University of Southampton

Doctoral Thesis

A Cost-Driven Process Planning Framework for Selective Laser Melting

Author:
Valeriya Griffiths

Supervisory Team:
Prof. James P. Scanlan,
Dr. Murat H. Eres,
Dr. A. Martinez-Sykora,
Dr. Phani Chinchapatnam

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Engineering

in the

Computational Engineering and Design Group
Faculty of Engineering and the Environment

August 13, 2018

Declaration of Authorship

I, Valeriya Griffiths, declare that this thesis titled, "A Cost-Driven Process Planning Framework for Selective Laser Melting" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.
- An article has been revised and submitted to the European Journal of Operational Research. The paper covers the Iterative Tabu Search Procedure presented in Chapter 6.

Signed:

Date:

Abstract

Valeriya Griffiths

A Cost-Driven Process Planning Framework for Selective Laser Melting

Selective Laser Melting (SLM) is an additive manufacturing (AM) method, capable of producing end-use metal parts by selectively melting layers of powder with a moving laser. The process can create complex lightweight geometries in a single build, and offers an opportunity to improve product value through greater design freedom, part consolidation, reduced tooling and reduced supply chain complexity for a range of different industries. However, SLM also poses a number of material- and process-related challenges, which prevent wider adoption of this process for end-use part production. Due to high residual stresses characteristic of this process, build failure and poor part quality are still common issues. The cost of this process is also relatively high, while the need for post-processing leads to additional, and sometimes unexpected, costs. Effective process planning can mitigate these problems, however, this is not an easy task due to the large number of interdependent process parameters and their combined effects on the process cost, time and quality. Part build orientation is one such important parameter, which affects the surface roughness, build cost and time, as well as the need for support structure. Due to the anisotropic material properties produced by SLM and the effects of part geometry on the residual stress profile, there is also a complex relationship between build orientation, part quality and the risk of build failure. Moreover, the build orientations of multiple parts determine how many SLM machines are required to build all those parts. The placement of parts inside AM machines is known as a bin packing problem, and its link to build orientation has been acknowledged throughout the literature. Existing approaches have optimised build orientation and bin packing as two separate problems, limiting the optimality of the overall solution.

Thus, this thesis provides a cost-driven framework for optimising build orientation and bin packing of multiple parts in SLM, and provides two heuristics for solving these two problems simultaneously. The cost model used to drive the framework is developed based on a thorough literature review, and incorporates build cost and post-processing cost. Additionally, by considering SLM in the context of lean manufacturing two distinct production scenarios are identified, and referred to as Identical Batch Production (IBP) and Mixed Batch Production (MBP). Thus, each heuristic

is aimed at a specific production scenario. For IBP, a Tabu search procedure is developed by solving the build orientation problem along with a cutting-stock problem, as bins are assumed to be identical. To improve the efficiency of this search heuristic an area-based approximation strategy is proposed, which can reduce the solution time by as much as 50%, as indicated by the computational results.

The effect of build orientation on residual stresses is also considered, by extending the above Tabu search to address the multi-objective problem of cost and residual stresses in the context of IBP. Due to the high computational cost of this problem, two alternative approximation strategies are proposed, and coupled with the multi-objective Tabu search. The two strategies are demonstrated on a single test part of medium complexity; the results demonstrate the validity of this approach, as better, albeit less obvious trade-off solutions were found by the proposed Tabu search than by an experienced SLM operator.

Finally, an Iterative Tabu Search Procedure (ITSP) is developed for the MBP scenario. Because the solution space of this problem is much larger and more complex than the IBP, the ITSP consists of six distinct stages, where each stage is aimed at exploring a different area of the solution space. The procedure is benchmarked against commercial software, indicating an average cost improvement of 14.6% for 27 test instances.

Acknowledgements

I would like to express my sincerest gratitude to all my academic supervisors, each of whom have contributed significantly to my personal and professional development during my time at the University of Southampton. I would like to thank Jim Scanlan for believing in me and for giving me the freedom to learn and explore, but promptly stepping in whenever I needed help; Hakki Eres for his helpful guidance, encouragement and moral support; and Toni Martinez-Sykora whose good advice, support and friendship have kept me inspired throughout this project.

Special thanks must go to my industrial supervisors: Dr Stuart Jinks, Dr Phani Chinchapatnam and the late Dr Steve Wiseall who sadly passed away during the first year of my EngD, but without whom I would not have had this invaluable opportunity in the first place.

I also have numerous other people to thank in the Computation Engineering and Design Group at the University, as well as the Product Cost Engineering, Materials and Process Modelling, and Component Design System teams at Rolls-Royce plc for sharing their knowledge and expertise.

An honorary mention should also go to Julie Cheung and Chris Massey, for putting up with endless conversations about my EngD and taking on the role of mentor numerous times, despite it not being in their job description!

Last but not least, I am eternally grateful to my parents for their unwavering patience, love and support – I could not have finished this project without them.

Contents

List of Figures

List of Tables

List of Abbreviations

2DBPP	Two-dimensional bin packing problem
2DIBPP	Two-dimensional irregular bin packing problem
ABC	Activity based costing
AM	Additive manufacturing
C&P	Cutting and packing
CAD	Computer-aided design
CNC	Computer numerical control
DMLS	Direct metal laser sintering
EBM	Electron beam melting
FDM	Fused deposition modelling
NFP	No-fit polygon
RMSE	Root mean squared error
RP	Rapid prototyping
SLA	Stereolithography
SLM	Selective laser melting
SLS	Selective laser sintering
STL	Stereolithography file format

Chapter 1

Introduction

Additive Manufacturing (AM) is a recently emerged group of methods, which has been formally defined as the process of joining materials to make objects from three-dimensional (3D) model data, usually layer upon layer, as opposed to subtractive manufacturing fabrication methodologies (Wohlers, 2014). In contrast to traditional subtractive methods, AM offers greater freedom of geometry, less material waste and less tooling, thereby, AM is able to produce highly complex parts, which would have been infeasible or cost-prohibitive in the past, relatively cheaply. This also means that fewer process steps are required in AM and lead times can be reduced from months to weeks and from weeks to days (Williams et al., 2016). These benefits are now being recognised by global market leaders such as BAE Systems, GE Aviation and Rolls-Royce plc.

However, despite the large and rapidly growing body of research relating to this technology there are still a number of gaps that need to be addressed. Some of the key challenges are outlined in Section 1.3 and Section 1.4 of this chapter, while the corresponding research objectives and contributions are outlined in Section 1.3 and Section 1.4, respectively. To provide some context to the chosen problem, a brief history and outline of different AM methods is provided in Section 1.1, and an overview of AM process steps is provided in Section 1.2.

1.1 Emergence of Additive Manufacturing

The origins of AM can be traced back to Rapid Prototyping (RP). A process known as Stereolithography (SLA) was the first commercial RP technology, developed and patented by Hull (1986), who later founded 3D Systems. The method works by curing UV-reactive resins in a layer-by-layer fashion, as shown schematically in Figure 1.1. A layer of liquid resin is spread evenly along the build plate, an ultra-violet (UV) laser then selectively scans the resin, creating a solid slice of the geometry. Following this the build plate shifts down by a distance equivalent to the resin layer thickness and a new layer of liquid resin is spread over the top. The process repeats until the 3D geometry is finished.

The most significant part of this technology was the creation of the Stereolithography (STL) file format, which remains the standard geometry format for AM machines

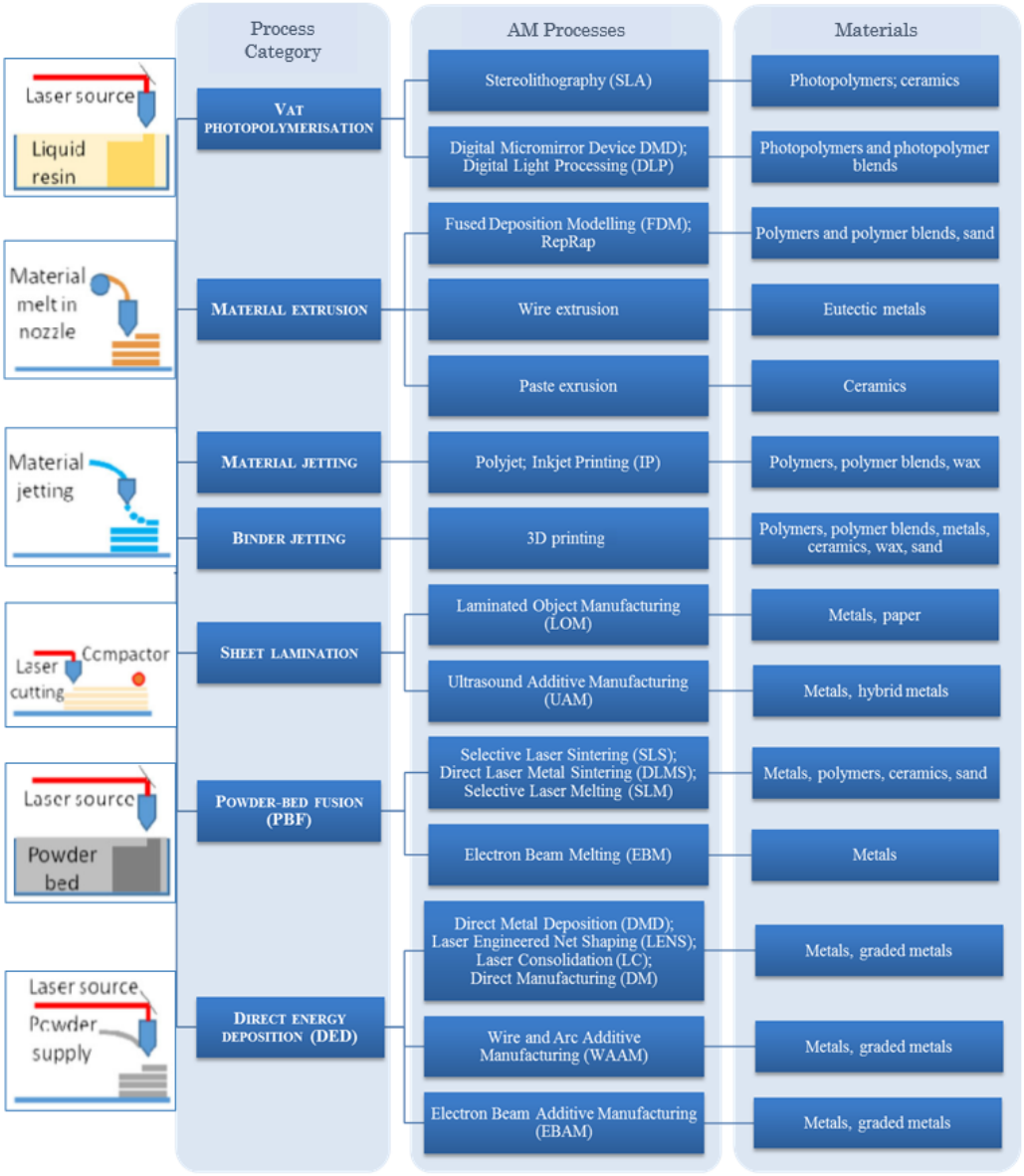


FIGURE 1.1: Summary and classification of AM processes.

today (Ma et al., 2001). The method continues to enjoy a wide range of uses such as medical models and moulds for investment-casting, as well as commercial uses such as jewellery. However, due to its restriction in material range and the development of other competing AM methods its unit sales have steadily declined in the last two decades (Wohlers, 2014). Selective Laser Sintering (SLS), a method which fuses polymer powders on a bed of powder using a carbon dioxide (CO_2) laser, was patented soon after, by researchers at the University of Texas at Austin (Deckard, 1989). This marked the birth of powder-bed fusion methods, which today have the potential to create functional end-use metal parts, as will be discussed in the following section.

Many other polymer-based methods and variants have been developed since, such as Fused Deposition Modelling (FDM), binder jetting and 3D printing. These methods enjoy a wide range of applications today such as prototyping and tooling,

consumer goods, medical implants, automotive and aircraft parts, and even multi-material parts (Goh et al., 2017).

1.1.1 Metal AM Processes

Following the success of SLS using polymer powders, attempts to apply the process to metal powders quickly followed. Early machines could not provide sufficient power or building conditions to produce high quality metal parts, and thus indirect methods were initially developed which rely on mixing (or coating) of metal powders with a polymer (Agarwala et al., 1995). The polymer binder melts much more readily and spreads in between the metal powder particles, acting as a binder. A post-processing sintering treatment removes the binder and the remaining highly porous structure can be infiltrated with a molten material, such as bronze.

In 1995 a company called EOS released the first commercial Direct Metal Laser Sintering (DMLS) machine, able to sinter metallic mixtures of steel-based and bronze-based powders, with a CO_2 laser (Khaing et al., 2001). While this was still a sintering process, which produced fragile and porous parts (30-45% porosity), the M250 was a crucial technological leap and the forerunner of todays Selective Laser Melting (SLM) machines.

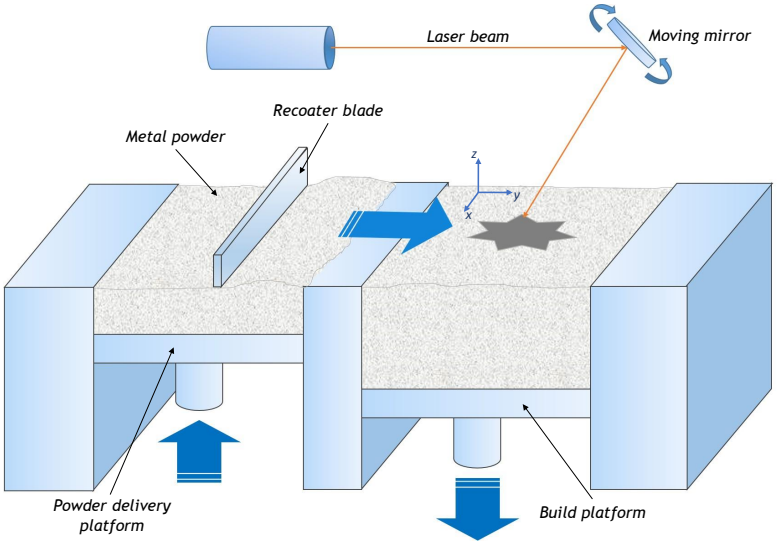


FIGURE 1.2: A schematic description of the Selective Laser Melting (SLM) process, where the build direction is defined by the z-axis.

Like DMLS, SLM is the process of selectively fusing powder by a moving laser in a layer-by-layer fashion, as shown in Figure 1.2, but this fusion is achieved through fully melting the powder (as opposed to sintering) resulting in densities of over 99% (Sames et al., 2016). This has become possible due to better alloy powder mixtures, better controlled build chambers filled with inert gas and improved laser technology, by replacing CO_2 lasers with Nd:YAG[1] and fiber lasers, which are much more readily absorbed by metals (Vandenbroucke and Kruth, 2007).

[1] neodymium-doped yttrium aluminum garnet.

In 2000 another powder-bed process, known as Electron Beam Melting (EBM), was developed in Sweden and licensed by Arcam AB, which remains the main EBM machine supplier today (Sames et al., 2016). The process is similar to SLM in many respects, however, instead of using a laser, it relies on an electron beam as its energy source, and thus, requires vacuum conditions inside the build chamber. The process is faster than SLM, due to the larger power input and larger spot diameter of the electron beam; however, this also leads to poorer surface quality and feature resolution. Furthermore, in EBM each layer of powder must be pre-scanned to a sintered state before it is melted, to prevent powder particles from being repelled and dispersed by the electron beam. The build-up of charge in the part requires helium gas to be added into the chamber while the higher process temperature requires significantly longer cooling times than SLM (Gong et al., 2014).

Another key metal AM process is Direct Energy Deposition (DED) which traces its history to welding. Rather than using mirrors (as in SLM) or magnets (as in EBM) to control the beam, DED effectively uses a welding torch mounted on a robotic arm and incorporates a wire or powder feed nozzle, which deposits material directly into the path of the welding torch. Thus, DED does not require a powder-bed and offers greater flexibility in terms of build size and platform; new parts can be built on a substrate or as additional features on an existing part. Although DED is faster and more flexible than the powder-bed processes, it offers the worst surface quality and dimensional accuracy, making it more suitable for repairs and production of large parts of low-to-medium complexity (Vayre et al., 2012).

	SLM	EBM	DED
Power (kW)	0.1 - 1	1 - 3	0.5 - 4
Building speed (cm^3/h)	5 - 70	55 - 80	2500
Surface roughness *Ra* (µm)	9 - 16	25 - 35	200
Layer thickness (µm)	20 - 100	50 - 200	200 - 300 (powder) 3000 (wire)
Dimensional accuracy (µm)	75 - 100	100 - 200	200 - 380 (powder) 16000 (wire)
Maximum build dimensions (mm)	800 x 400 x 500 (*x*-*y*-*z*)	350 x 380 (∅-*H*)	4000 x 1981 x 1778 (*x*-*y*-*z*)
Material delivery system	Powder feeder and re-coater	Powder feeder and re-coater	Nozzle-fed wire or powder
Referenced manufacturers	EOS, Concept Laser	Arcam	Sciaky, Optomec

TABLE 1.1: Summary and comparison of metal AM processes.

Table 1.1 summarises the key properties and trade-offs of the three main metal AM methods; DED offers faster build times and greater flexibility in terms of part size and material, but is limited in the relatively accuracy and geometry it can achieve; SLM offers the highest resolution and build accuracy out of the three but is slightly slower; both processes suffer from high residual stresses due to large thermal gradients during the build cycle. EBM falls in between the two in terms of building speed and accuracy, and does not suffer from residual stresses as significantly, but offers a limited choice

of materials and incurs high running costs due to the vacuum and heating conditions, as well as material waste.

1.2 General Process Steps

All AM and RP processes follow the same basic steps, consisting of file preparation, part building and post-processing. The process begins with creating a 3D computer-aided design (CAD) file of the part geometry. The majority of designs are now created using CAD software packages, of which there are many. Most CAD systems are also capable of converting the geometry into STL file format, which defines the external closed surfaces of the CAD geometry, using a mesh of tiny triangles (no other information about the solid geometry or features is stored). Before being transferred to the machine, the STL file must be positioned and oriented in the build environment; the rotational position of the part geometry with respect to the z-axis (as shown in Figure 1.2) is called the build orientation. Selecting a suitable build orientation is an important decision as it has a direct effect on the production cost, time and quality, as well as the required post-processing. Additionally, for processes such as SLM and FDM, sacrificial support structures must be generated to prevent part distortion and failure during the build. As shown in Figure 1.3, the shape and volume of support structure is also closely related to the build orientation.

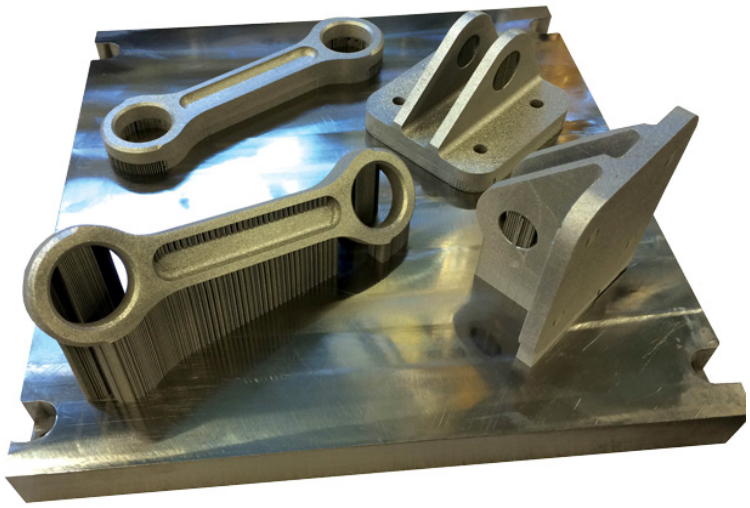


FIGURE 1.3: Example of different build orientations and resulting support structure for two different part geometries built via SLM (taken from Waterman (2017)).

Next, the oriented part geometry and support structures must be sliced into horizontal cross-sectional layers, the thickness of which is predetermined by the process, typically within a range of $20 - 100$ µm for most SLM machines (Vayre et al., 2012), and selected by the user to meet the optimal input energy requirements for the given material (Kempen et al., 2015). Commercial software packages such as Magics (produced by Materialise) and Nefabb (produced by Autodesk) can be used to slice the file, generating the sliced data in a new file format which stores the thickness and

surface area of each layer. The sliced file format is dependent on the system manufacturer, for instance, 3D Systems use System Layer Interface (SLI), while Common Layer Interface (CLI) is compatible with EOS machines. The next stage involves determining a scan strategy and generating a tool-path for each layer, which is often done by the same software as the slicing. The standard output format for this is written in the computer numerical control (CNC) language called G-code which contains ordered vectors of the tool-path. Some important decisions have to be made at this point such as selecting the scan pattern and hatch spacing (spacing between the vectors). Finally, the G-code file and machine build parameters are uploaded to the AM machine and the part is built. Choosing the correct build parameter values is vital for building a part of sufficient quality and accuracy, as well as minimising production time and cost. During the build, commercial powder-bed machines will typically provide the operator with in-process monitoring data, such as the melt-pool temperature signature and melt-pool dimensions, using a high frame-rate camera and photodiode, respectively (Berumen et al., 2010). This allows for post-build error examination and fine-tuning of the process parameters.

After the build chamber has cooled, the parts are extracted from the machine and cleaned of any loose material. Parts produced by metal powder-bed processes are often cut away from the substrate using wire Electric Discharge Machining (EDM), though other tools such as a hacksaw can be used, depending on part specifications, geometry and ease of removal (Atzeni and Salmi, 2012). Current metal parts produced by SLM typically require post heat treatments to alleviate residual stresses and improve part ductility; Hot Isostatic Pressing (HIP) may also be used to reduce porosity and seal micro-cracks (Vrancken et al., 2012). The final stage includes support structure removal and surface finishing.

1.3 Project Background and Motivation

1.3.1 Design Benefits of AM

In current aerospace manufacturing, Buy-to-Fly ratios as high as $20:1$ (i.e. with 95% of material turned into scrap) are not uncommon when using subtractive methods, where the weight of a part is inversely proportional to the volume of scrap. This poses a particular challenge for the aerospace industry, which relies on high-value alloys, like nickel and titanium, and is under constant pressure to produce lighter and more efficient aircraft with lower emissions (Huang et al., 2016). In contrast, AM methods can yield Buy-to-Fly ratios of $2:1$ or less (English et al., 2010; Horn and Harrysson, 2012), as these methods only add material where it is needed rather than removing unwanted material from a billet.

Furthermore, the additive "tool-less" nature of AM processes eliminates many of the conventional manufacturing constraints of both subtractive methods such as

machining (e.g. fixtures, tool access) and formative ones such as casting (e.g. uniform wall thickness, split line location) (Yang and Zhao, 2015). Complex lightweight geometries can be created using topology optimisation, which is a computational method of distributing material economically within a predefined design space (Watts and Hague, 2006).

FIGURE 1.4: Concept heat exchanger design build in one piece via SLM (taken from 3T RPD (2018))

Part consolidation can also reduce labour and tooling costs, assembly time, as well as improve the overall part quality, as demonstrated by General Electric (Kellner, 2014) who used AM to create a single-part fuel injector nozzle, which previously consisted of 18 components. The new design reduced the number of welds from 25 to just five, and is two thirds lighter and five times more durable than the original assembly. Other examples of how AM can be used to improve design functionality include complex heat exchange systems (as shown in Figure 1.4), parts with conformal cooling channels and integrated oil delivery systems, all built in a single process (Horn and Harrysson, 2012).

1.3.2 Supply Chain Benefits of AM

The above design benefits also lead to two key supply chain benefits. Firstly, without the need for tooling, on demand production of customised parts is no more costly than mass production, and perhaps even cheaper, via AM (Garrett, 2014). This in turn enables AM production to be driven by demand, rather than the need to justify tooling costs, reducing the costs and risks associated with large inventories and overproduction. Secondly, due to its geometric flexibility AM can produce a wide range of geometries simultaneously in a single build; this is referred to as Mixed Batch Production (MBP) throughout this thesis, and can be used to maximise machine utilisation and produce small quantities of parts efficiently. According to Ruffo et al. (2006) MBP is already common practice employed by RP bureaus and specialised AM service providers, who often receive small orders for parts of various sizes, functions and complexities from a number of different customers.

Both the design and the supply chain benefits make AM especially promising for the aerospace industry, which is characterised by relatively low production volumes, high part variety and highly complex systems (Webb et al., 2015).

1.3.3 AM Process Planning

Process planning is an important and currently time consuming step in SLM. Suitable laser parameters and thermal conditions are critical for robust and high quality production; meanwhile alloy properties, powder quality, part geometry and part positioning in the SLM build chamber all have a significant and combined effect on build quality, cost and time. Commercial software like Magics and Netfabb, as well as a number of academic models offer process planning support, however, these tools are limited to one process parameter at a time and typically require the user to provide suitable weights or an order of priority for a number of competing objectives.

Thus, existing tools can only provide limited decision support, instead of fully automated process planning. This means that process parameters are often selected manually by experienced users, making the process planning step significantly more time consuming and labour intensive, while also making the quality and productivity of SLM highly dependent on the level of skill and experience of individual designers and manufacturing engineers. Poorly informed decisions can lead to time consuming and costly trial-and-error during the development of a product; Sames et al. (2016) termed this the "rule of 4", meaning it can take up to four iterations to achieve a successful part and reliable build process. Furthermore, these decisions affect not only the SLM process but also the subsequent post-processing stage, which can have a significant effect on the final unit cost and lead time of a part and thus must be considered as part of process planning (Mellor et al., 2014). The risks of build failure and significant post-processing raise the current barrier to the adoption of SLM and other AM processes, especially in industries such as aerospace, where quality control and certification requirements are particularly stringent.

One of the key process parameters, as mentioned in Section 1.2, is the part build orientation, which affects many aspects of SLM (and other AM methods), including machine run time and cost; part surface roughness, accuracy and required post processing; material anisotropy, support structures and residual stresses in the parts; and finally batch size, which is a driving factor of machine productivity and overall manufacturing cost. This clearly presents a large number of trade-offs making it difficult, not only to locate optimal build orientations, but to define a measure of "optimality" to evaluate these build orientations in the first place, especially when dealing with MBP.

Furthermore, the high residual stresses in SLM are currently one of the biggest challenges of this process, adding significant build failure risk and limiting its suitability for end-use structural parts. The need for additional sacrificial support structures and their subsequent removal results in additional material waste, build time and time-consuming manual finishing operations, while the need for stress relief heat

treatments further increases the lead time and cost. Although numerous simulation models have been proposed (Roberts, 2012; Megahed et al., 2016; Peng et al., 2017) for predicting residual stresses and distortion during SLM, they share two major limitations: firstly, simulations of realistic parts are extremely time consuming and computationally expensive; secondly, these simulations can only provide a simplified approximation of the real process, as numerous assumptions must be made to make them tractable. For this reason, applying process modelling and optimisation heuristics to minimising residual stresses is a particularly difficult task, and existing research relating to this problem is limited (Megahed et al., 2016).

1.3.4 SLM in Rolls-Royce

The potential benefits of metal AM processes have been recognised and already exploited to some degree by Rolls-Royce. For instance, the lead-time of the XWB-97 front bearing housing (FBH) was reduced by more than 30%, by manufacturing the FBH vanes via EBM (Stuart, 2015); while, DED is a well-established method of Blisk repair used by a number of aero-engine manufacturers (Hellemann et al., 2003; Ford and Despeisse, 2016). However, based on the trade-offs discussed in Section 1.1.1, the focus of this thesis is on the SLM process, due to its particular suitability for the production of intricate medium-sized aero-engine parts.

To use SLM effectively and to be able to assess its suitability against other methods of manufacture, Rolls-Royce must be able to accurately predict and minimise the cost of the entire production process. Rolls-Royce has already developed a holistic cost model of SLM, but currently has no tool to optimise the process or minimise the cost; as discussed above, this leads to a lot of manual effort going into the design and process planning steps, which lack a clear and consistent procedure.

1.4 Challenges

Based on the above discussion, the challenges addressed in this thesis are summarised in the following four points:

Automated process planning in a realistic production context: Despite the existence of commercial software tools such as Magics and Netfabb, as well as a number of academic methodologies, fully automated process planning is yet to be achieved. Additionally, the role and benefits of AM methods in the supply chain of adopting companies such as Rolls-Royce are currently not fully appreciated or understood. Thus, to encourage mainstream adoption of AM, process planning must be not only robust and fully automated, but must also address different, realistic supply chain and production scenarios.

Holistic objective function for build orientation optimisation: Part build orientation affects a number of important factors such as build cost, build time, support structure and build surface roughness. Consequently, existing literature offers numerous competing objective functions, but lacks agreement on which is the most suitable one, while existing multi-objective functions require the user to provide appropriate weight coefficients or target values, which is a difficult task in practice.

Consideration of residual stresses and post-processing: Other important challenges in SLM include high residual stresses and poor surface finish, both of which have an adverse effect on the part fatigue life, and thus, necessitate additional post-processing steps following SLM, which in turn increase lead time and cost. Furthermore, high residual stresses can lead to build failure, microcracking and deterioration of build accuracy during the SLM build process. As suggested by a number of works (Mercelis and Kruth, 2006; Protasov et al., 2016; Peng et al., 2017) there is a non-trivial relationship between the part geometry, build orientation and residual stress profile; however, this relationship is yet to be fully explored and exploited.

Simultaneous parameter optimisation for more optimal solutions: While build orientation and bin packing are two important and strongly interdependent parameters in AM, there is currently no approach which addresses the two parameters simultaneously. Existing works address the two parameters as two separate problems, typically with limited movements in the bin packing problem, thus, limiting the scope of the optimisation search and missing out many potentially good solutions.

1.5 Research Objectives

To address the four challenges outlined above, this thesis proposes an automated process planning framework to minimise the overall production cost, which includes SLM build cost and post-processing cost. In this way, the framework should unite the competing decision criteria for build orientation and bin packing into a single objective function, making the problem simpler and more intuitive to solve, with only a few inputs and no supervision required from the user. The framework should also include the effects of residual stresses and should be applied in the context of realistic supply chain scenarios. Thus, the research entails meeting the following five objectives:

1. Develop an automated cost-driven process planning framework, which simultaneously optimises build orientation and bin packing for SLM.

2. Develop suitable optimisation models that apply the framework to the IBP and MBP scenarios, respectively.

3. For IBP, the optimisation model should incorporate both build cost and post-processing cost.

4. For MBP, the optimisation model should consider the bin assignment of parts, as well as build orientations and bin packing.

5. Finally, develop a model which uses finite element analysis (FEA) of SLM to optimise both the unit cost and residual stresses in the context of IBP.

1.6 Contributions

The contributions made by this thesis are summarised as follows:

1. A process planning optimisation framework has been developed, which considers production cost as a function of part build orientation and 2D bin packing inside identical SLM machines. This approach unites a number of competing objectives, such as surface roughness, support structure and projected area of the part, into a single objective, and removes the need for guessing appropriate objective weights by assigning realistic cost coefficients to the different objective criteria.

2. A simple approach has been proposed, which enables target surface roughness to be provided as a design requirement by the user and treated as a hard constraint; in this way, the build surface roughness of SLM parts can be considered as a post-processing cost, rather than a separate objective. The benefits of this are two-fold: firstly, this approach supports contribution 1 above, in unifying different build orientation criteria into a single objective function; secondly, it includes post-processing costs into the process planning stage, which reduces the risk of hidden or unforeseen additional costs at the later stages of the manufacturing process.

3. In order to effectively deal with different demand patterns, the framework has been applied to two different production scenarios, namely IBP and MBP, as discussed in Section 3.1.1. The two production scenarios are addressed in Chapter 4 and Chapter 6, respectively. Since the framework has been developed in the context of the aerospace industry, the presence of safety-critical parts has been considered and a consequent build orientation constraint has been proposed, which is discussed in Section 3.1.2.

4. For the IBP scenario, the cost model includes both build cost and post-processing cost as a function of build orientation and 2D bin packing. Similar works by Alexander et al. (1998) and Byun and Lee (2006) propose holistic cost models to compare different build orientations in AM processes such as FDM and SLA; however, these models only consider single prototype production and do not include batch packing. No existing works have addressed the issue of modelling total production cost as a function of build orientation and 2D bin packing in the context of large-scale, end-use parts produced via SLM. Moreover, Chapter

5 extends the framework, by considering the multi-objective problem of minimising total cost and residual stresses in SLM. Two methodologies have been explored, and the resulting Pareto sets have been compared for a single geometry of medium complexity. The results indicate that the best build orientation may not necessarily be the most obvious one, even for an experienced SLM user, which demonstrates the usefulness of solving this problem.

5. For the MBP scenario, a novel Iterative Tabu Search Procedure (ITSP) has been proposed to solve the combined problem of build orientation and 2D bin packing of mixed geometries in a coupled way. To reduce the complexity of the problem the ITSP uses build cost to drive the search, without including post-processing, which is highly dependent on target surface roughness and surface finishing method chosen for each part. The ITSP works by iterating between the two problems, aiming to improve the bin packing solution, while also incrementally improving the build orientations of individual parts in each bin. This coupled approach is benchmarked against a de-coupled approach, equivalent to a single iteration of the ITSP (i.e. where the build orientation and bin packing problems are solved separately and only once) using the commercial software Magics. The results show a significant improvement in build cost (14.6% average), supporting the above hypothesis that optimising build orientation and bin packing in a coupled manner is more advantageous than solving the two problems separately. This work has been submitted to the European Journal of Operational Research.

1.7 Thesis Structure

This thesis consists of seven chapters, the contents of which are summarised as follows:

- Chapter 2 provides a literature review of key process challenges and costs in SLM, as well as the main process planning parameters affecting them.
- Chapter 3 covers the methodology of the proposed process planning framework, and provides further context and practical applications of the IBP and MBP scenarios.
- Chapter 4 describes the Tabu search (TS) procedure developed to optimise the unit cost of IBP.
- Chapter 5 extends the TS strategy of Chapter 4 and provides a methodology for finding trade-off solutions for unit cost and thermal residual stresses.
- Chapter 6 describes the Iterative Tabu Search Procedure (ITSP) developed to minimise the build cost of MBP.
- Chapter 7 summarises the research findings and drawn conclusions, and suggests potential directions for future research.

Chapter 2

Literature Review

Careful process planning is a crucial step in the SLM process as it affects part surface quality, dimensional accuracy, material microstructure and mechanical properties, as well as production cost and lead time. Selecting optimal parameters can produce cost-effective and high quality functional parts, while a poorly planned process is likely to result in build failure and waste (Berumen et al., 2010).

Thus, this chapter consists of three parts. The first provides an overview of the key characteristics and challenges related to part quality and process parameters. The second provides an overview of key process parameters relevant to this thesis, providing a critical literature review regarding modelling and optimisation of build orientation, support structure and batch packing. The third and final part looks at the key drivers and the overall nature of AM cost. Build time is covered in this section, since it is both a driver of cost and an important objective in its own right, and existing modelling approaches and current challenges with respect to manufacturing time and cost are discussed.

2.1 SLM Process Characteristics and Challenges

2.1.1 Material Porosity

Material porosity is a key challenge in SLM, as it has a detrimental effect on fracture toughness, fatigue performance and overall mechanical strength of the produced material. Figure 2.1 shows two types of pores commonly found in SLM materials – spherical and irregular. Spherical pores typically form due to gas bubbles getting trapped in a pool of locally melted powder, which cools before the gas has a chance to escape; these gas bubbles may simply be inert gas from the build chamber atmosphere or gas bubbles released by the porous powder particles as they turn into molten metal. Irregularly-shaped pores usually occur as a result of process-specific phenomena, namely balling and key-holing (Townsend et al., 2016).

Balling and key-holing are two persistent challenges in laser-related processes and are commonly observed in welding (Madison and Aagesen, 2012). Each track of newly melted powder is subjected to very high thermal stresses, which create local gradients in the surface tension of the molten metal (known as Marangoni effect) and cause it to break up into balls of liquid; this is known as balling (Strano et al., 2013b).

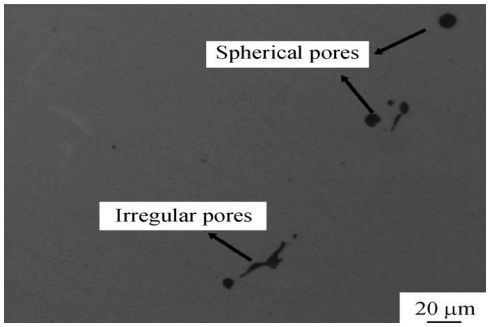


FIGURE 2.1: Electron microscope image showing different types of pores. Image taken from Song et al. (2015).

