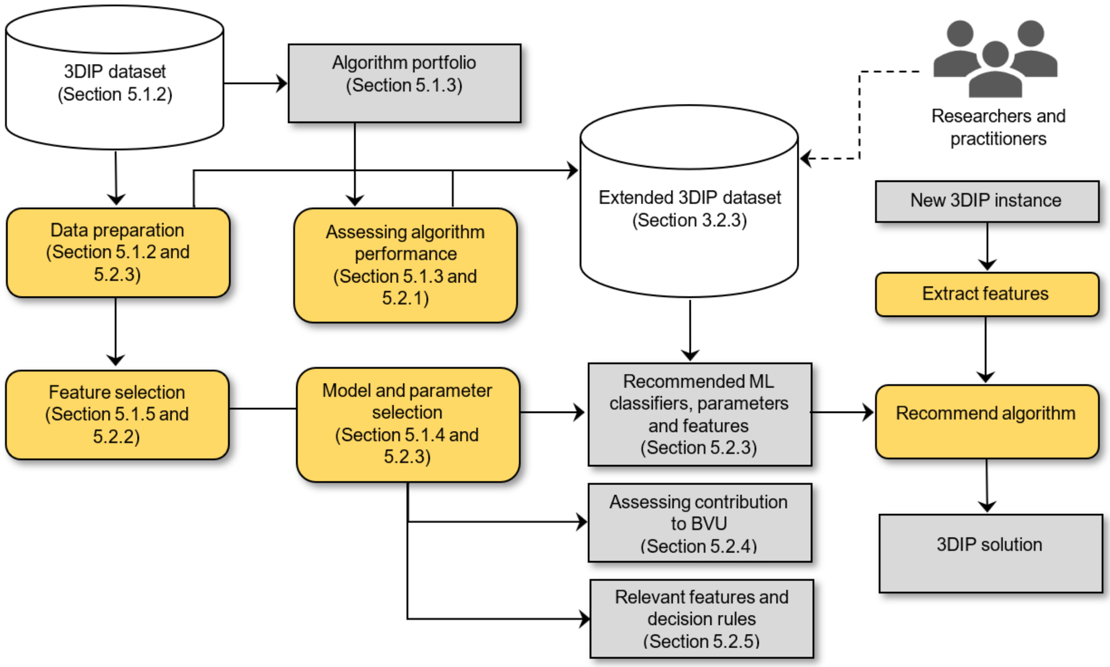


Figure 5.1: The methodology for implementing an algorithm selection for 3DIP problems.

5.1.1 The three-dimensional irregular knapsack problem

Three-dimensional irregular knapsack (3DIK) problems are packing problems in which a subset of items must be placed in containers in such a way that the utilised volume within the container is maximised (Wäscher et al., 2007). These problems are analogous to the task of choosing the subset of parts that maximises the utilised volume within the manufacturing build volume given an exceeding number of parts (Dickinson and Knopf, 2002; Baumers et al., 2013). Three-dimensional irregular knapsack is encoded by the tuple $3/B/O/$ in Dyckhoff's typology (Dyckhoff, 1990), and $3|Ou|Of|A$ in Araújo et al.'s extended taxonomy for C&P problems (Araújo et al., 2018).

Algorithms for tackling 3DIK problems are particularly valuable for AM technologies that permit a relatively unconstrained layout configuration for the parts within the build chamber, such as LS (Mueller, 2012; Gibson et al., 2014). One characteristic of 3DIK is that it results in a bounded output variable, which is the percentage of the effectively utilised volume capacity. This differs from the unbounded results observed in other kinds of packing problems, such as the three-dimensional strip packing problem, which aims to minimise build-volume height (Wäscher et al., 2007; Araújo et al., 2018).

5.1.2 Instances and features

This study uses the 3DIP dataset introduced in chapter 3, which consists of instances generated to resemble realistic AM requirements regarding parts and demands. Each 3DIP instance is represented in the dataset by an entry or row. This dataset is comprised of a comprehensive set of 2,343 entries and the 32 features shown below with their corresponding labels:

- Number of parts (num-parts)
- Number of models (num-models)
- Mean of demands (number of parts divided by the number of models) (mean-demand)
- Relative standard deviation (RSD) of the demands (rsd-demand)
- Mean of the file size (bytes) of models (mean-file-bytes)
- RSD of the file size (bytes) of models (rsd-file-bytes)
- Mean of the values for Mean Connectivity Value (MCV) of the parts (mean-MCV)
- RSD of the MCV of the parts (rsd-MCV)
- Mean v-value of the parts (mean-v-value)
- RSD of the v-value of the parts (rsd-v-value)
- Mean Spies ratio (SR) of the parts (mean-SR)
- RSD of the SR of the parts (rsd-SR)
- Mean of MCV multiplied by SR of each part (mean-MCVSR)
- RSD of MCV multiplied by SR of each part (rsd-MCVSR)
- Mean Minimum Bounding Box (MBB) volume of the parts (mean-vol-MBB)
- RSD of the MBB volume of the parts (rsd-vol-MBB)
- Mean number of faces of the parts (mean-num-facets)
- RSD of the number of faces of the parts (rsd-num-facets)
- Mean surface area of the parts (mean-surface-area)

- RSD of the surface area of the parts (rsd-surface-area)
- Mean volume of the parts (mean-volume)
- RSD of the volume of the parts (rsd-volume)
- Volume of the container (vol-container)
- Surface area of the container (surface-container)
- Mean of relative volumes of the parts, which is the volume of the part divided by the volume of the container (mean-rel-volume)
- RSD of the relative volumes of the parts (rsd-rel-volume)
- Mean of relative surface (surface of the part divided by the surface of the container) of the parts (mean-rel-surface-area)
- RSD of the relative surface of the parts (rsd-rel-surface-area)
- Mean of relative MBB volumes (MBB volume of the part divided by the volume of the container) of the parts (mean-rel-vol-MBB)
- RSD of the relative MBB volumes of the parts (rsd-rel-vol-MBB)
- Mean of connectivity scores (MCV divided by max{v-value, 0.1}) of the parts (mean-connectivity-score)
- RSD of the connectivity scores of the parts (rsd-connectivity-score)

An additional set of 27 features was generated to indicate the percentages of parts that have a specific feature value within a discretised range of intervals, as presented in Procedure 2. Despite the fact the computational runtime required to calculate each feature is relevant, such information is not within the scope of the current study. In total, each entry contains information on 59 features.

After data collection, the data is processed before being passed on to the learning algorithm. Data pre-processing is essential for cleaning and removing inconsistencies in the data. Although it is a time-consuming task, it can contribute to better quality predictions and a rapid learning rate (Kotsiantis et al., 2006). Furthermore, pre-processing influences the effectiveness of the models by preventing large magnitude features dominating the objective function. For example, while the number of facets of the parts in the 3DIP dataset ranges from 4 to 447,578 units, MCV and SR range between 0 to 1. Three data pre-processing strategies are tested in this study. These are listed below:

Algorithm 2: Pseudocode compute and feature values.

List of intervals = [[0, 0.2] , (0.2, 0.4] , (0.4, 0.6] , (0.6, 0.8] , (0.8, 1.0]]
for *each* interval *in* List of intervals **do**
    Add feature Percentage of parts with Mean Connectivity Value *in* interval
    Add feature Percentage of parts with V-value *in* interval
    Add feature Percentage of parts with Spies Ratio *in* interval
    Add feature Percentage of parts with relative volume *in* interval
    Add feature Percentage of parts with relative Minimum Bounding Box *volume in* interval
end
Add feature Percentage of parts with relative volume > 1.0
Add feature Percentage of parts with relative MBB volume > 1.0

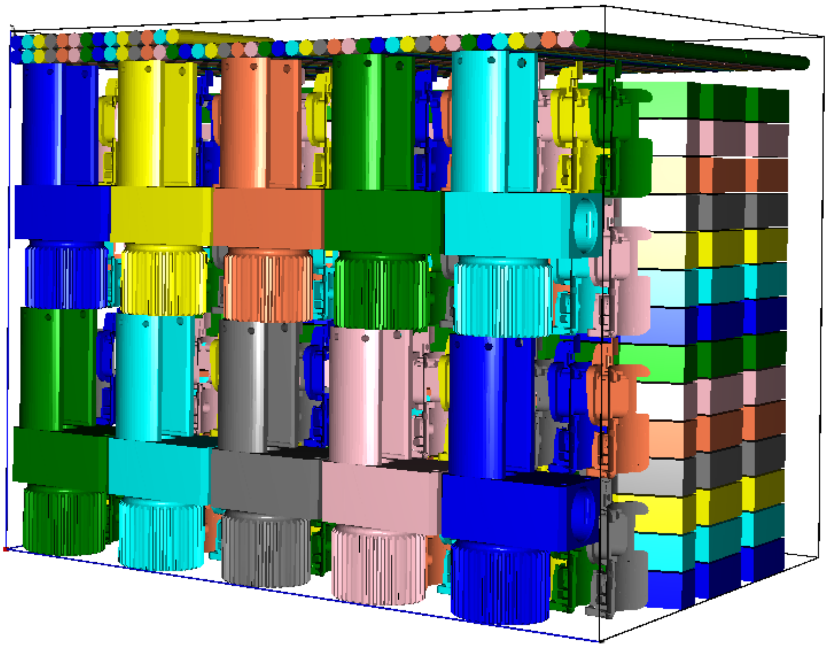
- Z-score, or standard: removes the mean and scales the values to the unit variance (Kotsiantis et al., 2006; Shalabi et al., 2006)
- Min-max: scales the feature values to the interval [0, 1] (Kotsiantis et al., 2006; Shalabi et al., 2006)
- Quantile: non-linear transformation based on the quantile functions to map the feature values to a uniform distribution with values in the interval [0, 1] (Bolstad et al., 2003)

5.1.3 Portfolio of packing algorithms and performance indicators

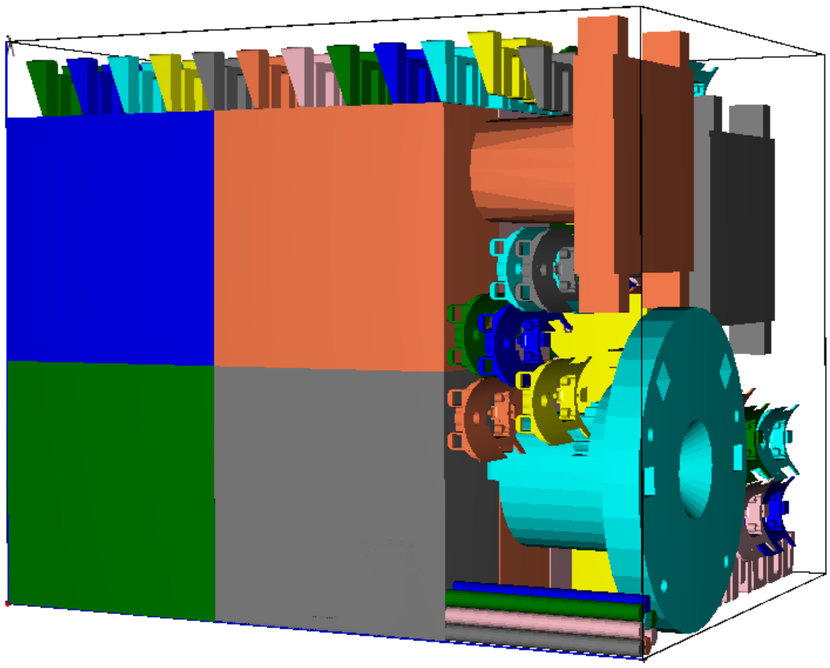
This initial study considers a portfolio with two deterministic packing algorithms for predicting the best performing algorithm based on the features of a given problem instance. The first algorithm is the WB approach presented by Araújo and Pinheiro (2010), which builds either one horizontal or vertical layer that maximises the total packed volume at each recursive call targeting the available space. The second algorithm is the DBLFD algorithm described in chapter 4. A preliminary overview of the solutions generated by both algorithms reveal that while WB arranges clusters of similar parts within the build volume, DBLFD can arrange different items close to each other, arranging smaller objects within the concavities of larger parts. Figure 5.2 illustrates the solutions obtained for one particular instance.

The aforementioned algorithms are used for solving all the instances in the 3DIP dataset. For each instance, the algorithms generate an output containing the specifications for the position and rotation of the packed parts within the container. From the set of packed parts, it is possible to extract the following performance metrics:

- Build volume utilisation (BVU), which is the ration between the total volume of packed parts and the volume of the container



(a) BVU = 0.3529; # of packed parts = 270; runtime = 2.1s



(b) BVU = 0.6380; # of packed parts = 247; runtime = 1053.4s

Figure 5.2: Solutions from the (a) WB and (b) DBLFD algorithms for the instance 1319 from the A2018 dataset (Araújo et al., 2018).

- Sum of the volume of the packed parts
- Number of packed parts
- Mean of the Mean Connectivity Value (MCV) of the packed parts
- Standard deviation of the MCV of the packed parts
- Mean of the V-values of the packed parts
- Standard deviation of the V-values of the packed parts
- Mean of the Spies ratio (SR) of the packed parts
- Standard deviation of the SR of the packed parts
- Mean of the number of faces of the packed parts
- Standard deviation of the number of faces of the packed parts
- Mean of the volume of the packed parts
- Standard deviation of the volume of the packed parts
- Runtime for solving the instance

Finally, each entry in the dataset is labelled with the indicators shown in Table 5.1.

Table 5.1: Additional indicators utilised as output for predictions.

Indicator	Value
Delta BVU	BVU(WB) - BVU(DBLFD)
Flag WB	+1 if BVU(WB) $\geq$ BVU(DBLFD); 0 otherwise
Flag DBLFD	+1 if *Flag WB* = 0; 0 otherwise
Flag Low BVU	+1 if BVU(WB) $<$ 35% and BVU(DBLFD) $<$ 35%; 0 otherwise

Many algorithm selection methods for combinatorial optimisation evaluate the performance of algorithms by comparing the runtime it takes each algorithm to achieve a certain objective value when solving a given instance (Hutter et al., 2015). By contrast, this study prioritises BVU as the objective and hence the performance indicator for packing approaches, given the economic relevance of full utilisation of productive capacity. Moreover, both algorithms tested in our algorithm selection approach demonstrated acceptable runtimes for typical AM processes. In fact, 'the lead times for tooling and fixturing are often longer than the time to the fabricate the product itself' (Conner et al., 2014, page 66). The authors have made the generated data on algorithm performance publicly available [1] in order to support and promote the development of computational tools for 3DIP in AM applications (Wuest et al., 2016).

5.1.4 Modelling and setting of parameters

Identifying the most appropriate machine learning model and finding its best parameter setting are extremely complex and computationally expensive tasks (Thornton et al., 2012). However, they are extremely beneficial for improving the performance of the ML algorithm. Identifying what is 'appropriate' is itself challenging and might vary according to the evaluation criteria, e.g., accuracy and precision rates. Furthermore, some stakeholders might favour learning algorithms capable of generating interpretable models or data structures. Table 5.2 presents some of the most commonly used classification models and parameter settings for the optimisation problems presented in Table 2.5.

A commonly used model selection method is the repeated k-fold cross-validation method, which is useful as it attributes a robust assessment score to the models (Arlot and Celisse, 2010). First, the k-fold cross-validation (CV) procedure splits the dataset into k folds. Each fold is used once to calculate an assessment metric (e.g. accuracy), while the remaining $k-1$ folds form the training data for the assessed model. The CV score is calculated as the mean of the score values associated with each fold (Kohavi and Others, 1995). This procedure is then repeated several times, with each repetition performing a random split to mitigate the influence of biased splits. This process leads to the selection of a model with a high generalisation capability, i.e., a classifier able to properly fit new data (Arlot and Celisse, 2010; Krstajic et al., 2014).

The classifiers exhibited in Table 5.2 accept a wide range of parameters which can significantly affect the predictive performance. This study implements an exhaustive grid search with repeated k-fold to select the parameter set from all possible combinations (Krstajic et al., 2014). Finally, Procedure 3 presents the algorithm for generating data, which yields the following attributes: data pre-processing strategy; classification model; parameter setting; number of folds (k); the name of the performance metric; mean and standard deviation of the performance metric collected through 30 repetitions; and mean and standard deviation of the training time.

5.1.5 Feature selection

Selecting the most relevant features ensures more accurate classification models and the generation of data structures that are easier to interpret (Kotsiantis et al., 2006). Moreover, pruning irrelevant or redundant features reduces the computational cost (Gu et al., 2012). However, identifying the subset that optimises the chosen performance metric is becomes more challenging and computationally expensive as the number of features increases (Chandrashekar and Sahin, 2014). Hence most feature selection methods are heuristic in nature, and are classified

[1] `http://www.cs.nott.ac.uk/~psxlja/algo-selec/`

Table 5.2: Summary of the main classifiers employed for algorithm selection.

Classifier	Parameters and their settings	Sample source(s)
Gaussian Naive Bayes	Non-specified prior probabilities of the predicted classes in the testing dataset	Kanda et al. (2010); Geschwender et al. (2016)
Logistic Regression	• Regularisation: L1 (Lasso) or L2 (Ridge), which are strategies for generating less complex models and prevent over-fitting • Regularisation strength (C^{-1}): a sequence of 10 numbers equally spaced on a log scale between 1 and 10^4	Pise and Kulkarni (2016); Kotthoff et al. (2015); Ilany and Gal (2016)
Decision Tree (CART)	• Splitting criterion at each node: Gini impurity (likelihood of an observation being incorrectly labelled) or Entropy (index to measure how much information is encoded in a particular decision) • Maximum depth: None (nodes are expanded until all the observations in the leaves are homogeneous), 1, 2, 3, 4, 5 or 6	Smith-Miles (2009); Kanda et al. (2010); King et al. (2013, 2015)
Decision tree with AdaBoost	• Number of estimators: 10, 50, 100 • Maximum depth: None	Davami and Sukthankar (2015)
Random Forest	• Number of trees: 1, 10, 50, 100 or 150 • Criterion: Gini impurity or Entropy • Maximum depth: None, 1, 2, 3, 4 or 5	King et al. (2013, 2015)
K-nearest neighbours	• Number of neighbours: 5, 10, 50 • Weights: Uniform (simple majority vote of the k nearest neighbours) or distance (weights are proportional to the inverse of the distance from the query observation)	Burke et al. (2006); Kanda et al. (2010); López-Camacho et al. (2014)
Multi-layer Perceptron	• Number of hidden layers: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50	Smith-Miles (2009); King et al. (2013, 2015)

according to two general strategies: filtering techniques, which iteratively drop less important features (Hall, 1999; Molina et al., 2002), and wrapper methods, which add the best or remove the worst feature at each interaction (Blum and Langley, 1997).

The most appropriate feature selection approach often depends on the characteristics of the data (e.g. the presence of outliers), the classification model and the domain of the problem to be addressed. Furthermore, there is no consensus on any one technique that performs best for all fields and models (Piramuthu, 2004). Therefore, this study assesses three widely used supervised feature selection methods:

- Removing the features with below-threshold value variance (John et al., 1994; Collins and Liu, 2003)

Algorithm 3: Generating data on the classifiers and parameters.

```
List of data preprocessing = [ Z-score, Min-max, Quantile Transformer ]
List of classifiers = [ Logistic Regression, Decision Tree, Random Forest, K-Neighbours,
  Multi-layer Perceptron, AdaBoost ]
List of kfolds = [ 5, 10 ]
List of scores = [ accuracy, precision, ROC AUC ]
foreach data preprocessing in List of data preprocessing do
    foreach classifier in List of classifiers do
        foreach kfold in List of kfolds do
            foreach score in List of scores do
                Run Grid Search for (data transformation, classifier, score) with Repeated (30
                  times) K-fold (#fold = kfold) cross-validation
                Collect tuples: data preprocessing, classifier, parameter setting, kfold, score,
                  mean score, standard deviation (SD), mean time, and SD of time
            end
        end
    end
end
```

- Removing the features with the lowest Fisher score, which is defined as the ratio of variance between classes to variance within classes (Gu et al., 2012; Aggarwal, 2014).
- Recursive feature elimination with repeated CV (RFECV): removes the least important features at each recursive call (Varma and Simon, 2006; Yan and Zhang, 2015)

5.2 Experimental results

This section presents the results of the algorithm selection approach described in Section 5.1. First, the standalone performance for each packing algorithm and the oracle performance, which constitutes the upper bound of the algorithm selection technique, are computed. Second, data pre-processing strategies, classification models and parameter settings is assessed againts the benchmark. Third, the outcomes from different feature selection methods are examined for patterns among the most relevant 3DIP features. Next, the selected predictive models are evaluated for gained volume utilisation, which is a useful indicator for operations management in AM. Lastly, easily interpretable decision rules are generated to provide insights into the main elements in the 3DIP domain.

5.2.1 Standalone algorithm performance and oracle performance

The WB and DBLFD algorithms were assessed to identify if an algorithm performs better than the other based on BVU. Both algorithms were run on all 2,000 benchmark entries. The results show that WB performed better than DBLFD on the 1,214 (60.7%) entries, while DBLFD performed better on the remaining 786 (39.3%). A between-class imbalance is a recurrent property in real-world data which increases the complexity of implementing classification models (Japkowicz, 2000). Table 5.3 compares the performances of WB and DBLFD, assuming that each algorithm is run on its own over all the problem instances. Not only is WB the best choice for packing in AM, but it is also proved faster than DBLFD, taking 32.7 sec as opposed to 5,969 sec to obtain a result an average BVU of 0.43.

Table 5.3: Number of instances with higher BVU, mean and standard deviation of both BVU and runtime.

Algorithm	Instances with higher BVU	Build volume utilisation		Runtime	
		Mean	Standard deviation	Mean (seconds)	Standard deviation
WB	1,214 (60.7%)	0.43	0.19	32.7	129.3
DBLFD	786 (39.3%)	0.39	0.19	5,969.0	15,312.6

When dealing with imbalanced data, and considering the significant cost of false positives when predicting *Flag DBLFD*, metrics other than accuracy are more suitable for assessing classification models (Han and Kamber, 2001). Moreover, some applications require a high precision rate (ratio between true positive observations and positive predictions) for the minority class (Chawla et al., 2002). This study focuses, therefore, on classifiers that result in high precision for the *Flag DBLFD* indicator, given that false positives require considerable runtime and lead to lower BVU. An alternative metric is the *computed area under the receiver operating characteristic curve* (ROC AUC), which is a technique for summarising trade-offs between true positive and false positive error rates that occur during classification (Swets, 1998; Bradley, 1997). This metric is also more suitable than precision for assessing multi-class classification, which is convenient for handling more than two packing algorithms (Hand et al., 2001).

An interesting insight can be derived from the fact that the indicator *Flag Low BVU* can be observed in 32.3% (646) of the problem instances. This result reinforces the need for the inclusion of more packing algorithms to address instances for which both WB and DBLFD deliver results below the adopted threshold (35%). The characteristics of such instances are investigated in Section 5.2.5.

Having discussed the evaluation of the algorithms individually, it is appropriate to use a metric which can assess how the algorithm selection approach performs when embedding the chosen algorithms. One option is the *oracle performance* measure (Wagner et al., 2018), which provides the theoretical optimal performance of an algorithm selection procedure. The *oracle performance* for the given portfolio and set of instances can be calculated as follows (Xu et al., 2008):

$$oracle(A,I) = \frac{1}{|I|}\sum_{i \in I} max_{a \in A} BVU(a) \tag{5.1}$$

Where A is the set of algorithms (portfolio), I is the set of 3DIP instances (Araújo et al., 2018) and $BVU(a)$ is the build volume utilisation obtained by algorithm a. The calculated *oracle performance* is 0.46 with a standard deviation of 0.18 assuming that the best algorithm is chosen for each problem instance.

5.2.2 Analysis of features

Feature selection using only variance and Fisher score often leads to a suboptimal subset as these functions analyse each feature independently (Gu et al., 2012). As a result, the comparison between ranked lists generated by the criteria mentioned above not rarely exhibits disjunctive sets, as illustrated in Table 5.4. Fisher score is often preferable as it quantifies how useful a particular feature is for discriminating samples in different classes (He et al., 2005).

Table 5.4: Highest variances and Fisher scores among the features.

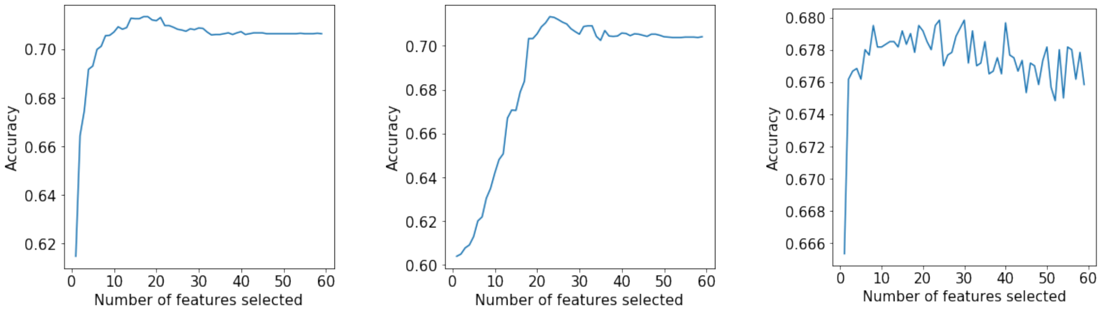
Highest variances		Highest Fisher scores	
Feature	Value	Feature	Value
Volume of the container	2.153E+15	Percentage of parts with relative volume in $(0.6, 0.8]$	35.41
Surface of the container	2.547E+11	RSD of demand	4.151
Mean of MBB volume of the parts	1.842E+11	Percentage of parts relative volume in $(0.4, 0.6]$	3.988
Mean file size of the parts (bytes)	6.986E+10	Mean V-value	0.842
Mean volume of the parts	5.841E+09	Mean SR	0.645
Mean of surface area of the parts	1.670E+08	Mean relative volume	0.614
Mean of number of faces of the parts	4.186E+07	RSD of volume	0.587
Mean of mean connectivity score of the parts	4.598E+05	RSD of relative surface area	0.502
Number of parts	3.636E+05	RSD of surface area	0.502
Mean of demand	1.791E+04	RSD of MCV	0.338

As mentioned in Section 5.1.5, another feature selection method tested in this study is RFECV, which iteratively removes the feature with less relevance for the employed classifier. In order to select such a classifier, preliminary experiments were conducted to assess the methods shown in Table 5.2 using all combinations of data preprocessing, number of folds and parameter settings. Table 5.5 presents the three classifiers with the highest mean accuracy, mean precision and mean ROC AUC scores after 30 runs.

Table 5.5: Models with the highest mean accuracy, mean precision and mean ROC AUC indicators.