On the other hand, key-holing occurs when sparks from the laser beam hit the melt-pool, ejecting droplets of liquid metal onto the surrounding surface (this is known as spatter). Key-holing can also occur as a result of metal vapour which may exert a pressure on the melt-pool, high enough to expel the liquid from the melt-pool cavity (Shifeng et al., 2014). Collapse of this cavity can leave voids in the wake of the laser beam (Madison and Aagesen, 2012; Kamath et al., 2014), and is often associated with the presence of un-melted powder particles in parts built via SLM (Olakanmi et al., 2015).

SLM porosity is still an ongoing research problem, since it is highly sensitive to alloy properties, powder quality and process parameters, and must be considered alongside other important factors such as microstructure. However, with the right combination of laser and scan parameters (please refer to Figure 2.8), densities of over 99% can be achieved for many industrial alloys, including those of aluminium (Rao et al., 2016; Tang and Pistorius, 2017), nickel (Carter et al., 2012; Dub et al., 2016) and titanium (Ali et al., 2017; Alfaify et al., 2018). The density can be further improved through a post-SLM HIP treatment, however, it should be noted that HIP also reduces material strength and hardness through coarsened microstructure; adds significant cost and time to the production process (Lewandowski and Seifi, 2016); and may not eliminate the presence of pores completely, particularly surface pores (Leuders et al., 2013).

2.1.2 Thermal Residual Stresses and Cracking

Arguably, an even more critical limitation of SLM is the occurrence of high residual stresses, due to their complex interaction with process parameters and microstructure. Even moderate residual tensile stresses can result in local micro-cracks and a significantly reduced fatigue life and toughness of the part (Leuders et al., 2013), while severe stresses can lead to complete build failure during SLM, due to severe cracking and distortion of the part (Yadroitsava et al., 2015).

The large input of thermal energy, combined with the relatively low conductivity of the cooling material underneath, causes a steep thermal gradient and heat accumulation in the top surface of the part. Due to this accumulated heat, the top of

the part tries to expand, while being restricted and subsequently compressed by the underlying solid material, resulting in a tensile stress field in the top surface as it cools. This process is shown in Figure 2.2. Additionally, each time a new melt-track is created by the laser, the molten metal shrinks as it cools, causing a more localised tensile stress field for each track (Mercelis and Kruth, 2006). Consequently, the residual stress profile is determined by a complex combination of thermal process settings, such as build chamber temperature and laser beam parameters; material properties, such as thermal expansion coefficient and elastic modulus; and the built geometry, which includes support structure.

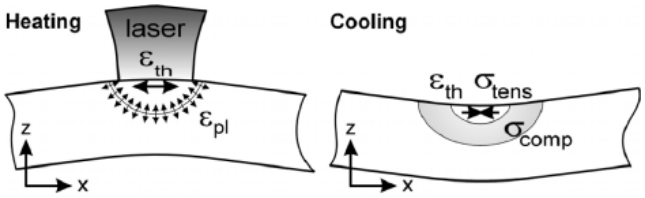


FIGURE 2.2: Residual stress fields caused by the thermal gradient. Image taken from Mercelis and Kruth (2006).

Preheating the powder bed can alleviate a significant portion of the residual stresses by reducing the thermal gradient. However, it also incurs an additional energy cost and can affect the properties of different materials in a number of ways, for instance, reducing the strength of tool steel and aluminium alloy (Kempen et al., 2014; Buchbinder et al., 2014), while reducing ductility in titanium alloy due to increased oxidation (Vrancken et al., 2016) and an undesirable microstructure (Ali et al., 2017).

The laser path can further influence residual stresses – the "chequerboard" or "island" scan strategy, shown in Figure 2.3, is a popular approach since it reduces the build-up of heat by shortening the scan vectors and distributing the heat input more evenly across each layer. Rotating the scan pattern for each layer, usually by $60^o - 90^o$, has also been shown to improve the overall distribution of heat, thus, further reducing the accumulation of residual stresses (Gusarov et al., 2011; Carter et al., 2014; Ali et al., 2018). Other proposed strategies include the use of a pulsed laser (Mumtaz and Hopkinson, 2010), layer re-melting (Yasa and Kruth, 2011; Ali et al., 2018) and laser shock peening (Kalentics et al., 2017).

Heat treatment after SLM can effectively alleviate residual stresses, while HIP can also seal the majority of micro-cracks. However, as already mentioned in Section 2.1.1, both methods add cost and time, and may not fully eliminate all stresses, while in some cases the heat-induced stress relaxation can actually produce new cracks and severely damage the part (Patterson et al., 2017). As with porosities, HIP is only able to seal internal cracks that do not reach the part's surface (Carter et al., 2012). Furthermore, the part must also be fixed to the build plate with support structure to prevent it from distorting before it can be post-processed, as well as to help dissipate

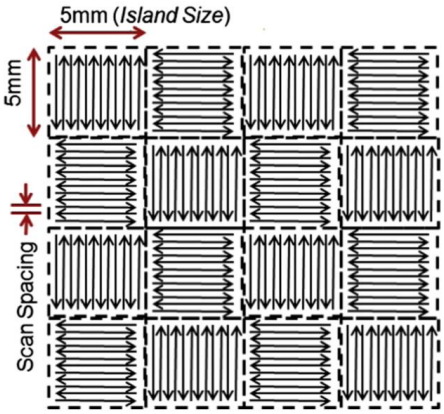


FIGURE 2.3: Schematic example of the island scan strategy. Image taken from Carter et al. (2014).

heat away from the part. As discussed in Section 2.2.1, support structure is a source of material waste and additional post-processing.

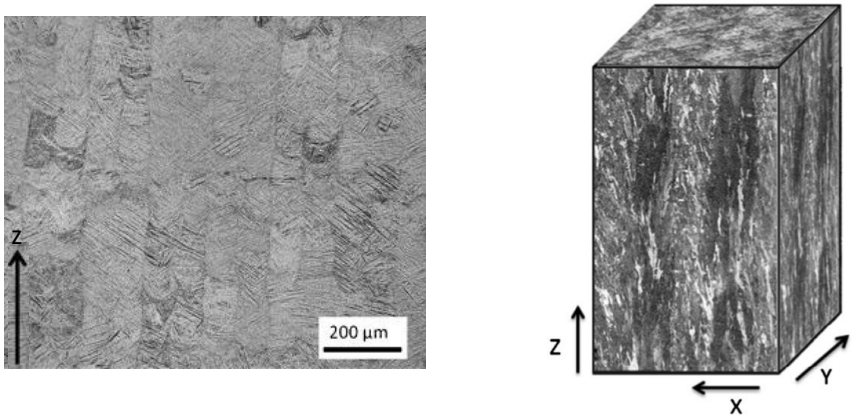
An interesting material-based approach has been proposed for building parts without the need for supports, known as anchorless SLM (ASLM) (Mumtaz et al., 2011; Vora et al., 2015, 2017). This approach blends powders in specific proportions, so that when these powders are melted together they form a eutectic alloy, which has a significantly lower melting temperature than its original constituents. By keeping the powder bed temperature raised above this new melting point the part can be maintained in a semi-solid state, thus, avoiding the onset of severe thermal gradients, residual stresses and resulting distortion. Despite its potential, this method has only been applied to a few alloys such as bismuth-zinc (Mumtaz et al., 2011) and aluminium (Vora et al., 2015, 2017), and therefore, requires further research to expand its scope. Furthermore, the ASLM approach does not account for other problems such as the effects of gravity on large overhangs.

Finally, the combined geometry of the part and the support structure clearly has a strong, albeit complicated relationship with the residual stress profile (Mercelis and Kruth, 2006; Zaeh and Branner, 2010; Keller et al., 2013). By extension, the influence of build orientation on residual stresses is twofold, since it determines not only the geometry of layers and height of the part, but also the location of support structures, as discussed in Section 2.2.1. The only known work to exploit this relationship is that of Peng et al. (2017), who optimise the build orientation for minimum thermal distortion, based on a simulated residual stress profile.

2.1.3 Microstructure and Mechanical Properties

Figure 2.4 shows the typical alloy microstructure produced by SLM, with largely equiaxial grain structure in the x-y plane and strongly elongated columnar grains in the z direction. This grain structure can be explained by the accumulation of heat in the top surface of the part, and the relatively low conductivity of the surrounding

unfused powder, causing a strong thermal gradient in the z direction, thus, encouraging columnar grain growth as the molten metal cools and as heat diffuses through the layers of material down into the substrate. Columnar grain structure has been observed in most commonly studied SLM metals, including alloys of titanium (Chlebus et al., 2011; Murr et al., 2012), nickel (Amato et al., 2012; Harrison et al., 2015), aluminium (Olakanmi et al., 2015) and stainless steel (Niendorf et al., 2013; Shifeng et al., 2014). In general, the fast cooling rates of SLM cause a fine martensitic crystal structure with subsequently high tensile strength and hardness, but poor ductility, when compared to wrought and cast alloys (Gorsse et al., 2017; Kok et al., 2018).

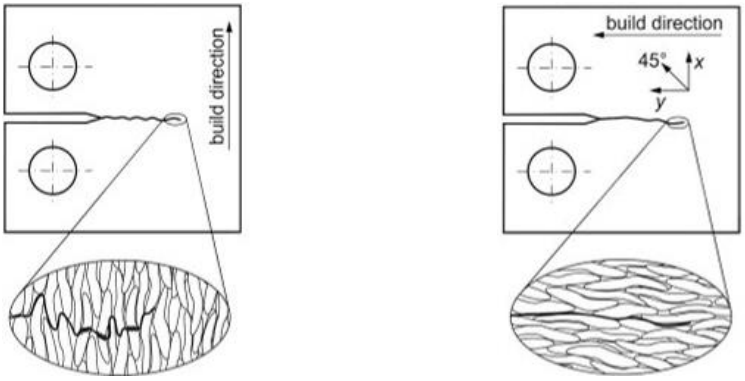


(A) Ti-6Al-4V columnar microstructure. (B) CM247LC microstructure in 3D.

FIGURE 2.4: Anisotropic microstructure of SLM alloys. Images taken from Cain et al. (2015) and Carter et al. (2014), respectively.

Most literature also reports anisotropic behaviour in SLM alloys, although the exact nature of this behaviour varies significantly between different materials and even between studies of the same alloy (Kok et al., 2018). For instance, Simonelli et al. (2014) reported Ti-6Al-4V to have the lowest strength in the z direction due to porosity between layers, compared to x and y directions; meanwhile Riemer et al. (2014) showed fatigue life of 316L stainless steel to be much better in the z direction due to a more tortuous crack path, as shown in Figure 2.5. Furthermore, the former reports poor ductility in all directions and reduced performance in the z direction due to tensile residual stresses, while the results of the latter indicate a relatively high ductility and minimal impact from residual stresses. Conversely, Cain et al. (2015) reported the tensile strength of Ti-6Al-4V to be highest in the z direction, in contrast to Simonelli et al. (2014).

Material anisotropy is generally believed to be the result of columnar grain structure, though other factors such as the residual stress profile, surface roughness, porosity and defects clearly play an important role as well (Rafi et al., 2013). Thus, the variation in existing literature on this topic can be explained by the fact that mechanical and anisotropic behaviour is caused by a combination of several complex factors,



(A) Tortuous crack path perpendicular to build direction.

(B) Non-tortuous crack path parallel to build direction.

FIGURE 2.5: Crack propagation perpendicular (A) and parallel (B) to the z direction (i.e. build direction); taken from Riemer et al. (2014)

where the dominance of each factor is determined by specific process parameters and alloy composition.

2.1.4 Dimensional Error and Surface Roughness

Irregularities in the surface offer natural crack initiation sites, making the part vulnerable to local loads and premature part failure due to fatigue. The local arithmetic mean surface roughness (Ra) of parts produced via SLM can range between 5 µm and 50 µm, depending on a number of process parameters such as build orientation, laser power, powder quality and layer thickness (Delgado et al., 2012; Snyder et al., 2015).

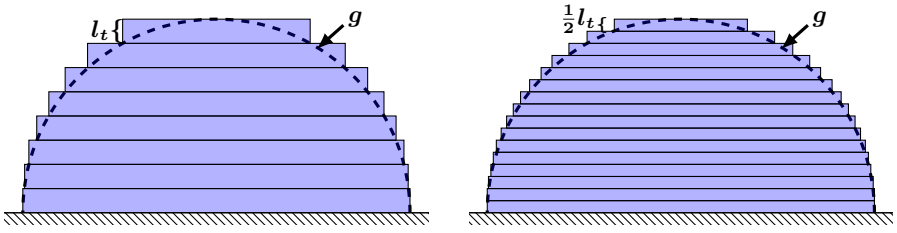


FIGURE 2.6: Schematic description of the stair-stepping effect, where $\boldsymbol{l_t}$ denotes the layer thickness and $\boldsymbol{g}$ denotes the original CAD geometry.

A common source of surface roughness and dimensional error shared by all AM processes, is the discretisation of the part into a finite number of layers, causing what is knowns as the "stair-stepping effect" (Pandey et al., 2004). This effect is strongly dependent on the layer thickness, as shown in Figure 2.6, and the part build orientation. An additional effect in the case of SLM (and other metal powder-bed processes), is the thermal diffusion from the heated sections during build into the adjacent and underlying loose powder, causing some of the loose powder particles to fuse to the surface of the part. This effect is dominant on near-vertical surfaces where the stair-step cusp becomes smaller than the size of adhering powder particles

(e.g. bottom section of the dome in Figure 2.6); the opposite is true for more sloped surfaces where the stair-stepping is more pronounced (e.g. middle and top sections of the dome) (Strano et al., 2013b). In the case of severe local heat build-up, part failure can occur as shown in Figure 2.7.

As mentioned in Section 2.1.2, support structure must be placed in such areas to reduce the risk of heat build-up, as well as to prevent distortion from residual stress relaxation. On the flip side, the removal of support structure leaves small protrusions on the surface of the part. These protrusions are reported to increase the surface roughness by up to an order of magnitude when compared with surfaces that do not require support (Snyder et al., 2015; Cooper et al., 2012).

Finally, flat top-facing surfaces are not affected by stair-stepping, powder adhesion or support structure. Nevertheless, even these surfaces exhibit a relatively poor surface finish (e.g. compared to machining) due to a rippling effect caused by the moving laser on the melt-pool surface, which cools and solidifies before it is able to settle back to its original state (Mumtaz and Hopkinson, 2009; Kruth et al., 2010).

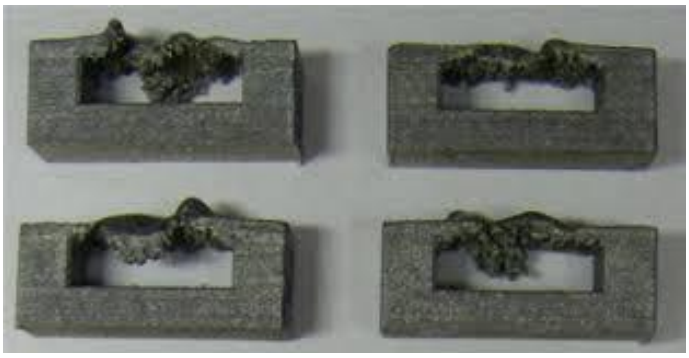


FIGURE 2.7: Example of specimen failure due to heat build-up in an unsupported overhanging section. Taken from Mertens et al. (2014).

As can be seen, the surface finish produced by SLM is the result of several different phenomena acting on different local surfaces. As a result, improving the surface finish of the as-built part is very difficult and often infeasible, instead requiring a compromise between SLM process parameters, and additional surface finishing operations post SLM. For example, a study by Mumtaz and Hopkinson (2009) compared the effects of different pulsed laser strategies on the *Ra* of top-facing and side-facing surfaces of a nickel alloy. While high peak power of the pulsed laser was found to improve the *Ra* of both surfaces, other parameters such as scan overlap, scan speed and pulse frequency were not able to reduce the *Ra* of one surface without increasing the *Ra* of the other. Therefore, the trade-off between the roughness of different local surfaces must be considered when selecting build orientation, as done by a number of researchers (Alexander et al., 1998; Hur and Lee, 1998; Masood et al., 2003; Ahn et al., 2007; Canellidis et al., 2009; Leary, 2016). A detailed review of these works is presented in Section 2.2.2.

2.2 Process Parameters Modelling and Optimisation

It is clear from the above discussion that SLM process planning is a difficult decision-making process, as numerous inputs and objectives must be considered. Section 2.1 is mainly concerned with material properties and build quality; however, the cost and time of production, which are discussed in detail in Section 2.3, are equally important and add further complexity and trade-offs to the problem. For instance, reducing the layer thickness can improve the surface roughness, which in turn can reduce the post-processing time; however, this can also increase the production time and the indirect cost of SLM. Moreover, thinner layers may affect the anisotropic material behaviour (for example, by introducing more porosity between layers (Savalani and Pizarro, 2016)), and require other parameters such as laser speed and power to be adjusted accordingly to avoid overheating and extensive fusion of loose particles on down-facing surfaces.

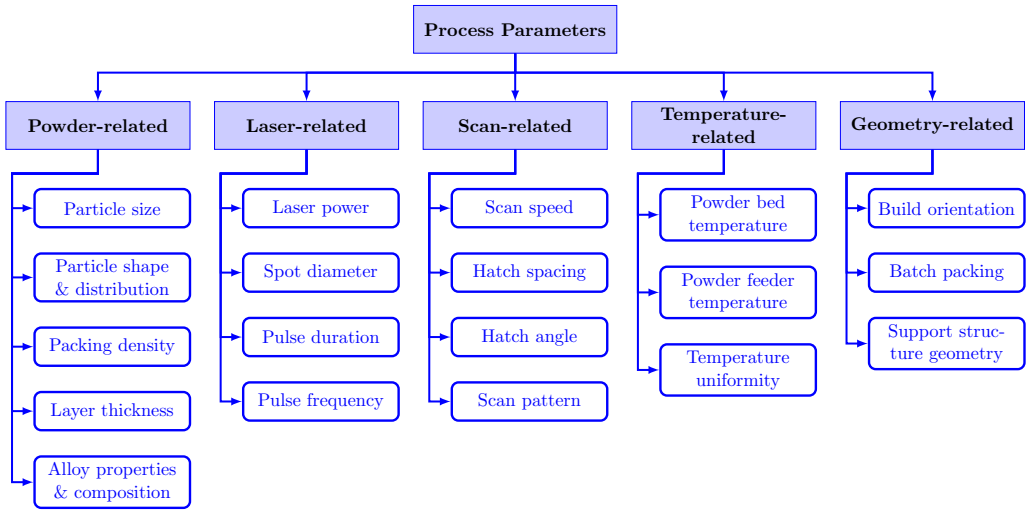


FIGURE 2.8: Key process parameters in SLM. Adapted from Aboulkhair et al. (2014).

Figure 2.8 provides a summary of key parameters grouped into five categories. It should be noted that while this diagram provides the most influential parameters with respect to build quality, cost and time, it is by no means exhaustive – according to Thomas (2009) there are at least 50 different parameters that could affect the final outcome of the build. As outlined in Section 2.1, laser-, scan- and temperature-related parameters are predominant drivers of build quality, and thus, must be tuned to minimise the presence of porosities, defects and residual stresses. While these parameters also influence cost (for example, due to energy consumption and build time) the costs of build failure, poor part quality or extensive post-processing that can result from inappropriate laser, scan and temperature settings are more than likely to outweigh any moderate savings these settings could produce during the SLM build.

On the other hand, selecting suitable optimisation criteria for geometry-related parameters is a challenge in itself, as they have a direct effect on production cost

and time, as well as a somewhat less transparent effect on build quality. This is reflected by the large number of different objective functions and criteria that have been proposed in literature and lack of agreement on the most suitable one. The problem is made worse by the fact that these parameters are not independent of each other; for instance, build orientation has an influence on both the packing solution and the support structure. To further demonstrate these challenges, each geometry-related parameter and relevant literature are discussed in detail below.

2.2.1 Support Structure

Support structure is a source of additional build time and cost, material scrap and post-processing. As a result of the latter, support structures also restrict the geometry freedom of SLM, as tool access for their removal and subsequent surface finishing must be factored into the design. However, support structure is often necessary in SLM, as it performs the following key functions:

- Preventing the part from moving or distorting, either by sinking due to gravitational forces (i.e. in overhanging sections) or warping due to high thermal gradients and residual stresses;
- Dissipating heat away from the part;
- Providing extra support especially for thin and delicate features, and preventing lateral distortion of layers caused by the powder re-coater;
- Anchoring the part to the build plate and facilitating its removal post-production (i.e. base support structure).

In addition to providing sufficient strength and stiffness to carry out these functions, support structure must also be relatively easy to remove in order to reduce the time, cost and risk of part damage during finishing operations. For this reason, support attachments on the part are often designed as tapered "teeth", as can be seen in Figure 2.9.

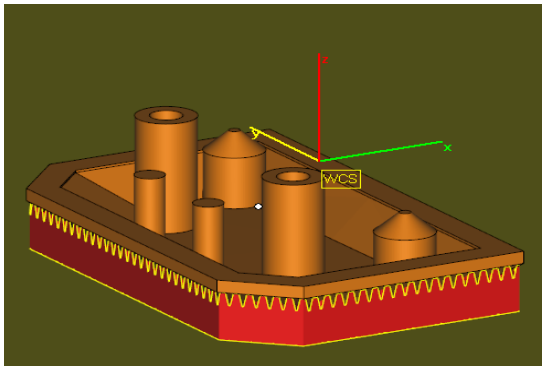


FIGURE 2.9: Support structure (red) with teeth attachments, automatically generated by Magics software for an end cap geometry (orange).

It is common to consider any overhanging surface of the part below a certain angle, typically between 30° and 45° (Thomas, 2009; Järvinen et al., 2014; Calignano, 2014; Levatti et al., 2014), as requiring support to prevent its distortion or collapse. Delicate features, such as thin walls and protrusions may also be supported. This is the logic used by the commercial software Magics, which is currently the industry standard process planning software for AM. The software contains a database of specifications for a number of existing AM machines provided by different manufacturers (e.g. EOS, Concept Laser, etc.), including manufacturer-specified overhang limits, which the software can use to automatically generate support structure for a given part geometry (2.9). As shown in Figure 2.10, Magics also offers several types of support geometries, which can be selected automatically based on the geometry of the surface to be supported, or can be chosen by the user. While the automatic support structure generated by Magics provides a good starting point, especially for an inexperienced user, it is not a definitive solution; for example, the software does not optimise support structure with respect to cost or its thermal profile, thus, providing scope for further research and improvement.

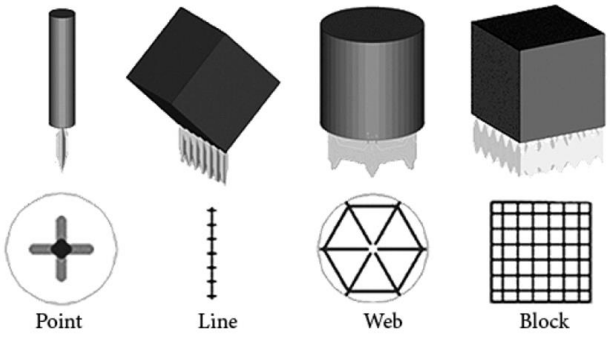
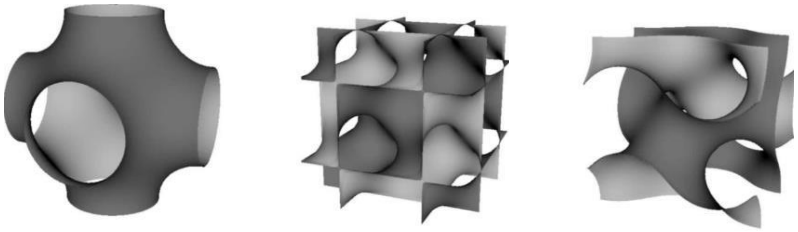


FIGURE 2.10: Types of generic support geometries provided by Magics software. Taken from Thomas (2009).

A number of researchers have investigated the trade-off between the functionality, volume and ease of removal of different support structure geometries. Poyraz et al. (2015) and Calignano (2014) tested the effects of overhang geometry and teeth dimensions on part distortion and ease of support removal for simple block support geometries. Similarly, Cloots et al. (2013) studied the effects of overhang angle and scan parameters on the accuracy and quality of overhanging sections, proposing a localised scanning strategy and optimised support structure geometry to reduce the unsupported overhang angle and minimise the need for post-processing. Järvinen et al. (2014) proposed the use of a pulsed laser strategy to built less dense supports which are easier to remove; they also found web supports (Figure 2.10) much easer to remove than solid struts (Figure 2.13), with less damage on the part's surface during removal.

More complex lattice topologies have also been considered, to further reduce support structure volume and facilitate its removal. For instance, researchers at the



(A) Schwarz primitive. (B) Schwarz diamond. (C) Schoen gyroid.

FIGURE 2.11: Three types of triply periodic minimal surface cells proposed for lattice support structure. Taken from Hussein et al. (2013).

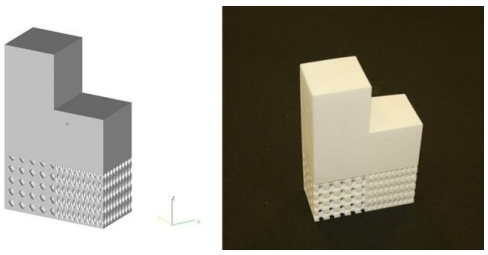


FIGURE 2.12: Graded lattice support structure using Schwarz primitive cells. Taken from Strano et al. (2013a).

University of Exeter have proposed triply periodic minimal surface (TPMS) cells, shown in Figure 2.11, as potential building blocks for lighter and easier to remove lattice support structures. Hussein et al. (2013) compared Gyroid and Diamond support lattices, studied the effects of different cell configurations (e.g. cell size, cell orientation etc.) and proposed a graded support structure strategy for improved heat dissipation and ease of removal. Strano et al. (2013a) proposed a similar graded lattice strategy based on the vertical load on the lattice using the Schwarz primitive unit cell, as shown in Figure 2.12. Despite the use of these advanced lattice structures in academic works, it is still common practice in industry to use solid support structures in SLM (Morgan et al., 2017), as shown in Figure 2.13.

An interesting method of creating dissolvable support structure for SLM has been proposed by Lefky et al. (2017). The method was applied to a 17-4 PH stainless steel part, which was coated with a sensitizing agent and heat treated. This process modified the chemical composition of the part's surface to a depth of 100 µm – 200 µm, while completely penetrating the lattice support structure. Subsequently, the part surface layer and the entire support structure was removed at an accelerated rate using chemical etching. Despite its potential, the scope of this method, its effects on surface roughness and material properties, as well as its implications on overall production cost, require further examination and development.

A number of researchers have also attempted to reduce or even eliminate the need for support structure by modifying the part geometry, for instance, by making tear-shaped holes (Thomas, 2009) and by modifying overhanging surfaces to be self-supporting (Hu et al., 2015). More recently, topology optimisation with additional

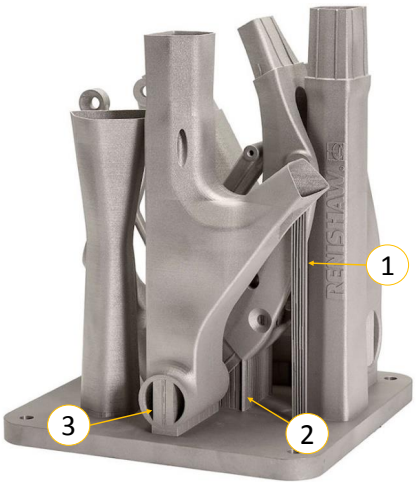


FIGURE 2.13: Mountain bike frame built by Renishaw, with custom-designed solid support structures, as indicated by yellow arrows: 1. Thin struts used to support relatively gentle overhang; 2. Thicker struts used to support the base of the part and facilitate part removal from the plate 3. The hole is a critical feature with severe vertical overhang at the top, thus solid supports are used to preserve its dimensional accuracy.

considerations of AM manufacturability, such as overhanging surfaces and minimum feature size, has become a popular area of research (Leary et al., 2014; Gaynor and Guest, 2016; Allaire et al., 2017; Zhang and Zhou, 2018; Langelaar, 2018). However, this paradigm comes with its own set of trade-offs, for example, self-supporting surfaces usually come at the expense of reduced stiffness or additional volume and weight in the re-designed part (Leary et al., 2014; Allaire et al., 2017; Zhang and Zhou, 2018). In addition to functional requirements, most parts (especially assembly parts) come with a set of geometric constraints, presenting another trade-off between design requirements and manufacturability. Furthermore, most of these methods are tested on 2D cases (Leary et al., 2014; Gaynor and Guest, 2016; Zhang and Zhou, 2018; Langelaar, 2018), and thus require further work to reduce their computational expense and make them applicable to more complex 3D geometries. Due to computational expense, current models are not able to consider thermal effects such as local heat build-up, residual stresses and distortion (Langelaar, 2018).

Finally, it is important to consider the build orientation before optimising the part or support structure, since an optimised design would become meaningless and would have to be repeated if the build orientation is changed. Several researchers (Alexander et al., 1998; Schwerdt et al., 2000; Strano et al., 2013a; Das et al., 2015, 2017) have used build orientation to minimise the presence of overhanging surfaces and thus minimise support structure, as discussed in the following section. Ideally, build orientation should be considered simultaneously alongside part design and support structure geometry, in order to minimise or eliminate the latter; however, this problem is currently computationally infeasible and only approximate solutions exist (Langelaar, 2018).

2.2.2 Build Orientation

The need for a decision support tool when selecting the build orientation was first identified by Frank and Fadel (1995), who proposed an interactive rule-based system, which suggests a favourable orientation based on the surface quality of up to two user-selected part features for Stereolithography (SLA). The tool can also provide estimates of supported part area and build time, however, only the surface quality was used as a selection criterion. Several other researchers have addressed surface finish: Schwerdt et al. (2000) proposed an analytical solution to minimise the part surface area requiring support; Masood et al. (2000); Hur et al. (2001); Chowdhury et al. (2017); Jaiswal et al. (2018) considered the surface error due to the stair-stepping effect; while Alexander et al. (1998) and Byun and Lee (2006) considered the sum of both. The method used by Masood et al. (2000) for measuring the stair-stepping effect involves slicing the STL geometry and measuring the volumetric deviation of individual slices, providing an accurate but time consuming solution. A much simpler and more common approach is to measure the stair-stepping error in 2D using the trigonometric function shown in Equation 2.1, as done by Alexander et al. (1998).

$$h_c = \begin{cases} l_t |\cos(\alpha)| & \text{if } |\cos(\alpha)| \neq 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

where h_c is the cusp height between two layers, l_t is the layer thickness and α is the overhang angle of the surface. Typically the stair-stepping error alone does not provide an accurate prediction of the actual surface roughness of the built part, as it neglects other thermal effects, which are discussed in Section 2.1.4. Thus, modelling surface roughness based on practical data is a more suitable approach, as done by Ahn et al. (2007); Strano et al. (2011) who interpolated surface roughness data provided for SLS; Leary (2016) who fit a logarithmic curve to a larger and more scattered data set gathered for titanium alloy; and Strano et al. (2013b) who proposed an analytical function based on an idealised model of spherical particles attaching to the stair-steps during SLM of stainless steel, although this model only considers the top-facing surfaces.

As discussed in Section 2.1.4, it is impossible to optimise the entire part surface, as there are trade-offs between different local surfaces. Thus, many existing methods consider the average surface roughness of the part as the objective (Masood et al., 2003; Byun and Lee, 2006; Delfs et al., 2016; Leary, 2016); however, a more suitable approach is to prioritise or weight different local surfaces based on their functional and geometric requirements. This is acknowledged by Ahn et al. (2007), who minimised the total difference between build surface roughness and target surface roughness of local surfaces. Leary (2016) also made a distinction between functionally "relevant" and "irrelevant" surfaces, and only considered the former in his objective function, however, this binary approach is less flexible. It should also be noted that all of the above approaches assume a local post-processing method such as manual polishing or

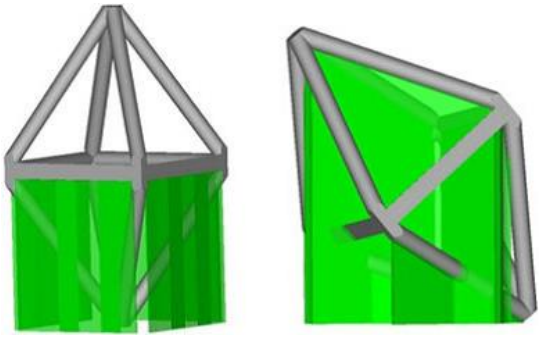


FIGURE 2.14: Example of predicted support volume, shown in green. Taken from (Strano et al., 2013a).

machining, whereas in many cases parts produced via SLM are finished using global methods such as barrel tumbling (Schmid and Levy, 2012; Gordon et al., 2016). In the latter case, optimising the build orientation for surface roughness is likely to have a minimal impact on the final cost and quality of the part.

Build time is another parameter that has been addressed by a number of researchers. Most existing models (Alexander et al., 1998; Canellidis et al., 2009; Singhal et al., 2009; Delfs et al., 2016) assume a constant layer thickness and use build height to estimate build time, as this is much faster and cheaper than computing the actual build time of each layer, as discussed in Section 2.3.3. Combined optimisation of build orientation and layer thickness has also been proposed to improve both the surface finish and build time (Hur and Lee, 1998; Masood et al., 2003). While this is reasonable in the context of RP, where dimensional accuracy, finish and build speed of prototypes can take priority over their mechanical strength, such an approach is not suitable for SLM of end-use parts, where challenges such as porosity and material properties must be considered first.

While the importance of cost is acknowledge throughout AM literature (Baumers et al., 2017; Laureijs et al., 2017; Fera et al., 2017), only the works of Alexander et al. (1998) and Byun and Lee (2006) were found to address cost directly as an objective of build orientation, while several others (Xu et al., 1999; Ahn et al., 2007; Zhang et al., 2017) alluded to cost but did not model it fully or explicitly. Moreover, Alexander et al. (1998) only used cost to break tie between build orientations of similar surface quality, while Byun and Lee (2006) considered cost, time and surface roughness as competing objectives. Conversely, many more works have focused on build time (Frank and Fadel, 1995; Hur and Lee, 1998; Lan et al., 1997; Byun and Lee, 2006; Canellidis et al., 2009; Delfs et al., 2016), surface roughness Frank and Fadel (1995); Alexander et al. (1998); Masood et al. (2000); Hur et al. (2001); Chowdhury et al. (2017); Delfs et al. (2016); Jaiswal et al. (2018) and support structure volume (Lan et al., 1997; Hur and Lee, 1998; Schwerdt et al., 2000; Strano et al., 2013a; Zhang et al., 2016; Das et al., 2015, 2017), which is typically calculated as the volume underneath the supported area as shown in Figure 2.14. Although both build time and support structure have a direct effect on cost, they do not provide the full picture,

especially since they are often optimised implicitly by minimising build height (for build time) and supported surface area (for support structure volume).

A number of works have also addressed the dependence of material properties and build quality on build orientation. The work of Thompson and Crawford (1997) was one of the first, using analysis of variance (ANOVA) to produce response surfaces relating ultimate tensile strength (UTS), build time and surface roughness to build orientation, laser power and layer thickness. Their experiments showed polymer parts built via SLS to be weaker in the z direction due to the relatively weaker bonding between layers. More recently, Ulu et al. (2015) used a combination of experimental data and finite element analysis (FEA) to predict the mechanical properties of a case study part as a function of build orientation. The latter proposed using a factor of safety based on the modelled part properties and design requirements to optimise the build orientation for best part performance. Peng et al. (2017) also used FEA to predict part distortion as a function of build orientation, while Jaiswal et al. (2018) developed a layer-by-layer model for evaluating the material composition error for functionally graded material (FGM) extruded via FDM. Although the layer-wise approach offers an accurate prediction, it is also very time consuming and computationally expensive; to deal with this Jaiswal et al. (2018) used a surrogate model to approximate the output of the expensive model. Similarly, both Ulu et al. (2015) and Thompson and Crawford (1997) used surrogate modelling and data fitting to deal with the large computational burden of FEA and time-consuming experiments, respectively. Meanwhile, Peng et al. (2017) used the dividing rectangles (DIRECT) optimisation approach, which avoids exploring large areas of the solution space by dividing it into rectangles at each iteration and only evaluating the centre point of each rectangle. At each iteration the best rectangle is divided into smaller rectangles and the process is repeated until convergence or until a termination criterion is satisfied.

Table 2.1 presents a summary of reviewed literature. When searching for an optimal build orientation, two types of approaches have been proposed in literature: continuous rotation and discrete. One common discrete method, is to consider each flat surface of the part as a potential base, corresponding to a candidate build orientation. However, this method is only suitable for relatively simple geometries. If the part geometry is relatively complex, a "gift-wrap" approach can be used, where a 3D convex hull is generated and each flat surface of the hull is considered as a candidate base of the part. Another method is to isolate important surfaces and design features (e.g. a hole) and generate the most favourable build orientation based on these critical features, as done by Frank and Fadel (1995) and Zhang et al. (2017). This method makes the optimisation search more "intelligent" but can also be quite restrictive and requires either a very large database or manual interaction from the user in order to identify relevant design features. Since discrete approaches produce a relatively small set of candidate solutions, these solutions can normally be evaluated via simple exhaustive methods.

Authors	Objectives						Method	AM Process
	BT	BC	SR	SA	SV	Other		
Frank and Fadel (1995)	⋆		×	⋆			Interactive decision support, can position up to two user-selected surfaces to be top- or side-facing.	SLA
Thompson and Crawford (1997)	×			×		×	Uses ANOVA to create response surfaces of BT, SR and UTS as functions of laser power, layer thickness and build orientation.	SLS
Lan et al. (1997)	×		×	×			Uses user-specified primary objective, with secondary objective to break ties. Discrete rotation.	SLA
Alexander et al. (1998)	⋆	⋆	×	⋆	⋆		Selects orientation with smallest SR, using cost to break ties. Discrete rotation.	SLA, FDM
Hur and Lee (1998)	×		×		×		User must rank objectives or use a WSM. Continuous rotation.	SLA, FDM
Schwerdt et al. (2000)				×			Uses ray intersection calculations to prevent supports on a user-specified facet.	Generic
Masood et al. (2003)			×				Uses slicing approach to get accurate stair-step error. Incremental rotation using 45^o and 5^o steps about two axes, respectively.	FDM, SLA, SLS
Byun and Lee (2006)	×	×	×				Solutions placed in a decision matrix and scored using WSM function. Discrete rotation.	FDM, SLA, LOM, SLS
Ahn et al. (2007)			×				Uses GA to minimise total difference between target SR and as-built SR. Continuous rotation.	SLS
Canellidis et al. (2009)	×		×	×			Uses GA and WSM function to indicate fitness. Continuous rotation.	SLA
Strano et al. (2011)				×	×		Uses GA to find Pareto front between energy consumption and SR. Incremental rotation with 5^o steps.	SLS
Ulu et al. (2015)						×	Uses FEA and surrogate modelling to minimise a predicted factor of safety.	Inkjet printing
Das et al. (2015, 2017)			×	×		×	Uses interior point method and WSM of SR and BT with additional tolerance constraints. Continuous rotation.	SLS
Peng et al. (2017)						×	Minimises average thermal distortion using FEA and DIRECT optimisation approach.	SLS
Jaiswal et al. (2018)			×	×		×	Uses surrogate modelling to optimise WSM of SR and material composition error for FGM.	FDM

TABLE 2.1: Summary of key literature on build orientation, where BT = build time; BC = build cost; SR = surface roughness; SA = supported surface area; SV = support structure volume. Primary and secondary objectives are marked with × and ⋆, respectively.

On the other hand, continuous methods are much more flexible and generally more automated, but come at the expense of a much larger solution space. Consequently, such methods must rely on more complex search heuristics, such as Genetic Algorithm (GA), which is a popular choice in literature as shown in Table 2.1. Many works use a weighted sum model (WSM) to solve multi-objective problems, as this is a simple and straightforward method. However, it cannot distinguish between different Pareto solutions with identical WSM scores and requires the user to provide

appropriate weights for different objectives, which can be difficult in practice and particularly misleading for objectives which are already interdependent (for instance, support structure volume and cost). Furthermore, in the case of discrete rotation, the problem is made more tractable by the limited number of candidate solutions, while using continuous rotation to optimise build orientation for multiple objectives presents a much bigger challenge, requiring a careful balance of solution space exploration against computational feasibility and solution time.

2.2.3 Batch Packing

Despite the absence of tooling costs, cost amortisation can still be achieved in AM through efficient machine utilisation, as demonstrated in Section 2.3. The need for an optimal arrangement of parts to maximise machine utilisation or minimise the number of used machines, presents what is known as a bin packing problem. This is a combinatorial problem belonging to a wider class of problems known as Cutting & Packing (C&P), which is a well-established field of research dating back to the 1960s (Gilmore and Gomory, 1961). The following section provides a brief overview of C&P, while the focus of Section 2.2.3 is specifically on bin packing in the context of AM. The focus of this review is on 2D methods, although 1D and 3D approaches are also mentioned where relevant.

The Bin Packing Problem

The generalised bin packing problem can be formally defined as follows: given a set of large objects and a set of small objects (i.e. pieces or parts[1]), assign all or some of the small objects to one or more of the large objects (i.e. bins), to optimise some given objective function. The placement of small objects must meet the geometric condition, which states that:

1. small objects may not intersect each other; and
2. small objects must lie entirely within the large object to which they are assigned.

This problem is known to be NP-hard, meaning that currently it cannot be solved in an exact or exhaustive way in reasonable time; instead, heuristic approaches must be used to find a "good enough" approximation (Whitwell, 2004).

Some of the earliest and simplest heuristics used for assigning pieces to bins include next-fit (NF), first-fit (FF) and best-fit (BF) algorithms (Coffman et al., 1999). NF opens a bin and places each piece in that bin sequentially. If the bin cannot accommodate the piece, NF closes that bin and opens a new one, where it places the piece. Each following piece is placed in the most recently-opened bin and once a bin is closed it cannot be used again.

In contrast, FF packs each piece in the lowest indexed bin that can accommodate the piece, and all partially filled bins are kept open until all pieces have been placed.

[1]The terms "piece" and "part" are used interchangeably from this point on.

Similarly to FF, BF keeps all partially filled bins open until the end, but instead of stopping at the first bin that can accommodate the piece, BF will check all available bins and select the one with the least available free space in which that piece can still fit. These heuristics were originally used to solve 1D bin packing problems, and were later extended to 2D packing, where pieces are typically ordered by decreasing height and placed in the lowest possible gap inside the rectangular bin (Burke et al., 2006).

A popular 2D packing method is the bottom-left (BL) heuristic, which seeks to sequentially place each object in a bottom left stable position. Figure 2.15 shows an example of this method, as implemented by Jakobs (1996), which first slides the object downwards, followed by a slide to the left until no further moves can be achieved.

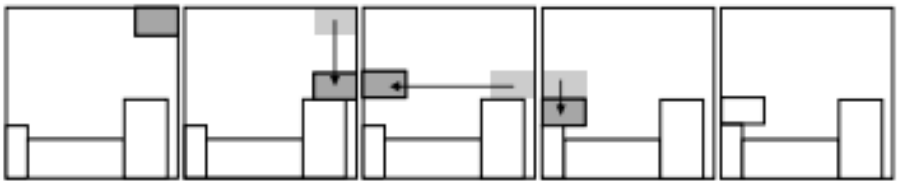


FIGURE 2.15: Bottom-left 2D bin packing heuristic. Taken from Jakobs (1996).

Packing of Irregular Shapes

Intersection checks in rectangular packing problems are relatively easy to compute, as only four orthogonal edges need to be considered for each piece. This task becomes much more challenging and computationally expensive when dealing with polygonal (i.e. irregular) objects, particularly when these objects are non-convex. Pixel-raster scanning, trigonometric functions, no-fit polygon and phi-functions are the most well-known and well-documented approaches of checking for intersections between irregular shapes (Bennell and Oliveira, 2008).

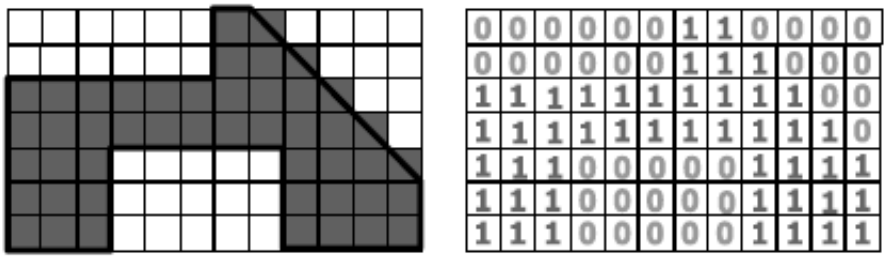


FIGURE 2.16: Example of a binary pixel-raster representation of an irregular object. Taken from Bennell and Oliveira (2008).

The basic idea behind the pixel-raster method is to discretise the 2D bin space into a grid, where each cell contains information about whether it is full or empty. Similarly, each irregular shape is represented by a discrete group of cells (i.e. pixels) as shown in Figure 2.16, and a raster scan through the bin is performed to determine an appropriate region of empty cells for that shape to be placed into. This method is reasonably fast at finding feasible piece placements, are simple to code and can

represent any irregular and non-convex shape. However, the storage of pixel information requires a lot of memory resources and can become prohibitive for very complex geometries or geometries which require very high resolution.

Trigonometric methods use vectors to represent piece topology and detect collisions between pieces. This approach provides much more accurate rendering of irregular geometries than the raster method, with the only source of error being in the rounding of float numbers by the computer (Whitwell, 2004), however, they are also very computationally expensive. In the past decade, the no-fit polygon (NFP) has become a very popular approach for intersection checking, the concept for which was first introduced by Art (1966). The basic idea is demonstrated in Figure 2.17: the reference point of polygon j is traced around the boundary of polygon i until it reaches its original starting point; the resulting traced polygon is the NFP, as shown in bold.

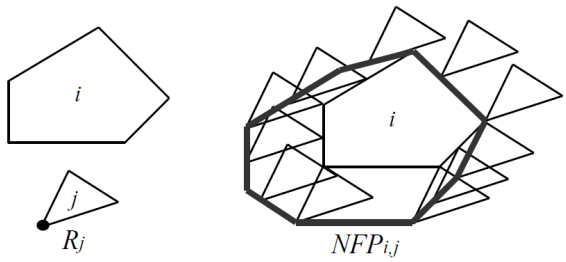


FIGURE 2.17: No-fit polygonNFP_{ij} of two convex polygons i and j. Taken from Imahori et al. (2006).

By determining the NFP of each pair of polygons prior to placing them in the bin, the intersection check becomes a relatively simple operation of checking whether the reference point of the current polygon lies inside the NFP boundary (i.e. the polygons intersect), outside (i.e. no intersection) or exactly on the NFP boundary (i.e. polygons are just touching) with regard to the one or more polygons that have already been placed in the bin. The main downside of this approach is when dealing with non-convex shapes, the NFP may result in degenerate cases or may miss internal cavities in the stationary polygon if the sliding polygon has no way to slide into that cavity (e.g. a hole or C-shaped cavity). While some researchers have addressed these cases, a robust method for dealing with degenerate NFPs is yet to be established (Burke et al., 2007; Dean et al., 2006).

Finally, the phi-function is a technique developed by Stoyan et al. (2004), which represents all mutual positions of two shapes through mathematical expressions: two objects are in contact if the phi-function value is exactly zero, intersect if the value is less than zero and are separated if the value is greater than zero. This approach could prove a very powerful tool able to deal with a large number of irregular shapes, however, currently its adoption is limited by the lack of a robust algorithmic procedure for arbitrary shapes (Bennell and Oliveira, 2008).

Bin Packing in AM

Both 2D and 3D bin packing problems have been solved in the context of AM. The former is aimed at processes such as FDM, SLA and particularly SLM, where the presence of high residual stresses, the need for support structures and the subsequently raised risk of build failure currently limits the possibility of stacking parts on top of one another, thus, reducing the problem to 2D. On the other hand, with some AM methods it is possible to avoid the need for support structures, for example in the case of SLS where parts may be sufficiently supported by the surrounding powder (although care should be taken when generalising this assumption for different geometries) (Zhang et al., 2016; Baumers et al., 2011), and therefore 3D bin packing is preferable in this case to make full use of the machine capacity.

Several approaches have used the BL placement heuristic to generate candidate packing solutions in 2D (Canellidis et al., 2010, 2013) and an equivalent heuristic in 3D, which places pieces as close to the bottom-left-front (BLF) corner of the bin as possible (Baumers et al., 2013; Gogate and Pande, 2008; Hur et al., 2001; Wu et al., 2014). Common criticisms of such sequential methods include limited solution exploration and inability of placing pieces inside large closed concavities of other pieces (Zhang et al., 2016), as shown in Figure 2.18. As an alternative, Zhang et al. (2016) proposed a parallel packing approach where a GA is used to drive the translation and rotation of pieces simultaneously, starting from an initial naive layout and searching for a solution with minimal piece overlap. However, this approach allows the overlap of pieces during the search and thus cannot guarantee that the final solution will be feasible. It is also unclear how the authors performed overlap checks.

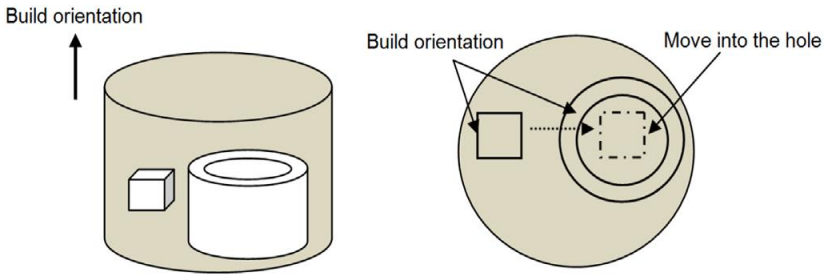
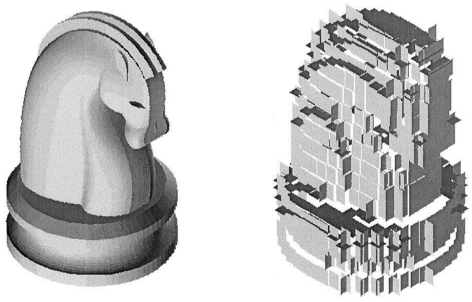


FIGURE 2.18: Example of how a smaller piece can be placed within the closed concavity of a larger piece. Taken from Zhang et al. (2016).

The overlap checks that must be carried out in packing problems, especially in 3D, are typically the most computationally expensive part of the algorithm. For this reason, many of the earlier works have used bounding boxes of pieces, both in 3D (Wodziak et al., 1994; Zhang et al., 2002) and in 2D (Canellidis et al., 2009). In a follow-up paper, Canellidis et al. (2013) also proposed a method which used the NFP approach to detect intersections between convex shapes, and used ray casting to extend this approach to non-convex shapes.



(A) Original STL (≈76,000 facets). (B) Depth map (≈1,000 facets).

FIGURE 2.19: Method of reducing the number of facets representing the STL geometry. Taken from Dickinson and Knopf (1998).

Several methods used voxels to represent 3D pieces (Hur et al., 2001; Gogate and Pande, 2008; Wu et al., 2014). This is a computationally expensive approach, which often requires a compromise between the resolution of voxels and running time of the algorithm – as shown by Gogate and Pande (2008) who used very low resolution of voxels early on in the algorithm, reducing the size of voxels later on, as the algorithm would start to converge on a final solution. Dickinson and Knopf (1998) took a slightly different approach by deriving six depth maps using the bounding box of the part geometry, and replacing the triangulated facets of the STL file with a reduced number of planar facets with six degrees of freedom as shown in Figure 2.19.

It can be seen that build orientation and bin packing are two closely related parameters in AM and should be considered as two parts of a single optimisation problem, since the former cannot be changed without updating the latter. While this fact is acknowledged by several researchers, existing literature addresses the two parameters in two separate stages and no works have been found to address the two parameters simultaneously. Furthermore, no works have considered the effects of the bin packing solution on AM-related objectives, such as surface roughness and build time; though a number of researchers consider these objectives implicitly through build orientation. For example, Ikonen et al. (1997) assumed parts to be provided with adequate build orientations and solved a 3D bin packing problem, using a Simulated Annealing (SA) algorithm to minimise overlap between pieces; Gogate and Pande (2008) provided an algorithm to rank candidate build orientations based on stair-step error and allowed the user to interactively select a build orientation, packing the resulting pieces in a separate stage; while Canellidis et al. (2006) provided a separate optimisation algorithm to generate build orientations based on optimum surface quality, support structure and build time (see Section 2.3) before applying a bin packing algorithm with the objective of maximising utilisation. A more recent work by Wu et al. (2014) used a BLF placement heuristic, along with a GA to generate a Pareto front of the maximum build height, surface roughness and the support volume of the batch. However, in their approach the build orientations of parts are not optimised and kept fixed throughout.

Authors	Problem Type	Piece Representation	Heuristic & Objective	Build Orientation	AM Process
Wodziak et al. (1994)	2D & 3D	Bounding box	GA; max. percentage of placed pieces or max. bin utilisation.	Considered fixed	SLA
Nyaluke et al. (1996)	3D	Bounding box	GA; max. number and size of placed pieces.	Considered fixed	SLS
Ikonen et al. (1997)	3D	Bounding box & STL facets	GA; min. distance from origin and percentage overlap between pieces.	Not considered	SLS
Dickinson and Knopf (1998)	3D	Depth map	SA; max. bin utilisation and compactness based on point moment of pieces.	Not considered	Generic
Hur et al. (2001)	3D	Voxels	GA; max. bin utilisation and min. build height.	Considered fixed	SLS
Gogate and Pande (2008)	3D	Voxels	GA; min. build height, stair-step error and supported area.	Considered fixed	SLS
Canellidis et al. (2009)	2D	Bounding box	GA; max. bin utilisation.	Considered fixed	SLA
Baumers et al. (2013)	3D	Voxels	Min. distance from bin centre & max. bin utilisation.	Not considered	SLM, EBM
Canellidis et al. (2013)	2D	Polygon	GA; max. bin utilisation.	Considered fixed	SLA
Wu et al. (2014)	3D	Voxels	GA; Support volume, height, surface roughness.	Considered fixed	Generic
Zhang et al. (2016)	2D	Polygon	GA; min. percentage overlap between polygons.	Considered fixed	SLA, SLM

TABLE 2.2: Summary of key literature on bin packing in AM.

Only the work of Zhang et al. (2017) has addressed the build orientation of more than one part at a time, by considering a limited set of candidate build orientations for a group of parts and using a GA to find a set of build orientations, where the fitness function is based on the closeness of the solution to a predefined aspiration criterion. In an earlier work, Zhang and Bernard (2014) also proposed a batch grouping methodology for multiple parts with fixed build orientations, based on similarity criteria such as part height and size. As can be seen, Zhang *et al.* consider different aspects of the problem in separate stages and as a result present a number of limitations, such as assuming similarity between pieces to be the best indicator of good bin assignment (which may not always be the case)[2]; and only dealing with a small enough set of pieces to fit into a single bin.

Finally, no works have been found to truly consider a multi-bin packing problem in AM and as a result, such approaches give no guarantee of a full placement of pieces; instead either assuming a sufficiently small group of pieces (Gogate and Pande, 2008; Hur et al., 2001), or designing the algorithm to select the best subset of

[2]For instance, separating large and small pieces is likely to result in a packing solution with much lower bin utilisation, than a solution where pieces of different sizes are mixed together.

pieces to maximise bin utilisation (Ikonen et al., 1997; Wodziak et al., 1994; Nyaluke et al., 1996; Dickinson and Knopf, 1998; Canellidis et al., 2009).For this reason, bin utilisation (i.e. proportion of bin capacity occupied by pieces) is the most commonly used objective in these packing algorithms, although other metrics such as piece compactness (Canellidis et al., 2009) and the combined centre-of-mass of packed pieces (Baumers et al., 2013) have also been used. The reviewed literature is summarised in Table 2.2.

2.2.4 Summary of Search Heuristics

As can be seen from tables 2.1 and 2.2, the majority of reviewed optimisation problems make use of GA. Simulated annealing (SA) is another popular and effective search heuristic (Szykman and Cagan, 1997; Dickinson and Knopf, 1998), which is based on the process of heat diffusion in metals. The movement of atoms from a less favourable state to a more favourable state in the metal crystal lattice is used as an analogy for the movement of an algorithm through solution space. Similarly, the algorithm is given a temperature schedule which determines the starting point, step size and termination point of the algorithm. The favourability of each new solution is evaluated based on the change in the value of the objective function, which in turn determines the probability of the algorithm moving on to this solution or staying with the current one. The general equation is given below:

$$P_{acc} = e^{\frac{-\triangle C}{T}} \tag{2.2}$$

where P_{acc} is the probability of accepting the new solution, $\triangle C$ is the change in objective function value and T is the temperature of the current solution.

Both SA and GA are meta-heuristics, meaning they are not problem-specific, and consequently, can address problems where relatively little is known about the solution space. However, unlike SA which moves iteratively from one solution to another, the GA search evaluates a population of solutions at each step (i.e. generation), where each solution is represented by a chromosome. Natural selection is the driving principle behind GA, thus each chromosome in a population is assigned a fitness value based on the objective function, which determines the probability of that chromosome mating and passing on its genes to the next generation. The apparent preference for GA in the literature could be due to the population-based nature of this heuristic, making it more suitable for parallelisation and potentially leading to faster exploration and more agressive convergence compared to the more gradual sequential search via SA. However, GA population size is also a sensitive factor which must be chosen carefully to ensure sufficient solution space exploration, which bearing in mind any computational and solution time constraints. Finally, for either SA or GA to work effectively the objective function must be computationally fast and efficient; on the other hand, if candidate solutions are computationally expensive (for instance, when using FEA) the chosen search heuristic can be coupled with a surrogate or approximation model,

otherwise, a different approach must be taken, for instance, the DIRECT method described in Section 2.2.2.

2.3 Cost Modelling

Cost is usually a key decision factor in selecting and planning a suitable manufacturing route; it is therefore critical to base such decisions on a realistic estimate of cost. The accuracy of predicted cost largely depends on the complexity and structure of the cost model, as well as data availability, which is often limited or incomplete. Existing cost modelling approaches can be grouped into the following categories (Dodd, 2015):

- *Parametric methods* search for statistical correlation between product cost and potential cost-driving variables, such as product geometry. The cost is expressed as an analytical function of the variables. Typically, the selection of relevant variables and their role in the function are determined based on expert judgement, while the function coefficients are determined through linear regression. An alternative to parametric modelling was first proposed by Cavalieri et al. (2004) in the form of artificial neural networks (ANN), a pattern recognition heuristic inspired by the human brain. Both methods can provide useful estimates of cost based on relatively limited knowledge of the product, particularly the ANN approach, which treats the cost function as a 'black box'. However, the results can be difficult to validate and can suffer from a large error margin, while statistically significant relationships do not necessarily indicate causality.

- *Analogy-based methods* estimate the cost of the target product based on a similar known product, adjusting for differences between the two. As with data-based approaches, analogy-based models require a degree of expert judgement, but tend to provide a slightly more reliable estimate (Myrtveit and Stensrud, 1999).

- *Detailed approaches* provide the most accurate estimation of cost, and include any method that tries to establish an analytical relationship between the product and the resources used in each step of the production process. According to Dodd (2015), activity based costing (ABC), bottom-up and feature-based methods all fall under this category.

2.3.1 AM Cost Modelling

A seminal model of AM build cost was produced by Hopkinson and Dicknes (2003), who used it to provide a first approximation break-even analysis of high volume production via FDM, SLA and SLS compared to injection moulding. This model considers labour, machine and material costs, while neglecting overheads and ancillary costs. The authors also make some limiting assumptions, namely, using an unrealistically high machine utilisation value of 90% and extrapolating unit cost at relatively high volumes with no direct consideration of low volume production.

These limitations were overcome by Ruffo et al. (2006) with a full cost model that includes administration, hardware and software overheads, and assumes an annual machine utilisation of 57%, which is regarded as typical in the industry. The authors propose Equation 2.3, which represents the cost of each AM build as a sum of direct and indirect costs.

$$C_{build} = C_d M + C_i T \tag{2.3}$$

where C_d is the direct cost coefficient (£/kg), C_i is the indirect cost coefficient (£/h), and M and T denote material (kg) and build time (h), respectively. The direct cost of Equation 2.3 only considers material consumption, while all other costs which cannot be measured directly based on the product (e.g. machine value absorption, administration etc.), are grouped into C_i and allocated to each build based on the time it takes to complete. Baumers et al. (2013) expanded Equation 2.3 to include energy consumption as another direct cost, calculated as a function of time and the part geometries in the build; however, their results indicate that energy consumption accounts for less than 1% of the total build cost.

It is widely acknowledged (Alexander et al., 1998; Lindemann et al., 2012; Rickenbacher et al., 2013) that certain pre- and post-processing activities tend to be an integral part of any AM process, thus, a true cost model is incomplete without these activities. Pre-processing activities mainly include file preparation (as mentioned in Section 1.2) and machine set-up. The cost and time of these activities are particularly difficult to quantify, as these can vary significantly between different AM methods, systems, software packages, and operators. Alexander et al. (1998) estimated these activities to account for roughly 10% of overall unit cost in the case of FDM and SLA, while Atzeni and Salmi (2012) estimated the cost of machine set-up to be 1.5% of overall SLM unit cost.

Post-processing activities typically include part extraction and cleaning, separation of parts from the substrate, removal of support structure and surface polishing. While post heat treatment is often required to mitigate the thermal residual stresses in parts produced via SLM, its cost is normally omitted, as it is considered to be a separate process. Arguably, the most detailed and complete model of SLM post-processing cost was provided by Rickenbacher et al. (2013). This ABC model considers the cost of part extraction from the machine; the cost of wire EDM allocated to each part based on its area of contact with the substrate; and the cost of support structure removal and surface finishing, as shown in Equation 2.4.

$$C_{finish}(P_i) = T_{finish}(G_i)(C_{op} + C_{tool}) \tag{2.4}$$

where C_{finish} is the total cost of support structure removal and finishing of part P_i with geometry G_i; T_{finish} is the total time spent on these activities, and C_{op} and C_{tool} are operator and tool cost rates, respectively. The same model also includes the pre-processing and build cost of SLM, and considers the presence of inert gas inside

the SLM build chamber; the latter is translated into a direct cost rate incurred during the build, and an additional effort factor included into the operator cost of material change and part extraction.

Finally, the costs associated with inventory, machine set-up and downtime, and build failure are typically significant but ill-structured making them very difficult to optimise and model accurately; according to Schmid and Levy (2012) such costs account for up to 20% of the overall production, while Baumers and Holweg (2016) estimates this figure to be as high as 38%. Specifically, Baumers and Holweg (2016) proposed a probability-based model to predict the cost associated with the risk of part failure in SLS. This model was used to evaluate the trade-off between vertically stacking a large number of parts to improve machine utilisation, and the increased risk of failure accompanying it. Literature on this topic was found to be very limited, likely due to the difficulty of defining and modelling ill-structured costs, and the relative novelty of this problem.

2.3.2 Economies of Scale

Unlike Hopkinson and Dicknes (2003), Ruffo et al. (2006) considered the cost per part as a function of demand, showing that economies of scale – the inverse relationship between production volume and the average part cost – which is typically present in traditional manufacturing, is also relevant to AM methods. This is indicated by the unit cost curve in Figure 2.20, which reduces dramatically and eventually plateaus as the optimal production volume is reached (i.e. machine utilisation is maximised). Each local peak in the curve represents a newly added part resulting in additional layers or an additional machine being used, followed by the subsequent cost decrease as more parts are added and the overall build cost is amortized.

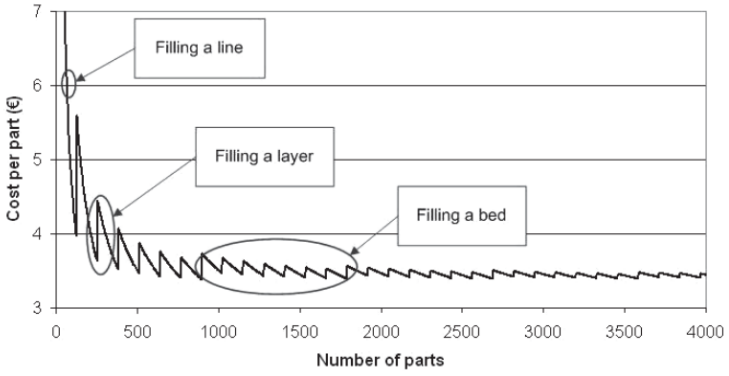


FIGURE 2.20: Unit cost vs. production volume. Taken from Ruffo et al. (2006).

Throughout the literature there is a consensus that machine cost tends to be the largest cost component; although the exact allocation varies significantly it can be as high as 90% (Atzeni and Salmi, 2012). This observation explains why machine utilisation and economies of scale play such a significant role in AM, as demonstrated by the findings of Ruffo et al. (2006), as well as other more recent research; a study by

Baumers et al. (2013) shows the unit cost of a partially full machine can be reduced by 25% by increasing machine utility to 92.6%; Rickenbacher et al. (2013) demonstrates unit cost a reduction of up to 90% for medium-size parts by replacing a single part with a batch size of five; Lindemann et al. (2012) observes a reduction in the unit cost of an automotive component from about 125 euros for a single part, down to about 10 euros for a batch of 190 parts.

However, it should be noted that most of the above results were accomplished with relatively small parts, while increasing the production volume of large parts via AM has very little effect on unit cost (Schröder et al., 2015). Schröder et al. (2015) suggests a way to overcome this is to fill machines with different components of various sizes, as done by Rickenbacher et al. (2013) and Baumers et al. (2013). This is also a common practice used by RP and AM service providers to maximise machine utilisation for low volumes of parts (Ruffo et al., 2006).

2.3.3 AM Build Time

It is evident that AM build time is both a driver of cost and an important objective in its own right. In general, AM build time is the sum of material scanning time, tool repositioning time and delay time in-between the deposition of each layer. Since the structural quality of support structure is less critical than that of the part, the scan rate of support structure can be increased to reduce build time. Thus, the scan time of the part and its support structure are often calculated separately. Di Angelo and Di Stefano (2011) proposed the following generic equation for AM build time:

$$T_{build} = T_{c-part} + T_{h-part} + T_{rep-part} + T_{c-sup} + T_{h-sup} + T_{rep-sup} + T_{delay} \quad (2.5)$$

where T_{c-part} is the total time to scan the contour of each layer in the part, T_{h-part} is the total time to hatch each layer in the part and $T_{part-rep}$ is the time to reposition the tool in between scanning. T_{c-sup}, T_{h-sup} and $T_{rep-sup}$ denote contour scan, hatching and repositioning time for support structure, respectively, and T_{delay} denotes the total unproductive time in between layers. In the case of SLM, T_{delay} represents the time taken to distribute a fresh layer of powder before it is scanned and $T_{rep-part}$ and $T_{rep-sup}$ can be considered negligible.

While a number of build time models have been proposed for different AM processes, the methods behind these models are generally similar and can be classified as either parametric or detailed analysis approaches (Di Angelo and Di Stefano, 2011).

The detailed model approach calculates the build time as an analytical function of the total distance travelled by the laser and the idling time during layer deposition. The laser path can be calculated by slicing the STL geometry and determining the topology of each layer, from which the contouring and hatching laser path and time can be deduced. Thus, such approaches must be able to accurately slice and compute the topology of each layer, or alternatively must rely on commercial slicing

and scan-path software that usually comes with AM machines. These approaches typically provides very detailed and accurate estimates of build time, but are much more cumbersome and computationally expensive than parametric methods. Also, most such methods assume a simple raster hatching pattern, and are not directly applicable to other scan patterns, such as the island scan strategy shown in Section 2.1.2. Examples of detailed build time models include Chen and Sullivan (1996); Giannatsis et al. (2001); Choi and Samavedam (2002) who provided detailed build time estimators for SLA.

Conversely, the parametric approach is generally lower in accuracy but offers a simpler and much faster solution. One of the simplest parametric relationships has been proposed by Cheng et al. (1995), who modelled the build time of SLA as a linear function of part height. Xu et al. (1999) proposed a slightly more sophisticated approach, which considers the average area of layers in the part as well as part height. To account for geometric complexity, Ruffo et al. (2007) considered the dimensions of the part's bounding box, in addition to height and volume. The model estimates the build time of SLS as the sum of total laser scanning time, total powder recoating time and total warm-up and cooling time before and after the build, respectively. Each of these three elements were linked to the input parameters using a statistical curve-fit approach, achieving an average predictive error of 15%. An alternative empirical approach was proposed by Zhang et al. (2015), who attempted to find a direct relationship between bounding box dimensions, volume and height of a part and its build time via linear regression. To improve the fidelity of this model its output is integrated with that of a differential model. The outputs of the two models are weighted based on their respective values of standard deviation and error.

Finally, several studies have proposed the use of ANN. Munguía et al. (2009) used part height, part volume and bounding box volume as inputs, though their findings indicated very low correlation with the latter. Di Angelo and Di Stefano (2011) used a number of geometry-based measures such as part volume, support structure volume, build height and layer thickness as ANN inputs. While these models were able to achieve reasonable predictive errors for their tested data sets (maximum errors of 10% and 16%, respectively), the main downsides of using ANN include the need for large data sets and a lack of transparency, as well as the need to re-train the model for different data sets.

2.4 Conclusion

This chapter provides a critical review of existing literature relevant to this thesis. Common SLM material and build quality challenges are discussed and include porosity; residual stresses which can lead to cracking and distortion; mechanical and microstructural anisotropy; surface roughness and dimensional error. High porosity and surface roughness result in poor fatigue performance, while anisotropy and residual stresses can lead to non-uniform mechanical properties, further deteriorating

mechanical performance in a particular direction. The exact nature of this anisotropy depends strongly on a combination of material properties, powder quality, laser- and scan-related parameters.

For most industrial alloys it is currently impossible to eliminate residual stresses completely, meaning that support structure is required not only to support overhangs but also to prevent thermal distortion, Meanwhile, some variable roughness is also bound to be present due to a number of geometrical (e.g. stair-step error) and thermal effects (e.g. melt-pool rippling) as discussed in Section 2.1.4. Consequently, post-processing operations, such as support structure removal, surface finishing and heat treatment are typically inevitable. All of these operations add significant cost and time, as a well as additional risk of part damage.

The problem is made worse by the fact that many of the above defects depend on the same process parameters such as laser power and scan velocity, scan pattern and layer thickness, and thus, act as competing objectives. For example, many alloys can now be produced with over 99% density, however, this requires high laser power and scan speed, which comes at the cost of even greater residual stresses and surface roughness.

Furthermore, optimising process parameters separately is likely to produce sub-optimal results, as they have a combined effect on SLM and are often interdependent. For example, reducing layer thickness can reduce the stair-step error, however, laser power and speed must be adjusted accordingly in order to prevent heat build-up and severe distortion, while the increased number of layers may also increase porosity and production time. Similarly, placing more support structure on the part can improve heat dissipation and reduce risk of distortion; however, it also increases cost, surface roughness and required post-processing. In addition, both support structure and the bin packing solution are strongly dependent on build orientation; thus, if the build orientation is changed both the bin packing solution and support structure geometry must be updated.

As shown in the literature, build orientation is closely related to build cost and time, surface roughness and post-processing, anisotropy and the residual stress profile. While build orientation in itself cannot change the inherent properties of SLM, it can be optimised for a specific part and its design requirements in a way that minimises any adverse effects of these properties (e.g. by placing critical surfaces and features in a favourable position, or by reducing the total number of layers in the part). As discussed in Section 2.1.2, the cross-sectional area, build height, support structure, and thus, the build orientation of a part have a significant effect on its residual stress profile. This is further supported by the work of Peng et al. (2017), who showed that thermal distortion can be reduced significantly with the right build orientation. Despite this, further research is required to explore and exploit this relationship in further detail.

Cost is clearly a key driver in manufacturing. However, as shown in Section 2.2.2, existing methods for optimising build orientation focus predominantly on surface

roughness, support structure and build time. Many of these works aim to reduce cost indirectly (e.g. by reducing post-processing and waste material), but relatively few researchers address cost explicitly.

This has several downsides. Firstly, minimising parameters such as build time or support structure alone, particularly in a simplified model of height and supported surface area, does not provide the full picture. For example, a solution optimised for support structure may minimise the cost of waste material, but result in greater production cost overall, due to longer build time, or vice versa.

Secondly, most of the proposed multi-objective approaches rely on the user to rank or weight the criteria, which can be misleading and difficult to do in practice. For example, when considering build cost and support structure volume as competing objectives, it is not intuitive or immediately obvious how they should be weighted, since build cost is also a function of support structure volume. Moreover, support structure volume and support structure removal are two different sources of build and post-processing cost, respectively; this is difficult to represent through a user-specified weighting of cost and support structure. Another approach is to use a utopian or goal solution, however, a problem formulated in this way can be very difficult to solve in practice, as it relies on a "sensible" set of goals, which are difficult to know in advance.

Thirdly, optimising the build orientation for surface roughness can reduce the cost and time of post-processing, but only for a local finishing process such as manual polishing or machining. If on the other hand, parts are finished via a global method such as barrel tumbling, the optimised build orientation becomes redundant, since the time and cost of global surface finishing is independent of local surface roughness variations.