	Preprocessing	Classifier	Folds	Parameters	Shorthand
Highest mean accuracy	Z-score	Logistic Regression	10	C: 1.0; penalty: L1	LR1
Highest mean precision	Min-max	Random Forest	10	criterion: entropy, max depth: 1, # estimators: 150	RF
Highest mean ROC AUC	Min-max	Logistic Regression	10	C: 10,000.0; penalty: L1	LR2

Having selected LR1, LR2 and RF, RFECV shows how the predictive metric varies according to the number of features, as shown in Figure 5.3. In this experiment, for consistency, accuracy is the adopted metric for the three classifiers.



(a) LR1 (C:1.0; penalty: L1)

(b) LR2 (C:10,000.0; penalty: L1)

(c) RF (Entropy; max. depth:1; # estimators: 150)

Figure 5.3: Accuracy per number of features during the recursive feature elimination with cross-validation.

As shown in Figure 5.3, it is not necessary to use the entire set of 59 features to achieve the best accuracy. The highest accuracy for the LR1, LR2 and RF classifiers is 71.42% (18 features), 71.35% (23 features) and 67.99% (features), respectively. One interesting observation is that the five feature ranking schemes (variance, Fisher, RFECV for LR1, RFECV for LR2 and RFECV for RF) that are generated show very different variations of the most relevant features, as illustrated in Figure 5.4.

One strategy to further analyse the most relevant features in 3DIP is to use the 18 features selected using RFECV for LR1, as the results of this scheme showed the highest accuracy. The set of 18 features, in order of relevance beginning with the most relevant, are: percentage of parts with relative volume in (0.6, 0.8], mean SR, mean MCV*SR, number of parts, number of models, RSD of volume, mean demands, mean surface area, mean volume, mean relative surface area, surface of container, percentage of parts with SR in [0.0, 0.2], percentage of parts with relative volume in (0.2, 0.4], percentage of parts with MCV in (0.8, 1.0], RSD of relative MBB volumes, RSD of relative volumes, RSD of connectivity scores and RSD of SR. The objective is to analyse how the use of this subset of features affects the performance of all the models and parameters shown in Table 5.8.

Next, the performance of classifiers using the reduced 18 features is compared to the full set of features for accuracy, precision and ROC AUC. As is evident from Table 5.6, feature selection leads to an effective dimensionality reduction, consistently resulting in higher predictive ability for all the classification models based on any indicator.

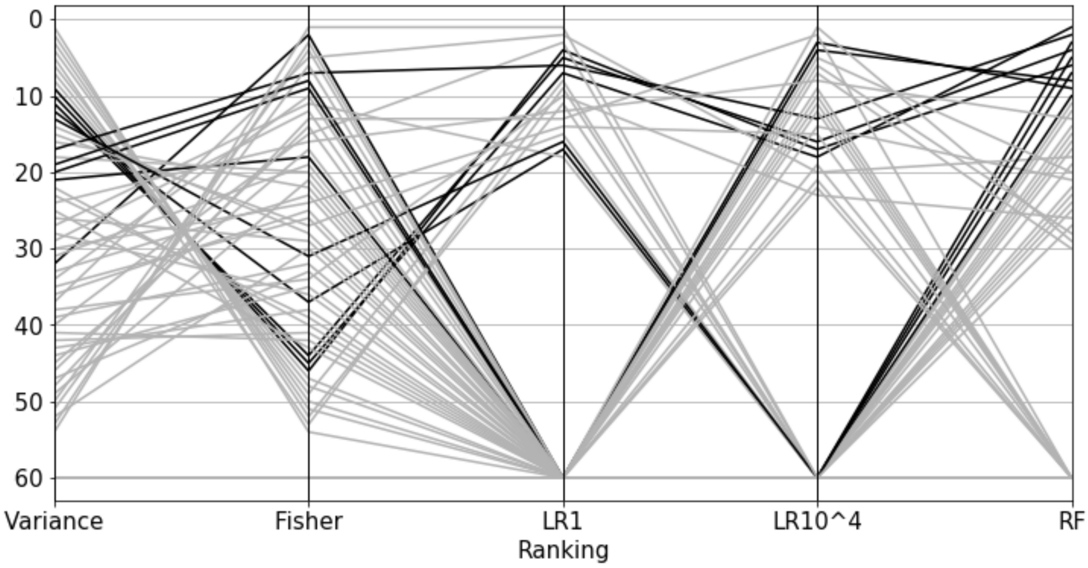


Figure 5.4: Parallel coordinates graph comparing feature ranking using variance and Fisher-based criteria and RFECV for LR1, LR2 and RF. The 10 more relevant features for RF are highlighted for the purposes of illustration.

Table 5.6: Performance variation after feature selection.

Model	Variation after feature selection[(a)]			
	Accuracy (%)	Precision (%)	ROC AUC (%)	Time[(b)]
AdaBoost	+0.706	+1.930	+1.005	-0.163
Decision Tree	+0.266	+0.251	+0.441	-0.016
Gaussian Naive Bayes	+0.286	+7.316	+1.078	-0.001
K-Neighbors	+2.040	+6.259	+0.617	-0.001
Logistic Regression	+1.830	+4.162	+0.697	-1.237
Multi-layer Perceptron	+1.523	+4.521	+1.958	-0.125
Random Forest	+0.763	+0.803	+1.092	-0.057

[(a)] Score with 18 selected features - score utilising the full set of features.
[(b)] Training time in seconds.

Table 5.6 shows that more precise and quicker predictive models can be obtained using the reduced set of features. The findings also reinforce the relevance of the features and methods introduced in Section 3.1. In fact, the ‘new’ adopted metrics, namely, MCV, v-value, SR and functions of these features (connectivity score and MBB volume) constitute 7 out of the 18 most relevant features and cannot be ignored when categorising and reporting 3DIP problems.

5.2.3 Data pre-processing, model selection and estimation of parameters

We have compared the performance of a number of combinations of methods and settings during training using the generated data for algorithm selection, joining pre-processing techniques, classifiers, parameter settings, the number of folds for cross-validation based on accuracy, precision and ROC AUC.

Table 5.7: Analysis of the predictive performance for each tested combination of methods.

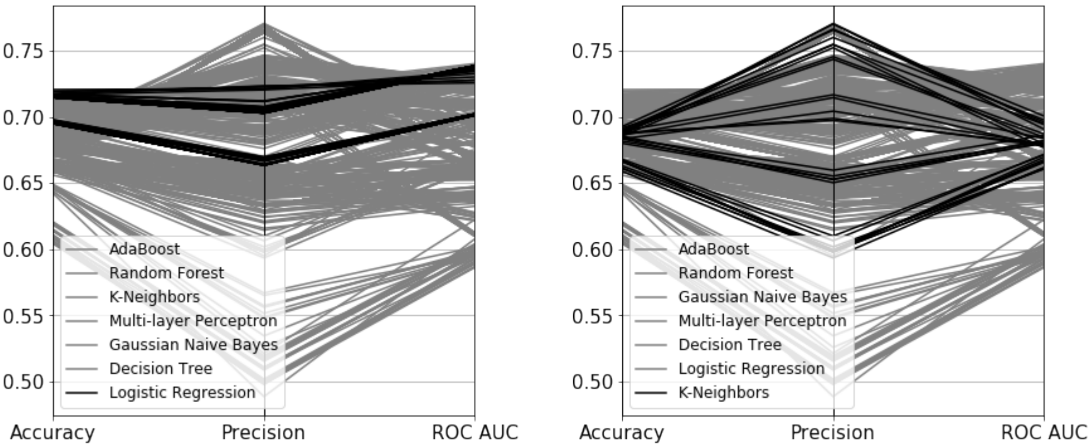
Method	Setting	Accuracy		Precision		ROC AUC	
		Mean	S.D.	Mean	S.D.	Mean	S.D.
Transformation	Min-max	**67.646**	0.138	76.463	0.291	**66.562**	0.147
	Quantile	66.187	0.278	64.303	1.058	65.768	0.139
	Z-score	67.581	0.156	**76.680**	0.299	**66.562**	0.132
Classifier	AdaBoost	69.033	0.261	65.188	0.529	70.070	0.860
	Decision Tree	66.666	2.074	64.079	5.776	64.429	2.569
	Gaussian Naive Bayes	67.137	0.699	**72.485**	5.820	66.295	0.375
	K-Neighbors	67.937	1.119	67.964	6.773	67.895	1.154
	Logistic Regression	**70.904**	0.936	69.285	1.990	**72.529**	1.694
	Multi-layer Perceptron	70.275	1.625	68.186	4.762	71.725	2.915
	Random Forest	68.117	2.010	69.620	5.638	68.806	3.907
Cross-validation	5-fold	66.965	0.744	**72.909**	5.359	66.276	0.408
	10-fold	**67.311**	0.611	72.055	6.221	**66.318**	0.390

(a) The highest mean score for each metric for each combination are displayed in bold.

Table 5.7 shows that min-max transformation resulted in higher mean accuracy and ROC AUC, resulting in a slightly lower mean precision when compared to the z-score transformation. Logistic Regression models for classification produced higher mean accuracy and ROC AUC than AdaBoost, decision tree, Gaussian naive Bayes, k-neighbours, LR, multi-layer perceptron and RF. Interestingly, Gaussian naive Bayes achieved the highest mean precision, suggesting that the 18 features used are weakly correlated. Cross-validation 5- and 10-fold settings achieved similar results, although 10-fold cross-validation produced slightly higher accuracy and ROC AUC. For this reason and because it is a common practice in most of the ML literature, this study adopted 10-fold cross-validation (Arlot and Celisse, 2010). Overall, as shown in Table 5.7, all the tested classifiers shown yield accuracy values that are consistently higher than the percentage of the majority class (60.7%). This observation is relevant as it confirms the usefulness of classification models in algorithm selection for 3DIP.

As previously mentioned, the performance of a classifier depends, among other aspects, on the parameter setting. Table 5.8 compares the performance of each classifier with various parameter settings, with the highest scores for mean accuracy, precision and ROC AUC displayed in bold. Only two configurations dominate the rest for all three predictive metrics: KNN with number of neighbours equal to 50 and RF using entropy as an impurity metric and 150 for the number of estimators.

An alternative strategy for model selection is to observe how often an individual classifier delivers the highest value based on the predictive metrics. The parallel coordinate graph presented in Figure 5.5 provides insights into the general performance of LR and RF classifiers. It can be observed, for example, that LR models are among those that result in the highest accuracy and ROC AUC scores. However, K-nearest neighbour models present the highest precision scores, being particularly useful for avoiding false positives when predicting instances to be solved by the DBLFD algorithm.



(a) Logistic Regression

(b) K-nearest neighbours

Figure 5.5: Parallel coordinates graph highlighting the higher precision obtained by several K-nearest neighbour models.

The following combination of algorithm choices and parameter settings are used in the next sections, unless mentioned otherwise. Min-max is used as the data pre-processing method. For the classifiers and their settings, the following ML models are used, based on the fact that each classifier and setting ranks first for each indicator:

- **LR3**: Logistic Regression with C= 2.782 using 10-fold cross-validation and an L2 penalty, which results in the highest accuracy at 72.06%

- **KNN1**: K-nearest neighbours with 50 neighbours using 10-fold cross-validation and uniform weights, which yields the highest precision at 77.05%

- **MLP1**: Multi-layer Perceptron with 50 hidden layers and 5-fold cross-validation, yielding the highest ROC AUC at 74.02%

Table 5.8: Average performance of each classifier grouped by parameter values.

Model	Parameter	Value	Mean accuracy (%)	Mean precision (%)	Mean ROC AUC (%)
AdaBoost	Number of estimators	10	68.77	**65.64**	69.04
		50	**69.36**	65.42	**71.02**
		100	68.97	64.50	70.16
Decision Tree	Criterion	Entropy	66.66	**64.13**	**64.63**
		Gini	**66.68**	64.03	64.23
	Maximum depth	1	67.70	**71.81**	61.11
		2	**67.92**	65.79	65.34
		3	67.66	66.01	**66.53**
		4	66.93	64.42	66.30
		5	67.41	65.15	66.02
		6	67.36	64.01	65.80
		None	61.69	51.37	59.92
K-nearest neighbours	Number of neighbours	10	68.49	68.03	68.10
		5	66.44	60.03	66.57
		50	**68.88**	**75.83**	**69.02**
	Weights	Distance	67.91	66.91	**68.02**
		Uniform	**67.97**	**69.02**	67.77
Logistic Regression	C	1.0	**71.00**	**69.89**	72.23
		2.782	**71.00**	69.60	72.48
		7.742	70.88	69.20	72.51
		21.5442	70.90	69.22	72.58
		59.948	70.83	69.02	72.55
		166.810	70.88	69.15	**72.60**
		464.158	70.88	69.15	**72.60**
		1291.549	70.88	69.15	**72.60**
		3593.813	70.81	68.99	72.56
		10,000.0	70.88	69.15	**72.60**
	penalty	L1	70.90	69.26	**72.54**
		L2	**70.91**	**69.31**	72.52
Multi-layer Perceptron	Number of hidden layers	1	65.45	53.50	64.09
		2	68.97	65.69	68.61
		3	70.04	67.36	70.68
		4	70.37	69.57	71.26
		5	70.49	70.07	71.96
		6	70.73	**70.39**	72.23
		7	70.80	70.32	72.54
		8	70.91	**70.39**	72.69
		9	70.96	70.21	72.98
		10	70.95	70.28	73.00
		20	**71.09**	69.62	73.45
		30	**71.09**	69.35	73.52
		40	71.05	68.86	**73.60**
		50	70.96	69.00	73.53
Random Forest	Criterion	Entropy	**68.14**	**69.81**	**68.84**
		Gini	68.10	69.43	68.78
	Maximum depth	1	67.11	69.00	65.48
		2	67.67	70.60	67.93
		3	68.05	71.07	69.32
		4	68.64	**71.38**	70.08
		5	**69.11**	71.12	**70.51**
		None	68.11	64.55	69.52
	Number of estimators	1	64.96	60.36	62.16
		10	68.38	70.46	69.30
		50	69.01	72.23	70.69
		100	69.11	72.47	70.90
		150	**69.14**	**72.58**	**70.98**

(*) The highest average scores for each parameter are shown in bold.