Finally, as discussed in Section 2.3, machine utilisation is also strongly related to the cost-effectiveness and productivity of SLM. Since this parameter cannot be solved without first solving build orientation, many works discussed in Section 2.2.3 consider the two parameters in two separate stages. This is a very restrictive approach, since build orientations cannot be changed once they have been selected. However, allowing build orientations to be updated during bin packing may significantly improve the overall solution. For instance, if a set of parts, where the build orientation of each part is optimised for minimum build height, cannot be placed into a single bin, it is logical to rotate one or several of the parts in a way that increases their height but allows sufficient space for all parts to be packed. Furthermore, only the tallest part determines the overall build height (and thus build time) of the whole batch, regardless of other parts in that batch.

It is clear that reducing the number of bins is another important objective in cost driver in the bin packing problem. Since existing methods only consider a small enough set of parts or a subset of parts to fit into a single bin, this problem is yet to be addressed. Although Zhang et al. (2015) have alluded to this problem, by grouping a large set of parts into several subsets based on similarity attributes such as height and size, this method has two key limitations. Firstly, similarity may not be the best

criterion, as it can have an adverse effect on bin utilisation and cost; secondly, this approach does not solve the actual bin packing problem; thus, assuming that each subset can fit into a single bin without checking this assumption. Moreover, their approach aims to select an optimal build orientation for each part in the set (also based on similarity), but as with other works, these build orientations are fixed in the final solution, with no way to update them during the subsequent bin packing problem.

Chapter 3

Methodology

This chapter proposes a cost-driven modelling framework for optimising process parameters in SLM, namely build orientation and bin packing. Figure 3.1 also provides a more detailed overview of the rest of this thesis.

As shown in Chapter 2, selecting a good part build orientation for AM is an important and complex multi-criteria decision problem. Consequently, existing literature offers numerous competing objective functions, but lacks agreement on which is the most suitable one. Many existing approaches, as well as commercial software tools such as Magics, require the user to provide appropriate relative weights for the different objectives, which can be difficult to do in practice and can be misleading since many of the objectives are interdependent (e.g. cost, time, support structure).

Therefore, a key contribution of this chapter lies in proposing a single objective function for build orientation optimisation, which unifies various competing objectives, such as support structure, surface roughness and build height. Unlike the previously proposed WSM methods, this approach offers realistic coefficients of build and post-processing cost, thus, removing the need to "guess" objective weights and instead providing the most cost-effective trade-off solutions.

Additionally, no works have been found to solve the build orientation and bin packing problem simultaneously. Thus, the framework presented in this chapter addresses both of these parameters and shows how they are linked through cost. Moreover, Section 3.1.1 presents two different production scenarios for SLM, hence two distinct optimisation methodologies are proposed in chapters 4 and 6, respectively, in order to apply the framework to each respective scenario.

3.1 Process Planning Framework

The proposed process planning framework is described schematically in Figure 3.1. The diagram shows key steps and parameters in the framework, and provides section and chapter references throughout this thesis. First, the user must specify the part demand and provide a STL file in ASCII format. In order for the framework can calculate the post-processing cost resulting from surface finishing in IBP, the user must also specify a desired target surface finish, as shown in Figure 3.1. Section 3.3.1 and Appendix B outline a simple preliminary procedure, which enables the user

to define the target surface finish of local faces and features in the part. This step requires a CAD file and outputs a modified STL file, to be input into the framework.

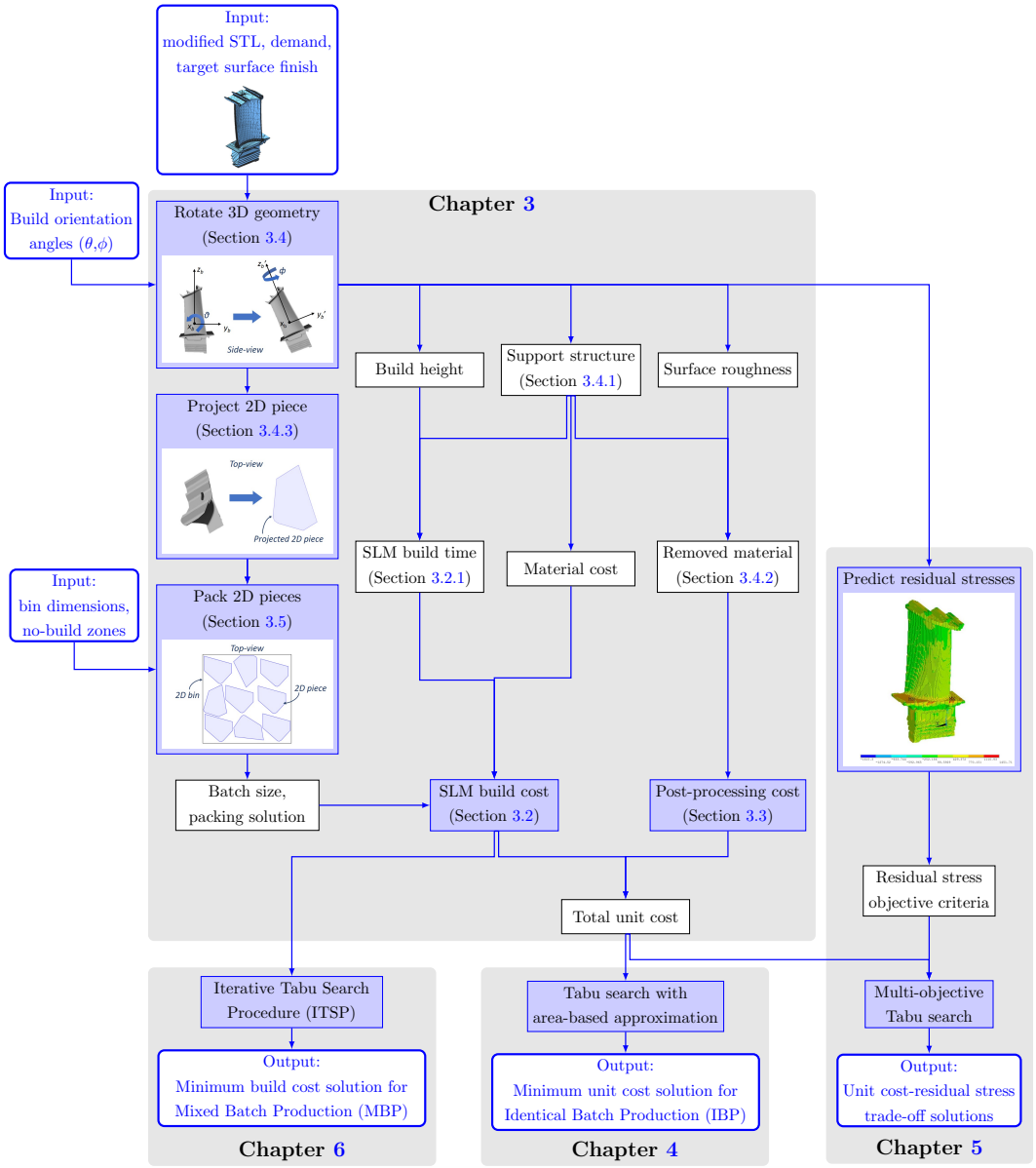


FIGURE 3.1: Flowchart of proposed cost-driven process planning framework. Key steps and sub-models are shown in blue boxes, while model inputs, outputs and key parameters are shown in white boxes. Grey boxes indicate how the framework is divided between chapters.

As discussed in Chapter 2, both discrete and continuous approaches have been proposed when searching for an optimal build orientation. Since the discrete approach is much more restrictive, the latter method is chosen in the proposed framework. Thus, part build orientation is treated as a continuous function of two angles of rotation, denoted by θ and ϕ.

Similarly to Zhang et al. (2016), vertical stacking and overlapping of parts is forbidden in the proposed framework, since it is likely to increase the complexity of support structure and the risk of damage to built parts due to residual stresses

and difficulty of part removal. This results in a 2D irregular bin packing problem (2DIBPP), where each part is represented by a 2D piece and the build plate of each machine is represented by a rectangular 2D bin (since most popular SLM machines feature a rectangular build plate). The 2D piece is a vertically projected bounding polygon of the part geometry, as shown by the box labelled "Project 2D piece" in Figure 3.1. It can be seen that the area and shape of the piece are determined by the build orientation, so both the piece and the packing solution must be updated for each new build orientation.

To prevent the build plate from moving or distorting during the SLM process, it is fixed down with bolts, typically in each corner of the build plate, as can be seen in Figure 3.3. These bolts are represented by four no-build zones in each 2D bin, where the size and position of each bolt can be specified by the user. Finally, to simplify the problem it is assumed that all bins are identical and that enough bins are available to accommodate the specified production demand. An optimal 2DIBPP solution places all required pieces into a minimum number of bins, where the number of pieces in each bin (i.e. batch size) is used to determine the SLM build cost associated with that bin.

3.1.1 Two Production Scenarios

The concept of "runners, repeaters and strangers" is a well-known lean manufacturing technique, where parts (or activities) are grouped into three categories based on their demand patterns (Reed, 2010). As shown in Figure 3.2, parts with low-variety high-volume (LVHV) demand are classed as "runners"; parts with high-variety low-volume (HVLV) demand are classed as "strangers"; while parts that fall in between LVHV and HVLV, with periodic or seasonal demands, are classed as "repeaters". Due to these demand characteristics, it is typical for runners to account for 80% of production volume and only 20% of manufacturing effort (i.e. jobs); while repeaters and strangers comprise the remaining 20% of production output and account for 80% of manufacturing jobs, due to their volatile demands and geometric oddities.

Custom implants and prosthetics in the medical industry, and replacement parts (including legacy components) in the transportation and defence industries, are typical examples of repeaters and strangers. In practice, the demand for aircraft spare parts is particularly difficult to forecast (Ghobbar, 2004; Liu et al., 2014), which often leads to overstocking, high inventory costs and risk of delays. As proposed by a number of researchers (Holmström et al., 2010; Khajavi et al., 2014; Thomas, 2016), SLM has the potential of significantly reducing the costs and effort expended on such parts, by producing them on demand relatively quickly and with minimal tooling. Moreover, as discussed in Chapter 1, mixed batch production (MBP) offers an economical way of meeting the low-volume, short-term demands of stranger and repeater parts, as demonstrated by most AM and RP specialists (Ruffo et al., 2006). Figure 3.3a provides an example of a MBP set-up.

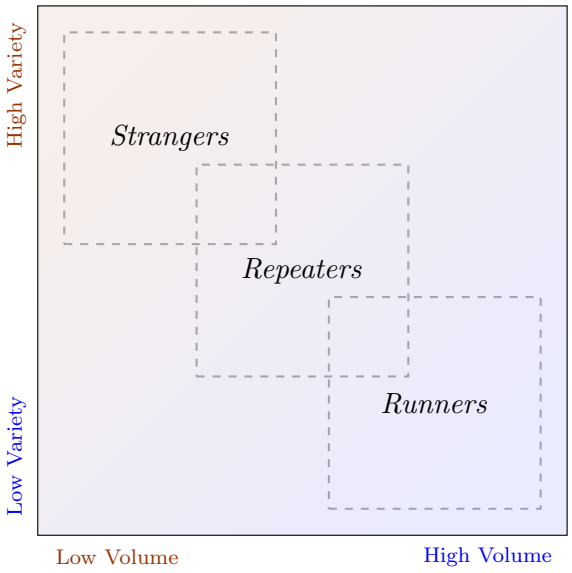
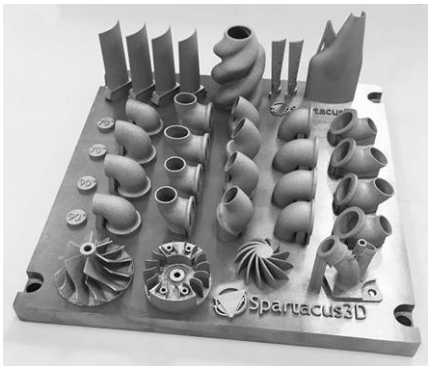
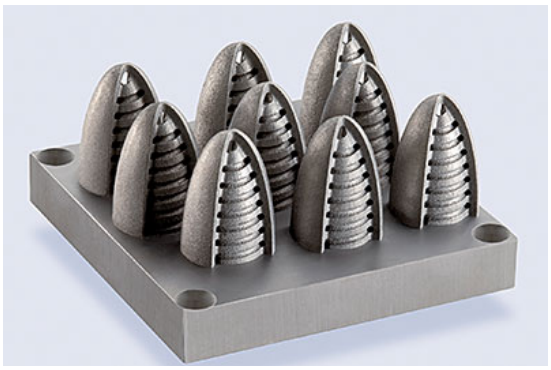


FIGURE 3.2: Product categorisation based on demand patterns.

Conversely, while MBP has obvious benefits for repeaters and strangers, these benefits do not necessarily apply to runners. Since the demand of runners is typically steady and predictable, mixing such parts with other heterogeneous geometries only increases the complexity and difficulty of planning and overseeing the production schedule, as well as tracking individual parts. For this reason, identical batch production (IBP) is suggested for building runner parts via SLM (Figure 3.3b). Existing literature (Atzeni and Salmi, 2012; López-Castro et al., 2017) suggests that the most suitable runners for IBP are high-complexity and high-value parts, which are redesigned to improve operational life and functionality or reduce manufacturing complexity, for instance, by reducing the part count of an assembly. The redesigned GE fuel injector nozzle is a good example, offering both a reduction in part count and improved performance (Kellner, 2014).



(A) Mixed batch production. Image taken from Farinia Group (2018).



(B) Identical batch production. Image taken from WJW Waterjet GmbH (2018).

FIGURE 3.3: Two production scenarios in SLM.

In summary, the proposed framework is applied to two different production scenarios: Identical Batch Production (IBP), which addresses steady, long-term demand of identical runner parts, and Mixed Batch Production (MBP), which addresses short-term, HVLV demand of different stranger and repeater parts. As shown in Figure 3.1, in Chapter 4 the IBP problem is optimised for total unit cost, which includes build and post-processing costs; while Chapter 5 extends this optimisation problem to include both unit cost and residual stress criteria. Similarly, the MBP optimisation problem is solved in Chapter 6. Because the MBP scenario presents a significantly more complex combinatorial problem than IBP, the proposed solution is based solely on build cost; this is explained further in the introduction of Chapter 6.

3.1.2 Build Orientation Constraint

When considering safety-critical parts the issue of certification must be considered. If two identical parts are built in two different build orientations, each part can be expected to exhibit different surface qualities, dimensional accuracy and mechanical properties (Ahn et al., 2007; Chlebus et al., 2011). Critical aircraft parts must go through a time-consuming and expensive certification process and meet stringent airworthiness requirements (Saha et al., 2013). If a change in the manufacturing process leads to different part properties, that part must be re-certified. Thus, even if the additional manufacturing variability is deemed acceptable by the user, the cost of certifying a part for each different build orientation is likely to outweigh any benefits.

Similarly, additional variability introduced during the production of instruments and tools (e.g. two moulds produced in two different build orientations), will translate into additional variability in the subsequent process that uses these tools, and thus, have an undesirable effect on the accuracy and performance of the final process.

It is also worth considering the possible benefits that may be gained from placing identical parts in different build orientations, provided these parts are non-critical. In the context of IBP, these benefits are likely to be negligible; assuming an optimal packing solution where all parts and all build orientations are identical, a single part cannot be rotated without either increasing the overall build height or decreasing the number of parts in the batch. Meanwhile, the additional manufacturing variability and complexity is still likely to be undesirable, even for non-critical parts.

This is not the case with MBP, where bins are allowed to contain various geometries of various sizes and so identical parts may be placed in different build orientations to improve machine utilisation, without increasing the build height or cost. By default, the proposed MBP optimisation heuristic imposes the build orientation constraint on all identical geometries; however, it can be easily bypassed by the user if needed as discussed in Chapter 6.

Finally, it should be noted that to fully meet this constraint, critical repeater parts which have been produced via SLM in the past, must be fixed in the build orientation that was used for the original order.

3.1.3 Limitations and Assumptions

Simplifying assumption, which have not already been mentioned, are outlined below:

1. Both IBP and MBP assume identical SLM machines and any variations between production time and quality of individual machines are considered negligible.
2. Material properties produced by each machine are consistent across the build plate and are not affected by the 2DIBPP solution.
3. Build quality is independent of the horizontal orientation of parts, hence, identical 2D pieces can be rotated in the horizontal plane during the 2DIBPP search.
4. A steady and consistent production flow is assumed. Only one order is dealt with at a time, and other orders which may compete with it are not considered.
5. Since delivery time constraints are outside the scope of this thesis, an infinite number of available bins can be assumed (in practice, each batch is built on the next machine that becomes available).
6. Powder can be fully recycled, and powder losses during extraction and cleaning are neglected.

3.2 SLM Build Cost Model

As discussed in Chapter 2, a number of models have been proposed to capture the build cost of ALM. This thesis uses the model proposed by Ruffo et al. (2006), as it provides a detailed outline of indirect costs and is suitable for capturing the cost of large scale production. Since support structure volume is a function of build orientation, the original equation is adapted to show support structure and part volume as two distinct material costs, as shown in Equation 3.1. Equation 3.1 outputs the cost of a single SLM build, c_b, assuming the machine is filled with n identical parts.

$$c_b = c_1 t_b + c_2 \rho (s_{tot} + v_{tot}) \tag{3.1}$$

where c_1 and c_2 are coefficients of indirect cost and material cost, respectively; t_b is the build time; ρ is the material density; and v and s denote the part volume and support structure volume, respectively.

3.2.1 SLM Build Time

To calculate the build time for a single bin, a linear regression model is proposed, which considers three components of build time, namely, total scan time of the part t_v, total scan time of the support structure t_s[1] and total powder recoat time t_r, as shown in Equation 3.2.

$$t_b = t_r + t_v + t_s + \beta_1 \tag{3.2}$$

$$\text{where} \quad t_r = \beta_2 h \quad t_v = \beta_3 v_{tot} \quad t_s = \beta_4 s_{tot}$$

As can be seen, t_v, t_s and t_r are approximated as linear functions of total part volume v_{tot}, total support structure volume s_{tot} build height h, respectively, where β_2 - β_4 are linear regression coefficients. To account for additional build time independent of the parts, for example, machine set-up, warm-up and cool-down time the constant coefficient β_1 is also introduced into Equation 3.2. The above equation can be re-written to provide Equation 3.3, which is used throughout this thesis.

$$t_b = \beta_1 + \beta_2 h + \beta_3 v_{tot} + \beta_4 s_{tot} \tag{3.3}$$

Two sets of data were used to calculate β_2, β_3 and β_4 for two different alloys – Ti-6Al-4V and CoCr, which are used in Chapter 4 and Chapter 6, respectively. The data has been provided by Rolls Royce plc and is presented in full in Appendix A.

[1] Support structure can be scanned more quickly since its quality is less critical than that of the part.

3.3 Post-Processing Cost Model

As outlined in Chapter 2, SLM post-processing typically consists of a number of steps, including the cleaning and separation of parts from the build plate, heat treatment, support structure removal and surface treatment.

While heat treatment is often necessary to relieve residual stresses and improve the ductility of parts produced via SLM, its cost is assumed to be largely independent of build orientation and batch packing[2]. Moreover, the cost of part cleaning is predominantly determined by user preferences, part geometry and functional requirements; therefore, it is not considered as a variable in this model. Finally, build plate removal time and cost is strongly dependent on the choice of removal method, and a multitude of process parameters, as well as the part's material properties. Thus, this cost is not modelled explicitly as a function of build orientation; instead a constant cost plate removal cost is estimated, based on wire-EDM cutting, and included into the indirect cost coefficient, c_1, as proposed by Baumers et al. (2013).

Consequently, the post-processing cost, shown in Equation 3.4, is treated as a function of support structure removal and required surface polishing. As discussed in Chapter 2, both of these factors have a clear functional relationship with build orientation (Townsend et al., 2016; Calignano, 2014).

$$c_{pp} = c_3 t_s + c_4 t_r \tag{3.4}$$

where t_s denotes surface finishing time and t_r denotes support structure removal time; c_3 and c_4 are time-dependent cost coefficients, which are determined by labour costs, chosen finishing and material removal methods, as well as the process parameters selected for those methods.

3.3.1 Defining Target Surface Roughness

Surface roughness is a complex parameter, which is typically defined by a combination of metrics, such as the size and distribution of microscopic peaks and troughs, surface waviness, and lay, which describes the patterns and direction of grains or cuts (in the case of subtractive methods) on the surface. By far the most common metric studied in SLM literature is the arithmetic mean roughness, or *Ra*, which is the arithmetic average of the absolute ordinate measurements of microscopic peaks and troughs within a given sampling length (Townsend et al., 2016). The main downside of this metric is that it is two-dimensional, and thus, may provide an incomplete or misleading representation of the surface topology. To address this, *Ra* should be measured several times, in different locations and orientations relative to the surface. By comparison, there is a lack of clear definition and available data for other metrics in SLM, such as area-based (i.e. 3D) roughness, waviness and lay. For this reason, only the *Ra* metric is used to define surface roughness throughout this thesis.

[2]However, the need for heat treatment may be reduced by minimising residual stresses which are addressed in Chapter 6.

The finishing time t_s in Equation 3.4 is assumed to be a function of the total difference between the target part surface roughness and the as-built surface roughness produced by SLM. The latter can be defined as a function of build orientation, as described in detail in Section 3.4.2.

The target surface roughness, denoted by Ra_t, must be provided by the user. Thus, the framework requires the user to choose one of the following options:

- Specify a local target value Ra_{t_f} for each facet f in the part's surface;
- Input a single global value of Ra_t directly into the model; or
- Omit Ra_t (for instance, if a global finishing method is being used), in which case the framework will only calculate the build cost.

In the case of the first option, a simple application has been developed using a CAD Application Programming Interface (API), which enables the user to conveniently and quickly input desired values of Ra_t for local faces and features of the CAD geometry. This application is described in more detail in Appendix B.

3.3.2 Surface Finishing Method and Time

As demonstrated by Gordon et al. (2016), there is a wide range of methods available for surface treatment. As mentioned in Chapter 2, these methods can be divided into two broad categories: global treatments and local treatments. The former category includes batch processes such as barrel tumbling and vibratory finishing, where the entire surface of each part in the batch is treated simultaneously and the relationship between the post-processing time and local variations in surface roughness produced by SLM can be considered negligible. The benefit of such approaches is relative speed and low cost, however, the shared downsides of these methods is the relative lack of control and resulting unwanted erosion of certain areas and features (e.g. corners and sharp features), limited media access to internal features and concavities, especially of complex parts. Also methods such as tumbling may not be suitable for fragile parts. Another global method is electrochemical polishing which can achieve a superior surface finish, but may produce a poorer, non-uniform finish inside deep concavities.

Common local approaches include manual or robotic grinding and Computer Numerical Control (CNC) machining. These are often slower than batch processes and unsuitable for complex internal structures, but offer a lot more precision and control over the resulting surface quality and dimensional accuracy of the part. Laser polishing is another promising local finishing approach, but is still relatively immature and difficult to control (Marimuthu et al., 2015).

Selecting the most effective post-processing method is a complex decision based on the part's geometry, requirements and cost. Currently, it is common practice to use several finishing steps, for instance, grit blasting followed by tumbling. If the entire surface of the part requires post-processing and the part does not contain any delicate

features, a global tumbling method may be used; while a part that contains complex internal concavities, may be more suitable for electrochemical polishing. However, for parts where only certain surfaces require finishing, or where a global method can have a negative impact on the mechanical performance and dimensional accuracy (for example, for a part with delicate features) of the part, a local method is more suitable. For the purpose of demonstrating the proposed framework, part surface finishing is assumed to be carried out in a single step using a medium-sized 5-axis CNC milling centre, since it is a local process capable of achieving $Ra < 1\,\mu m$.

The total time of CNC milling is modelled in Equation 3.5. Other local finishing processes can be substituted for CNC machining, by adjusting Equation 3.5 accordingly. As mentioned above, the post-processing time of global methods is not affected by local variations in surface roughness, and thus, can be considered unrelated to SLM build orientation. Therefore, while the cost of different global post-processing methods can be used to select the cheapest one, it will have no effect on the solution produced by the framework and optimisation heuristics described in this thesis.

$$t_s = \frac{v_s}{q} + \left\lceil \frac{v_s}{t_{tl} q} \right\rceil t_{load} \tag{3.5}$$

where q is the process material removal rate for surface finishing; v_s is the total volume of material removed during this step; t_{tl} is the expected tool life and t_{load} is the tool loading time, where the number of tool changes must be rounded up before multiplying by t_{load}. As can be seen, the above equation is a simplification of the real process, that assumes constant cutting speed, fixed tool life and fixed tool change time, and ignores all other factors such as machine set-up time and swarf removal.

3.3.3 Support Structure Removal

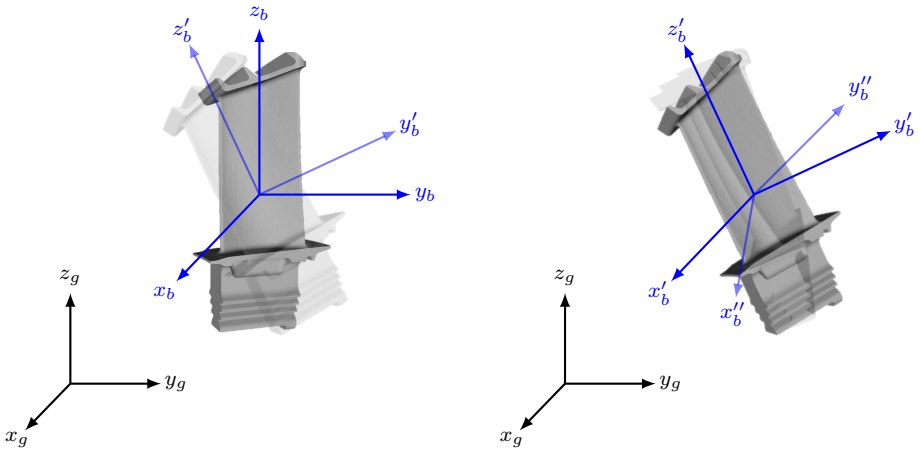
Support structures are typically removed manually, thus, the process is highly dependent on the skill and experience of the operator, as well as the size and complexity of the part geometry. A model of support structure removal time is proposed by Alexander et al. (1998), who argue that support struts that are attached to the part at both ends are more difficult to remove than struts that are attached to the build plate. This principle applies to all AM processes, thus, their model is adapted in Equation 3.6 to describe the time of manual support structure removal for SLM parts.

$$t_r = w_1 A_d + w_2 A_u \tag{3.6}$$

where A_d denotes the total area of down-facing part surfaces in contact with support structure; A_u denotes the total area of upward-facing part surfaces in contact with support structure; and w_1 and w_1 are material removal coefficients. A_d and A_u are calculated in Section 3.4.1.

3.4 Build Orientation Generation

As already mentioned, the STL file defines the surface of the 3D geometry using a set of triangular facets, denoted by F. Each facet $f \in F$ can be defined as $f(v_1, v_2, v_3, \hat{n})$ where v_1, v_2 and v_3 are three distinct vertices, and $\hat{n}$ is the unit normal vector of the facet pointing outward from the solid geometry; v_1, v_2, v_3 and $\hat{n}$ are each defined in 3D Cartesian coordinates. In the proposed model each build orientation is achieved by two consecutive rotations about the geometry's body-centric axes, as demonstrated in Figure 3.4, where the body-centric coordinates of the geometry are denoted by x_b, y_b and z_b, respectively. The first rotation is shown in Figure 3.4a, where the geometry is rotated about x_b; the second rotation is shown in Figure 3.4b, where the geometry is rotated about z'_b. The global axes are shown for reference and denoted by x_g, y_g and z_g, respectively.



(A) R_θ rotates geometry about x_b axis. (B) R_ϕ rotates geometry about z'_b axis.

FIGURE 3.4: Schematic description of build orientation generation.

Therefore, each build orientation can be defined as $p(\theta, \phi)$, where θ and ϕ are two rotation angles, and the rotated geometry is defined by a set of facets F'. Each rotated facet $f' \in F'$ is produced by applying the R_{xz} transformation to the three vertices v_1, v_2 and v_3, and the normal vector $\hat{n}$ of the original facet f, as shown in Equation 3.7.

$$f'(v'_1, v'_2, v'_3, \hat{n}') = R_{\theta\phi} f(v_1, v_2, v_3, \hat{n}) \tag{3.7}$$

where v'_1, v'_2, v'_3 and $\hat{n}'$ denote the rotated vertices and rotated normal vector of f, respectively. The matrix R_{xz} is defined in Equation 3.8.

$$R_{xz} = R_\phi R_\theta \tag{3.8}$$

$$\text{where} \quad R_\theta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \quad \text{and} \quad R_\phi = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Using this transformation, all unique build orientations can be produced within input boundaries of $\theta \in [0, 180]$ and $\phi \in [0, 360)$. It should be noted, that if a part geometry is rotated about the global z-axis, its key geometric properties, such as height and supported area, and therefore its cost, will remain unchanged. Thus, build orientations achieved by such rotations are considered to be duplicates of one another. For example, when using Equation 3.8 for any geometry, all build orientations corresponding to $p(0^o, ^*)$ and $p(180^o, ^*)$ can be considered duplicates of $p(0^o, 0^o)$ and $p(180^o, 0^o)$, respectively, where * denotes any arbitrary value of ϕ.

Once the geometry has been rotated, a number of variables, which are driven by the build orientation, are calculated and passed into the cost equation. These variables include support structure volume and contact-area, geometry height, and surface roughness. Additionally, the area and shape of the 2D piece is required to determine the batch size and number of bins required to meet the demand. Given a set of unique vertices V, which can be easily derived from F, the height of the geometry can be easily calculated by sorting the vertices in V by non-increasing z-coordinate and taking the difference of z_{max} and z_{min}. Support structure, surface roughness and 2D piece generation are discussed in the following sections.

3.4.1 Support Structure Prediction

Predicting support structure volume is a difficult task for a number of reasons. Firstly, determining the volumes created between the surfaces of the STL geometry is a non-trivial computational task. Secondly, there is no definitive set of guidelines and rules for designing support structures. Current commercial software relies on broad generic rules to generate support structure automatically, for instance, using an overhang threshold below which surfaces are supported; meanwhile, experienced users often resort to manually modelling a support structure which is tailored to a specific part.

For the purpose of demonstrating the proposed cost model, a simple rule is adopted for calculating the volume of support structure. As discussed in Chapter 2, this rule is commonly used in literature and states that any surface with a build angle of less than α_{sup} must be supported, where α_{sup} is a threshold value that typically lies between 30° and 45°. Figure 3.5 provides a 2D example for this support structure generation approach. The build angle of each facet $f \in F$ is calculated using Equation 3.9.

$$\alpha_f = \cos^{-1}(\text{-}\hat{z} \cdot \hat{n}_f) \tag{3.9}$$

where α_f and $\hat{n}_f$ denote the build angle and unit normal vector of f, respectively; and $\text{-}\hat{z}$ is a unit vector with coordinates (0, 0, -1).

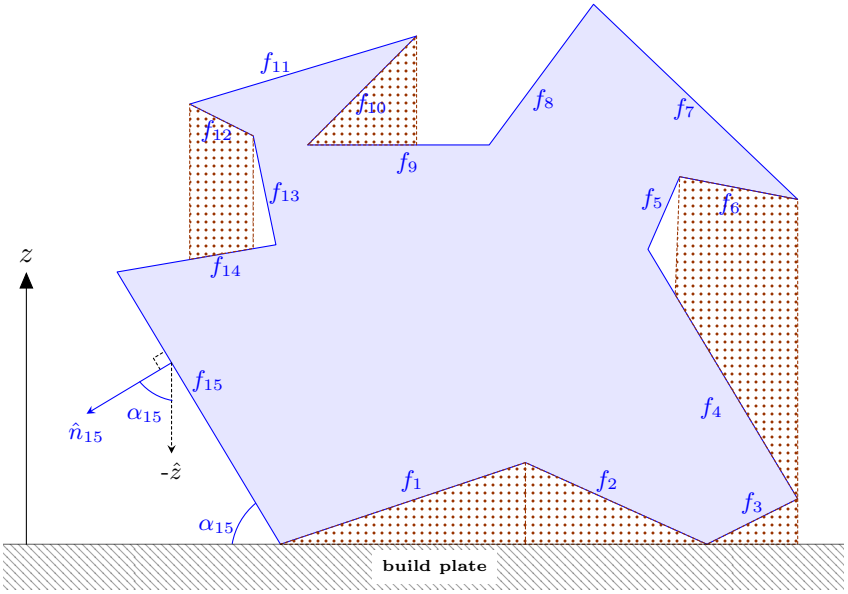


FIGURE 3.5: Example of support structure (red dots) for an arbitrary geometry (blue).

Thus, let f be defined as a supported facet with $\alpha_f < \alpha_{sup}$, and let F_S denote a list of all supported facets in the part geometry, where $F_S \subset F$. Also, let f be defined as an upward-facing facet if $\alpha_f > 90^o$, and let F_U denote a list of all upward-facing facets, where $F_U \subset F$. It is assumed that the top surface of the build plate is represented by a horizontal plane at z_{min}, where z_{min} is the z-coordinate of the lowest vertex in V.

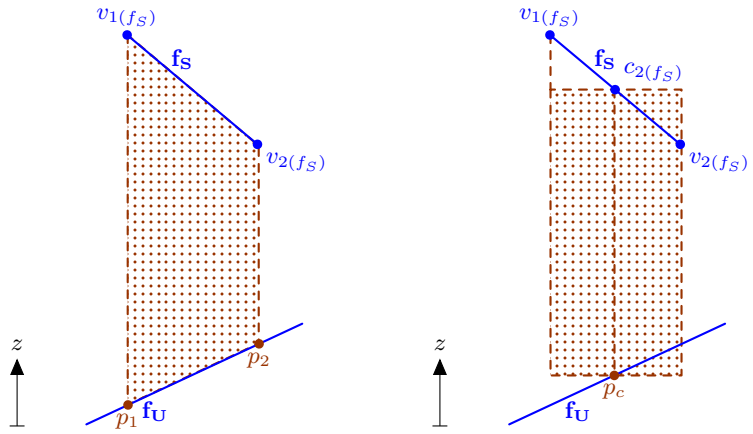
It can be imagined that each facet $f \in F_S$ is supported by a prism-shaped strut, and the total support structure volume is the sum of the volume of these struts. Each support strut is formed by three vertical line segments, where each line segment is drawn along the -$\hat{z}$ direction between a vertex of f and the surface of the build plate. Since these line segments may intersect the part geometry, an additional intersection check must be carried out between each edge of the support strut and each facet $f \in F_U$.

Given a vertex v_{f_S} belonging to a supported facet f_S, and an upward-facing facet, f_U, the intersection check consists of two steps. First, it determines whether a vertical line segment projected from v_{f_S} intersects the plane containing f_U; if this condition is true, the second step determines if the point of the line-plane intersection, denoted by p, lies inside f_U. If the latter is true, the edge of the support strut is updated to p; otherwise, the edge is extended all the way to the build plate.

Model Efficiency and Validation

It can be seen that the worst-case time complexity of the above support structure prediction approach is $O(3SU)$, where S denotes the size of F_S and U denotes the size of F_U. To improve the computational time, F_U is ordered into a tree structure, based on the location and bounding box of each facet, so that for a given vertex v_{f_S}

and facet $f \in F_U$, the intersection check is performed only if v_{f_S} lies inside the x-y bounding box of f, and f does not lie above v_{f_S}. To further improve the solution time, a simplified, low fidelity approach is proposed, which checks only the centre point of each supported facet, instead of its three vertices; this reduces the worst-case time complexity to $O(SU)$. The original high fidelity (HF) approach and the low fidelity (LF) version are summarised schematically in Figure 3.6.



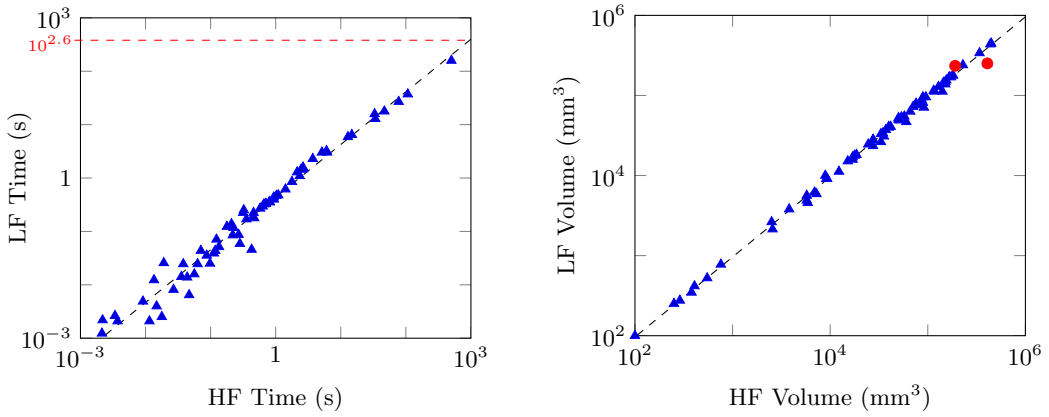
(A) High fidelity approach. (B) Low fidelity approach.

FIGURE 3.6: Schematic comparison of high fidelity and low fidelity modelling of a single support strut in 2D (red dots). f_S denotes the supported facet; f_U denotes an upward-facing facet; v_{1f_S}, v_{2f_S} and c_{f_S} denote two vertices and the centre point of f_S, respectively; p_1, p_2 and p_c denote intersection points corresponding to v_{1f_S}, v_{2f_S} and c_{f_S}, respectively.

The LF and HF approaches were tested on 68 geometries of various sizes, where each geometry was placed in a randomly selected build orientation, and α_{sup} was set to 45°. Figure 3.7 shows a comparison of the computational time and accuracy of each approach. The two worst errors produced by the low fidelity model had values of 38% and -24%, respectively, and are marked by red circles in Figure 3.7b. However, 62 of the 68 data points had errors below 10%, while the whole data set yielded a root-mean-square error (RMSE) of 5% (normalised over the range of data). Moreover, the low fidelity model showed an average reduction in computational time by a factor of 3 (taken from the gradient of the line of best fit), as expected.

It should also be noted that the high fidelity approach can only return the correct support strut, if all three edges of the strut intersect the same upward-facing facet, as shown in Figure 3.6a; if this is not the case the model will also introduce some error, despite the additional computations, as shown in Figure 3.8. Based on these observations, the low fidelity model is chosen over the high fidelity model and used throughout this thesis.

The support structure volume predicted by the adopted low fidelity model has been validated against support structure produced by a commercial software, Magics v20.03. The same 68 test instances and settings were used to compare the proposed model and Magics; the results are shown in Figure 3.9. Despite producing an over 30% error in three test instances, with 48% being the worst, the model produced an



(A) Computational time of low fidelity (LF) vs. high fidelity (HF) approach.

(B) Support structure volume predicted by low fidelity (LF) vs. high fidelity (HF) approach.

FIGURE 3.7: Performance comparison between the low fidelity and high fidelity models of support structure, shown on a logarithmic scale.

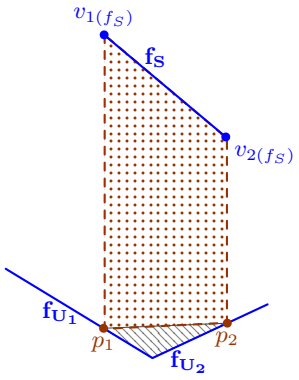


FIGURE 3.8: Example of support strut error (grey lines) produced by the HF approach.

average error of 7% and RSME of 3% for the whole data set, indicating generally good predictive accuracy in the majority of cases. The occasionally large errors are likely a result of the intersection error discussed above, and demonstrated in Figure 3.8.

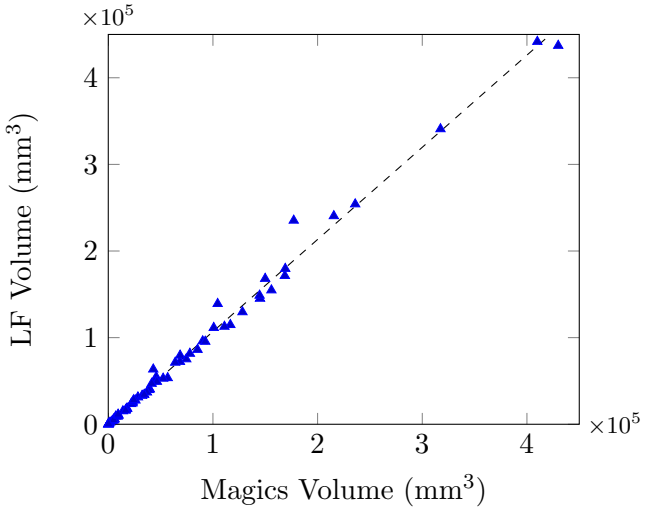
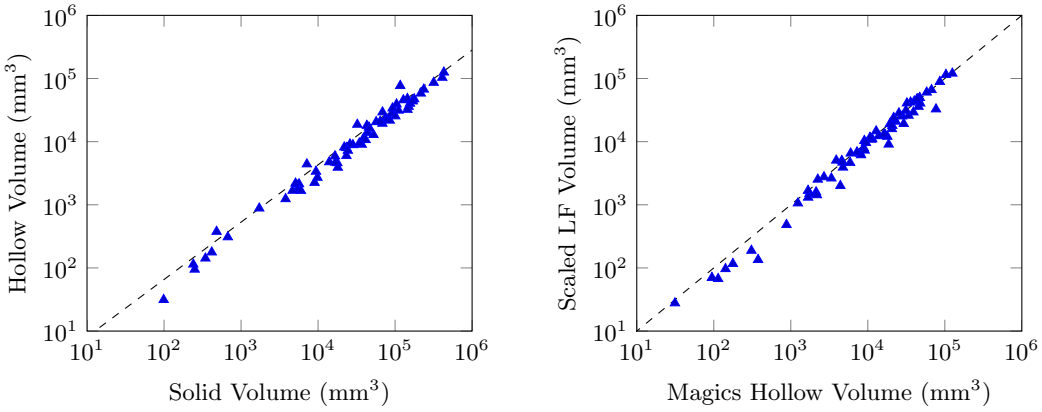


FIGURE 3.9: Comparison of support structure volume predicted by the low fidelity model vs. Magics software, shown on a logarithmic scale.

Finally, it should be noted that the proposed model assumes fully solid support structure, whereas hollow and lattice-shaped support structures are often used in practice to reduce waste material. Other features, such as contact teeth and offsets around edges, are also not considered. A simple and quick way to approximate lattice-shaped support structure is to apply a density fraction to the calculated solid support volume. This idea was tested by creating and comparing both hollow and solid support structures in Magics, using the same 68 geometries and the same build orientations, as above. Figure 3.10a shows a regression line for the two types of support structure, with a R^2 of 0.93, RMSE of 5.47%, and a gradient of 0.28. Figure 3.10b shows the support structure volume predicted by the LF model and scaled by a factor of 0.28, compared against hollow support structure created by Magics. Due to its simplicity, this approach can only provide an approximation of hollow supports; the maximum predictive error of data shown in Figure 3.10b is 61.0%, with a RMSE and mean absolute error of 5.96% and 16.5%, respectively. Despite these errors, the proposed LF model of support structure volume is still a useful tool for comparing and selecting different build orientations, while the suggested scaling factor of 0.28 provides a more realistic weighting of support structure volume in the proposed cost equation, and resulting objective functions presented in chapters 4 and 5.

Support Structure Removal

To find the time of support structure removal, the contact area between the support structure and the upward-facing and downward-facing facets, denoted by A_u and A_d, respectively, must be known. Following the procedure described above, A_d and A_u can be easily calculated by summing up the areas of relevant facets, as shown in equations 3.10 and 3.11, respectively.



(A) Volume of hollow vs. solid support structure predicted by Magics.

(B) Magics hollow support structure volume vs. scaled volume predicted by LF model.

FIGURE 3.10: Prediction of hollow support structure volume, shown on a logarithmic scale.

$$A_d = \sum_{f=1}^{F_S} a_f \tag{3.10}$$

$$A_u = \sum_{f=1}^{F_U} \sigma_f a_f \tag{3.11}$$

$$\text{where} \quad \sigma_f = \begin{cases} 1 & \text{if } f \text{ is intersected by a projected support strut} \\ 0 & \text{otherwise} \end{cases}$$

where F_S denotes a list of supported facets; F_U denotes a list of upward-facing facets and a_f is the area of facet f.

For clarity, Figure 3.11 provides a simplified 2D example, where the total support structure contact area is calculated as the sum of the supported facet f_S and its underlying upward-facing facet f_U, which have areas A_d and A_u, respectively. As can be seen, a tiny offset is assumed between the support strut and the adjacent vertical surface of the part, which is assumed to be support-free.

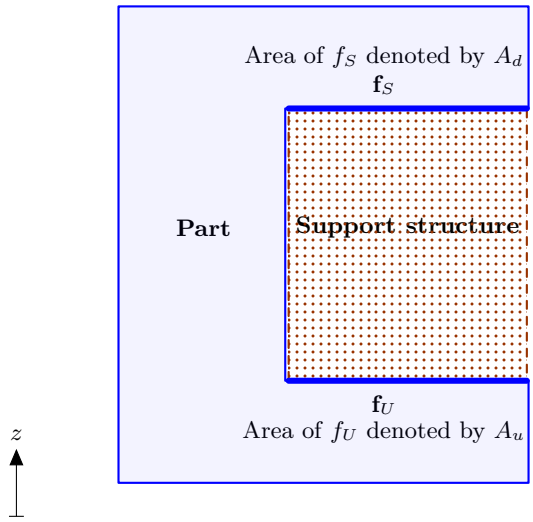


FIGURE 3.11: Schematic example of supported facet, f_S, its underlying upward-facing facet, f_U, and their corresponding areas (highlighted by thick blue lines), A_d and A_u, respectively.

3.4.2 Removed Material Prediction

As described in Section 3.3, the post-processing time and cost of SLM is modelled as a function of total volume of removed material, v_{SF}. In order to calculate v_{SF}, the original surface roughness, denoted by Ra_f, for each facet $f \in F$ of the part geometry must be known. Several works in literature (Cooper et al., 2012; Strano et al., 2013b; Triantaphyllou et al., 2015) have reported experimental data that relates the surface build angle, denoted by α, and surface roughness, Ra. These data vary significantly due to differences in laser parameters, layer thickness, materials, powder properties and SLM machines. A thorough investigation of surface roughness produced for Ti-6Al-4V alloy was carried out by Vandenbroucke and Kruth (2007); their results are shown in Figure 3.12. Unlike most of the above studies which provide only a few sampling points (e.g. at 45^o increments), Vandenbroucke and Kruth (2007) cover the whole range of α and provide a relatively large data set of 18 sampling points (within 15^o increments or less). For this reason, the data in Figure 3.12 are used to interpolate surface roughness in this thesis.

As proposed by Ahn et al. (2007), for a build angle α^* that falls in between the sampling points, the surface roughness value, denoted by Ra^*, can be interpolated using Equation 3.12.

$$Ra^* = Ra^f + \frac{\alpha^*(Ra^c - Ra^f)}{\alpha^c - \alpha^f} \tag{3.12}$$

[3] A layer thickness of 30 µm is assumed for Ti-6Al-4V throughout this thesis. The sharp peak in Ra on the top-facing surface (i.e. $\alpha > 150^o$) can be reduced significantly (to below 20 µm) by using a layer thickness of 20 µm, leading to smaller and more stable melt pools (Vandenbroucke and Kruth, 2007).

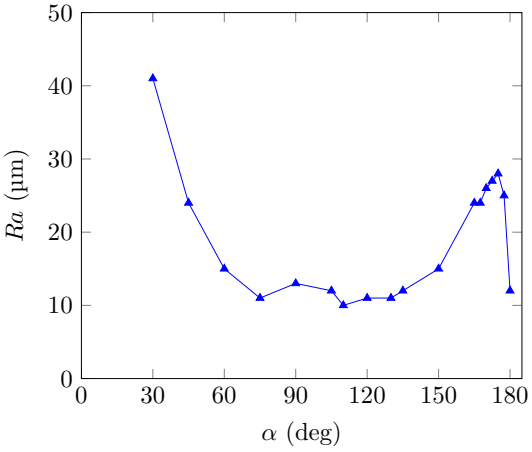


FIGURE 3.12: Surface roughness of Ti-6Al-4V alloy produced by an M3 Concept Laser machine with a layer thickness of 30 µm[3]; adapted from Vandenbroucke and Kruth (2007).

where α^f and α^c are the floor and ceiling values of the interval containing α^*, respectively; and Ra^f and Ra^c are the roughness values corresponding to α^f and α^c, respectively.

The data in Figure 3.12 accounts for process characteristics such as the stair-stepping effect, melt-pool rippling and additional powder particles fusing to downward-facing surfaces. However, another key source of surface roughness is support structure, which leaves marks and protrusions after it is removed. As can be seen, the roughness of supported surfaces (where $\alpha < 30^o$) was not reported.

As discussed in the previous section, the threshold build angle for support-free surfaces, denoted by α_{sup}, typically lies between 30^o and 45^o. Although it is still possible to successfully build support-free surfaces within this range, a significant deterioration in the build accuracy and surface finish can be expected, as indicated by the sharp rise in Ra for $\alpha < 45^o$ shown in Figure 3.12. Conversely, supporting these surfaces can actually improve the overall surface finish and build accuracy, despite the residual marks and protrusions, due to the additional structural support and heat dissipation (Calignano, 2014).

Thus, for each support-free facet $f \in F$, where $\alpha_{sup} \leq \alpha_f \leq 180^o$, the surface roughness of f, denoted by Ra_f, is determined using Equation 3.12 and the data in Figure 3.12. Meanwhile, for each supported facet f_S, where $0 < \alpha_{f_S} < \alpha_{sup}$, the surface roughness Ra_{f_S} is set to Ra_{sup}, which is a constant value denoting the roughness of supported surfaces.

It is assumed that all parts are placed directly onto the build plate (i.e. without base support structure[4]), and that the parts are removed from the build plate using wire-EDM (Section 3.3). Thus, for any facet $f \in F$, where $\alpha_f = 0^o$, f is considered to be in full contact with the build plate, and Ra_f is set to Ra_{rem}, which denotes a constant surface roughness resulting from part removal via wire-EDM.

[4] Additional support structure placed between the base of the part and the build plate, in order to improve build quality and facilitate part separation from the build plate.

Values of Ra_{sup} and Ra_{rem} can be specified by the user. The roughness of supported surfaces produced in Ti-6Al-4V alloy has been reported (Cooper et al., 2012; Triantaphyllou et al., 2015) to range between 15 µm and 32 µm; while typical surface roughness produced by wire-EDM falls between 1 µm and 5 µm.

Once the roughness Ra_f of each facet $f \in F$ is known, the total removed material, v_s, can be calculated using Equation 3.13.

$$v_s = \sum_f \kappa a_f (Ra_f - Ra_{t_f}) \tag{3.13}$$

$$\text{where} \quad \kappa = \begin{cases} 1 & \text{if} \quad Ra_f > Ra_{t_f} \\ 0 & \text{otherwise} \end{cases}$$

where Ra_f denotes the SLM surface roughness of facet f, a_f is the area of f and Ra_{t_f} is the target surface roughness of f, which is defined by the user as described in Section 3.3.1.

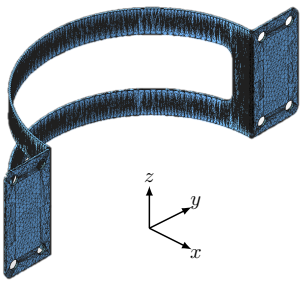
3.4.3 2D Piece Generation

The 2D piece represents the 3D part geometry in the 2D bin packing problem. Since overlapping and stacking of parts is avoided in this model, the piece can be though of as a shadow of the part on the build plate. In this model, the 2D piece is created in four stages, which are outlined in Figure 3.13.

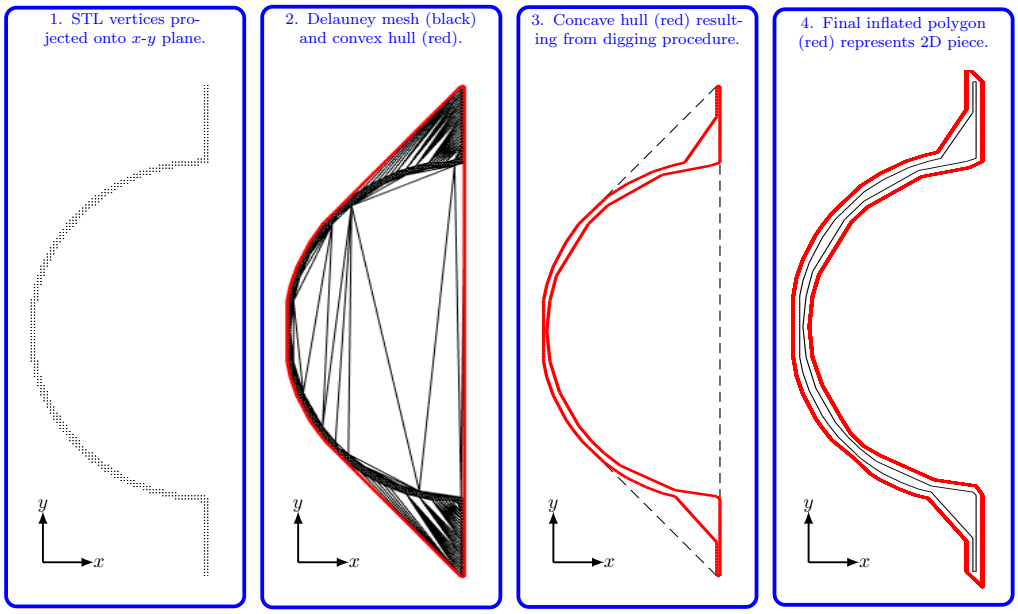
Using the set of unique vertices V, each vertex $v \in V$ is projected vertically onto the x-y plane, such that $v_{3D}(x, y, z) \leftarrow v_{2D}(x, y)$. The resulting set of 2D points is pruned for duplicates, and a Delaunay triangulation is performed (Shewchuck, 2002).

Each edge in the resulting triangular mesh can either be defined as an internal edge, if it is shared by two triangles, or an outer edge, if the edge belongs to only one triangle. The convex hull of the projected points is defined by these outer edges. In some cases the convex hull can be significantly larger than the actual outline of the geometry, as demonstrated by the example in Figure 3.13. To get the concave hull, a boundary digging procedure is adopted from Duckham et al. (2008), which removes outer edges exceeding a specified maximum length. The procedure is described as 'digging', because it can be imagined to physically remove an outer edge from the mesh, leaving a small concavity and two exposed internal edges. Figure 3.14 provides a pseudo-code of this procedure, where $List_{oe}$ denotes the list of outer edges and L_{max} denotes the maximum edge length.

When using this procedure, care must be taken to avoid removing too many edges and creating a degenerate or discontinuous polygon. To do this, the algorithm in Figure 3.14 checks the status of the newly exposed pair of edges, e_1 and e_2; if one of these edges is already classed as an outer edge, the process stops and returns the most recent iteration of the list of polygon vertices, $List_{oe}$. Furthermore, a suitable value of L_{max} must be used. Preliminary testing showed this value to be around 20 mm, where the algorithm managed to sufficiently capture large concavities without digging



(A) STL file of bracket.



(B) Stages in the 2D piece generation procedure.

FIGURE 3.13: The 2D piece generation procedure tested on a bracket, with a maximum edge limit of 20 mm (stage 3), and an inflation offset of 4 mm (stage 4).

too far. Removing edges below this limit increases the risk of creating degenerate or multiple disconnected polygons, and causing numerical errors in the no-fit polygon procedure during bin packing (Lindmark, 2013), as well as increasing computational cost. More importantly, capturing very small concavities is unlikely to provide a significant improvement in the 2DBPP solution, as very few parts would fit in such a small space.

It should be noted that closed internal cavities are not captured by this procedure. While internal cavities can affect the 2DBPP solution in certain cases (for instance, if very small parts can be successfully placed inside a large annulus), such cases are considered rare, and thus, are left for future research.

Finally, to avoid part damage during the SLM process and to facilitate removal of built parts from the SLM build substrate, a small gap must be maintained between the parts when they are placed on the build plate. To maintain a minimum distance d_o between parts packed in SLM, the 2D piece of each part can be inflated by $\frac{1}{2}d_o$.

The process is shown schematically in Figure 3.15. Vertices in the concave polygon

Input: $List_{oe}, L_{max}$

Result: updated $List_{oe}$

Sort $List_{oe}$ by non-increasing length;
Set $e_0 \leftarrow$ first edge in $List_{oe}$;
while $e.Length > L_{max}$ **do**
    Find triangle T, which contains the edges e_0, e_1 and e_2;
    if $e_1, e_2 = inner\_edge$ **then**
        Remove e_0 from $List_{oe}$ and add e_1, e_2 to $List_{oe}$;
    end
    else
        Terminate procedure;
    end
    Sort $List_{oe}$ by non-increasing length;
    Set $e \leftarrow$ first edge in $List_{oe}$;
end

FIGURE 3.14: Pseudo-code of polygon boundary digging procedure.

(produced in stage 3) are ordered in an anti-clockwise order, thus, the polygon can be described by a set of edges, where each edge contains two vertices and a direction. To inflate the polygon, the vertices of each edge can be translated by a vector, with magnitude $\frac{1}{2}d_o$ and direction perpendicular to the edge. The resulting polygon is likely to contain edges which intersect or form a gap. To fix this, the vertex shared by each pair of edges is replaced with an intersection point, computed for the two vectors corresponding to these edges, as shown in Figure 3.15.

3.5 2D Irregular Bin Packing Problem (2DIBPP)

From the above procedure, it follows that the resulting 2DIBPP is likely to contain highly irregular pieces, often with concavities. In order to exploit these concavities and find high quality bin packing solutions, continuous and unrestricted rotation of pieces in the $x - y$ plane should be permitted. Moreover, as discussed in Section 3.1 the framework assumes an infinite number of available machines, containing identical rectangular build platforms. Thus, the method of Martinez-Sykora et al. (2017) is used in this thesis, which addresses the 2DIBPP, characterised by irregular concave pieces, continuous piece rotation and the presence of multiple bins.

The aim of this procedure is to pack a given set of 2D irregular pieces, denoted by P, into a set of bins, such that the utilisation of each bin is maximised. In the case of MBP, the algorithm consists of the following three steps. The first step tries to assign all pieces to a minimum number of bins by solving a 1D bin packing problem, where each piece is represented by its area only.

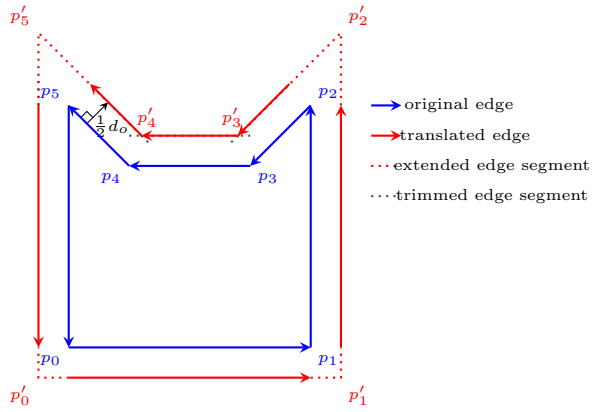
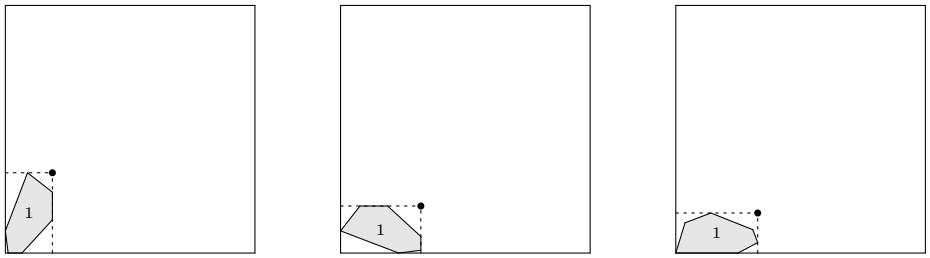


FIGURE 3.15: Schematic description of polygon inflation.

The second step tries to place each potential subset of pieces into its corresponding bin. In this step, the subset of pieces is sorted by non-increasing area and each piece is placed sequentially into the bin. A mixed integer programming (MIP) model is used to evaluate different rotational positions of each piece, by selecting a rotation which minimises the overall bounding box dimensions of all placed pieces in the bin, as shown in Figure 3.16. If all pieces in the current subset are packed successfully, the subset is removed from P, the successful bin is saved to the final solution and the algorithm moves onto the next subset of pieces until all pieces in P have been packed. Otherwise, the algorithm proceeds to the third step.



(A) Rotation 1 (B) Rotation 2 (C) Rotation 3

FIGURE 3.16: Three potential rotations for the first piece in the bin, where Rotation 3 is considered the most favourable. Taken from Martinez-Sykora et al. (2017).

It is clear that the first step in the procedure is likely to produce very ambitious and sometimes infeasible piece subsets. Thus, the third and final step is designed to mend the assignment of pieces to bins, should the second step fail to place one of the pieces in a given subset. This is done by sorting all remaining untried pieces in P and attempting to place each piece sequentially into the partially packed current bin. Once all pieces have been tried, the bin is saved to the final solution and the packed pieces are removed from P. Because this step disrupts the original bin assignment, the procedure must go back to the first step and solve the 1D bin packing problem again for all remaining pieces in P.

In the case of IBP, the problem is much simpler, since all pieces are identical and bin assignment is not required. As discussed in Chapter 4, only the batch size is required to find the unit cost of identical pieces, which can be found using only the second step of the procedure described above.

Finally, as described in Section 3.1, each bin must have four no-build zones which represent the bolts fixing the build plate to the SLM machine platform. As shown in Figure 3.17, these no-build zones are represented by four rectangular "dummy" pieces, which may not be overlapped by any of the real pieces placed inside the bin.

3.6 Implementation

The framework is implemented in C# using the Visual Studio 2012 development environment, and parallelised where appropriate. To solve the MIP problem for the 2DIBPP the commercial solver Gurobi (version 6.5.2) is used. All experimental tests throughout this thesis are done on the same i7-3820 four core machine, with 2.70 GHz maximum frequency, eight logical processors and 16GB of RAM.

Throughout this thesis, the SLM build envelope (i.e. viable building space inside the build chamber) is assumed to have 240 mm width, 240 mm length and 275 mm height. Thus, the 2D bin dimensions used in the 2DIBPP are shown schematically in Figure 3.17, where W and L denote bin width and length, respectively; d_1 and d_2 denote the location and dimensions of four symmetrical squares representing the no-build zones, as shown; and d_o denotes the minimum allowable distance between the SLM parts (as discussed in Section 3.4.3).

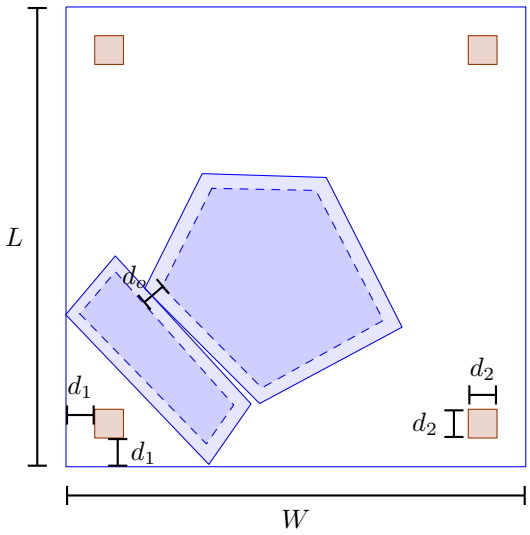


FIGURE 3.17: Bin dimensions used in the 2DIBPP. The diagram shows a 2D bin packed with two inflated pieces, shown in blue, and no-build zones shown in red.

3.7 Conclusion

This chapter describes a cost-driven process planning framework for SLM. Section 3.1.1 outlines two SLM production scenarios, IBP and MBP, and discusses the suitability of each scenario in the context of lean manufacturing and different demand patterns which are commonly seen in many industries, including aerospace, automotive and the medical sector. The presence of safety critical parts is also considered and a new build orientation constraint is proposed.

Sections 3.2 and 3.3 cover the cost modelling methodology, which includes both SLM build cost and post-processing cost. The build cost is an adaptation of Ruffo et al. (2006), where the coefficient values are taken from Baumers et al. (2016) who adapt the same model in their work. To predict the build time of SLM, a simple linear regression model is proposed, based on build height, part volume and support structure volume, where the latter two parameters are driven by build orientation. While this method is less accurate than an analytic approach, it offers a fast and sufficient approximation of build time, by bypassing the need to compute the exact laser path, which is too time consuming and computationally expensive for the optimisation models proposed in this thesis. Furthermore, given the limited training data, the fitted regression model is considered to be acceptable for the purpose of demonstrating the methodology proposed in this thesis; however, future work should aim to build a larger set of data to properly train and validate this model.

As discussed in Chapter 1, to avoid unexpected additional costs and risks of delay at the later stages of the manufacturing process, post-processing costs should be taken into consideration as early as possible i.e. at the process planning stage. As discussed in Section 3.3, the key post-processing costs associated with build orientation are surface finishing and support structure removal. This thesis argues target surface finish should be treated as a constraint and Section 3.3.1 describes a simple application designed to enable the user to include local target roughness into a standard STL file.

Section 3.3 also outlines different surface finishing methods available for SLM, which can be classified as either global or local methods. Following the above logic, the framework assumes a local surface finishing method, where the difference between the target surface roughness and build surface roughness produced by SLM has a direct effect on the post-processing time and cost. Conversely, a global surface finishing method treats the entire surface simultaneously, thereby, de-coupling the build surface roughness (and hence the build orientation) from the resulting post-processing cost, and negating the effects of surface finishing on the optimisation problem. It should be noted that this model considers only the operating costs and tool costs, based on Ti-6Al-4V and a medium-sized 5-axis milling machine[5], while other costs such as tool-path planning by a skilled operator and machine set-up are not considered.

The support structure removal model assumes a manual process and is based on

[5]milling data provided by Rolls-Royce plc.

the assumption of Alexander et al. (1998), who suggest that removing support structure attached to the part at both ends is more difficult and time consuming than removing support structure attached at only one end. To reflect this, the model proposes two different weighting factors for downward-facing and upward-facing surfaces, respectively, which are in contact with support structure.

Section 3.4 outlines in detail the build orientation generation procedure, which assumes continuous rotation about two perpendicular axes. Continuous rotation is proposed as a way of exploring more potential (and less obvious) solutions in the design space. Methods of calculating support structure, surface roughness and 2D piece generation are discussed in detail in this section. The support structure prediction was found to be the most time consuming aspect of this part of the framework, so a simplifying technique is proposed in Section 3.4.1 and validated against the commercial software Magics, producing a RMSE of 5.47% for a set of 68 test geometries, with a maximum absolute error of 48%. It was found that, combined with a density fraction of 0.28 (for the same 68 test geometries), the model can also be used to predict hollow support structure, although this increases the maximum error to 61.0% when compared to Magics. In both cases, despite the large maximum errors, most of the predicted data points were found to be reasonably accurate. As part of future work, a surrogate modelling or machine learning approach could be used to predict support structure more quickly and accurately, although such a method would require a very large and robust data set, based on a variety of parts of different sizes and geometric complexities.

Chapter 4

Cost Optimisation for Identical Batch Production

The first part of this chapter outlines a formal problem description for process planning in the IBP scenario. Next, the Tabu search heuristic is proposed to solve the problem, due to its explorational attributes and ability to escape local minima. Finally, Tabu search parameters, such as the size of local neighbourhoods, are tested on three different realistic part geometries.

A key contribution of this chapter is a new objective function for the combined build orientation and bin packing problem, which includes both build and post processing cost, derived from equations described in Chapter 3.

Moreover, manufactured parts often contain geometric symmetries, which typically lead to symmetries in the build orientation solution space. This is acknowledged in relevant literature (Peng et al., 2017), but no works explicitly address this challenge. Therefore, a second contribution of this chapter is a thorough discussion of solution space symmetry, its effects on optimisation performance, and ways of addressing it. Following on from this discussion, a new approximation strategy is proposed to reduce the frequency of time-consuming, and potentially redundant, bin packing solutions. The test results indicate that the approximation strategy reduces solution time of the Tabu search by at least 50%.

4.1 Problem Description

Consider N identical geometries, which are to be placed into identical bins of width W, length L and height H. The geometry is denoted by g, and its surface is defined by a set of facets F, where each facet $f \in F$ has a target surface roughness denoted by Ra_{t_f}.

The objective is to find a build orientation of g with the minimum unit cost. In order to calculate the unit cost the batch size must be known, thus, a bin packing problem must be solved for each new build orientation of g. As discussed in Chapter 3, vertical stacking of parts is not allowed, therefore, the bin packing problem is modelled in 2D, where each new build orientation of g is represented by a 2D piece.

Thus, let p denote the 2D piece representing g in its current build orientation, and let h_p denote the current build height of p; s_p denotes the current support structure volume of p; n_p denotes the batch size resulting from p; $v_s(p)$ denotes the resulting volume of material removed during surface polishing; and $A_d(p)$ and $A_u(p)$ denote the total downward-facing and upward-facing areas, respectively, which are in contact with support structure resulting for p.

It should be noted that h_p may not exceed the bin height H. Furthermore, as discussed in Section 3.1.2, a solution to the IBP problem must place all instances of g in the same build orientation to avoid any additional process complexities and variations. In other words, the bin packing solution must place N identical instances of p into M bins, where $n_pM \geq N$.

This presents a simple version of the cutting stock problem, which can be solved by placing n_p instances of p into a single unique bin and duplicating this bin M times to meet the demand N. Since the process planning framework assumes an unlimited number of available bins (Section 3.1.3), a packing solution for a single bin is sufficient to determine the unit cost corresponding to piece p, without considering the $n_pM \geq N$ condition. Thus, the objective function of this problem can be summarised as follows:

$$\min_{p(\theta,\phi)} C(p) \tag{4.1}$$

$$\text{for} \quad \theta \in [0^o, 180^o], \quad \phi \in [0^o, 360^o)$$

$$\text{where} \quad C(p) = \frac{c_b(p)}{n_p} + c_{pp}(p) \tag{4.2}$$

where $c_b(p)$ denotes the SLM build cost of a batch of n_p identical instances of piece p, and $c_{pp}(p)$ denotes the post processing cost of p. Definitions of c_b and c_{pp} are provided in Equation 3.1 and Equation 3.4, respectively, and substituting these two equations into Equation 4.2 results in Equation 4.3.

$$C(p) = \omega_1 + \omega_2 s_p + \omega_3 t_s(p) + \omega_4 t_r(p) + \omega_5 \frac{h_p}{n_p} + \omega_6 \frac{1}{n_p} \tag{4.3}$$

$$\omega_1 = v_g(c_2\rho + c_1\beta_3), \quad \omega_2 = c_1\beta_4 + c_2\rho, \quad \omega_3 = c_3, \quad \omega_4 = c_4, \quad \omega_5 = c_1\beta_2, \quad \omega_6 = c_1\beta_1$$

where s_p, $t_s(p)$, $t_r(p)$, h_p and n_p are the decision criteria, which are driven by the build orientation and resulting packing solution; and $\omega_1, \ldots, \omega_6$ are constant coefficients, which determine the weight of each criterion in the objective function. For the reader's convenience, equations 3.5 and 3.6 which define support structure removal time, $t_s(p)$, and surface finishing time, $t_s(p)$, in Chapter 3, are shown below:

$$t_s(p) = \frac{v_s(p)}{q} + \left\lceil \frac{v_s(p)}{t_{tl}q} \right\rceil t_{load} \quad (3.5) \;, \qquad t_r(p) = w_1 A_d(p) + w_2 A_u(p) \quad (3.6)$$

where q is the material removal rate of the surface polishing method, t_{tl} is the tool life, t_{load} is the tool loading time, and w_1 and w_2 are time coefficients associated with manual support structure removal. The detailed methodologies for calculating $A_d(p)$

and $A_u(p)$, and $v_s(p)$ are provided in Section 3.4.1 and Section 3.4.2, respectively.

The method for determining n_p is discussed in Section 3.5, which tries to place as many instances of p as possible into a 2D bin of width W and length L, and produces a bin solution that can be denoted by $b(n, O, X, Y)$, where n denotes the batch size. During the search the pieces are allowed to rotate continuously about the vertical z-axis, hence, $O = \{o_1, \ldots, o_n\}$ defines the 2D orientation of each piece. Finally, $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$ define the x and y coordinates of the reference point of each piece, respectively. The reference point of a piece is the bottom left corner of its enclosing rectangle, corresponding to its 2D orientation. Every piece must be placed completely inside the bin and may not overlap with any other pieces or the no-build zones (shown in Figure 3.17).

4.2 Tabu Search Procedure

It can be seen that the above problem is highly non-linear, not only as a result of Equation 4.3, but also due to the complex relationships between the criteria and the different surfaces in the part geometry. To deal with the presence of multiple local minima, this section proposes the use of Tabu search (TS) (Glover, 1989, 1990), which is a well-known meta-heuristic used to guide a local search procedure and aid it in exploring the solution space beyond local minima. The effectiveness of TS stems from its two defining principles; firstly, it can move from a better solution to a worse solution to escape a local minimum; secondly, previously visited solutions are marked as "taboo", which means the search is forbidden from re-visiting these solutions[1], and gives the algorithm its name.