The classifiers above are used to validate the algorithm selection approach, along with the previously selected classifiers (LR1. LR2 and RF1).

5.2.4 Assessing the contribution of the portfolio for volume utilisation

This section focuses on the usefulness of the portfolio with regard to BVU compared to the 'winner- takes-all' approach. For each 3DIP instance, the three selected classifiers were trained with the remaining data and used to predict the packing algorithm that would deliver the highest BVU. Table 5.9 shows the BVU obtained by DBLFD and WB, the best algorithm and the packing algorithm predicted by each classifier for a subset of instances. The last two rows present the mean and standard deviation of BVU for each algorithm, the optimal algorithm selection and the BVU obtained from each classifier.

Table 5.9: Comparing the performance of DBLFD, WB and the ML algorithms used for predicting the best packing heuristic in terms of BVU on the benchmark problem instances (Due to the large number of instances only 10 of them are provided for the purposes of illustration).

Instance	BVU of DBLFD (%)	BVU of WB (%)	Classifier					
			LR1	LR2	RF1	LR3	KNN1	MLP1
# 0	18.96	18.96	x	x	x	x	x	x
# 1	13.29	18.86	✓	x	✓	✓	x	✓
# 2	08.59	08.58	x	x	✓	x	x	✓
# 3	18.10	16.62	✓	✓	✓	✓	✓	✓
# 4	12.65	25.09	✓	✓	✓	✓	✓	✓
...	...	...	...	...	...	...	...	...
# 1995	80.19	45.78	✓	✓	✓	✓	✓	✓
# 1996	78.29	39.92	✓	✓	✓	✓	✓	✓
# 1997	94.78	70.41	✓	✓	✓	✓	✓	✓
# 1998	52.62	41.71	✓	✓	✓	✓	✓	✓
# 1999	80.88	74.81	✓	✓	✓	✓	✓	✓
Mean BVU (%)	39.15	43.52	45.00	44.88	44.49	44.99	44.88	44.91
SD of BVU	19.03	19.20	19.84	19.85	19.62	19.83	19.85	19.62

* Performance of classifiers resulting in correct (✓) or incorrect (x) prediction of the best performing packing heuristic for a sample of 10 benchmark instances.

Table 5.9 shows that the mean BVU values obtained by the classifiers LR1, LR2, RF1, LR3, KNN1 and MLP1 are 45.00%, 44.88%, 44.49%, 44.99%, 44.88%, 44.91%, respectively, where BVU is averaged over 2,000 problem entries. All those values are consistently higher than the standalone performance of each packing heuristic on the benchmark instances, with DBLFD and WB returning mean BVU values of 39.15% and 43.52%, respectively. The performance of the best performing algorithm, i.e., LR1 (45.00%), is competitive against the performance of the *oracle*, which returned a mean BVU of 46.98%, as discussed in Section 5.2.1. This result confirms the potential of the proposed algorithm selection approach for delivering increased volume utilisation in 3DIP applications.

5.2.5 Decision rules for algorithm selection

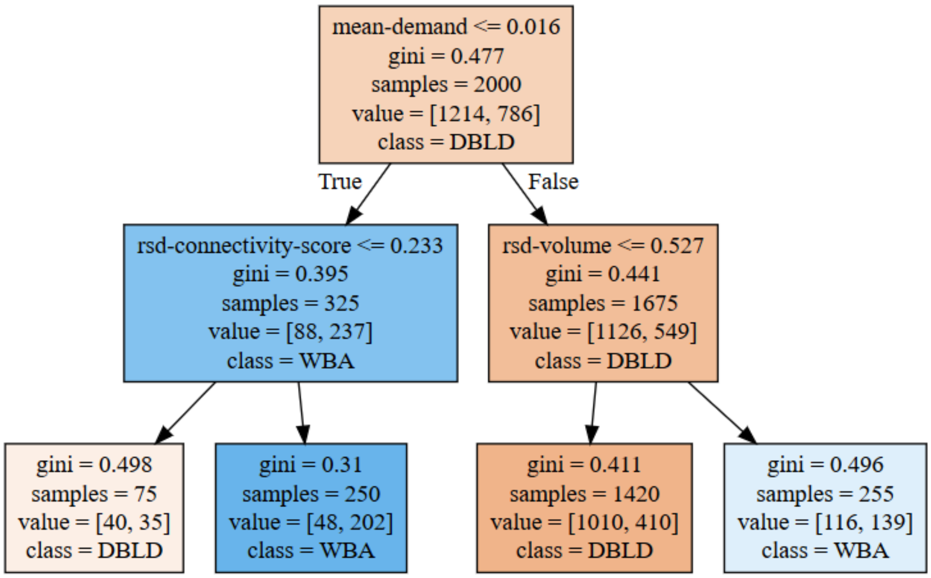
Classification and regression trees are procedures that generate predictive models describing easily interpretable rules to formalise the acquired knowledge. Such methods use heuristics that perform feature selection at each level of the tree, which helps researchers understand the underlying dataset (Guyon and Elisseeff, 2003). Based on the benchmark data discussed in Section 5.2.3, Gini index and maximum depth of the two are features that result in highest accuracy for decision trees. This configuration is used to generate models that can (i) identify the characteristics that favour either the WB or the DBLFD algorithm, and (ii) distinguish the features that indicate poor performance ($< 35\%$) for both algorithms. Figure 5.6 presents the resulting decision trees.

The decision trees illustrate how features that capture part complexity are, along with demand variation, relevant for predicting the best packing algorithm. For example, *relative standard deviation of connectivity score* (rsd-connectivity-score) was the best splitting variable (Gini equal to 0.395) of a node with depth 2 in the decision tree. A second observation that can be made from Figure 5.6a is that low *mean demand* favours the WB algorithm, while higher values for that feature favours the DBLFD approach. Regarding instances for which both WB and DBLFD result in low ($< 35\%$) BVU, Figure 5.6b shows that MCV, SR and number of parts play an essential role in the identification of such instances. For future development of packing algorithms, this data could be used increase the overall performance of the portfolio.

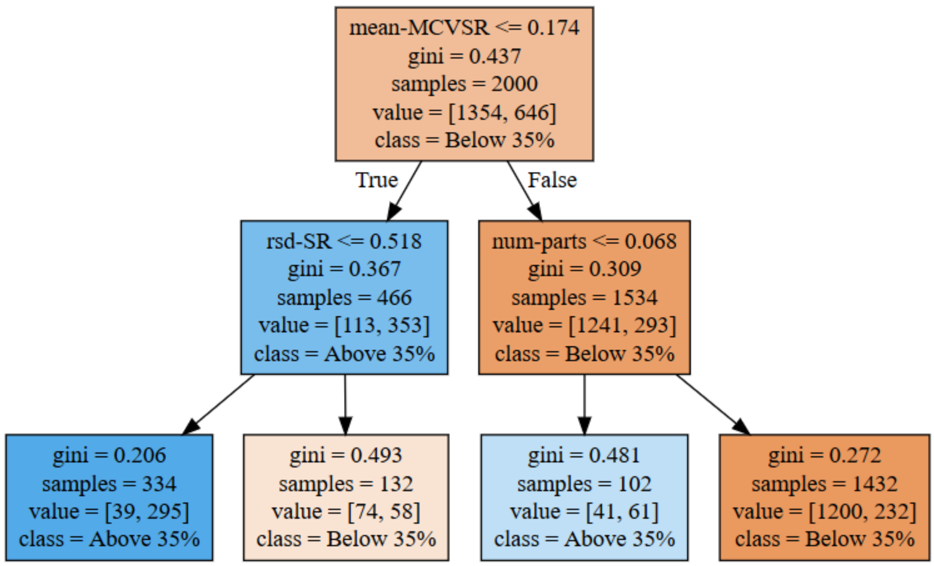
5.3 Discussion

The experiments conducted in this study indicate that, as with other combinatorial optimisation problems, 3DIP can benefit from the use of algorithm selection methods (Burke et al., 2010; Pillay, 2012; López-Camacho et al., 2014). To date, studies on algorithm selection for C&P have largely been restricted to one- and two-dimensional problems dealing with simple rectangular objects. This is one of the first studies that illustrates the viability of algorithm selection for the 3DIP domain, incorporating ML and supporting intelligent planning and scheduling in AM. As a side benefit, this study also contributes to an enhanced understanding of the main characteristics of 3DIP problems.

The first experiment demonstrates the advantage of WB algorithms over the DBLFD heuristic with regard to BVU and runtime (George and Robinson, 1980; Pisinger, 2002; Karabulut and Murat, 2004; Araújo and Pinheiro, 2010). This outcome was primarily determined by the simplified boundary representation for complex 3D geometries, which allows rapid geometric



(a) Generated rules for algorithm selection for a portfolio consisting of the WB and the DBLFD algorithms.



(b) Characteristics of instances with poor performance for the portfolio.

Figure 5.6: Decision trees for algorithm selection and complex instances.

feasibility testing and efficient geometric transformations such as rotation and translation. This study also draws attention to the high cost associated with false positives when predicting instances to be solved using the DBLFD heuristic, which can result in considerably higher runtime and lower volume utilisation. This result suggests that evaluation metrics other than accuracy rate are more suitable for assessing the quality of the classifiers to be integrated into the algorithm selection approach.

Most of the studies on algorithm selection for combinatorial optimisation problems (see Section 2.7) offer little or no justification for the use of a particular learning model or parameter settings. The approach taken here and the data made publicly available as part of this study aim to support researchers and practitioners in the 3DIP domain.

This study addresses the observation made by some authors that WB algorithms are more suitable for weakly heterogeneous demands (Abdou and Elmasry, 1999; Gehring H., 1997; Pisinger, 2002; Wäscher et al., 2007; Araújo and Pinheiro, 2010; Bortfeldt and Gehring, 1998). This study identifies precise decision rules that support such an affirmation, diminishing ambiguity in 3DIP categorisation. Moreover, it emphasises that part features like the volume of the bounding box and part complexity play a crucial role in predicting algorithm performance and in recognising instances with expected low BVU for both WB and DBLFD algorithms.

From an operations management perspective, the use of algorithm selection for 3DIP is attractive due to its capacity to increase BVU in AM processes. For example, the results reported in this study demonstrate that the use of an LR classifier led to an average BVU of 45.00% BVU, which is 1.48% higher than that achieved by using only the WB algorithm (43.52%). Although this increase might not seem impressive, the reader must consider that better results can be obtained when integrating alternative packing algorithms into the portfolio. This finding illustrates the potential of such a method to increase efficiency in manufacturing. Future research on packing methods could, therefore, focus on instances that resulted in low efficiency in the portfolio implemented in this thesis (Leyton-Brown et al., 2003).

Chapter 6

Conclusions and future work

This chapter summarises the key contributions of the thesis, presents the most important results, and suggests directions for future research.

6.1 Summary

The main objective of this study is to promote more efficient AM processes by enabling increased volume utilisation during the configuration stage of production. In order to achieve this objective, multifaceted research was conducted which required a range of skills in areas including operations research, computer graphics, software development and data science. The results obtained during the research push the boundaries of existing knowledge in the 3DIP domain, particularly within AM and address certain practical issues that can arise when manufacturing high-value and complex objects (see section 6.2).

The use of packing algorithms to improve AM processes is not a new concept. The first studies date back to the 1990s and were motivated by the economic benefits that can be achieved by fully utilising the available space (Ikonen et al., 1996). A variety of approaches for solving C&P problems have since been introduced, each of which has advantages and drawbacks. However, most of these methods have addressed regular packing problems and are not entirely aligned with contemporary requirements regarding part complexity and high demand such as those found in AM processes. Some solutions capable of packing arbitrary three-dimensional geometries have been published, but these have considered fewer and less complex parts than are considered here. Algorithms for both two and three-dimensional problems mostly implement heuristics or metaheuristics to cope with the high computational effort required to process a large number of parts or highly convoluted objects. In summary, there is a wide variety of packing algorithms, and identifying the best alternative is not straightforward, since it depends on the characteristics of a particular problem.

Despite the potential practical benefits of analysing the relationship between the efficiency of packing algorithms and classes of 3DIP problems, a study of this kind has not yet appeared in the OR literature. The reasons for this gap include the consideration of an insufficient number of features to describe the kind of three-dimensional objects that are relevant for AM, the sometimes ambiguous categorisation and terminology in the existing C&P taxonomies, and the lack of a significant amount of data on algorithm performance for realistic problem instances. These issues are addressed in Chapter 3, in which two new features for part complexity are introduced and a practical method for complexity measurement is proposed; an extended C&P taxonomy is also presented to address the limitations of the previous categorisations. Finally, a comprehensive dataset has been made publicly available to stand as a useful resource for future practitioners and researchers to analyse packing algorithms across a wider variety of problems. To increase its usefulness for the research community and save users' time, the features of the data have been calculated.