The pseudo-code of the proposed algorithm is presented in Figure 4.1. At each iteration, the TS must evaluate the local neighbourhood $P^N(p_c)$ of the current solution p_c and move to the most favourable neighbour $p' \in P^N(p_c)$, which becomes the new current solution. Equation 4.3 is used to evaluate each solution p in the neighbourhood, such that $C(p') \leq C(p)$ for all $p \in P^N(p_c)$. This process is repeated until the termination criterion of $I_n \geq I_{n_{max}}$ is satisfied, where I_n is the number of consecutive non-improvement iterations and $I_{n_{max}}$ is a threshold value set by the user. Since p' is not necessarily better than p_c, the procedure must also track the overall best solution encountered at each iteration, which is denoted by p_b and must be updated each time the condition $C(p_b) > C(p_c)$ is met.

As shown in Figure 4.1, each visited solution (p_c) is marked as "taboo" and stored in a Tabu list, preventing it from being selected in the following iterations. To avoid exceeding the computational memory budget and to avoid the search cornering itself into a dead-end, the maximum number of entries in the Tabu list is limited to a fixed

[1] Variants of the TS procedure may incur a penalty for, or reduce the probability of, moving to a marked solution (instead of forbidding it), and can also mark unexplored solutions with potentially undesirable attributes as "taboo" (in addition to visited solutions).

Input: $p_{in}, \mathcal{E}, L_{size}, I_{n_{max}}$
Result: p_b
$p_c \leftarrow p_{in}$;
$p_b \leftarrow p_c$;
$I_n \leftarrow \varnothing$;
$TabuList \leftarrow \emptyset$;
while $I_n < I_{n_{max}}$ **do**
  Construct neighbourhood $P^N(p_c)$ for p_c using step size $\mathcal{E}$;
  Return best solution $p' \in P^N(p_c)$;
  if $TabuList.Size > L_{size}$ **then**
    Remove oldest entry in $TabuList$;
  end
  Add p_c to $TabuList$;
  Set $p_c \leftarrow p'$;
  if $C(p_b) > C(p_c)$ **then**
    $I_n \leftarrow \varnothing$;
    $p_b \leftarrow p_c$;
  end
  else
    $I_n \leftarrow I_n + 1$;
  end
end

FIGURE 4.1: Tabu search procedure, where p_{in}, p_c and p_b denote the initial, current and best solution, respectively; and $C(p_b)$ and $C(p_c)$ denote the unit cost of p_b and p_c, respectively.

size, denoted by L_{size}; when this size is reached, the oldest entry must be removed from the list before a new entry can be added.

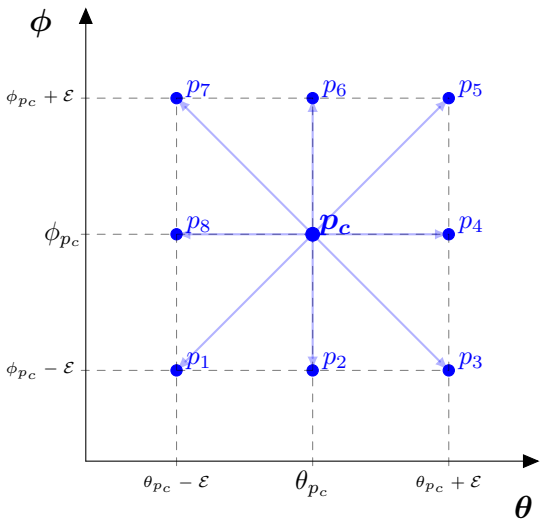


FIGURE 4.2: Structure of local neighbourhood $P^N(p_c) = \{p_1, \ldots, p_8\}$, where p_c denotes the current solution, $\mathcal{E}$ denotes the step size and θ and ϕ are two angles of rotation.

The structure of the local neighbourhood $P^N(p_c)$, is a fundamental element of TS, and has a significant effect on the algorithm's performance. In the proposed procedure, the neighbourhood has a fixed size and a simple structure, as shown in Figure 4.2. For each piece p_c, the local neighbourhood $P^N(p_c)$ consists of a total of eight neighbouring solutions, where each neighbour is produced by taking a step

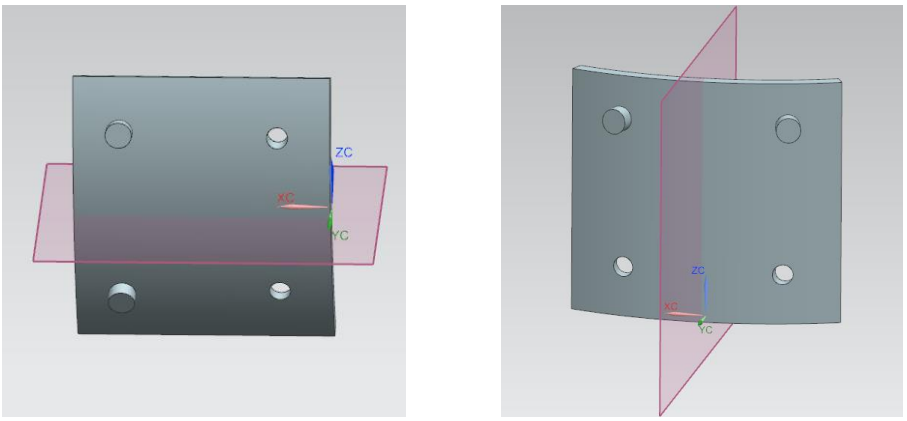
of size $\mathcal{E}$ away from p_c, in any direction in θ, in ϕ, or in both dimensions. This simple structure allows each neighbourhood to be constructed and evaluated quickly, with a relatively small computational expense, while the step size $\mathcal{E}$ can be changed between iterations to encourage the procedure to either explore the solution space, or to converge on a minimum.

4.2.1 Solution Space Symmetry

As stated in Section 3.4, sampling points along $\theta = 0^{\circ}$ and $\theta = 180^{\circ}$ result in duplicate build orientations. If one of these build orientations happens to be a local minimum the Tabu search is likely to get stuck. To deal with this issue, once a solution at $\theta = 0^{\circ}$ has been visited, a new entry denoted by $p(0^{\circ},^{*})$ is added to the Tabu list, where * represents any arbitrary value of ϕ. This marks all solutions at $\theta = 0^{\circ}$ as "taboo" and prevents the search from moving to another solution located at $\theta = 0^{\circ}$; the same is done for solutions at $\theta = 180^{\circ}$.

Furthermore, any symmetries in the geometry can be expected to produce corresponding symmetries in the solution space. This is likely to have a detrimental effect on the efficiency of the optimisation search and quality of the resulting solution. To ensure that only unique areas of the solution space are explored in the presence of reflectional symmetries, it is suggested that the initial build orientation of the geometry is calibrated in one of two ways:

- by aligning the plane of symmetry to the global x-y plane, in which case the input range of θ can be bisected, as shown in Figure 4.3a, or;
- by aligning the plane of symmetry to the global y-z plane, in which case the input range of ϕ can be bisected, as shown in Figure 4.3b.



(A) Bisect interval $\theta \in [0, 180] \leftarrow [0, 90]$. (B) Bisect interval $\phi \in [0, 360) \leftarrow [0, 180]$.

FIGURE 4.3: Two possible initial build orientations and resulting input intervals for a geometry containing a single plane of reflectional symmetry.

Alternatively, if the geometry contains a plane with multiple lines of rotational symmetry, the same principle can be applied by aligning one of the axes of rotation to the normal vector of that plane and dividing the corresponding input range by the degree of rotational symmetry, as shown in Figure 4.4.

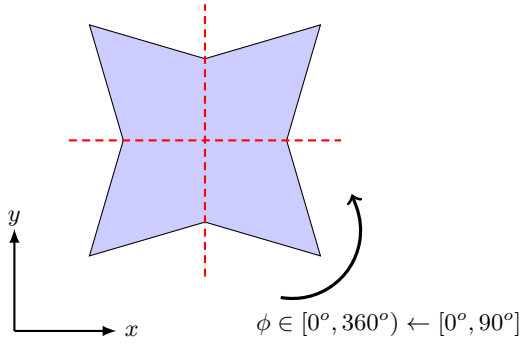
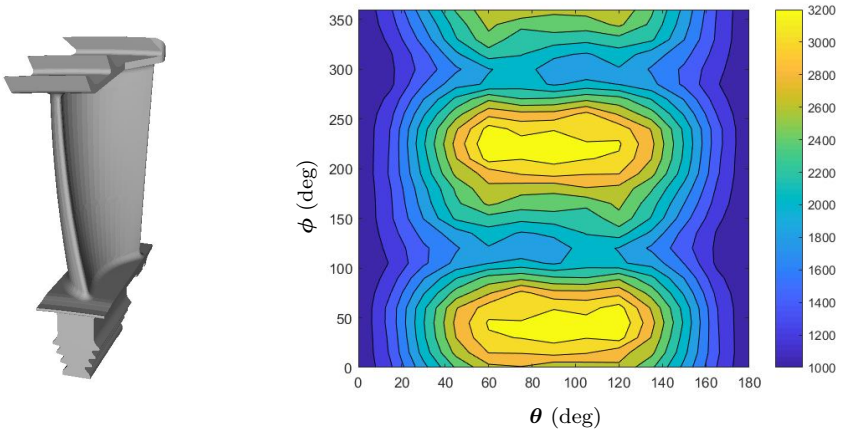


FIGURE 4.4: Example of rotational symmetry in the cross-section of a prism. Since the x-y plane contains four degrees of rotational symmetry (dashed red lines) the interval of ϕ can be reduced from $[0^o, 360^o)$ to $[0^o, 90^o]$.

Finally, it should be noted that even if the geometry contains no symmetries, the build height and piece area can be expected to follow a recurring pattern with respect to θ and ϕ, due to the periodic nature of the build orientation problem. This is demonstrated in Figure 4.5b, which shows the a_p solution space for a non-symmetrical turbine blade geometry. It can be seen that the yellow region in the lower part of the figure can be mapped onto the top yellow region with two reflection transformations, one in the θ axis and another in the ϕ axis. It can be easily deduced that such repetitive behaviour in the a_p domain is highly likely to be present in the n_p domain as well. Since the 2D bin packing solution makes n_p significantly more expensive to compute than all the other piece attributes, this behaviour of the solution space should be used to avoid duplicate bin packing solutions where possible during the Tabu search.



(A) Turbine blade geometry. (B) Solution space of piece area a_p (mm^2).

FIGURE 4.5: Solution space of a_p shown for a non-symmetrical turbine blade geometry.

4.2.2 Efficient Search Strategy

Following the above discussion, this section proposes a strategy that aims to reduce the number of times the bin packing problem is solved, especially for pieces of similar area. This is achieved by comparing the projected area a_p of a current piece p to the areas of previously visited pieces, and either solving the bin packing problem to find the exact value of n_p, or approximating n_p based on an existing solution with an area similar to a_p.

More precisely, each exact solutions must be stored in a separate list denoted by *SolutionList*, and each new solution p must be compared against all solutions in *SolutionList*. If *SolutionList* contains a piece p_S with area a_{p_S}, such that $a_p \approx a_{p_S}$, then set $n_p \leftarrow n_{p_S}$; otherwise, if a_p is significantly different from all existing solutions in *SolutionList*, the bin packing problem must be solved to find n_p, and p must be added to *SolutionList*. The difference between a_p and a_{p_S} is considered small, if $\frac{|a_p - a_{p_S}|}{a_p} \leq A_{lim}$, where A_{lim} is an approximation parameter specified by the user. Figure 4.6 provides pseudo-code of the local neighbourhood construction procedure, which uses this strategy.

Input: p_c, $\mathcal{E}$, A_{lim}, *SolutionList*
Result: updated p_c, updated *SolutionList*
Create an empty list $P^N(p_c)$ which denotes the neighbourhood of p_c, using step size $\mathcal{E}$;
foreach $p \in P^N$ **do**
    Find build height h_p of p;
    Find support structure volume s_p of p;
    Find area a_p of p;
    foreach $p_S \in$ *SolutionList* **do**
        $\delta a(p_S) = \frac{|a_p - a_{p_S}|}{a_p}$;
    end
    Take solution $p_S \in$ *SolutionList* with minimum δa;
    if $\delta a \leq A_{lim}$ **then**
        set $n_p \leftarrow n_{p_S}$;
        Calculate cost $C(p)$ of p;
    end
    else
        Solve 2DBPP to find n_p;
        Calculate cost $C(p)$ of p;
        Add p to *SolutionList*;
    end
end
Order all $p \in P^N$ by non-decreasing $C(p)$, and set $p_c \leftarrow P^N[1]$;

FIGURE 4.6: Neighbourhood construction procedure, where A_{lim} is a hyper-parameter provided by the user and *SolutionList* denotes a list which stores all solutions, for which the exact value of n_p is known.

Finally, let $C_a(p)$ denote the approximated cost of a solution p, and $C(p_b)$ denote the cost of the current best solution p_b. If $C_a(p) < C(p_b)$, then the exact cost of p, denoted by $C(p)$, must be computed and p must be added to *SolutionList*. If $C(p) < C(p_b)$, update the best solution $p_b \leftarrow p$, otherwise continue to the next iteration. It should be noted that to find the approximated cost $C_a(p)$ only the

number of pieces in the bin n_p is approximated, while the bin height h_p and support structure volume s_p are still calculated exactly, since these values are relatively cheap to compute when compared to the bin packing problem.

4.3 Implementation

Table 4.1 provides a summary of all parameters, settings and cost modelling coefficients used in the computational results provided in Section 4.4. For surfaces which are not supported or in contact with the build plate, the SLM surface roughness is predicted using the method and data provided in Section 3.4.2, where the latter is taken from Vandenbroucke and Kruth (2007). The model assumes that all parts are built directly on the SLM build plate without any base support structure.

Parameter	Value	Units	Definition
c_1	26.64	£/h	Indirect cost coefficient
c_2	258.2	£/kg^3	Material cost rate
c_3	20.0	£/h	Labour rate
c_4	60.0	£/h	Machining cost coefficient
ρ	4.41	g cm^{-3}	Ti-6Al-4V density produced by SLM
β_1	3.00	h	SLM constant time coefficient
β_2	1.03×10^{-1}	h mm^{-1}	SLM recoat time coefficient
β_3	1.81×10^{-4}	h mm^{-3}	SLM part exposure time coefficient
β_4	7.50×10^{-5}	h mm^{-3}	SLM support structure time coefficient
w_1	4×10^{-4}	h mm^{-2}	Support removal coefficient for downward-facing surfaces
w_2	8×10^{-4}	h mm^{-2}	Support removal coefficient for upward-facing surfaces
q	1×10^5	mm^3 h^{-1}	Machining material removal rate
t_{tl}	40.0	min	Tool life
t_{load}	0.25	min	Tool loading time
Ra_{sup}	27.0	µm	Surface roughness after support structure removal
Ra_{rem}	5.00	µm	Surface roughness after wire-EDM
α_{sup}	40.0	°	Minimum unsupported build angle
ρ_{sup}	28	%	Density fraction for lattice support structure
d_0	10.0	mm	Minimum distance between pieces in 2DIBPP
L	50.0	mm	Maximum edge length in 2D piece polygon

TABLE 4.1: Summary of cost coefficients and modelling parameters.

The TS procedure is implemented in C# using the Visual Studio 2012 development environment. The code is parallelised on four i7-3820 cores, with 2.70 GHz maximum frequency. In order to allow for more thorough exploration of the design space, the TS is set to terminate after 30 non-improvement iterations. The length of the Tabu list, denoted by L_{size}, is also set to a relatively large value of 100 to further encourage exploration. Finally, when varying the size of local neighbourhoods, the minimum value of $\mathcal{E}$ is restricted to 1^o, as cost changes below this step size are expected to be negligible.

4.3.1 Test Geometries

The above TS procedure is implemented on three test geometries, shown in Figure 4.7 below. As can be seen, different target surface roughness values are assigned to

different local surfaces of each geometry. For simplicity, each local surface of each geometry is assigned one of two different target roughness values. High tolerance or critical surfaces (e.g. mating faces and gas-washed surfaces) are assigned a low target roughness value, denoted by $Ra_t(1)$, while all other surfaces are allowed a higher target roughness, denoted by $Ra_t(2)$ and ranging between 4 µm and 8 µm.

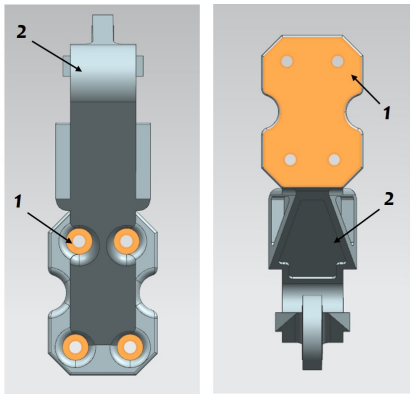
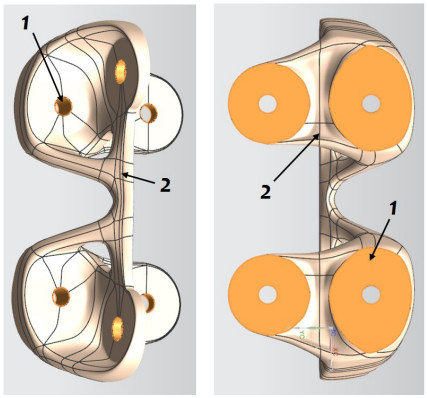
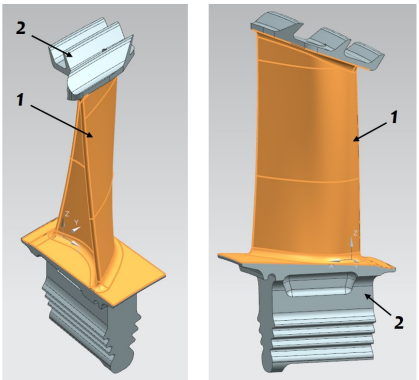
Aero bracket (AB)		Volume (mm^3) 40,806 Surface area (mm^2) 20,037 No. of facets 32,686 No. of holes 6 $Ra_t(1)$ (µm) 1 $Ra_t(2)$ (µm) 8
Space bracket (SB)		Volume (mm^3) 30,540 Surface area (mm^2) 13,682 No. of facets 75,982 No. of holes 6 $Ra_t(1)$ (µm) 1 $Ra_t(2)$ (µm) 5
Turbine blade (TB)		Volume (mm^3) 13,216 Surface area (mm^2) 10,478 No. of facets 31,230 No. of holes 0 $Ra_t(1)$ (µm) 1 $Ra_t(2)$ (µm) 4

FIGURE 4.7: Test geometries and their properties. $Ra_t(1)$ and $Ra_t(2)$ denote the target surface roughness values associated with local surfaces labelled 1 (highlighted in orange) and 2 (grey), respectively.

It can be seen that the AB and SB geometries both contain a single plane of symmetry. As discussed in Section 4.2.1, each geometry is positioned such that its plane of symmetry is aligned with the global y-z plane and all unique solutions lie within $\theta \in [0^o, 180^o]$ and $\phi \in [0^o, 180^o]$. Since the TB geometry contains no symmetries, its solution space is bounded by the intervals $\theta \in [0^o, 180^o]$ and $\phi \in [0^o, 360^o)$.

4.4 Computational Results

This section consists of two parts. The first part evaluates the effects of the approximation strategy on the speed and accuracy of the TS, and proposes the most suitable value of A_{lim} based on the above test geometries. The second part considers the effects of the local neighbourhood size on the TS, considering four different strategies with both fixed and varying values of $\mathcal{E}$.

4.4.1 Approximation Parameter Study

As shown in Table 4.2, four values of A_{lim} are tested for each geometry, where the step size $\mathcal{E}$ is fixed at 10^o throughout all tests. Figures 4.8 - 4.10 provide plots of the TS procedure for each test; where the solution cost at each iteration is denoted by $C(p)$; the cost of approximated solutions is denoted by $C_a(p)$; and the exact cost computed for every other approximated solution is denoted by $C_a^*(p)$.

The latter is used only for calculating the error in cost that results from using the area-based approximation method discussed in Section 4.2.2 (thus, the solution time of $C_a^*(p)$ is not recorded); the cost error for an approximated solution p is calculated as the difference between the approximated cost $C_a(p)$ and exact cost $C_a^*(p)$. This cost approximation error is recorded throughout each test and used to calculate statistical measurements, namely, RMSE, mean error, denoted by $\overline{E}$, and maximum error, denoted by E_{max}. These measurements of error are used along with solution time and the number of iterations before convergence to evaluate the performance of each test. These data are summarised in Table 4.2. For a consistent comparison, RMSE is calculated as a percentage with respect to the initial solution of each geometry.

Geometry	A_{lim} (%)	$C(p_{in})$ (£)	$C(p_b)$ (£)	Solution Time (s)	Iterations	$\overline{E}$ (%)	E_{max} (%)	RMSE (%)
AB	0.01	696	641	4,301	55	-0.35	-3.43	1.54
AB	0.05	696	641	2,185	64	0.17	3.42	1.33
AB	0.1	696	641	1,797	53	0.55	3.59	1.64
AB	0.2	696	650	1,244	49	0.84	3.65	1.48
SB	0.01	417	369	3,736	39	-0.26	2.60	1.40
SB	0.05	417	369	1,987	39	-0.69	-3.56	1.34
SB	0.1	417	369	1,719	39	-1.07	-3.59	1.67
SB	0.2	417	369	1,358	39	-1.22	-3.59	1.78
TB	0.01	967	834	6,793	71	-1.03	3.56	2.35
TB	0.05	967	834	2,837	54	-0.54	4.14	2.11
TB	0.1	967	834	2,287	57	-1.07	-4.11	2.13
TB	0.2	967	834	2,217	46	-0.42	4.14	2.39

TABLE 4.2: Overview of results, where A_{lim} denotes the approximation parameter; and $C(p_{in})$ and $C(p_b)$ denote the initial and best unit cost, respectively.

While there appears to be a very slight increase in the magnitude of $\overline{E}$ with increasing A_{lim}, no clear correlation can be seen between the other two measures of error and the approximation parameter. Moreover, the results indicate no particular bias towards either overestimating or underestimating solutions, as both positive and negative values of $\overline{E}$ and E_{max} can be seen in Table 4.2 and in figures 4.8 - 4.10.

Therefore, it is important to note the different effect each type of error has on the TS performance.

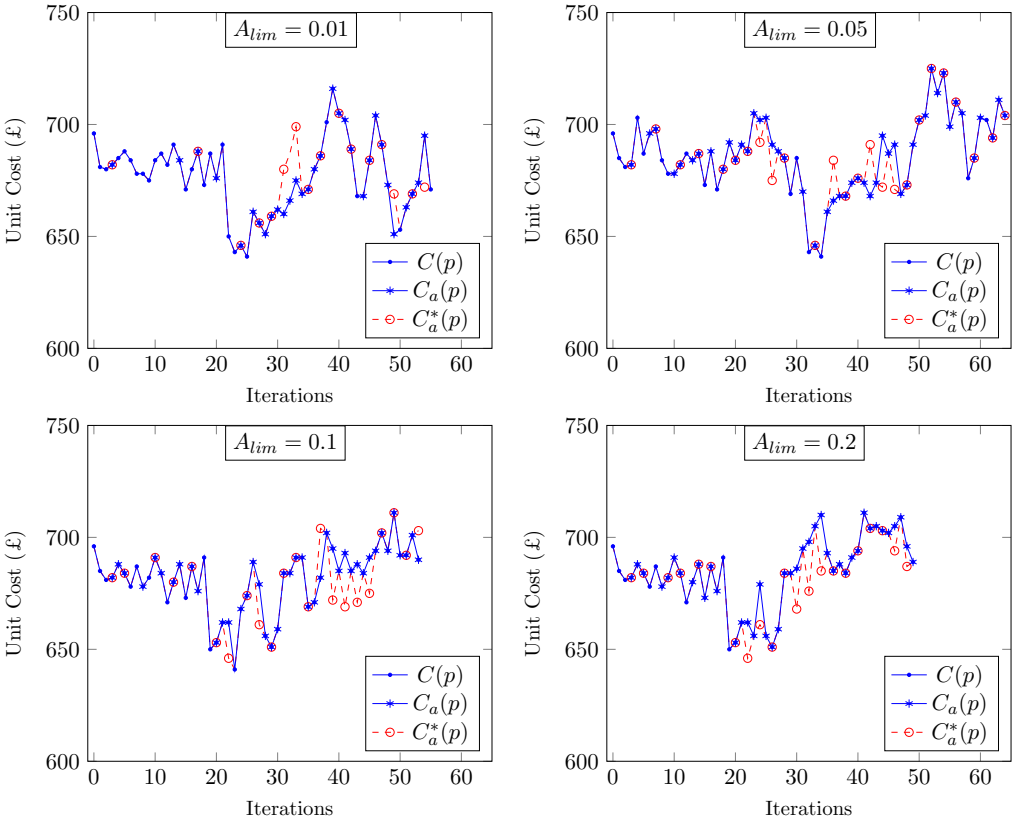


FIGURE 4.8: Tabu search results for aero bracket (AB) geometry.

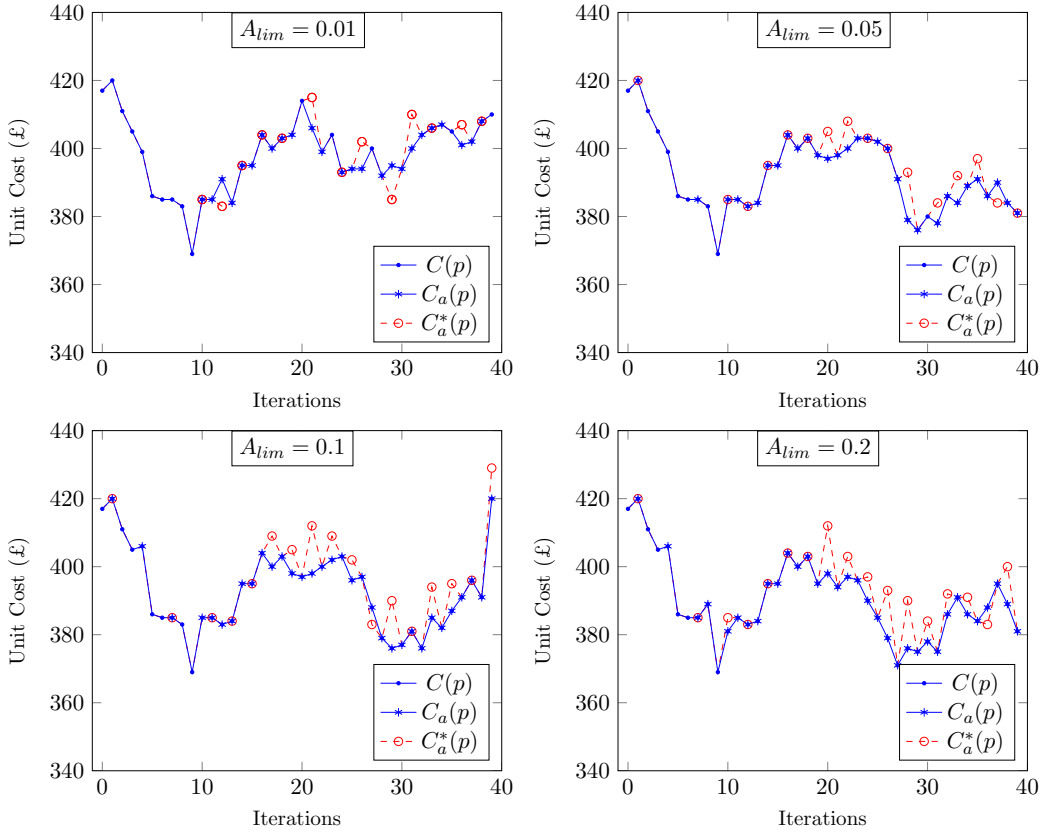


FIGURE 4.9: Tabu search results for space bracket (SB) geometry.

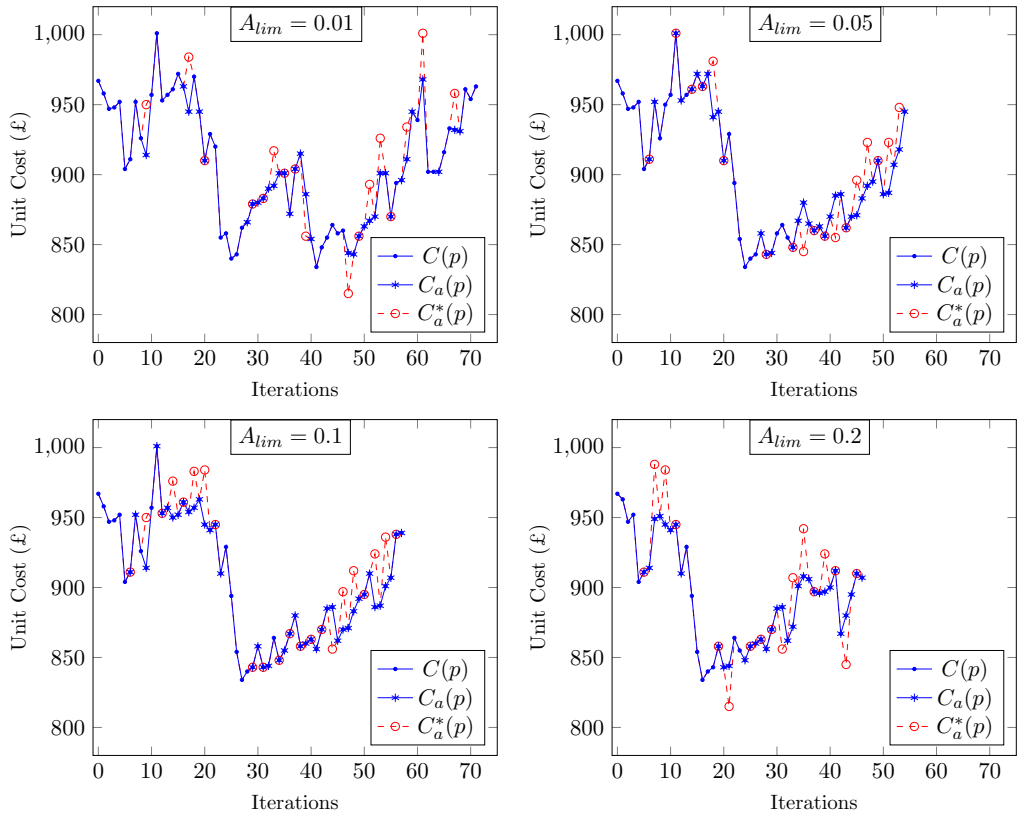


FIGURE 4.10: Tabu search results for turbine blade (TB) geometry.

Clearly, the TS must accumulate a sufficiently large and diverse list of exact solutions, before this list can be used to approximate similar solutions encountered later on in the procedure. Furthermore, as outlined in Section 4.2.2, the strategy is forced to calculate the exact cost for every cost-improvement solution, regardless of its similarity to other recorded solutions. Consequently, the proposed approximation strategy tends to provide accurate solutions in the beginning of the TS and during its convergence to a local minimum, but as the search starts to diverge away from this point, the frequency of approximations rises and the accuracy declines, as shown in figures 4.8 - 4.10.

It follows that whenever the cost of p is underestimated, such that $C^a(p) < C(p_b)$, the exact solution of p must be computed by solving the bin packing problem. In this way, negative errors can increase the solution time by forcing the TS to solve the bin packing problem more frequently, even for solutions which are not in fact better than the best solution. Positive errors pose an arguably bigger risk. If the real cost of $C(p)$ is an undiscovered local minimum, such that $C(p) < C(p_b)$, but the cost of p is overestimated, such that $C^a(p) > C(p_b)$, the TS will miss the solution and return a sub-optimal p_b. Examples of this can be seen in Figure 4.8 for $A_{lim} = 0.02$, and Figure 4.10 for $A_{lim} = 0.01$ and $A_{lim} = 0.2$.

In addition, both positive and negative approximation errors pose the risk of misguiding the TS, either away from potentially good solutions (in case of the former), or towards inferior ones (in case of the latter). Conversely, it is also possible for these errors to guide the TS towards a better solution "by chance". The AB geometry provides examples of both scenarios; as shown in Figure 4.8, $A = 0.2$ produced a relatively poor result, as the TS was guided towards a sub-optimal solution; meanwhile in the case of $A = 0.1$, the TS was guided towards the same best solution in fewer iterations than $A = 0.01$ and $A = 0.05$. Thus, the risk of errors misguiding the TS is counteracted by the possibility of these errors guiding the TS towards better solutions.

Furthermore, the main advantage of the approximation procedure is the saving of computational time and resources, particularly on duplicate solutions and solutions which yield no significant cost improvement, as shown in Figure 4.9. The tests indicate that up to 56% of solutions are approximated in the case of $A_{lim} = 0.01$; this percentage goes up to 79% for $A_{lim} = 0.05$ and up to 87% when $A_{lim} \geq 0.1$. This has two implications:

- Since an approximated solution is typically computed within milliseconds, while the bin packing problem is solved on the order of 100 seconds, it can be inferred that using the approximation strategy with $A_{lim} = 0.01$ reduces the solution time of the regular TS procedure (i.e. where the bin packing problem is solved for each solution) by at least 50%.

- The reduction in solution time becomes smaller as A_{lim} is increased. This can be seen in Figure 4.11, where the solution time is halved when A_{lim} is increased

from 0.01 to 0.05, but does not change significantly for $A_{lim} \geq 0.1$.

Based on these observations, it is suggested that the value of A_{lim} is kept in the range 0.01 - 0.05. This conclusion can only be applied with certainty to the three geometries tested in this thesis; to apply the procedure to other geometries, such geometries should be tested first, using the approach described in this section.

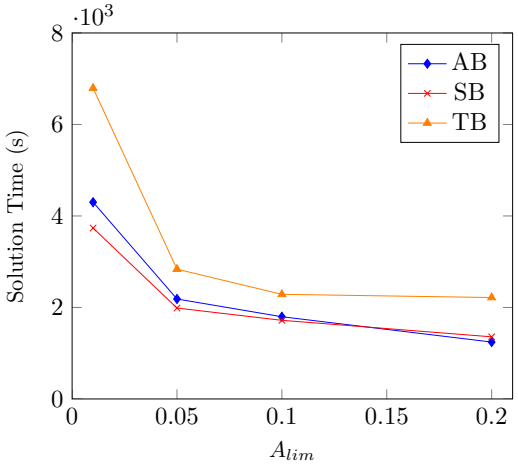


FIGURE 4.11: Reduction in solution time with respect to the approximation parameter A_{lim}.

Finally, as can be seen from figures 4.8 - 4.10, the search converged quickly (within 10-30 iterations) in all three test cases despite the relatively large size of the Tabu list, suggesting that the starting point of each test was within close proximity of a very good local minimum solution. One way of addressing this is to expand the Tabu list to include not only previously visited solutions, but also unvisited solutions with characteristics that are undesirable or very similar to those of previously visited solutions. However, to minimise the risk of adding good solutions to the Tabu list, such an approach would require very careful parameter tuning, using a larger set of test geometries.

4.4.2 Local Neighbourhood Size Study

To test the effects of the neighbourhood size, the TS is performed with three constant step sizes, 5^o, 10^o and 15^o, and compared against two variable step strategies, namely, random step, denoted by R, and decreasing step, denoted by D. The above approximation procedure is used in all tests, with A_{lim} set to 0.05.

In the R strategy the value of $\mathcal{E}$ is allowed to range between 0^o and 180^o, enabling the TS to cover potentially large regions of the solution space, but also significantly reducing the likelihood of converging on a local minimum. This is indicated by the scattered appearance of the R sampling points in figures 4.12a - 4.12c. Figure 4.12a also provides an example of how the R strategy can miss good quality solutions. As can be seen, one of the R solutions lies very close (within $\sim 5^o$) to the best local minimum found by the 15^o step strategy; however, despite their proximity, the cost of the R solution exceeds the cost of the local minimum by almost 5%. Since the R

solution is approximated, it is possible that the significant cost difference is actually caused by the approximation strategy overestimating the cost of the R solution, as discussed above. However, in this case the approximation error makes little difference, as the cost of solutions does not have a strong guiding effect on the random step TS (unlike the other step strategies).

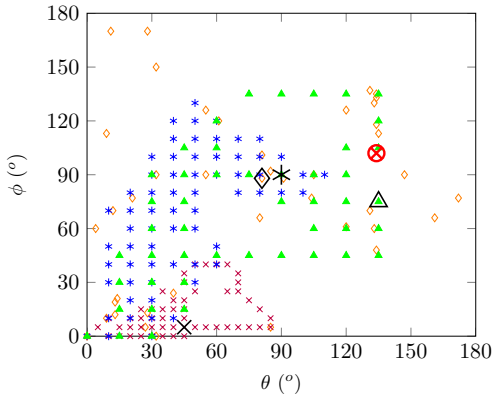
Geometry	$\mathcal{E}$ (°)	$C(p_{in})$ (£)	$C(p_b)$ (£)	Improvement (%)	Solution Time (s)	Iterations
AB	R	696	651	6.5	2,104	41
AB	5	696	663	4.7	1,967	52
AB	10	696	641	7.9	2,185	64
AB	15	696	640	8.0	1,715	46
AB	D	696	632	9.2	2,177	59
SB	R	417	368	11.8	2,300	41
SB	5	417	369	11.5	1,247	48
SB	10	417	369	11.5	1,987	39
SB	15	417	369	11.5	1,847	36
SB	D	417	369	11.5	2,337	46
TB	R	967	846	12.5	3,052	62
TB	5	967	886	8.4	1,664	48
TB	10	967	834	13.8	2,780	54
TB	15	967	829	14.3	1,522	44
TB	D	967	827	14.5	3,289	60

TABLE 4.3: Overview of results, where $\mathcal{E}$ denotes the step size, R denotes random step size, D denotes decreasing step size, $C(p_{in})$ denotes the initial unit cost and $C(p_b)$ denotes the best unit cost.

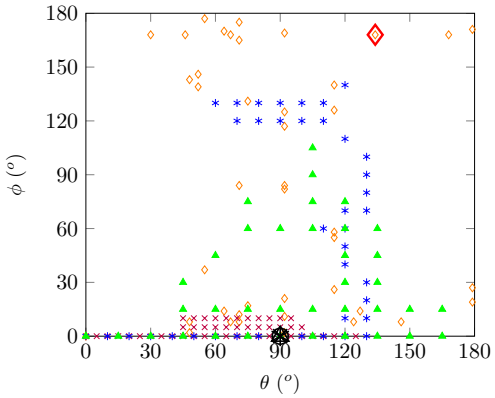
Conversely, in the case of the SB geometry, the R strategy found a slightly better solution, with a 0.3% greater cost improvement than the fixed step strategies. As can be seen from Figure 4.12b, this solution is located in the top right region of the solution space, which none of the fixed step strategies managed to explore before terminating.

From Table 4.3, it can be seen that the 5^o step strategy has the worst performance, in terms of both cost improvement (relative to $C(p_{in})$) and number of iterations. As shown in Figure 4.12 this strategy explored only a very small region of the solution space in all three test cases, and found only one very good local minimum (Figure 4.12b), which happens to lie very close to the starting point of the search. This is not unexpected, as a small value of $\mathcal{E}$ has the following two effects on the TS performance: firstly, the local neighbourhoods are much smaller, so it takes longer to explore the solution space; secondly, as shown clearly in 4.12a, the search path tends to make a lot of U-turns and circle back on itself as it struggles to escape the local minimum.

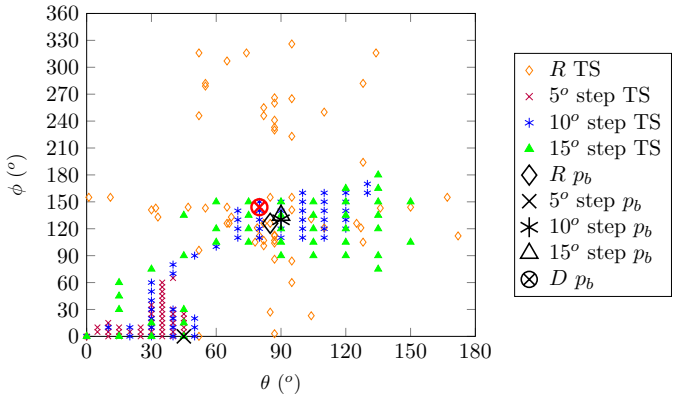
On the other hand, the 15^o step strategy has the best performance out of the three fixed step strategies, followed closely by the 10^o step strategy (Table 4.3). As can be seen from Figure 4.12, both strategies managed to explore a substantial portion of the solution space; and in the case of the SB geometry, the 10^o step strategy came relatively close to the best local minimum found by the R strategy.



(A) Aero bracket (AB) geometry.



(B) Space bracket (SB) geometry.



(C) Turbine blade (TB) geometry.

FIGURE 4.12: Solution space exploration by different step strategies. The best solution is shown for each strategy and is denoted by p_b; the best local minimum found for each geometry is highlighted on each graph in red.

Decreasing Step Strategy

The main downside of using a fixed step size lies in discretising the solution space and consequently missing potentially good solutions located in between the TS steps; this problem is clearly worse for large values of $\mathcal{E}$. Conversely, constantly varying the step size can result in the search moving in a zigzag or circling pattern in between "taboo" solutions, making it much more difficult to drive the TS to escape local minima. With this in mind, the decreasing step strategy, denoted by D, is designed with the following steps:

1. Perform the TS with $\mathcal{E}$ fixed at 15^o, until the maximum non-improvement limit, $I_{n_{max}}$, is exceeded;
2. Update the current solution to the best solution found so far, $p_c \leftarrow p_b$;
3. Reset $I_n \leftarrow \varnothing$, and reduce $\mathcal{E}$ by a factor ϵ_r;
4. Continue the TS, reducing the step size at each iteration, such that $\mathcal{E} \leftarrow \epsilon_r \mathcal{E}$, and terminate when $I_{n_{max}}$ is exceeded.

As can be seen, the first stage of the D strategy is identical to the 15^o step strategy; the key difference is that after this stage is completed, the D strategy attempts to improve p_b by further exploring its vicinity and by reducing the step size as the search converges. Since the two strategies follow a similar search path, and for the purpose of clarity, only the final best solution of D is shown for each geometry in Figure 4.12. Since the aim of this strategy is to improve a local solution, rather than explore the whole solution space, the value of $I_{n_{max}}$ is reduced to 10, after the first stage. The reduction factor ϵ_r is set to 0.95.

As shown in Table 4.3, the D strategy improved the best solution for two of the three geometries, AB and TB, although the latter improvement is very small – outperforming the 15^o step solution by 0.2%. As shown in Figure 4.12, the slightly better SB solution provided by the R strategy is out of reach of the D strategy and no further improvement over the 15^o step strategy was achieved. It can also be seen that the solution time of the D strategy is longer than R in all three test cases, and the worst on average. This downside can be addressed by further reducing $I_{n_{max}}$ in the latter stage, but at the risk of reducing the likelihood of finding a local cost improvement, which is the main purpose of the D strategy.

Arguably, even the longest D solution time in Table 4.3 is not particularly long (just under an hour for TB), since the TS runs automatically and does not require any supervision. Furthermore, even the small unit cost reduction of £2 over the 15^o step strategy (for TB geometry) is a worthwhile trade-off, when considered in the context of IBP of several thousand parts.

4.5 Conclusion

The problem of process planning in the IBP scenario is outlined in this chapter and solved using the Tabu search (TS) heuristic. Since realistic part geometries often contain symmetries, which lead to symmetries in the solution space, a build orientation calibration method is proposed to minimise the effects of these symmetries on the TS performance. Furthermore, a new area-based approximation strategy is proposed to reduce the number of bin packing solutions and improve TS efficiency.

This strategy is driven by an approximation parameter, A_{lim}; thus, three different values of A_{lim} are tested on three different geometries. The results indicate that even the lowest tested value of $A_{lim} = 0.01$ speeds up the TS by as much as 50%. To be able to approximate solutions, the strategy must first build up a sufficiently large and diverse pool of exact solutions, on which each approximation is based. For this reason, the improvement in solution time increases with A_{lim}, but starts to plateau after $A_{lim} \geq 0.1$. Thus, it is suggested that the value of A_{lim} should lie between 0.01 and 0.05. The results show that even small values of A_{lim} pose a risk of missing good local solutions, because of approximation errors introduced into the search. However, as demonstrated by $A_{lim} = 0.1$ in Figure 4.8, these errors can equally lead the search to better solutions, thus, mitigating this risk.

To explore the effects of local neighbourhood size, the TS is tested with five different step strategies, including random step, R, decreasing step, D and three fixed step strategies. For all three test geometries, the 15^o step strategy outperformed the other fixed step strategies. Meanwhile, for one of the geometries the R strategy happened to find a slightly better solution, which lies outside the solution space region explored by the other four strategies. To tackle this, one of the other strategies can be run multiple times, from different starting points, at the expense of longer solution time.

As discussed, a key downside of a fixed step strategy is the risk of missing good solutions that lie in between the TS steps. Meanwhile, varying the step size too much or too early enables the TS to move in between "taboo" solutions, significantly decreasing the effectiveness of the TS. The purpose of the D strategy is to mitigate both of these problems; first, by using a fixed step size to explore the solution space; and, second, by looking for a further cost improvement in the local vicinity of the best found solution with a gradually decreasing step size. The D strategy improves the cost of two of the three test geometries, at the expense of increasing the solution time by up to 30 min (in the case of the TB geometry). However, since the procedure runs automatically without the need for any supervision, the additional solution time can be considered an acceptable price to pay for a potential increase in cost savings.

Chapter 5

Optimisation of Cost and Residual Stresses

5.1 Introduction

As discussed in Chapter 2, there exists a clear and complex relationship between the build orientation and the residual stress profile of a part. Despite this, very limited work has been done in this area and no works have explicitly addressed the problem of residual stress minimisation as a function of build orientation. Moreover, cost is another equally important objective, as discussed throughout this thesis. Therefore, a multi-objective optimisation problem is addressed in this chapter, the aim of which is to find a set of trade-off solutions based on residual stresses and unit cost in the context of IBP.

The reason this problem has received so little attention is likely due to the prohibitive computational expense of FEA, which is typically required to predict the residual stress profile (Peng et al., 2017). To deal with this issue a number of alternative approximation approaches are explored, namely, a linear interpolation (LI) approach and a nearest-neighbour (NN) approach. As expected, the LI approach predicts the results with less variability than NN, as well as providing a more diverse Pareto set. However, the NN method is relatively fast and simple, and provides a useful benchmark for the LI results. Attempts were also made to create a more sophisticated Kriging model of build orientation and residual stresses. Although this approach did not produce successful results, it provides a number of useful insights into the problem, as discussed in Appendix C.

5.2 Problem Description

As in Chapter 4, this problem considers N identical geometries, which are to be placed into identical bins of width W, length L and height H. The surface of the geometry is defined by a set of facets F, where each facet $f \in F$ has a target surface roughness denoted by Ra_{t_f}. As before, let p denote the geometry in its current build orientation, and let $C(p)$ denote the unit cost of the geometry resulting from p.

In addition to the unit cost, the presence of residual stresses must also be considered. It is well known that tensile residual stresses reduce the mechanical performance and fatigue life of metals, as they tend to further open cracks, while compressive stresses have the opposite effect (Leuders et al., 2013; Simonelli et al., 2014). It follows that a reduced overall tensile stress in the part is desirable. Thus, two additional optimisation criteria are introduced, namely, mean tensile stress, denoted by $\bar{\sigma}_t$, and maximum tensile stress, denoted by σ_{max}. While $\bar{\sigma}_t$ and σ_{max} are clearly related, each criterion provides different information about the magnitude and distribution of the residual stress profile and are therefore treated as separate competing objectives. Additionally, since $\bar{\sigma}_t$ and σ_{max} are independent of geometry, they can be calculated easily for a variety of parts regardless of their geometric complexities. The resulting objective function $\boldsymbol{F(p)}$ (which is a vector of the three competing objectives) is shown in Equation 5.1.

$$\min_{p(\theta,\phi)} \boldsymbol{F(p)} = [C(p), \bar{\sigma}_t(p), \sigma_{max}(p)] \tag{5.1}$$
$$\text{for} \quad \theta \in [0^o, 180^o], \quad \phi \in [0^o, 360^o)$$

where $\bar{\sigma}_t(p)$ and $\sigma_{max}(p)$ denote the mean residual tensile stress and the maximum residual tensile stress corresponding to p, respectively.

Given two feasible solutions p_1 and p_2, p_1 is said to dominate p_2 if $\boldsymbol{F(p_1)} \leq \boldsymbol{F(p_2)}$ and at least one of the following is true: $C(p_1) < C(p_2)$, $\bar{\sigma}_t(p_1) < \bar{\sigma}_t(p_2)$, $\sigma_{max}(p_1) < \sigma_{max}(p_2)$. If the solution space contains no other solutions that dominate p_1, then p_1 is considered a non-dominated (also known as Pareto-optimal) solution. Following these definitions, the aim of this problem is to provide a set of non-dominated feasible solutions, which is known as a Pareto set.

The values of $\bar{\sigma}_t(p)$ and $\sigma_{max}(p)$) are calculated using a Finite Element Analysis (FEA) approach, which is described in Section 5.3. Because SLM metals are particularly prone to brittle fracture (Rafi et al., 2013; Carter et al., 2014), the two proposed objectives, $\bar{\sigma}_t(p)$ and $\sigma_{max}(p)$ are calculated using the maximum principal stress (MPS) of each discrete element in the FEA solution. It should be noted that only elements belonging to the part geometry (i.e. excluding build plate and support structure elements are considered in these calculations. The FEA is performed separately for each solution p, and thus, it is important to note two key assumptions in this problem. Firstly, it is assumed that the FEA model is a sufficiently accurate representation of the "real" residual stress profile of p. Secondly, it is assumed that the residual stress profile is identical for all instances of p and is independent of the bin packing solution (e.g. neighbouring pieces do not affect one another).

5.3 Thermo-Mechanical FEA Model

FEA is a popular numerical method for solving complex problems and performing complex process simulations. This approach is able to simulate the temperature and

residual stresses in the SLM process by discretising the part geometry (and a section of the build plate) into small mesh elements, as discussed in Section 5.3.1, and solving a system of equations to determine the local temperature and stress corresponding to each individual element.

The simulation model assumes that the part geometry has a strong influence on the residual stress profile, as proposed by Keller et al. (2013), and consists of two stages. Firstly, a transient thermal analysis is performed to obtain the temperature distribution profile in the part geometry; secondly, a transient stress analysis is performed using the temperature profile determined in the previous stage to define the loading conditions. This two-stage thermo-mechanical analysis is known as loosely-coupled, since data only flows in one direction.

Although it is technically possible to simulate the laser path on each individual layer of powder, this is not computationally tractable for a real-size part. Thus, during the thermal analysis a whole layer of mesh elements is activated simultaneously and a heat flux of constant density is applied instantly to the whole layer. The heat flux density and exposure time are based on a number of parameters such as mesh element size, laser settings and cross-sectional area of the layer, and chosen so that the temperature in the current top layer of elements just exceeds the material melting temperature (to avoid convergence issues). Once the heat flux has been applied to an element, the material properties of that element are changed from "loose powder material" to "solid bulk material".

To represent the idling time in between laser scanning (e.g. during powder re-coating and repositioning of the laser), the heating step described above is followed by a cooling step. The cooling time during step is determined as a function of the layer cross-sectional area, laser scan speed and scan spacing. The heating and cooling steps are repeated for each layer of elements, until all layers have been activated. It should be noted that modelling the actual layer thickness used in SLM (e.g. 0.3 µm) would result in a prohibitive number of mesh elements and layers. Thus, to make the model feasible, each layer in the FEA represents a group of 10 - 30 layers in the actual SLM process. To account for convection heat losses during the thermal analysis, a heat transfer coefficient is applied to each activated layer of elements, while assuming ambient temperature inside the build chamber.

The stress analysis follows a very similar procedure to the thermal analysis. The same grouping of layers is used, and the same heating and cooling steps are applied to each layer of elements. The results of the thermal analysis are stored in a database, so the temperature of each element can be applied as a load on that element during the stress analysis. To prevent rigid body motion during the simulation, the part is assumed to be fixed to a section of the build plate, which in turn, is fully constrained throughout the process. A more detailed description of the full FEA model can be found in Palumbo and Granville (2016).

5.3.1 Mesh Generation

While a traditional tetrahedral mesh retains the most geometric detail and accuracy, it can also result in a very large number of elements when dealing with real-life components (i.e. $\gg 10^6$), making the FEA model computationally infeasible. This type of mesh also increases the risk of non-convergence due to its unstructured nature. A swept hexahedral mesh can significantly improve the tractability and robustness of the FEA model, however, it can be difficult to generate for very complex geometries (Wu et al., 2017). Consequently, a regular "voxel" mesh is used in this thesis, which approximates the geometry using identical cuboid elements. This provides a robust and simple mesh generation approach, and reduces the risk of non-convergence and intractability in the FEA, at the cost of some geometric error. This error is considered to be acceptable, as it is relatively small compared to the errors introduced by the assumptions discussed in Section 5.3.

The voxel mesh is generated in two stages: firstly, the part geometry must be rotated and meshed (the latter is done using a commercial meshing software called Harpoon); secondly, the steps shown in Figure 5.3 are used to generate additional mesh elements which represent the support structure and a section of the build plate. To create the additional mesh elements, the method generates a regular 2D grid in the horizontal plane, denoted by C, where each 2D cell $c \in C$ contains a "column" of all 3D elements in E whose x-y coordinates correspond to the centre point of c, as shown in Figure 5.3a-5.3b. The second step marks the bottom element of each column as supported, e_s, or unsupported, e_u, and generates nodes in the empty space underneath each e_s element as shown in Figure 5.3b. These nodes are then connected to create a set of vertically stacked 3D elements that form a support strut, denoted by s_c, as shown in red in Figure 5.3c.

To determine whether an element is supported or unsupported, the part mesh is compared to the STL geometry of the rotated part. As described in Section 3.4.1, each facet in the STL file is defined as supported if its build angle α_f exceeds α_{sup}. Thus, any element $e \in E$ that lies in close proximity of a supported facet is considered to be supported. The "close proximity" condition is considered to be true if the element e lies inside the bounding box of the supported facet f_s, and only applies to the bottom element (i.e. with minimum z-coordinate) of each column c, regardless of any concavities or "gaps" that may be present in c, as shown in Figure 5.1. In this way, the support structure mesh is prevented from obscuring the part geometry in the FEA model (e.g. by closing up holes and concavities). Elements which are very close to the build plate (e.g. $\leqslant 2\,\text{mm}$) are supported regardless of the surface angle.

As discussed in Section 3.4.1, performing an exact distance or intersection check a large number of times is computationally expensive and time consuming; while the above approach can occasionally result in additional support structure, as in the case of the two elements highlighted in red in Figure 5.1, it is fast and simple to implement, and provides sufficient contact between the part and the build plate, which is required

for a successful FEA solution.

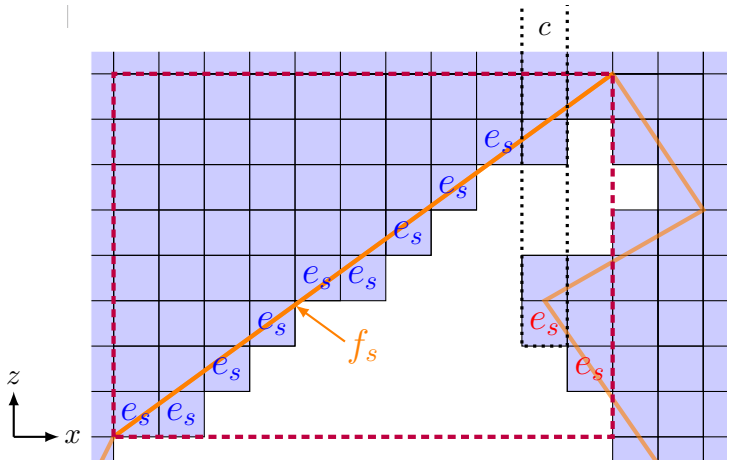
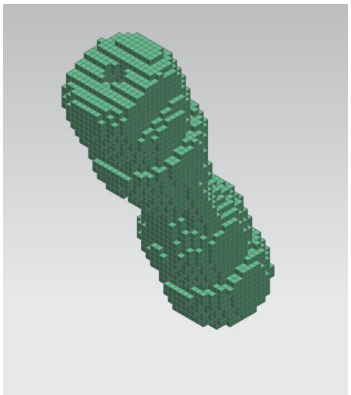
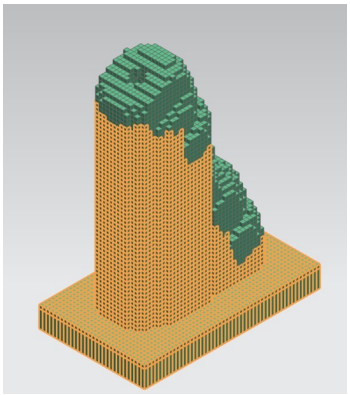


FIGURE 5.1: Schematic example of voxel mesh elements (blue) compared against the STL surface (orange). Each bottom element in the part mesh is classified as supported, e_s, or unsupported, e_u, based on the bounding box (dashed purple line) of the supported facet f_s. The two elements which are incorrectly associated with facet f_s are highlighted in red.

The third and final step generates the mesh representing a small section of the build plate, beneath the part and support structure. As shown in Figure 5.3d, this is done by generating a uniform 2D grid of nodes within user-specified dimensions, namely, an offset d around the horizontal bounding box of the part mesh. Since there cannot be two nodes in the same location, each node location in the 2D grid must be cross-referenced with the bottom nodes in the support structure. If a location already contains a support structure node, that node is marked as the top node in the build plate; otherwise, if the location is empty, a new node is generated to fill it, as shown in Figure 5.3e. As before, mesh elements are generated from each column of neighbouring nodes in the grid, where the height of each column is determined by the user-specified build plate thickness t, as shown in Figure 5.3f. An example of the final resulting 3D mesh can be seen in Figure 5.2.

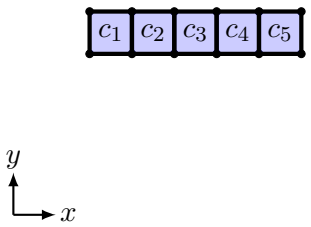


(A) Stage 1: Mesh of rotated part generated by Harpoon software.

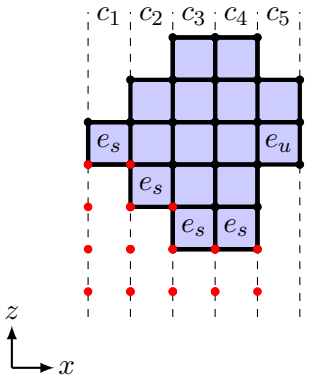


(B) Stage 2: Additional support structure and build plate mesh.

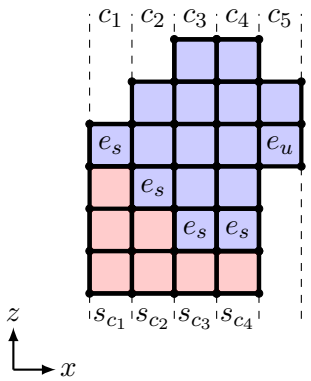
FIGURE 5.2: Example of the two-stage meshing procedure applied to a Venturi pipe geometry.



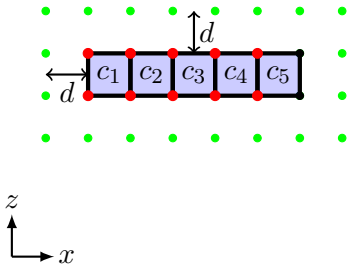
(A) Part mesh elements (blue) are sorted into a 2D grid, with columns denoted by $c_1, \ldots, c_5$.



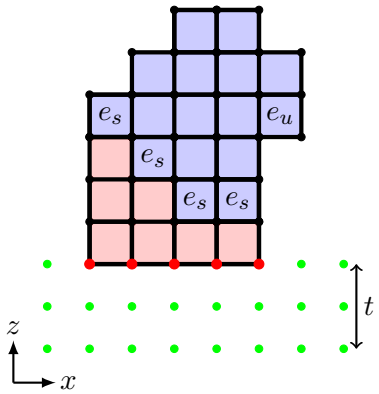
(B) Support structure nodes (red) are created below each supported element, denoted by e_s.



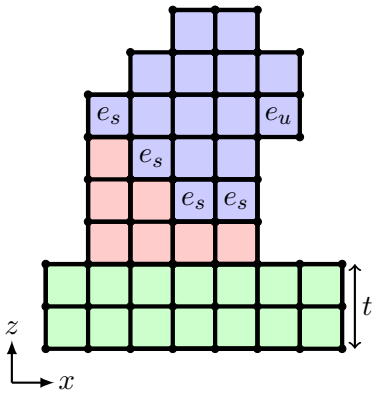
(C) A column of support structure elements, denoted by s_c, is generated for each column c.



(D) A 2D grid of build plate nodes (green) is created with an offset d around the part.



(E) Bottom support structure nodes (red) are marked as top build plate nodes. A 3D node grid is generated with height t.



(F) Columns of build plate elements (green) are generated from the 3D grid of new nodes.

FIGURE 5.3: Schematic description of support structure and build plate mesh generation. For simplicity, the part is shown as a single row of elements.

5.4 Approximated Search Strategies

As shown in Chapter 4, Tabu search is an effective and flexible approach for optimising the build orientation of identical parts in SLM. TS is also a popular choice in the wider literature for addressing single-objective optimisation problems (Hajji et al., 2004, 2005; Karimi et al., 2010; Basu, 2012; Saka et al., 2016). In contrast, relatively few works (Baykasoglu, 2006; Jaeggi et al., 2008; Yang et al., 2013) have considered TS in the context of multi-objective optimisation (MOO), where evolutionary algorithms such as the non-dominated sorting genetic algorithm II (NSGAII), are often preferred. However, to work effectively GAs require a relatively large and diverse population, which comes at the expense of reduced computational efficiency (for example, the time complexity of NSGAII is $O(MN^2)$, where M is the number of objectives and N is the population size). Meanwhile, TS also explores several solutions (i.e. local neighbourhood) at each iteration, but is not limited by population size in the same way as GAs. Moreover, TS has the potential of finding good Pareto solutions in the highly non-linear build orientation solution space due to its intensified exploration of local neighbourhoods (Yang et al., 2013). Thus, this chapter builds upon the TS procedure described in Chapter 4, proposing two modified versions to deal with the MOO problem, as discussed below.

Despite the simplifying assumptions discussed in Section 5.3, the FEA model is still very expensive in terms of computational memory and time. This means that only a very limited number of solutions can be explored in a reasonable computational budget. To address this limitation, two alternative approximation-based search strategies are explored; namely, a nearest-neighbour Tabu search (NNTS) strategy, discussed in Section 5.4.3, and a linear interpolation (LITS) strategy, discussed in Section 5.4.2. A third approach based on a Kriging surrogate model has also been considered, however, this approach was not successful as discussed in Appendix C.

Each approximation strategy starts by computing a small preliminary sample of exact solutions P, which is evenly distributed across the solution space and can be used to predict other unknown solutions during the optimisation search. The sampling method, known as design of experiments (DOE), is discussed next.

5.4.1 Design of Experiments

An optimal DOE is one that captures as much information as possible with as few sampling points as possible. One of the most popular and flexible DOE strategies is the Latin Hypercube (LH) method proposed by McKay et al. (1979).

To produce N sampling points, the LH approach equally divides the design range of each input parameter into N levels and ensures that only one point can be placed in each level. Because levels are seeded and combined randomly, a good design is not guaranteed. Consequently, numerous strategies have been proposed (Ye, 1998; Shields and Zhang, 2016) to optimise the initial LH point distribution based on some given criteria (e.g. space-filling, distance between points, etc.), leading to what is known

as the Optimal Latin Hypercube (OLH) approach. This approach can produce an effective sampling plan relatively quickly and cheaply when generating a small DOE (for instance, up to 100 points and fewer than 10 input parameters (Liefvendahl and Stocki, 2006)), making it suitable for the problem described in this chapter. For the build orientation solution domain, the OLH approach provides an additional advantage since it avoids placing multiple redundant points along the lines of $\theta = 0^{\circ}$ and $\theta = 180^{\circ}$, where all solutions are identical as discussed in Section 3.4. As shown in Figure 5.4, the DOE procedure is parallelised to further save time on the FEA.

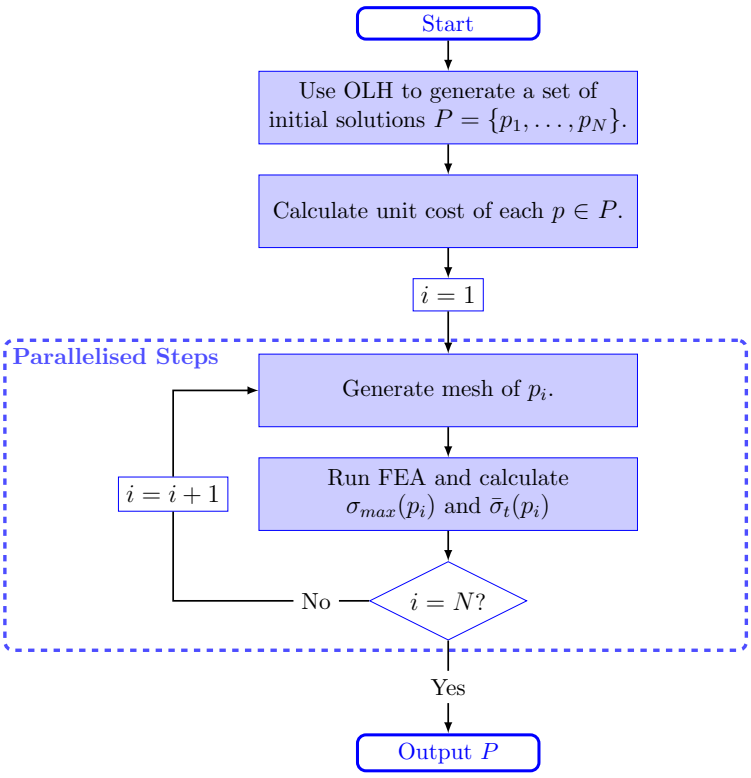


FIGURE 5.4: Flow diagram of DOE procedure.

Finally, the solution space symmetry discussed in Section 4.2.1 is particularly important to consider when generating the DOE and performing the optimisation search, since duplicated solutions can waste a lot of time and computational effort. To avoid this, the initial orientation of the part geometry and the input ranges of θ and ϕ should be adjusted following the approach discussed in Section 4.2.1.

5.4.2 Nearest-Neighbour Tabu Search

The NNTS algorithm and its neighbourhood construction procedure are shown in figures 5.5 and 5.6, respectively. As shown in Figure 5.5, the first step of the procedure is to generate an initial set of exact solutions P using the DOE procedure discussed in Section 5.4.1. The second step is to find an initial Pareto set denoted by P^*, where $P^* \subseteq P$. As can be seen in line 3 of Figure 5.5, the TS is performed multiple times, starting from each initial non-dominated solution $p^* \in P^*$. This helps to diversify the search and makes it suitable for parallelisation. The termination condition is met

when the number of consecutive no-improvement iterations I_n exceeds $I_{n_{max}}$; a no-improvement iteration is one where not a single non-dominated solution was found in the local neighbourhood P^N.

Input: $\mathcal{E}, L_{size}, I_{n_{max}}$
Result: Final Pareto set P^*
1 Generate a set of exact solutions P using DOE procedure in Figure 5.4;
2 Find initial non-dominated Pareto set P^*;
3 **foreach** $p^* \in P^*$ **do**
4 $p_c \leftarrow p^*$;
5 $I_n \leftarrow \varnothing$;
6 $TabuList \leftarrow \emptyset$;
7 $AddSolList \leftarrow \emptyset$;
8 $k \leftarrow \varnothing$;
9 **while** $I_n < I_{n_{max}}$ **do**
10 Construct neighbourhood $P^N(p_c)$ for p_c using procedure in Figure 5.6;
11 Order P^N by non-decreasing cost $C(p)$ and return first solution $p_1^N \in P^N$ with minimum C;
12 Set $p_c \leftarrow p_1^N$;
13 **if** $TabuList.Size > L_{size}$ **then**
14 Remove oldest entry in $TabuList$;
15 **end**
16 Add p_c to $TabuList$;
17 **if** P^N *contains non-dominated solutions* **then**
18 $I_n \leftarrow \varnothing$;
19 **end**
20 **else**
21 $I_n \leftarrow I_n + 1$;
22 **end**
23 $k \leftarrow k + 1$;
24 **if** $k \geq K$ **then**
25 Add p_c to $AddSolList$;
26 $k \leftarrow \varnothing$;
27 **end**
28 **end**
29 **end**
30 **foreach** $p^* \in P^*$ **do**
31 **if** $p^* \notin P$ **then**
32 Run FEA and update $\bar{\sigma}(p^*)$ and $\sigma_{max}(p^*)$;
33 Add p^* to P;
34 **end**
35 **end**
36 **foreach** $p \in AddSolList$ **do**
37 **if** $p \notin P$ **then**
38 Run FEA and update $\bar{\sigma}(p)$ and $\sigma_{max}(p)$;
39 Add p to P;
40 **end**
41 Compare p against P^* and add p to P^* if p is non-dominated;
42 **end**

FIGURE 5.5: Pseudo-code of NNTS procedure, where p_c denotes the current solution and p^* denotes a non-dominated solution.

To determine if a solution p^N in the current local neighbourhood P^N is non-dominated, p^N must be checked against each current non-dominated solution $p^* \in P^*$. This non-dominance check is described by lines 11-18 in Figure 5.6. If the condition in line 12 is true for any solution p^* then P^N is non-dominated and must be added

Input: p_c, $\mathcal{E}$, ξ_{crit}, P, P^*
Result: P^N, updated P^*
1 Create an empty list $P^N(p_c)$ which denotes the neighbourhood of p_c, using step size $\mathcal{E}$;
2 **foreach** $p^N \in P^N$ **do**
3 Calculate the unit cost $C(p^N)$;
4 **foreach** $p \in P$ **do**
5 Find distance $\delta \leftarrow \sqrt{(\theta_p - \theta_{p^N})^2 + (\phi_p - \phi_{p^N})^2}$;
6 **end**
7 Order P by non-decreasing $\delta(p)$ and take the first solution $p_1 \in P$ with minimum δ;
8 Set $\sigma_{max}(p^N) \leftarrow \sigma_{max}(p_1)$;
9 Set $\bar{\sigma}(p^N) \leftarrow \bar{\sigma}(p_1)$;
10 **foreach** $p^* \in P^*$ **do**
11 **if** $C(p^*) > C(p^N)$ *or* $\bar{\sigma}(p^*) > \bar{\sigma}(p^N)$ *or* $\sigma_{max}(p^*) > \sigma_{max}(p^N)$ **then**
12 **if** $p^N \notin P^*$ *and* $\psi(p^N) < \xi_{crit}$ **then**
13 Add p^N to P^*;
14 **end**
15 **if** $C(p^*) \geq C(p^N)$ *and* $\sigma_{max}(p^*) \geq \sigma_{max}(p^N)$ *and* $\bar{\sigma}(p^*) \geq \bar{\sigma}(p^N)$ **then**
16 Remove p^* from P^*;
17 **end**
18 **end**
19 **end**
20 **end**
21 Return P^N and updated P^*;

FIGURE 5.6: NNTS neighbourhood construction procedure, where p_c denotes the current solution, P denotes a set of exact solutions and P^* denotes a set of non-dominated solutions.

to P^*. Additionally, if any solution $p^* \in P^*$ is dominated completely by p^N, that solution must be removed from the list as shown in lines 15-17.

The local neighbourhood structure used by NNTS is the same as in Chapter 4 and can be found in Figure 4.2. It should be noted that the area-based approximation strategy discussed in Chapter 4 is also incorporated into the NNTS procedure, although it is not shown in Figure 5.5 for the sake of clarity. As in Chapter 4, the cost of a solution p may be approximated based on the difference in area between p and a previous exact solution. However, if an approximated solution p appears to be non-dominated, its cost must be calculated exactly (i.e. by solving the 2DIBPP) and p must be checked for non-dominance once again.

For an unknown solution p, the stress criteria $\bar{\sigma}(p)$ and $\sigma_{max}(p)$ are approximated based on the nearest known solution $p \in P$ in the solution space (i.e. nearest neighbour). As shown in Figure 5.6, the distance δ between an unknown solution p^N in the local neighbourhood P^N, and each known solution p is denoted by δ, and calculated as a function of the respective θ and ϕ coordinates of the two solutions. Since no new values of $\bar{\sigma}$ and σ_{max} can be introduced during the NNTS procedure (other than those already contained in P), only the unit cost C is used to direct the movements of the search, as shown in line 11 of Figure 5.5. In order to diversify the search, the NNTS also records the current solution p_c at every K^{th} iteration in a separate list, denoted by $AddSolList$. At the end of the procedure, the FEA is solved for each solution $p \in AddSolList$, along with the Pareto set P^*. Any non-dominated solutions in $AddSolList$ are then added to P^*.