Using the enhanced knowledge on the 3DIP-domain obtained from the extended taxonomy and the new data, the research investigates how the performance of 3DIP algorithms can vary according to problem features. This is aimed at demonstrating the strengths and pitfalls of existing 3DIP algorithms for solving realistic instances. The first research direction is an examination of the usefulness of DBLF-based methods, which serves as the basis for several, if not most, existing 3DIP algorithms. The integration of genetic algorithms and placement policies, sometimes referred as GARP, occupy a special position among packing approaches for 3DIP. Chapter 4 demonstrates that, contrary to what might be presumed, higher degrees of freedom for part rotation during packing do not translate into significantly gains in volume utilisation. One of the conclusions is that when employing a DBLF-based algorithm, individual optimisation of each part is recommended before packing.

The second research direction concerns the implementation of machine-learning-based algorithm selection for 3DIP problems, discussed in Chapter 5. To the best of the author's knowledge, this study stands as the first work in the literature on the relationship between the 3DIP search space and packing performance. An additional outcome is the generation and sharing of a large amount of data, which others can use to evaluate their own approaches or for developing their own classification models and feature selection techniques for the 3DIP domain. This work focuses, therefore, on offering resources to enable more precise and well-informed decision-making during the modelling task that occurs in the early stages of an AM process.

The results obtained during the research enabled a performance comparison of three-dimensional irregular packing algorithms based on DBLF and wall-building and DBLFD heuristics. Although heuristics are fast, the other approaches, such as metaheuristics with DBLF embedded are slower and require significant computational effort for collision detection between non-rectangular objects. For example, although GA can successfully contribute to the identification of suitable packing sequences for DBLF and yield improved results, its use as a population-based metaheuristic requires additional computational effort due to the need to process multiple candidate solutions during the search, as discussed in Chapter 4. Chapter 5 demonstrated that WB resulted in higher BVU for the majority of benchmark in- stances, which can be attributed to the capability of the WB heuristic to accommodate rotation of the bounding boxes of parts.

6.2 Key results

This section summarises the key results achieved in this thesis, which contribute to an increased comprehension of problems and algorithmic performance in the 3DIP domain.

New features for part complexity. It is reasonable to assert that part complexity is among the main aspects that influence packing efficiency, and this is confirmed by the results in this thesis. This study resulted in two complexity measurements to quantify degrees of non-convexity for three-dimensional objects: mean connectivity value (MCV) and v-value. A practical method for calculating a third commonly considered measure, the SR, was also proposed. The comparison of these metrics to the existing ones demonstrated that the new features successfully measure higher complexity and irregularity, while they does not result in data redundancy. Moreover, the use of the proposed measurements resulted in higher accuracy of the most of the predictive models presented in Chapter 5.

An extended taxonomy for cutting and packing. Notwithstanding their undisputed relevance for C&P research in the field of OR, the existing taxonomies contain ambiguous and subjective criteria for some classes of problems, as discussed in Section 2.4. This inhibits straightforward identification of the proper literature and solutions for particular 3DIP problems. The extended taxonomy proposed in this thesis addressed the limitations of the previous taxonomies, resulting in a taxonomy that facilitates categorisation of packing problems. Moreover, the new taxonomy allows the description and representation of variants that have not been commonly addressed in the literature and are recurrent in AM processes. For example, while the three-dimensional irregular bin packing problem with distinctive containers has not been addressed in the literature, for the best of the authors' knowledge, it occurs within manufacturing environments with different 3D printing machines.

A comprehensive 3DIP dataset aligned with AM requirements. Among the factors that inhibit the identification of the most suitable algorithm for a particular 3DIP problem, are the limited number of available datasets and the limited heterogeneity of the feature values within those datasets. Also, most of the datasets appear to be generated for the purpose of presenting features that are suited to the proposed packing algorithm and do not apply to different approaches. The dataset introduced in Section 3.2.3 is by far the largest that has been made available to researchers in the 3DIP domain. Furthermore, it emulates the challenging real-world AM requirements and incorporating a wide range of feature values and industrial-like parts.

Assessment of the main DBLF-based methods within the AM sector. As shown in this thesis, most of the solutions for 3DIP problems combine a placement policy, notably the DBLF heuristic, with a search algorithm which aims to identify suitable sequences and orientations for the parts. A recurring characteristic of DBLF implementation is the use of orthogonal part rotation or no part rotation at all. This study examined the usefulness and feasibility of DBLF with higher degrees of freedom for part rotation. This investigation included a comparison of three selected strategies (brute force search, DBLF decreasing and a genetic algorithm) which provides valuable insight into parameter settings for such procedures. Contrary to intuition, higher degrees of freedom for part rotation, i.e. smaller angle increments for rotation, do not translate into gained volume utilisation. On the other hand, the use of orthogonal rotation by the DBLF heuristic demonstrated to yield reasonable results regarding runtime and volume utilisation.

The application of an algorithm selection pipeline to 3DIP. Algorithm selection has been widely used to improve problem-solving for several combinatorial optimisation problems. This technique uses significant amounts of data on historical algorithm performance and data science techniques to predict the most suitable algorithm for a particular problem. To the best of the author's knowledge, this thesis presents the first application of such an approach to the 3DIP domain. The reported results illustrate how the use of algorithm selection for challenging 3DIP instances achieve better BVU than the 'winner-takes-all' method often adopted in the C&P literature. The findings of this thesis reinforce the potential of such a method for commercial applications. This work also illustrates that the use of data science, particularly ML techniques, are certainly viable in AM and opens up new research directions in the area.

Identification of the most relevant 3DIP problem features for algorithm selection. An additional benefit that can be gained from the algorithm selection method implemented in this research is the identification of the most relevant 3DIP problem features. The experiments using different feature selection methods demonstrated that the features introduced in this study are among those that contribute to higher accuracy when predicting the most suitable 3DIP algorithm. As a result, such measurements cannot be omitted by studies on 3DIP problems and techniques. Moreover, the decision trees generated during the experiments facilitate the generation of decision rules regarding algorithm selection for 3DIP that are easily understandable by stakeholders and practitioners.

6.3 Future work

This thesis concludes with recommendations for future work, although these do not exhaust the possible research directions due to the applied nature of the subject.

Implement a multi-objective approach for packing density and surface quality. One research direction that was partially explored during the research was the concurrent optimisation of surface quality and BVU. Several authors have demonstrated that build orientation, followed by layer thickness, plays a major role in surface quality (Tumer et al., 1996; Byun and Lee, 2006; Vetterli et al., 2014). Moreover, experiments reported in the literature provide evidence that, for several AM technologies such as laser sintering, large vertical areas contribute to high surface roughness due to the *stair case effect* (Bacchewar et al., 2007). Figure 6.1 illustrates an output from the software developed by the author to calculate overall surface quality for a packing configuration. However, further analysis is necessary to demonstrate the presence of trade-offs between the minimisation of vertical surfaces and the maximisation of build volume and to analyse the prospective benefits of adopting a multi-objective approach to AM processes (Li and Zhang, 2013).

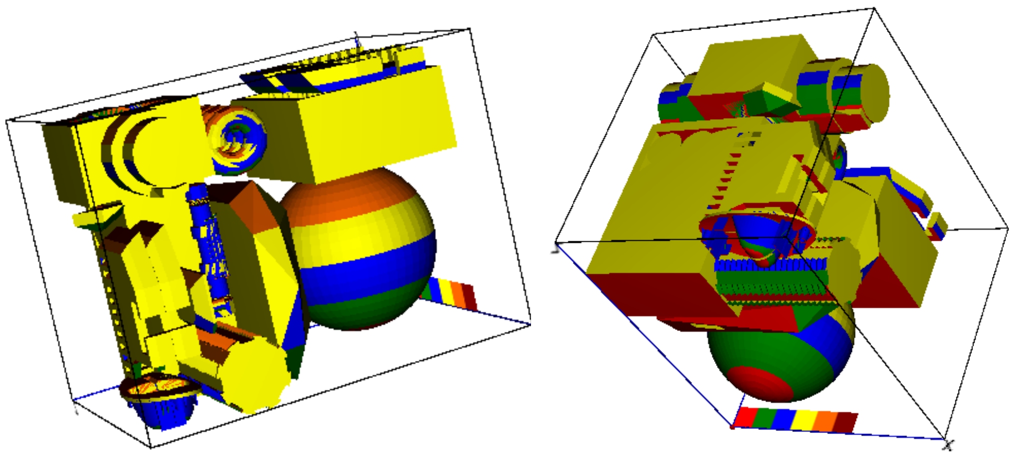


Figure 6.1: Rendering from the software developed during the research for measuring the overall surface roughness of a packing configuration.

Include stochastic algorithms in the portfolio. A useful property of the algorithms used in the algorithm selection pipeline shown in Section 5.1.3 is a deterministic output measured by the resultant build volume utilisation. However, most of the packing approaches reported in the literature combine stochastic search algorithms and placement policies. While the addition of such methods to the portfolio can lead to increased efficiency, it also adds a degree of complexity to the pipeline.

Test of layout configurations in real-world operations. The research conducted for this thesis provides satisfactory outcomes of several computer simulations and insights into 3DIP features. However, practical problems are likely to be encountered in real-world applications. For example, layer thickness is a machine parameter that, to some extent, affects the DBLF procedure during the sliding of each object. In addition, packing algorithms often ignore the fact that smaller parts cannot be manufactured within enclosed empty spaces of larger objects, as this would inhibit access to the contained components during post-production.

Commercial exploitation. Follow-up research is already planned to investigate the reduction of manufacturing cost for components of helicopter engines. Moreover, the use of the MCV complexity measurement developed during this research enables the detection of critical regions in mechanical parts, which is an additional business requirement for the planned research.

Bibliography

G. Abdou and M. Elmasry. 3D random stacking of weakly heterogeneous palletization problems. *International Journal of Production Research*, 37(7):1505–1524, 1999.

C. Achillas, D. Tzetzis, and M. O. Raimondo. Alternative production strategies based on the comparison of additive and traditional manufacturing technologies. *International Journal of Production Research*, 55(12):3497–3509, 2017.

C. C. Aggarwal. *Data classification: algorithms and applications*. CRC Press, 2014.

R. Al-Raoush and M. Alsaleh. Simulation of random packing of polydisperse particles. *Powder Technology*, 176(1):47–55, 2007.

A. Albert and J. A. Anderson. On the existence of maximum likelihood estimates in Logistic Regression Models. *Biometrika*, 71(1):1–10, 1984.

P. Alexander, S. Allen, and D. Dutta. Part orientation and build cost determination in layered manufacturing. *Computer-Aided Design*, 30(5):343–356, 1998.

S. Allen, E. K. Burke, M. Hyde, and G. Kendall. Evolving reusable 3d packing heuristics with genetic programming. *Proceedings of the 11th Annual conference on Genetic and evolutionary computation GECCO 09*, pages 931–938, 2009.

A. Ambrosi and M. Pumera. 3D-printing technologies for electrochemical applications. *Chemical Society Reviews*, 45(10):2740–2755, 2016.

A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 459–468, 2006.

L. J. Araújo, E. Özcan, J. A. Atkin, and M. Baumers. Analysis of irregular three-dimensional packing problems in additive manufacturing: a new taxonomy and dataset. *International Journal of Production Research*, pages 1–15, oct 2018.

L. J. P. Araújo and P. Pinheiro. Combining Heuristics Backtracking and Genetic Algorithm to Solve the Container Loading Problem with Weight Distribution. In *Soft Computing Models in Industrial* *and Environmental Applications, 5th International Workshop (SOCO 2010)*, pages 95–102. Springer, 2010.

L. J. P. Araújo, E. Özcan, J. J. Atkin, M. Baumers, C. Tuck, and R. J. M. Hague. Toward better build volume packing in additive manufacturing: classification of existing problems and benchmarks. *Proceedings of the Solid Freeform Fabrication Symposium*, pages 401–410, 2015.

L. J. P. Araújo, E. Özcan, J. A. D. Atkin, M. Baumers, C. Tuck, and R. Hague. A part complexity measurement method supporting 3D Printing. *NIP & Digital Fabrication Conference*, 2016.

S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

R. Art. *An approach to the two dimensional irregular cutting stock problem.* PhD thesis, Massachusetts Institute of Technology, 1966.

P. B. Bacchewar, S. K. Singhal, and P. M. Pandey. Statistical modelling and optimization of surface roughness in the selective laser sintering process. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 221(1):35–52, 2007.

S. Bagchi, S. Uckun, Y. Miyabe, and K. Kawamura. Exploring Problem-Specific Recombination Operators for Job Shop Scheduling. *Proc. of the Fourth Int. Conf. on Genetic Algorithms*, pages 10–17, 1991.

M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscek, I. Fasel, and J. R. Movellan. Recognizing facial expression: machine learning and application to spotaneous behavior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 568–573. IEEE, 2005.

M. Baumers, C. Tuck, R. Wildman, I. Ashcroft, E. Rosamond, and R. Hague. Transparency Built-in: Energy Consumption and Cost Estimation for Additive Manufacturing Baumers et al. Energy and Cost Estimation for Additive Manufacturing. *Journal of Industrial Ecology*, 17(3):418–431, 2013.

M. Baumers, P. Dickens, C. Tuck, and R. Hague. The cost of additive manufacturing: machine productivity, economies of scale and technology-push. *Technological forecasting and social change*, 102: 193–201, 2016a.

M. Baumers, C. Tuck, R. Wildman, I. Ashcroft, and R. Hague. Shape complexity and process energy consumption in electron beam melting: A case of something for nothing in additive manufacturing? *Journal of Industrial Ecology*, 2016b.

M. Baumers, L. Beltrametti, A. Gasparre, and R. Hague. Informing additive manufacturing technology adoption: total cost and the impact of capacity utilisation. *International Journal of Production Research*, 55(23):6957–6970, 2017a.

M. Baumers, C. Tuck, R. Wildman, I. Ashcroft, and R. Hague. Shape Complexity and Process Energy Consumption in Electron Beam Melting: A Case of Something for Nothing in Additive Manufacturing? *Journal of Industrial Ecology*, 21(S1):S157–S167, nov 2017b.

J. Beck and E. Freuder. Simple Rules for Low-Knowledge Algorithm Selection. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, 3011:50–64, 2004.

M. L. Benedikt. To Take Hold of Space: Isovists and Isovist View Fields. *Environment and Planning B: Planning and Design*, 6(1):47–65, 1979.

J. Bennell, G. Scheithauer, Y. Stoyan, and T. Romanova. Tools of mathematical modeling of arbitrary object packing problems. *Annals of Operations Research*, 179(1):343–368, 2010.

J. A. Bennell and J. F. Oliveira. A tutorial in irregular shape packing problems. *Journal of the Operational Research Society*, 60(sup1):93–105, may 2009.

J. A. Bennell and X. Song. A comprehensive and robust procedure for obtaining the nofit polygon using Minkowski sums. *Computers and Operations Research*, 35(1):267–281, 2008.

A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271, 1997. ISSN 00043702.

M. Boccia, S. di Muro, F. Mosca, A. Sforza, and C. Sterle. A fast heuristic for a three-dimensional non-convex domain loading problem. *4or*, 9(1):83–101, 2011.

B. M. Bolstad, R. A. Irizarry, M. Åstrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.

A. Bortfeldt and H. Gehring. A tabu search algorithm for weakly heterogeneous container loading problems. *OR SPEKTRUM*, 20(4):237–250, 1998.

A. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.

E. Burke, G. Kendall, J. Newall, E. Hart, P. Ross, and S. Schulenburg. Hyper-heuristics: An emerging direction in modern search technology. In *Handbook of metaheuristics*, pages 457–474. Springer, 2003a.

E. Burke, G. Kendall, E. S. J. of Heuristics, and U. 2003. A tabu-search hyperheuristic for timetabling and rostering. *Springer*, 2003b.

E. K. Burke, G. Kendall, and G. Whitwell. A New Placement Heuristic for the Orthogonal Stock-Cutting Problem, 2004.

E. K. Burke, S. Petrovic, and R. Qu. Case-based heuristic selection for timetabling problems. *Journal of Scheduling*, 9(2):115–132, 2006.

E. K. Burke, G. Kendall, and G. Whitwell. A simulated annealing enhancement of the best-fit heuristic for the orthogonal stock-cutting problem. *INFORMS Journal on Computing*, 21:505–516, 2009.

E. K. Burke, M. Hyde, G. Kendall, and J. Woodward. A genetic programming hyper-heuristic approach for evolving 2-D strip packing heuristics. *IEEE Transactions on Evolutionary Computation*, 14(6): 942–958, 2010.

E. K. Burke, G. Kendall, M. Misir, and E. Özcan. Monte Carlo hyper-heuristics for examination timetabling. *Annals of Operations Research*, 196(1):73–90, 2012.

E. K. Burke, M. Gendreau, M. Hyde, G. Ochoa, G. Kendall, E. Özcan, R. Qu, G. Ochoa, E. Özcan, and R. Qu. Hyper-heuristics : A Survey of the State of the Art. *Journal of the Operational Research Society*, 64(12):1695–1724, 2013.

H.-S. S. Byun and k. H. Lee. Determination of the optimal build direction for different rapid prototyping processes using multi-criterion decision making. *Robotics and Computer-Integrated Manufacturing*, Vol.22:69–80, 2006.

J. Cagan, D. Degentesh, and S. Yin. A simulated annealing-based algorithm using hierarchical models for general three- dimensional component layout. *Science*, 30(10):781–790, 1998.

J. Cagan, K. Shimada, and S. Yin. A survey of computational approaches to three-dimensional layout problems. *Computer-Aided Design*, 34(8):597–611, 2002.

V. Canellidis, V. Dedoussis, N. Mantzouratos, and S. Sofianopoulou. Pre-processing methodology for optimizing stereolithography apparatus build performance. *Computers in Industry*, 57(5):424–436, 2006.

V. Canellidis, J. Giannatsis, and V. Dedoussis. Genetic-algorithm-based multi-objective optimization of the build orientation in stereolithography. *International Journal of Advanced Manufacturing Technology*, 45(7-8):714–730, 2009.

V. Canellidis, J. Giannatsis, and V. Dedoussis. Effective nesting of layer manufacturing fabricated parts using a genetic algorithm and a bottom-left ray casting procedure. *IEEM2010 - IEEE International Conference on Industrial Engineering and Engineering Management*, pages 547–551, 2010.

V. Canellidis, J. Giannatsis, and V. Dedoussis. Efficient parts nesting schemes for improving stereolithography utilization. *CAD Computer Aided Design*, 45(5):875–886, 2013.

I. Carlbom, I. Chakravarty, and D. Vanderschel. A Hierarchical Data Structure for Representing the Spatial Decomposition of 3-D Objects. In *Frontiers in Computer Graphics*, pages 2–12. Springer Japan, Tokyo, 1985.

Y. Caseau, F. Laburthe, and G. Silverstein. A Meta-Heuristic Factory for Vehicle Routing Problems. *Cp'99*, pages 144–159, 1999.

V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985.

G. Chandrashekar and F. Sahin. A survey on feature selection methods. *Computers and Electrical Engineering*, 40(1):16–28, 2014.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

C. S. Chen, S.-M. Lee, and Q. S. Shen. An analytical model for the container loading problem. *European Journal of Operational Research*, 80(1):68–76, 1995.

N. Chernov, Y. Stoyan, and T. Romanova. Mathematical model and efficient algorithms for object packing problem. *Computational Geometry*, 43(5):535–553, jul 2010. ISSN 09257721.

V. Cicirello and S. F. Smith. The max k-armed bandit: A new model of exploration applied to search heuristic selection. *Proceedings of the National Conference on Artificial Intelligence*, 20:1355, 2005.

P. W. Cleary and M. L. Sawley. DEM modelling of industrial granular flows:3D case studies and the effect of particle shape on hopper discharge. *Appl.Math.Model.*, 26(2):89–111, 2002. ISSN 0307-904X.

R. T. Collins and Y. Liu. On-Line Selection of Discriminative Tracking Features. *IEEE International Conference on Computer Vision*, 2003.

B. P. Conner, G. P. Manogharan, A. N. Martof, L. M. Rodomsky, C. M. Rodomsky, D. C. Jordan, and J. W. Limperos. Making sense of 3-D printing: Creating a map of additive manufacturing products and services. *Additive Manufacturing*, 1:64–76, 2014.

M. Cotteleer and J. Joyce. 3D opportunity: Additive manufacturing paths to performance, innovation, and growth. *Deloitte Review*, 14:5–19, 2014.

T. M. Cover and P. E. Hart. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

D. R. Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 215–242, 1958.

T. G. Crainic, G. Perboli, and R. Tadei. TS2PACK: A two-level tabu search for the three-dimensional bin packing problem. *European Journal of Operational Research*, 195(3):744–760, 2009.

A. Criminisi, O. Pauly, B. Glocker, D. Mateus, A. M. Möller, S. Nekolla, and N. Navab. Regression Forests for Efficient Anatomy Detection and Localization in CT Studies. *MICCAI'11 Proceedings of the 14th international conference on Medical image computing and computer-assisted intervention - Volume Part III*, pages 239–247, 2011.

I. Daly. A review of feature selection techniques in BCI. *IEEE Trans. Systems, man and cybernetics. Part c; applications and reviews (submitted)*, pages 1–11, 2009.

E. Davami and G. Sukthankar. Improving the Performance of Mobile Phone Crowdsourcing Applications. *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 145–153, 2015.

C. Davatzikos, K. Ruparel, Y. Fan, D. G. Shen, M. Acharyya, J. W. Loughead, R. C. Gur, and D. D. Langleben. Classifying spatial patterns of brain activity with machine learning methods: application to lie detection. *Neuroimage*, 28(3):663–8, 2005. ISSN 1053-8119.

T. De La Rosa, S. J. Celorrio, and D. Borrajo. Learning Relational Decision Trees for Guiding Heuristic Planning. In *Icaps*, pages 60–67, 2008.

T. De La Rosa, S. Jiménez, R. Fuentetaja, and D. Borrajo. Scaling up heuristic planning with relational decision trees. *Journal of Artificial Intelligence Research*, 40:767–813, 2011.

H. T. Dean, Y. Tu, and J. F. Raffensperger. An improved method for calculating the no-fit polygon. *Computers and Operations Research*, 33(6):1521–1539, 2006.

J. K. Dickinson and G. K. Knopf. Serial packing of arbitrary 3d objects for optimizing layered manufacturing. In *Intelligent Robots and Computer Vision XVII: Algorithms, Techniques, and Active Vision*, volume 3522, pages 130–139. International Society for Optics and Photonics, 1998.

J. K. Dickinson and G. K. Knopf. Packing subsets of 3d parts for layered manufacturing. *International Journal of Smart Engineering System Design*, 4(3):147–161, 2002.

C. Domshlak, E. Karpas, and S. Markovitch. Max or not to max: Online learning for speeding up optimal planning. *Proceedings of the National Conference on Artificial Intelligence*, 2(Helmert):1071–1076, 2010.

K. A. Dowsland, S. Vaid, and W. B. Dowsland. An algorithm for polygon placement using a bottom-left strategy. *European Journal of Operational Research*, 141(2):371–381, 2002.

H. Dyckhoff. A typology of cutting and packing problems. *European Journal of Operational Research*, 44(2):145–159, 1990.

J. Egeblad. *Heuristics for Multidimensional Packing Problems*. PhD thesis, University of Copenhagen, 2008.

J. Egeblad. Placement of two- and three-dimensional irregular shapes for inertia moment and balance. *International Transactions in Operational Research*, 16(6):789–807, nov 2009.

J. Egeblad, B. K. Nielsen, and A. Odgaard. Fast neighborhood search for two- and three-dimensional nesting problems. *European Journal of Operational Research*, 183(3):1249–1266, dec 2007.

J. Egeblad, B. K. Nielsen, and M. Brazil. Translational packing of arbitrary polytopes. *Computational Geometry*, 42(4):269–288, may 2009.

J. Egeblad, C. Garavelli, S. Lisi, and D. Pisinger. Heuristics for container loading of furniture. *European Journal of Operational Research*, 200(3):881–892, 2010.

F. Eisenbrand, S. Funke, A. Karrenbauer, J. Reichel, and E. Schömer. Packing a trunk: now with a twist! *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, 14186:197–206, 2005.

E. Falkenauer and S. Bouffouix. A Genetic Algorithm for Job Shop. *CRIF - Research Center for Belgian Metalworking Industry*, pages 1–6, 1991.

G. Fasano. Cargo Analytical Integration in Space Engineering: A Three-dimensional Packing Model. In *Operational Research in Industry*, pages 232–246. Palgrave Macmillan UK, London, 1999.

G. Fasano. MIP Models for Solving 3-Dimensional Packing Problems Arising in Space Engineering. In *Operations Research in Space and Air*, pages 43–56. Springer, 2003.

G. Fasano. A MIP approach for some practical packing problems: Balancing constraints and tetris-like items. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 2(2): 161–174, jul 2004. ISSN 1619-4500.

G. Fasano. MIP-based heuristic for non-standard 3D-packing problems. *4OR*, 6(3):291–310, 2008.

Fasano, Giorgio and A. Novelli. A Cargo accommodation problem for a space vehicle: the CAST project. *Operations research in space and air*, 2003.

M. Fischetti and I. Luzzi. Exact and Heuristic MIP Models for Nesting Problems, 2003.

R. J. Fowler, M. S. Paterson, and S. L. Tanimoto. The complexity of packing and covering in the plane and related intersection graph problems. *Dep. Comput. Sci., Univ. of Washington, Seattle, TR*, pages 5–80, 1980.

R. J. Fowler, M. S. Paterson, and S. L. Tanimoto. Optimal packing and covering in the plane are NP-complete. *Information Processing Letters*, 12(3):133–137, 1981. ISSN 00200190.

F. Y. Fraige, P. A. Langston, and G. Z. Chen. Distinct element modelling of cubic particle packing and flow. *Powder Technology*, 186(3):224–240, 2008.

A. Garbajosa, T. De La Rosa, and R. Fuentetaja. Planning with ensembles of classifiers. *Frontiers in Artificial Intelligence and Applications*, 263:1007–1008, 2014.

J. Gardan. Additive manufacturing technologies: state of the art and trends. *International Journal of Production Research*, 54(10):3118–3132, 2016.

M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

B. A. Gehring H. A genetic algorithm for solving the container loading problem. *International Transactions in Operational Research*, 4(5-6):401–418, 1997.

M. Gell-Mann. What is complexity? In *Complexity and industrial clusters*, pages 13–24. Springer, 2002.

M. Gendreau and J.-Y. Potvin. *Handbook of metaheuristics*, volume 2. Springer, 2010.

M. Gendreau, M. Iori, G. Laporte, and S. Martello. A Tabu Search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks*, 51(1):4–18, jan 2008.

J. A. George and D. F. Robinson. A heuristic for packing boxes into a container. *Computers and Operations Research*, 7(3):147–156, 1980. ISSN 03050548.

D. J. Geschwender, R. Woodward, B. Choueiry, and S. Scott. A Portfolio Approach for Enforcing Minimality in a Tree Decomposition. *Proceeding of the International Conference on Principles and Practice of Constraint Programming*, 2016.

I. Gibson, D. Rosen, and B. Stucker. *Additive manufacturing technologies: 3D printing, rapid prototyping, and direct digital manufacturing*. Springer, 2014.

B. Glaser and A. Strauss. Grounded theory: The discovery of grounded theory. *Sociology The Journal Of The British Sociological Association*, 12:27–49, 1967.

F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5):533–549, 1986.

F. Glover. Tabu search - part I. *ORSA Journal on computing*, 1(3):190–206, 1989.

F. W. Glover and G. A. Kochenberger. *Handbook of Metaheuristics*, volume 57. Springer Science & Business Media, 2003.

a. S. Gogate and S. S. Pande. Intelligent layout planning for rapid prototyping. *International Journal of Production Research*, 46(20):5607–5631, 2008.

D. E. Goldberg and K. Deb. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. *Fga1*, 1:69–93, 1991.

J. J. Grefenstette. Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 1986.

P. M. Grignon and G. M. Fadel. A GA Based Configuration Design Optimization Method. *Journal of Mechanical Design*, 126(1):6, 2004. ISSN 10500472.

Q. Gu, Z. Li, and J. Han. Generalized Fisher Score for Feature Selection. *arxiv.org*, 2012.

I. Guyon and A. Elisseeff. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research (JMLR)*, 3(3):1157–1182, 2003.

R. Hague, I. Campbell, and P. Dickens. Implications on design of rapid manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 217(1): 25–30, 2003.

R. Hague, S. Mansour, and N. Saleh. Material and design considerations for rapid manufacturing. *International Journal of Production Research*, 42(22):4691–4708, 2004.

M. A. Hall. Correlation-based feature selection for machine learning. 1999.

J. Han and M. Kamber. Data Mining: Concepts and Techniques, chapter Mining association rules in large databases. 2001.

D. D. J. Hand, R. J. Till, R. T. M. Learning, and undefined 2001. A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. *Machine Learning*, 45(2):171–186, 2001.

X. He, D. Cai, and P. Niyogi. Laplacian Score for Feature Selection. In *Advances in Neural Information Processing Systems 18*, pages 507–514, 2005.

B. Hiller. Space is the Machine: A configurational theory of architecture. *Journal of Women s Health*, 1996.

B. Hillier and J. Hanson. The social logic of space. *Cambridge: Press syndicate of the University of Cambridge*, 1984.

D. S. Hochbaum. *Approximation algorithms for NP-hard problems*. PWS Publishing Co., 1996.

N. Hopkinson and P. Dickens. Analysis of rapid manufacturing - using layer manufacturing processes for production. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 217(1):31–39, 2003.

C. W. Hull. Apparatus for production of three-dimensional objects by stereolithography. *U.S. Patent No. 4,575,330*, 1986.

E. B. Hunt, J. Marin, and P. J. Stone. Experiments in induction. *Academic press*, 1966.

S. M. Hur, K. H. Choi, S. H. Lee, and P. K. Chang. Determination of fabricating orientation and packing in SLS process. *Journal of Materials Processing Technology*, 112:236–243, 2001.

F. Hutter, L. Xu, H. H. Hoos, and K. Leyton-Brown. Algorithm runtime prediction: Methods & evaluation. *IJCAI International Joint Conference on Artificial Intelligence*, 2015-Janua(Ijcai):4197–4201, 2015.

I. Ikonen and W. E. Biles. A genetic algorithm for optimal object packing in a selective laser sintering rapid prototyping machine. In *International conference on flexible automation and intelligent manufacturing*, pages 751–759, 1997.

I. Ikonen, W. W. E. Biles, A. Kumar, and R. K. R. Ragade. *Concept for a genetic algorithm for packing three dimensional objects of complex shape*. Citeseer, 1996.

I. Ikonen, W. Biles, and A. Kumar. A genetic algorithm for packing three-dimensional non-convex objects having cavities and holes. *Icga*, pages 591–598, 1997.

I. Ikonen, W. E. Biles, J. E. Lewis, A. Kumar, and R. K. Ragade. GARP : Genetic algorithm for part packing in a rapid prototyping machine. *spiedigitallibrary.org*, 3517(November 1998):54–62, 1998.

L. Ilany and Y. Gal. Algorithm selection in bilateral negotiation. *Autonomous Agents and Multi-Agent Systems*, 30(4):697–723, 2016.

M. Iori and S. Martello. Routing problems with loading constraints. *Top*, 18(1):4–27, 2010.

P. Jackson. *Introduction to expert systems*. Addison-Wesley Longman Publishing Co., Inc., 1998.

P. F. Jacobs. *Rapid prototyping & manufacturing: fundamentals of stereolithography*. Society of Manufacturing Engineers, 1992.

D. M. Jaeggi, G. T. Parks, T. Kipouros, and P. J. Clarkson. The development of a multi-objective Tabu Search algorithm for continuous optimisation problems. *European Journal of Operational Research*, 185(3):1192–1212, 2008.

N. Japkowicz. Learning from Imbalanced Data Sets: A Comparison of Various Strategies. In *AAAI workshop on learning from imbalanced data sets*, volume 68, pages 10–15. Menlo Park, CA, 2000.

X. Jia and R. Williams. A packing algorithm for particles of arbitrary shapes. *Powder Technology*, 120 (3):175–186, oct 2001.

X. Jia, M. Gan, R. A. Williams, and D. Rhodes. Validation of a digital packing algorithm in predicting powder packing densities. *Powder Technology*, 174(1-2):10–13, 2007.

G. H. John, R. Kohavi, and K. Pfleger. Irrelevant Features and the Subset Selection Problem. *Machine Learning Proceedings 1994*, pages 121–129, 1994.

H. John. Holland, Adaptation in natural and artificial systems, 1992.

J. Joseph, O. Sharif, A. Kumar, S. Gadkari, and A. Mohan. Using big data for machine learning analytics in manufacturing. *Tata Consultancy Services Limited White Paper*, 2014.

H. Kagermann, W.-D. Lukas, and W. Wahlster. Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution. *VDI nachrichten*, 13:11, 2011.

J. Kanda, A. Carvalho, E. Hruschka, and C. Soares. Using meta-learning to classify Traveling Salesman Problems. *Proceedings - 2010 11th Brazilian Symposium on Neural Networks, SBRN 2010*, pages 73–78, 2010.

J. Kanda, A. Carvalho, E. Hruschka, and C. Soares. Selection of algorithms to solve traveling salesman problems using meta-learning1. *International Journal of Hybrid Intelligent Systems*, 8(3):117–128, 2011.

K. Karabulut and M. Murat. A Hybrid Genetic Algorithm for Packing in 3D with Deepest Bottom Left with Fill Method. *Advis 2004, Lncs 3261*, pages 441–450, 2004.

J. King, P. Taylor, and S. Jupe. Autonomic Control Algorithm Selection In Decentralised Power Systems: A Voltage Control Case Study. *22nd International Conference on Electricity Distribution (CIRED)*, pages 10–13, 2013.

J. E. King, S. C. E. Jupe, and P. C. Taylor. Network State-Based Algorithm Selection for Power Flow Management Using Machine Learning. *IEEE Transactions on Power Systems*, 30(5):2657–2664, 2015.

S. Kirkpatrick, C. D. J. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, Vol.*, 220pp:671–680, 1983.

D. G. Kleinbaum and M. Klein. *Logistic regression: a self-learning text*. Springer Science & Business Media, 2010.

R. Kohavi and Others. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

A. Kolli, J. Cagan, and R. A. Rutenbar. Packing of generic, three-dimensional components based on multi-resolution modeling. In *Proceedings of the 22nd ASME Design Automation Conference (DAC-1479)*, pages 19–22, 1996.

S. B. Kotsiantis. Supervised Machine Learning: A Review of Classification Techniques. *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pages 3–24, 2007.

S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas. Data preprocessing for supervised learning. *International Journal of Computer Science*, 1(2):111–117, 2006.

L. Kotthoff. Algorithm selection for combinatorial search problems: A survey. *Ai Magazine*, 35(3): 48–60, 2014.

L. Kotthoff. Algorithm selection for combinatorial search problems: A survey. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10101, pages 149–190. Springer International Publishing, 2016.

L. Kotthoff, P. Kerschke, H. H. Hoos, and H. Trautmann. Improving the state of the art in inexact TSP solving using per-instance algorithm selection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8994:202–217, 2015.

C. Koulamas. A new constructive heuristic for the flowshop scheduling problem. *European Journal of Operational Research*, 105(1):66–71, 1998.

D. Krstajic, L. J. Buturovic, D. E. Leahy, and S. Thomas. Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of Cheminformatics*, 6(1):10, 2014.

P. T. Lan, S. Y. Chou, L. L. Chen, and D. Gemmill. Determining fabrication orientations for rapid prototyping with stereolithography apparatus. *CAD Computer Aided Design*, 29(1):53–62, 1997.

P. Langley, W. Iba, K. Thompson, et al. An analysis of bayesian classifiers. In *Aaai*, volume 90, pages 223–228, 1992.

E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, and D. B. Shmoys. Sequencing and scheduling: Algorithms and complexity. *Handbooks in operations research and management science*, 4:445–522, 1993.

Y. Lee, C.-T. Yang, and C.-S. Chien. A 3D ellipsoid-based model for packing of granular particles. *International Journal of Computer Applications in Technology*, 17(3):148, 2003.

Y. Lee, C. Fang, Y.-R. Tsou, L.-S. Lu, and C.-T. Yang. A packing algorithm for three-dimensional convex particles. *Granular Matter*, 11(5):307–315, 2009.

B. Leo, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and regression trees. *Wadsworth International Group*, 1984.

K. M. Leung. Naive bayesian classifier. *Polytechnic University Department of Computer Science/Finance and Risk Engineering*, 2007.

K. Leyton-Brown, E. Nudelman, and Y. Shoham. Learning the Empirical Hardness of Optimization Problems: The Case of Combinatorial Auctions. In *Principles and Practice of Constraint Programming - CP 2002*, pages 91–100. Springer, 2002.

K. Leyton-Brown, E. Nudelman, G. Andrew, J. McFadden, and Y. Shoham. A portfolio approach to algorithm selection. In *IJCAI*, volume 3, pages 1542–1543, 2003.

N. Li, J. Cha, Y. Lu, and G. Li. A simulated annealing algorithm based on parallel cluster for engineering layout design. *Complex Systems Concurrent Engineering: Collaboration, Technology Innovation and Sustainability*, 10:83–89, 2007.

N. Li, J. Cha, and Y. Lu. A parallel simulated annealing algorithm based on functional feature tree modeling for 3D engineering layout design. *Applied Soft Computing Journal*, 10(2):592–601, 2010.

Y. Li and J. Zhang. Multi-criteria GA-based Pareto optimization of building direction for rapid prototyping. *International Journal of Advanced Manufacturing Technology*, 69(5-8):1819–1831, 2013.

A. Lodi, S. Martello, and D. Vigo. TSpack: A Unified Tabu Search Code for Multi-Dimensional Bin Packing Problems. *Annals of Operations Research*, 131(1-4):203–213, oct 2004.

E. López-Camacho, H. Terashima-Marin, P. Ross, and G. Ochoa. A unified hyper-heuristic framework for solving bin packing problems. *Expert Systems with Applications*, 41(15):6876–6889, 2014.

M. Luckert and M. Schaefer-Kehnert. Using machine learning methods for evaluating the quality of technical documents, 2016.

E. Lutters, D. ten Dam, and T. Faneker. 3D Nesting of Complex Shapes. *Procedia CIRP*, 3:26–31, 2012.

S. Marsland. *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC, 2011.

V. J. Milenkovic and H. Schmidl. Optimization-based animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 37–46. ACM, 2001.

P. Min. Binvox, a 3d mesh voxelizer. *Available on: http://www. cs. princeton. edu/˜ min/binvox*, 2004.

H. D. Mittelmann. Decision Tree for Optimization Software, 2015.

L. C. Molina, L. Belanche, and A. Nebot. Feature selection algorithms: a survey and experimental evaluation. In *2002 IEEE International Conference on Data Mining*, pages 306 – 313. IEEE, 2002.

L. Monostori. AI and machine learning techniques for managing complexity, changes and uncertainities in manufacturing. *Engineering Applications of Artificial Intelligence*, 16(4):277–291, 2003.

B. Mueller. Additive Manufacturing Technologies - Rapid Prototyping to Direct Digital Manufacturing. *Assembly Automation*, 32(2):151–154, apr 2012.

M. A. Muñoz, M. Kirley, and S. K. Halgamuge. The Algorithm Selection Problem on the Continuous Optimization Domain. In *Computational Intelligence in Intelligent Data Analysis*, pages 75–89. Springer, 2013.

K. P. Murphy. Naive Bayes classifiers. *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series)*, 18:82 – 87, 2012.

S. K. Murthy. Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. *Data Mining and Knowledge Discovery*, 2(4):345–389, 1998.

M. N. Murty, P. J. Flynn, A. K. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: A Review. *dl.acm.org*, 2000.

A. Nyaluke, B. Nasser, H. R. Leep, and H. R. Parsaei. Rapid prototyping work space optimization. *Computers & industrial engineering*, 31(1-2):103–106, 1996.

E. O'Mahony, E. Hebrard, A. Holland, C. Nugent, and B. O'Sullivan. Using case-based reasoning in an algorithm portfolio for constraint solving. In *Irish Conference on Artificial Intelligence and Cognitive Science*, number 05, pages 210–216, 2008.

M. Padberg. Packing small boxes into a big box. *Mathematical Methods of Operations Research*, 52(1): 1–21, 2000.

R. Paul and S. Anand. Process energy analysis and optimization in selective laser sintering. *Journal of Manufacturing Systems*, 31(4):429–437, 2012.

R. Pearl and L. J. Reed. On the rate of growth of the population of the United States since 1790 and its mathematical representation. *Proceedings of the National Academy of Sciences*, 6(6):275–288, 1920.

P. Peduzzi, J. Concato, E. Kemper, T. H. J. of clinical ..., and U. 1996. A simulation study of the number of events per variable in logistic regression analysis. *Elsevier*, 1996.

I. Peko, A. Bašić, and A. Aljinović. Computer aided design, additive manufacturing and 3d product scanning. In *Inovativno pametno poduzeće*. Fakultet elektrotehnike, strojarstva i brodogradnje, 2018.

C. Peter-Orth. All solutions of the Soma cube puzzle. *Discrete Mathematics*, 57(1-2):105–121, 1985.

S. Petrovic and R. Qu. Case-based reasoning as a heuristic selector in a hyper-heuristic for course timetabling problems. *Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies*, pages 336–340, 2002.

J. Pihera and N. Musliu. Application of Machine Learning to Algorithm Selection for TSP. *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, pages 47–54, 2014.

N. Pillay. A study of evolutionary algorithm selection hyper-heuristics for the one-dimensional bin-packing problem. *South African Computer Journal*, 48(48):31–40, 2012.

S. Piramuthu. Evaluating feature selection methods for learning in data mining applications. *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, 5:294–301, 2004.

N. Pise and P. Kulkarni. Algorithm selection for classification problems. *2016 SAI Computing Conference (SAI)*, pages 203–211, 2016.

D. Pisinger. Heuristics for the container loading problem. *Elsevier*, pages 382–392, 2002.

D. Pisinger and J. Egeblad. Heuristic approaches for the two and three dimensional knapsack packing problems. Technical report, Technical report, 2006.

P. W. Poon and J. N. Carter. Genetic algorithm crossover operators for ordering applications. *Computers and Operations Research*, 22(1):135–147, 1995.

S. Psarra and T. Grajewski. Describing shape and shape complexity using local properties. In *Proceedings 3rd International Space Syntax Symposium*, pages 28–1. Citeseer, 2001.

A. Ravindran. *An Octree Based Genetic Algorithm for Three-dimensional Packing of Irregular Parts*. PhD thesis, University of Cincinnati, 2003.

S. Rémond and J. L. Gallias. A 3D semi-digital model for the placing of granular materials. *Powder Technology*, 148(1):56–59, 2004.

F. Rengier, A. Mehndiratta, H. Von Tengg-Kobligk, C. M. Zechmann, R. Unterhinninghofen, H. U. Kauczor, and F. L. Giesel. 3D printing based on imaging data: Review of medical applications. *International Journal of Computer Assisted Radiology and Surgery*, 5(4):335–341, jul 2010.

J. R. Rice. The algorithm selection problem. *Advances in Computers*, 15:65–117, 1975.

J. R. Rice. The Algorithm Selection Problem*. In *Advances in computers*, volume 15, pages 65–118. Elsevier, 1976.

A. Rosato, K. J. Strandburg, F. Prinz, and R. H. Swendsen. Why the Brazil nuts are on top: Size segregation of particulate matter by shaking. *Physical Review Letters*, 58(10):1038–1040, 1987.

P. Ross. Hyper-Heuristics. *Search Methodologies*, pages 529–556, 2005.

C. Rowe and R. Banham. *Mathematics of the Ideal Villa and Other Essays*, volume 31. TIMES NEWSPAPERS LTD PO BOX 479 VIRGINIA ST, LONDON, ENGLAND E1 9XU, 1977.

R. M. Royall. The effect of sample size on the meaning of significance tests. *The American Statistician*, 40(4):313–315, 1986.

M. Ruffo and R. Hague. Cost estimation for rapid manufacturing - simultaneous production of mixed components using laser sintering. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 221(11):1585–1591, 2007.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

M. Safe, J. Carballido, I. Ponzoni, and N. Brignole. On stopping criteria for genetic algorithms. In *Brazilian Symposium on Artificial Intelligence*, pages 405–413. Springer, 2004.

P. V. Sander, Z. J. Wood, S. J. Gortler, and H. Hoppe. Multi-chart geometry images. *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 146–155, 2003.

J. Schaeffer, P. Lu, D. Szafron, and R. Lake. A re-examination of brute-force search. In *Proceedings of the AAAI Fall Symposium on Games: Planning and Learning*, pages 51–58, 1993.

G. Scheithauer, Y. G. Stoyan, and T. Romanova. Mathematical modeling of interaction of primary geometric 3D objects. *Cybernetics and Systems Analysis*, 41(3):332–342, 2005.

R. W. Schmenner and M. L. Swink. On theory in operations management. *Journal of operations management*, 17(1):97–113, 1998.

K. Schwab. *The fourth industrial revolution*. Crown Business, 2017.

G. D. Scott and D. M. Kilgour. The density of random close packing of spheres. *Journal of Physics D: Applied Physics*, 2(6):863–866, 1969.

L. Sean. Essentials of metaheuristics. *Lulu: Department of Computer Science, George Mason University,*, 2009.

B. Selman and C. P. Gomes. Hill-climbing search. *Encyclopedia of Cognitive Science*, 2006.

S.-S. Shai and B.-D. Shai. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

L. A. Shalabi, Z. Shaaban, and B. Kasasbeh. Data Mining: A Preprocessing Engine. *Journal of Computer Science*, 2(9):735–739, 2006.

N. Sherwani. *Algorithms for VLSI physical design automation*. Springer, 2012.

S. N. Sivanandam and S. N. Deepa. *Introduction to genetic algorithms*. Springer Science & Business Media, 2007.

S. S. Skiena. The algorithm design manual, 1997. *Stony Brook, NY: Telos Pr*, 504, 1997.

D. Slansky. Integrating PLM with the Shop Floor: Validating As-Built to As-Designed. *ARC Advisory Group, http://www. arcweb. com*, 2008.

S. K. Smit and A. E. Eiben. Comparing parameter tuning methods for evolutionary algorithms. In *2009 IEEE congress on evolutionary computation*, pages 399–406. IEEE, 2009.

K. Smith-Miles and J. van Hemert. Discovering the suitability of optimisation algorithms by learning from evolved instances. *Annals of Mathematics and Artificial Intelligence*, 61(2):87–104, 2011.

K. A. Smith-Miles. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys (CSUR)*, 41(1):6, 2009.

A. Smola and S. V. N. Vishwanathan. *Introduction to machine learning*, volume 32. Cambridge University Press, 2008.

Y. Song, R. Turton, and F. Kayihan. Contact detection algorithms for DEM simulations of tablet-shaped particles. *Powder Technology*, 161(1):32–40, 2006.

K. Spies. *Die Zwischenformen beim Gesenkschmieden und ihre Herstellung durch Formwalzen*, volume 728. Springer-Verlag, 1957.

M. Srinivas and L. M. Patnaik. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 1994.

Y. G. Stoyan and L. D. Ponomarenko. Minkowski sum and hodograph of the dense placement vector function. *Reports of the SSR Academy of Science, SER. A*, 10, 1977.

Y. G. Stoyan and G. N. Yaskov. Mathematical methods for geometric design. In *Proceedings of International Conference PROLAMAT 82*, volume 82, pages 67–86, 1983.

Y. G. Stoyan, N. Gil, A. Pankratov, and G. Scheithauer. Packing non-convex polytopes into a parallelepiped. *MATH-NM-06-2004 (Preprint)*, 06(June), 2004.

Y. G. Stoyan, N. I. Gil, G. Scheithauer, A. Pankratov, and I. Magdalina. Packing of convex polytopes into a parallelepiped. *Optimization*, 54(2):215–235, 2005.

I. Stroud. *Boundary Representation Modelling Techniques*. Springer Science & Business Media, 2006.

R. Sutton and A. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

J. A. Swets. Measuring the accuracy of diagnostic systems. *Science*, 4857(240):1285–1293, 1998.

S. Szykman and J. Cagan. A simulated annealing-based approach to three-dimensional component packing. *... .Asmedigitalcollection.Asme.Org*, 117:308, 1995.

S. Szykman and J. Cagan. Constrained three-dimensional component layout using simulated annealing. *Journal of Mechanical Design*, 119(1):28–35, 1997.

R. Tavakkoli-Moghaddam, G. Moslehi, M. Vasei, and A. Azaron. Optimal scheduling for a single machine to minimize the sum of maximum earliness and tardiness considering idle insert. *Applied Mathematics and Computation*, 167(2):1430–1450, 2005.

H. Terashima-Marín, P. Ross, and M. Valenzuela-Rendón. Evolution of Constraint Satisfaction Strategies in Examination Timetabling. *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 635–642, 1999.

C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown. Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 847–855. ACM, 2012.

S. Tiwari, G. Fadel, and P. Fenyes. A Fast and Efficient Compact Packing Algorithm for SAE and ISO Luggage Packing Problems. *Journal of Computing and Information Science in Engineering*, 10(2): 021010, 2010.

G. D. Tourassi, E. D. Frederick, M. K. Markey, C. E. F. Jr, E. D. Frederick, and M. K. Markey. Application of the mutual information criterion for feature selection in computer-aided diagnosis Application of the mutual information criterion for feature selection in computer-aided diagnosis. *Wiley Online Library*, 2394(2001), 2012.

C. J. Tuck, R. J. M. Hague, M. Ruffo, M. Ransley, and P. Adams. Rapid manufacturing facilitated customization. *International Journal of Computer Integrated Manufacturing*, 21(3):245–258, 2008.

I. Y. Tumer, D. C. Thompson, K. L. Wood, and R. H. Crawford. Quantification of part surface quality: application to selective laser sintering. In *The 1996 World Automation Conference Proceedings*, pages 731–736, 1996.

A. Turner and A. Penn. Making isovists syntactic : isovist integration analysis. *2nd International Symposium on Space Syntax*, pages 1–9, 1999.

J. L. Udy, R. J. Balling, S. E. Benzley, and M. D. Landon. Computation of interferences between three-dimensional objects and the optimal packing problem. *Advances in Engineering Software (1978)*, 10 (1):8–14, 1988.

B. Valentan, T. Brajlih, I. Drstvensek, and J. Balic. Evaluation of shape complexity based on STL data. *Journal of achievements in materials and manufacturing engineering*, 17(1):293–296, 2006.

B. Valentan, T. Brajlih, I. Drstvensek, and J. Balic. Basic solutions on shape complexity evaluation of STL data. *Methods*, 26(1):73–80, 2008.

B. Valentan, T. Brajlih, I. Drstvenšek, and J. Balič. Development of a part-complexity evaluation model for application in additive fabrication technologies. 2011.

J. Vanek, J. A. G. Galicia, B. Benes, R. Měch, N. Carr, O. Stava, and G. S. Miller. PackMerger: A 3D print volume optimizer. *Computer Graphics Forum*, 33(6):322–332, sep 2014.

S. Varma and R. Simon. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7(1):91, 2006.

D. J. Velleman. Exponential vs. Factorial. *The American Mathematical Monthly*, 113(8):689–704, oct 2006.

M. Vetterli, M. Schmid, and K. Wegener. Comprehensive Investigation of Surface Characterization Methods for Laser Sintered Parts. *Proceedings of the Fraunhofer Direct Digital Manufacturing Conference*, 2014.

W. M. Visscher and M. Bolsterli. Random packing of equal and unequal spheres in two and three dimensions. *Nature*, 239(5374):504–507, 1972.

C. Voudouris, E. P. Tsang, and A. Alsheddy. Guided Local Search. *Springer*, pages 321–361, 2010.

M. Wagner, M. Lindauer, M. Misir, S. Nallaperuma, and F. Hutter. A case study of algorithm selection for the traveling thief problem. *Journal of Heuristics*, 24(3):295–320, 2018.

S. H. Walker and D. B. Duncan. Estimation of the probability of an event as a function of several independant variables. *Biometrika*, 54:167–179, 1967.

W. H. Wan. Introduction to STL format. *polytechnical university of Hong Kong.*, 1999.

G. Wäscher, H. Haußner, and H. Schumann. An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3):1109–1130, 2007.

G. J. Woeginger. Exact algorithms for NP-hard problems: A survey. In *Optimization*, volume 2570, pages 185–207. Springer, 2003.

D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

K. V. Wong and A. Hernandez. A Review of Additive Manufacturing. *ISRN Mechanical Engineering*, 2012:1–10, 2012.

S. Wu, M. Kay, R. King, A. Vila-Parrish, and D. Warsing. Multi-objective optimization of 3D packing problem in additive manufacturing. *IIE Annual Conference and Expo 2014*, i:1485–1494, 2014.

T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben. Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research*, 4(1):23–45, jan 2016.

L. Xu, F. Hutter, H. Hoos, and K. Leyton-brown. SATzilla: Portfolio-based Algorithm Selection for SAT. *J. Artif. Intell. Res.(JAIR)*, 32:565–606, 2008.

K. Yan and D. Zhang. Sensors and Actuators B : Chemical Feature selection and analysis on correlated gas sensor data with recursive feature elimination. *Elsevier*, 212:353–363, 2015.

B. Zadrozny and C. Elkan. Learning and making decisions when costs and probabilities are both unknown. *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '01*, pages 204–213, 2001.

C. Zhang and X. Zhou. Evaluation of the packing density of non-spherical particles using discrete element cluster algorithm. *Journal of Information and Computational Science*, 9(16):4969–4977, 2012.

X. Zhang, B. Zhou, Y. Zeng, and P. Gu. Model layout optimization for solid ground curing rapid prototyping processes. *Robotics and Computer-Integrated Manufacturing*, 18(1):41–51, 2002.