The errors in $\bar{\sigma}$ and σ_{max} produced by this approach can be expected to be relatively large due to its oversimplification of the solution space. Therefore, as shown in Figure 5.5, the final stage of the NNTS solves the FEA model to obtain exact values of $\bar{\sigma}$ and σ_{max} for each approximated solution in the new Pareto set P^*.

5.4.3 Linear Interpolation Tabu Search

Due to the high predictive error of the NN approach, the LITS method is considered as an alternative. Similarly to NNTS, LITS starts with a set of exact solutions P generated by the DOE, from which an initial Pareto set P^* is derived. Once again, a multi-start Tabu search is performed starting from each non-dominated solution $p^* \in P^*$, as shown in Figure 5.7, and each search is terminated after $I_{n_{max}}$ consecutive iterations fail to find a non-dominated solution. LITS also uses the area-based approximation strategy proposed in Chapter 4, with the exception of non-dominated solutions for which the exact cost equation must be solved.

To approximate the stress criteria, the LITS model uses the set of exact solutions P to create two triangulated surfaces – one corresponding to $\bar{\sigma}$ and the other to σ_{max}. Each vertex of the triangulated surface corresponds to a solution $p \in P$; thus, an unknown solution p^N is assumed to lie on the same plane as the three solutions in P, which make up the vertices of the triangular facet containing p^N.

In addition to the different approximation strategies, a key difference between LITS and NNTS is the evaluation function used to drive the movements of the two searches. Because LITS is able to introduce new values of $\bar{\sigma}$ and σ_{max} during the optimisation stage, the evaluation function of each neighbour $p^N \in P^N$ is determined by the change in all three optimisation criteria, denoted by δF, as shown in line 9 of Figure 5.7 and defined in Equation 5.2.

$$\delta F(p^N) = \delta C + \delta\bar{\sigma} + \delta\sigma_{max} \tag{5.2}$$

$$\delta C = \frac{C(p^N) - C(p_c)}{C(p_c)}, \quad \delta\bar{\sigma} = \frac{\bar{\sigma}(p^N) - \bar{\sigma}(p_c)}{\bar{\sigma}(p_c)}, \quad \delta\sigma_{max} = \frac{\sigma_{max}(p^N) - \sigma_{max}(p_c)}{\sigma_{max}(p_c)}$$

where p_c is the current solution and p^N is a neighbouring solution of p_c, and C, $\bar{\sigma}$ and σ_{max} denote the three optimisation criteria corresponding to each solution.

Although the estimation errors introduced by LITS are expected to be smaller than NNTS, the FEA model must still be solved for all new solutions in the updated Pareto set P^* after the optimisation stage, as shown in Figure 5.7.

5.5 Implementation

The FEA model was provided by the Process Modelling Group within Rolls-Royce plc and was coded in ANSYS Mechanical APDL (version 14). All FEA results were computed using a mesh size of 0.5 mm and a layer thickness of 1 mm. For the meshing procedure described in Section 5.3.1, the software Harpoon (version 5.6) was used.

Input: $\mathcal{E}, L_{size}, I_{n_{max}}$
Result: Final Pareto set P^*
Generate a set of exact solutions P using DOE procedure in Figure 5.4;
Find initial non-dominated Pareto set P^*;
foreach $p^* \in P^*$ **do**
    $p_c \leftarrow p^*$;
    $I_n \leftarrow \varnothing$;
    $TabuList \leftarrow \emptyset$;
    while $I_n < I_{n_{max}}$ **do**
        Construct neighbourhood $P^N(p_c)$ for p_c using procedure in Figure 5.6;
        Order P^N by non-decreasing $\delta F(p^N)$ and return the first solution $p_1^N \in P^N$ with min. δF;
        Set $p_c \leftarrow p_1^N$;
        if $TabuList.Size > L_{size}$ **then**
            Remove oldest entry in $TabuList$;
        end
        Add p_c to $TabuList$;
        if P^N *contains non-dominated solutions* **then**
            $I_n \leftarrow \varnothing$;
        end
        else
            $I_n \leftarrow I_n + 1$;
        end
    end
end
foreach $p^* \in P^*$ **do**
    if $p^* \notin P$ **then**
        Run FEA and update $\bar{\sigma}(p^*)$ and $\sigma_{max}(p^*)$;
        Add p^* to P;
    end
end

FIGURE 5.7: Pseudo-code of NNTS procedure, where p_c denotes the current solution and p^* denotes a non-dominated solution.

When creating the build plate mesh, the offset d and thickness t (defined in Figure 5.3d) were both set to 3 mm. All meshing and FEA solutions were computed in parallel using 15 nodes on a Windows HPC cluster, with 16 cores in each node.

The LITS and NNTS models were applied to a single test geometry of medium complexity, which is shown in Figure 5.8. For both procedures, the step size $\mathcal{E}$ was set to 5^o and the non-improvement termination condition $I_{n_{max}}$ was set to 5 iterations. Since both LITS and NNTS are run from multiple starting solutions, a more thorough and shorter local search can be performed around each starting solution. Thus, each procedure was performed with a step size of $\mathcal{E}$, and each search was terminated after 5 consecutive no-improvement iterations. The Tabu list size L_{size} was set to 100 in both cases. Additionally, in the case of NNTS the counter K was set to 5, meaning every 5^{th} current solution in NNTS was recorded and checked for non-dominance at the end of the procedure.

Both procedures were ran with the same initial set P, which was generated with 50 solutions, shown in Figure 5.9, using the OLH method provided by the commercial software Isight (version 5.6). Due to the symmetry of the curved plate geometry, the solution space was limited to the ranges $\theta \in [0^o, 180^o]$ and $\phi \in [0^o, 180^o]$.

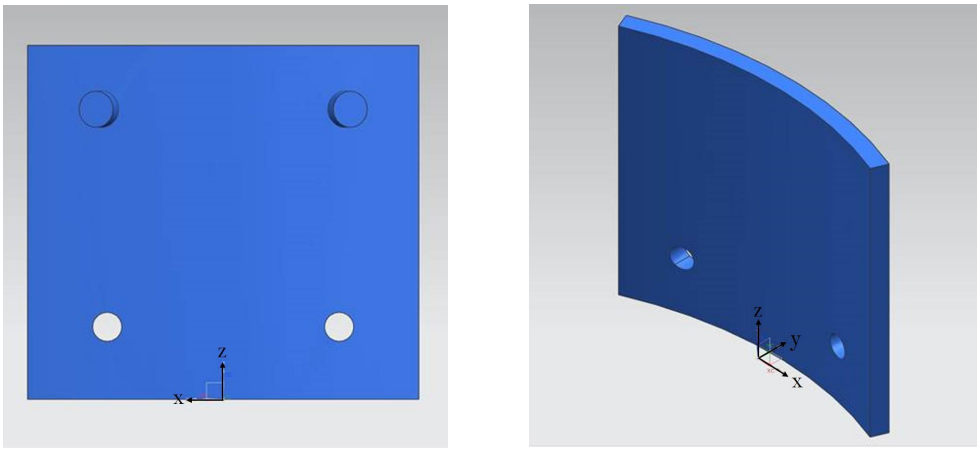


FIGURE 5.8: Curved plate test geometry containing two bosses and two holes.

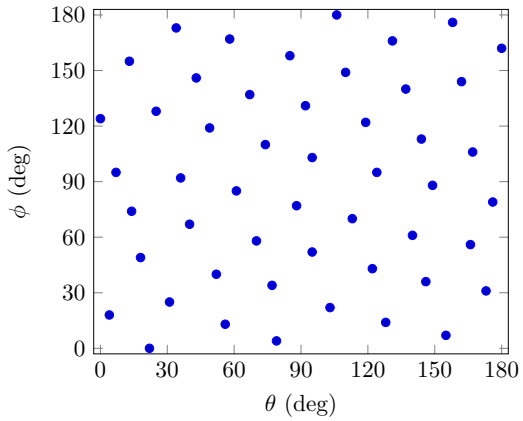
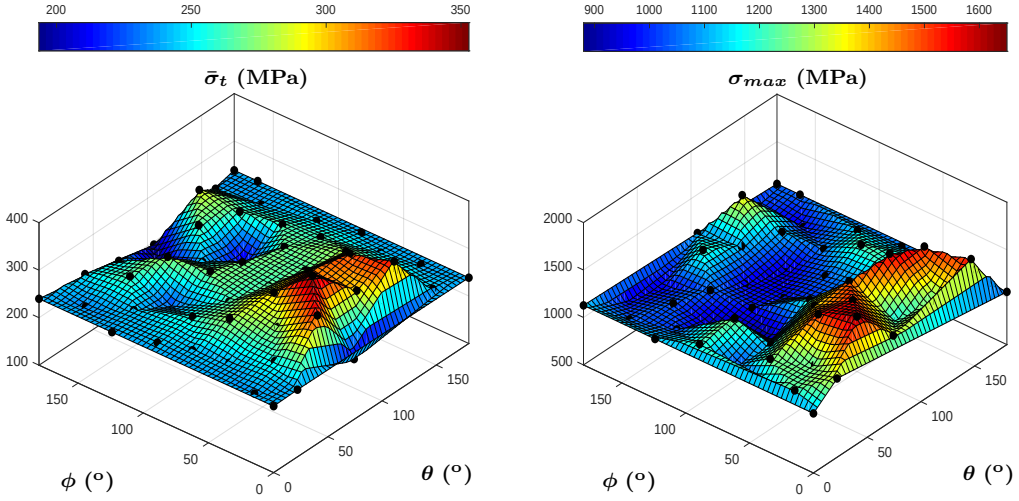


FIGURE 5.9: Set P produced by the DOE with 50 points.

The mesh generation and TS optimisation models were all coded in C# using the Visual Studio 2012 development environment, and were parallelised where appropriate. The linear interpolation surfaces were created using an in-built function in MATLAB (version 9.3). All optimisation results (excluding FEA) were performed on a four i7-3820 core machine, with 2.70 GHz maximum frequency. All cost modelling coefficients are the same as in Chapter 4 and can be found in Table 4.1. Based on the results in Chapter 4, the value of A_{lim} is set to 0.05.

5.6 Computational Results

The set of solutions P and the two resulting linear interpolation surfaces for $\bar{\sigma}$ and σ_{max} can be seen in Figure 5.10. An initial Pareto set, denoted by P^*, was obtained from P with 13 non-dominated solutions. Each of these solutions was used as a starting point in both search strategies, resulting in 13 local searches during each strategy. NNTS terminated after a total of 94 iterations, taking 4.5 h to perform the search, while LITS terminated after 171 iterations and 6.5 h. At the end of each strategy, 32 FEA instances were solved for NNTS and 50 FEA instances were solved for LITS (in addition to P).



(A) Plot of $\bar{\sigma}_t$ in MPa.

(B) Plot of σ_{max} in MPa.

FIGURE 5.10: Linear interpolation surfaces produced using P (black points).

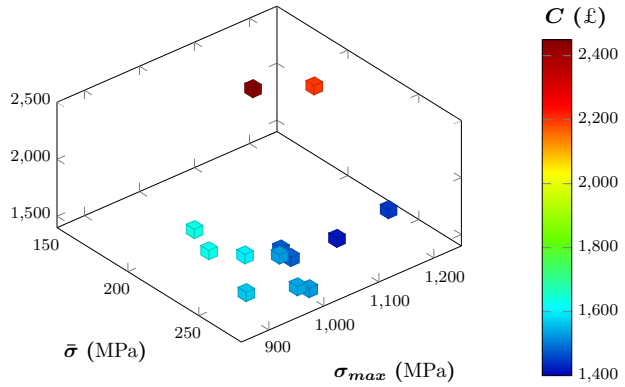
The initial approximated Pareto set produced during LITS contained 58 solutions; this set was reduced to 18 solutions after solving the FEA and updating the results. In the case of NNTS a much smaller Pareto set of 10 solutions was produced during the search. However, as discussed in the previous section, NNTS also recorded every 5^{th} current solution, producing an additional list of 23 solutions. After solving the FEA for all recorded solutions, the updated NNTS Pareto set contained a total of 13 solutions. It is interesting to note that only two solutions remained from the original Pareto set, and another two solutions were added from P^* back into the updated NNTS Pareto set, which were initially discarded due to predictive errors in $\bar{\sigma}$ and σ_{max}.

	LITS		NNTS	
	$\bar{\sigma}$	σ_{max}	$\bar{\sigma}$	σ_{max}
$\bar{E}$ (%)	3.8	1.7	-1.5	-2.5
ME (%)	31.9	15.3	24.3	11.6
RMSE (MPa)	16.42	78.43	20.26	80.25

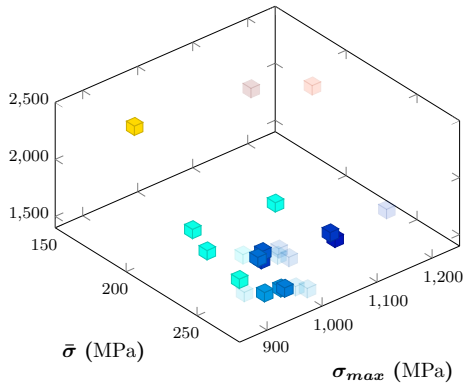
TABLE 5.1: Comparison of predictive errors for NNTS and LITS, where $\bar{E}$ is the mean error, ME is the maximum error and RMSE is the root-mean-square error.

Table 5.1 provides a summary of predictive errors produced by NNTS and LITS. As can be seen, LITS resulted in higher maximum errors and $\bar{\sigma}$ mean error; conversely, the σ_{max} mean error and the RMSE values are lower for LITS than for NNTS. This shows the variability in predictions provided by LITS is lower than those provided by NNTS, although high errors can result from both strategies.

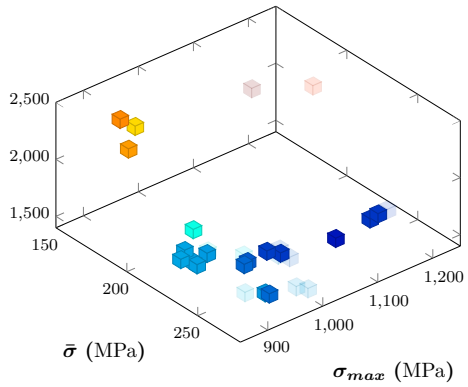
The predictive accuracy of both strategies can be improved by repeating each strategy multiple times, particularly NNTS, which is a more naive approach than LITS. At the end of each repetition, the updated exact solutions should be added to P, thus, increasing the pool of known solutions. This is also expected to encourage the approximated Pareto sets produced by NNTS and LITS to converge to the true global Pareto set. However, such repetitions come at the expense of significantly increased solution time.



(A) Initial Pareto set.



(B) NNTS Pareto set.



(C) LITS Pareto set.

FIGURE 5.11: 3D Pareto plots of the initial, NNTS and LITS Pareto sets. For comparison, figures (B) and (C) also show the initial Pareto set (opaque marks), and the same colour map is used to indicate the unit cost C in all three plots.

The final Pareto sets produced by NNTS, denoted by P^*_{nn}, and LITS, denoted by P^*_{li}, are shown in Figure 5.11. which includes the initial Pareto set P^* for comparison. As can be seen, the majority of solutions from P^* were discarded by both strategies, with the exception of two relatively strong solutions, which are highlighted in Figure 5.12 and discussed later. It can also be seen that LITS produced slightly stronger and

more diverse solutions in comparison to NNTS. This is not surprising, since LITS used all three criteria to drive the search, while NNTS relied solely on cost and random sampling. Figure 5.11 also indicates some clustering of solutions in both Pareto sets. This is generally undesirable, but can be addressed easily by discarding very similar solutions in P^*_{li} or by introducing an additional function during the LITS strategy to penalise any non-dominated solution that does not add significant diversity to the existing Pareto set.

NNTS Pareto Set (P^*_{nn})

	θ	ϕ	$\bar{\sigma}$	σ_{max}	C
1	0	124	240	1,127	1,417
2	3	180	235	1,006	1,446
3	4	23	237	1,126	1,445
4	10	163	232	1,016	1,474
5	15	128	236	997	1,491
6	35	128	256	912	1,634
7	40	123	264	937	1,525
8	49	119	227	928	1,627
9	70	88	265	964	1,519
10	85	93	265	971	1,502
11	86	180	156	979	2,063
12	114	117	215	1,083	1,627
13	119	122	209	948	1,632

Combined Pareto Set ($P^*_{li} \cup P^*_{nn}$)

	θ	ϕ	$\bar{\sigma}$	σ_{max}	C
p_1	0	124	240	1,127	1,417
p_2	3	180	235	1,006	1,446
p_3	4	8	239	1,192	1,443
p_4	4	13	238	1,209	1,444
p_5	4	23	237	1,126	1,445
p_6	8	180	231	1,025	1,450
p_7	10	163	232	1,016	1,474
p_8	15	149	234	982	1,474
p_9	23	180	236	971	1,485
p_{10}	40	123	264	937	1,525
p_{11}	45	119	221	942	1,528
p_{12}	51	115	223	912	1,530
p_{13}	54	109	226	911	1,532
p_{14}	55	103	213	922	1,535
p_{15}	56	100	218	902	1,536
p_{16}	59	95	263	947	1,486
p_{17}	86	180	156	979	2,063
p_{18}	90	178	173	923	2,113
p_{19}	91	180	152	962	2,129
p_{20}	119	122	209	948	1,632

LITS Pareto Set (P^*_{li})

	θ	ϕ	$\bar{\sigma}$	σ_{max}	C
1	0	124	240	1,127	1,417
2	3	175	236	1,037	1,446
3	4	8	239	1,192	1,443
4	4	13	238	1,209	1,444
5	8	180	231	1,025	1,450
6	15	149	234	982	1,474
7	23	180	236	971	1,485
8	40	123	264	937	1,525
9	45	119	221	942	1,528
10	51	115	223	912	1,530
11	54	109	226	911	1,532
12	55	103	213	922	1,535
13	56	100	218	902	1,536
14	59	95	263	947	1,486
15	86	180	156	979	2,063
16	90	178	173	923	2,113
17	91	180	152	962	2,129
18	119	122	209	948	1,632

FIGURE 5.12: Schematic comparison of Pareto sets produced by NNTS and LITS. The combined Pareto set resulting from the union $P^*_{li} \cup P^*_{nn}$ is shown in the middle, where solutions belonging to P^*_{nn} and P^*_{li} are colour-coded in red and blue, respectively, and solutions shared by P^*_{nn} and P^*_{li} are shown in black.

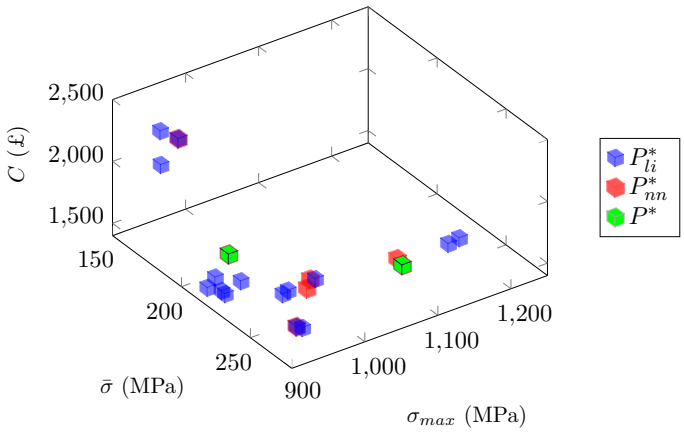
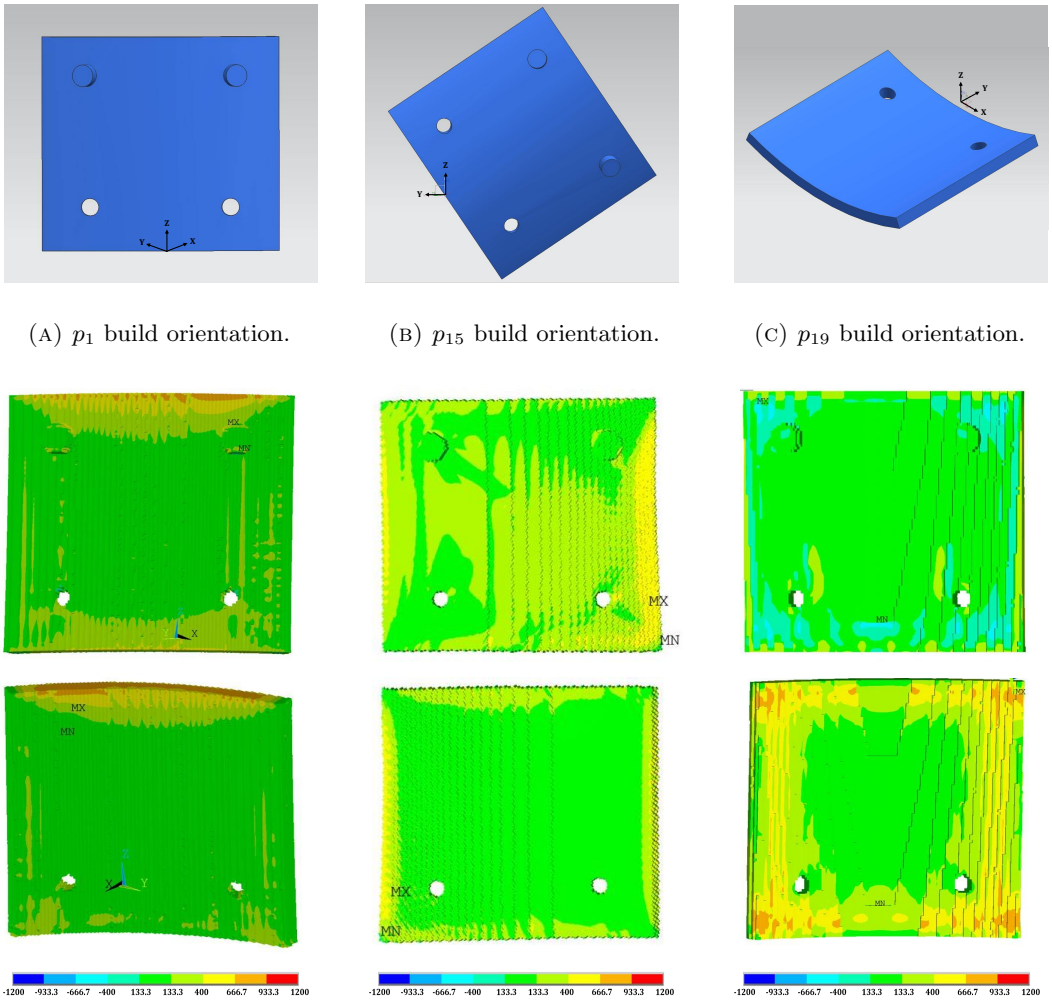


FIGURE 5.13: 3D plot of the combined NNTS and LITS Pareto set, where P^*_{li} denotes the LITS Pareto set, P^*_{nn} denotes the NNTS Pareto set, and P^* is the initial Pareto set.

To compare the LITS and NNTS strategies directly, the union $P^*_{li} \cup P^*_{nn}$ of the respective Pareto sets was considered, where all dominated solutions in $P^*_{li} \cup P^*_{nn}$ were discarded. The resulting combined Pareto set can be seen in Figure 5.12, where each solution p is colour-coded blue if $p \in P^*_{li}$, or red if $p \in P^*_{nn}$. Figure 5.13 provides a 3D plot of all non-dominated solutions found in the union $P^*_{li} \cup P^*_{nn}$. As can be seen, 15 of the 20 non-dominated solutions in $P^*_{li} \cup P^*_{nn}$ were produced by the LITS strategy. Moreover, two of the solutions in $P^*_{li} \cup P^*_{nn}$ were found in the initial Pareto set P^*,

and another two were found by both strategies. The two surviving solutions from P^* are highlighted in Figure 5.12; one of these solutions provides the lowest value of unit cost in $P^*_{li} \cup P^*_{nn}$ (p_1 in Figure 5.13), while the other provides well-rounded scores for all three criteria (p_{20} in Figure 5.13). As can be seen from Figure 5.12, two very good solutions of $\bar{\sigma}$ and σ_{max} were produced by LITS, and another one is shared by both strategies (p_{17} in Figure 5.13). However, it can be seen that these three solutions also have the highest values of C.



(A) p_1 build orientation. (B) p_{15} build orientation. (C) p_{19} build orientation.

(D) p_1 FEA. (E) p_{15} FEA. (F) p_{19} FEA.

FIGURE 5.14: Visual comparison of solutions p_1, p_{15} and p_{19} in $P^*_{li} \cup P^*_{nn}$, which are the best found solutions in terms of C, σ_{max}, $\bar{\sigma}$, respectively.

Figure 5.14 provides a visual comparison of the best solutions in terms of C, σ_{max}, $\bar{\sigma}$. As can be seen, Solution p_1 allows more parts to be placed in a single SLM machine, and thus, provides the best cost for this geometry; however, as shown by the FEA contour plot in Figure 5.14d, the flat top-facing edge contains a relatively high concentration of residual stresses, as can be expected. Conversely, making the concave face of the geometry top-facing, as done by p_{19} results in a smaller and

more gradual input of heat in each layer of material during SLM. As can be seen, this solution also produces regions of tensile stress in the top layers of the geometry, however, these regions are smaller and more distributed than in p_1. Solution p_{19} also has the added benefit of inducing compressive residual stresses in the convex surface of the geometry, as shown in Figure 5.14f; however, p_{19} can only fit two parts in a single build, making it relatively uneconomical. Meanwhile, it can be seen in Figure 5.14b that p_{15} provides the lowest value of σ_{max}, and a good overall residual stress profile, with the majority of tensile stresses not exceeding 400 MPa. This solution also provides a good value of C (£ 1,536), which is only moderately higher than the cost of p_1 (£ 1,417).

It is interesting to note that several; solutions in $P^*_{li} \cup P^*_{nn}$ (e.g. p_{15}), provide a better residual stress profile with only a moderate increase in cost over solution p_1, which was the "obvious" choice for a number of experienced SLM users in Rolls-Royce. This demonstrates the importance of considering the residual stress profile when selecting part build orientation in SLM, as well as considering a continuous build orientation solution space, since the best solution may not necessarily be the most obvious one.

Finally, it should be noted that σ_{max} alone may not be the best indicator of build failure risk for two reasons. Firstly, as shown in Figure 5.14 local concentrations of stress should also be taken into consideration, along with stress magnitude. Secondly, values of σ_{max} predicted by the FEA range between 900 MPa and 1200 MPa, which closely overlaps with the yield strength of Ti-6Al-4V (Kok et al., 2018) making it difficult to distinguish between high quality and low quality solutions. Thus, a more informative criterion could be based on the volume fraction of the part exceeding a critical stress value (e.g. yield stress or UTS). However, it is not known what this critical stress value should be exactly and what the acceptable fraction exceeding it should be; the former is strongly dependent on specific build and material conditions and so varies significantly in the literature, while the latter must be low enough to avoid build failure but high enough to avoid over-constraining the problem. Further experimental research is required to answer these questions, which is outside the scope of this thesis.

5.7 Conclusion

The multi-objective optimisation problem of unit cost and residual stresses in SLM is presented in this chapter. Since tensile stresses are known to be particularly damaging in metals, two simple measures are used to evaluate residual stresses based on the maximum principal stress criterion: mean tensile stress $\bar{\sigma}$ and maximum tensile stress σ_{max}.

Due to the prohibitive computational cost of the FEA used to predict residual stresses, two well-known approximation methods are applied to the problem, namely, a nearest-neighbour model and a linear interpolation model, based on an initial DOE.

The two approximation strategies are coupled with a multi-start Tabu search, which is based on the work presented in Chapter 4 and is extended to solve the multi-objective problem. The nearest-neighbour Tabu search (NNTS) provides a simple and fast albeit naive method for finding a Pareto set. Because this approach is unable find new values of stress (other than those already provided by the DOE) NNTS uses unit cost only to drive the movements of the search. Conversely, the linear interpolation Tabu search (LITS), which can interpolate new values of $\bar{\sigma}$ and σ_{max}, uses all three objective criteria simultaneously to select movements at each iteration.

Both strategies were demonstrated on a single geometry of medium complexity, and candidate solutions were discussed. It was shown that very good solutions (in terms of residual stresses and cost) may not be the most obvious ones, even to an experienced AM user. Thus, it is important to consider the residual stress profile when selecting a build orientation for SLM; moreover, it is suggested that continuous rotation should be selected over discrete methods (such as the 3D convex hull approach discussed in Chapter 2), as the latter risks discarding good solutions.

Despite slightly higher maximum errors, LITS showed lower variability in the predicted results compared to NNTS, as can be expected. LITS also explored almost twice as many candidate solutions as NNTS, and produced a more diverse Pareto set with more non-dominated solutions than NNTS, which can be explained by the more balanced movement strategy used by LITS. However, as mentioned previously, the diversity of both strategies could be improved further with an additional diversification function that penalises very similar non-dominated solutions.

Finally, it is suggested that a more informative stress criterion could be based on the volume fraction of the part exceeding a critical stress value, which could be the subject of future research.

Chapter 6

Cost Optimisation for Mixed Batch Production

This chapter outlines the problem of cost-driven process planning for the MBP scenario. The problem consists of three sub-problems: build orientation of multiple parts, bin assignment and bin packing of pieces. As discussed in Chapter 2, existing literature has addressed the build orientation of individual parts and bin packing (into a single bin) as two separate problems, while bin assignment in AM is yet to be addressed properly. Furthermore, there is currently no method to address all three aspects of this problem simultaneously.

Thus, the main contribution of this chapter lies in formalising the above problem and providing an iterative heuristic to optimise build orientation and bin packing in a coupled way. The results of the proposed iterative heuristic are compared against benchmark results provided by a commercial software, which solves the build orientation and bin packing problems separately. The heuristic outperformed the benchmark by an average of 14.6%, in terms of build cost improvement, based on 27 test instances.

It should be noted that since the MBP problem is much more complex than the IBP problem (addressed in Chapter 4), post-processing cost is not considered in this chapter to avoid adding further complexity to the MBP problem. However, the proposed heuristic can be easily extended to include post-processing as part of future work. The work presented in this chapter has been converted into an article and submitted to the European Journal of Operational Research.

6.1 Problem Description

Let $G = \{g_1, \ldots, g_N\}$ be a set of N unique 3D geometries, which are to be placed into identical bins; it is assumed that enough bins are available to accommodate all provided geometries, and the width, length and height of each bin are denoted by W, L and H, respectively. To account for multiple instances of identical geometries, each $g \in G$ corresponds to a quantity q_g in $Q = \{q_1, \ldots, q_N\}$. Because G contains different unique geometries, the algorithm solving the MBP problem, must not only consider the combined build orientation and bin packing problem, as in Chapter 4, but also

the bin assignment of different geometries. Thus, the MBP problem is addressed in the following two stages.

In the first stage, a favourable build orientation must be selected for each $g \in G$. Each build orientation of g is represented by a 2D piece, denoted by p_g. The quality of each piece p_g is determined based on its area a_p, build height h_p and total support structure volume s_p. As before, h_p may not exceed H. A solution to this stage produces a set of unique pieces $P = \{p_1, \ldots, p_N\}$, where each piece $p_g \in P$ corresponds to a geometry $g \in G$.

The second stage must solve a 2D bin packing problem (2DBPP) for the set of pieces P, by assigning and packing all pieces into a minimum number of identical bins of width W and length L. Since each unique geometry g has a quantity q_g, the 2DBPP must pack q_g copies of each piece $p_g \in P$. This means that, by default, the solution will place all instances of g in the same build orientation, which addresses the build orientation constraint for critical parts discussed in Section 3.1.2. To allow multiple instances of a non-critical geometry g to have different build orientations in the same solution, each instance of g may be added separately to the set G; if this is done, the quantity q_g must also be updated. For example, if g has a quantity $q_g = 4$, then it can be added as four separate geometries in G, where each geometry has a quantity $q_g = 1$.

A solution to the second stage is given by a set of bins $B = \{b_1, \ldots, b_M\}$, where M is the total number of bins. Each bin is denoted by $b_i(P_i, Q_i, p_i, X_i, Y_i)$, where $P_i \subseteq P$ and the vector Q_i lists the quantity of each piece in P_i. During the search for a packing solution (described in Section 3.5) the pieces are allowed to rotate continuously about the vertical z-axis, hence, $p_i = \{p_1, \ldots, p_n\}$ defines the 2D orientation of each piece. As before, $X_i = \{x_1, \ldots, x_n\}$ and $Y_i = \{y_1, \ldots, y_n\}$ define the x and y coordinates of the reference point of each piece, respectively. The reference point of a piece is the bottom left corner of its enclosing rectangle, corresponding to its 2D orientation. Every piece in P must be placed completely inside the bins and may not overlap with any other pieces or the no-build zones[1].

The objective is to find a solution with M bins which yields the minimum total production cost, as shown in Equation 6.1, where c_i denotes the cost of bin i. Equation 3.1, which defines SLM build cost in Chapter 3, is adapted for the MPB problem and provided in Equation 6.2.

$$\min \sum_{i=1}^{M} c_i \tag{6.1}$$

$$c_i = c_1 t_i + c_2 \rho \sum_{k=1}^{n_i} q_k (v_k + s_k) \tag{6.2}$$

where c_1 and c_2 are coefficients corresponding to indirect cost and material cost, respectively; t_i is the build time of bin i; and ρ is the material density. The number

[1] Please refer to Figure 3.17 in Chapter 3)

of pieces in bin i is denoted by n_i, where each piece k has a geometry volume v_k, support structure volume s_k, and corresponds to a quantity q_k. Similarly, Equation 6.3 provides an adapted version of build time taken from Equation 3.3.

$$t_i = \beta_1 + \beta_2 h_i + \beta_3 \sum_{k=1}^{n_i} q_k v_k + \beta_4 \sum_{k=1}^{n_i} q_k s_k \tag{6.3}$$

where β_1 - β_4 are regression model coefficients and h_i is the build height of bin i. Combining equations 6.2 and 6.3 provides the following definition of c_i for the MBP problem:

$$c_i = \omega_1 + \omega_2 h_i + \omega_3 \sum_{k=1}^{n_i} q_k s_k \tag{6.4}$$

$$\omega_1 = c_1\beta_1 + (c_1\beta_3 + c_2\rho) \sum_{k=1}^{n_i} q_k v_k, \quad \omega_2 = c_1\beta_2, \quad \omega_3 = c_1\beta_4 + c_2\rho \tag{6.5}$$

where $\omega_1, \ldots, \omega_3$ are objective function coefficients. As before each piece in the solution is defined by the angles θ and ϕ, such that $\theta \in [0^o, 180^o]$ and $\phi \in [0^o, 360^o)$. For any geometry $g \in G$ which contains symmetries, the initial build orientation and input ranges of θ and ϕ should be modified as discussed in Section 4.2.1.

6.2 Iterative Tabu Search Procedure (ITSP)

From equations 6.1 and 6.4, it can be observed that the total build cost for a MBP order is driven by the number of bins, the build height, and the total support structure volume of each bin. As mentioned above, the problem must consider not only the build orientation of each unique geometry $g \in G$, but also the bin assignment and bin packing solution of their corresponding pieces. This results in a very large and complex solution space, and what is known as an NP-hard problem. Currently, it is infeasible to solve NP-hard problems in reasonable time by any exhaustive approaches, and so a carefully designed heuristic approach is required.

makes the solution space much larger and more complex than in the case of the IBP problem. Consequently, the Tabu search heuristic, which is proposed in Chapter 3, is unlikely to be able to explore the solution space sufficiently in reasonable time. Thus, a new heuristic approach is developed in this chapter, which aims to guide the TS to different promising regions of the solution space. This approach is called the Iterative Tabu Search Procedure (ITSP) and consists of six distinct stages, where each stage focuses on a different component of SLM build cost (Equation 6.4).

In the first stage, the ITSP generates an initial solution and establishes a lower bound for the number of bins. This is achieved in the following steps. Firstly, an initial set of 2D pieces P is generated for the input set of geometries G. The local Tabu search is performed on each piece $p \in P$, to find an improved piece p^* with the minimum piece area a^*. Secondly, a 2DIBPP is solved for the resulting set P^*,

producing a solution with the smallest set of bins B. The remaining five stages attempt to improve the cost of each bin $b \in B$, as follows:

Stage 2, the Single Bin Build Height Reduction (SHR) strategy aims to reduce the build height of partially full bins, without disrupting the piece assignment.

Stage 3, the Single Bin Support Structure Volume Reduction (SSVR) strategy reduces the support structure volume of each piece inside each partially full bin, without disrupting the piece assignment.

Stage 4, the Pairwise Bin Build Height Reduction (PHR) strategy reduces the build height of bins in pairs, allowing pieces to move between paired bins, in the hope of improving the bin assignment.

Stage 5, the Pairwise Bin Support Structure Volume Reduction (PSVR) strategy reduces the support structure volume of bins in pairs; once again, allowing pieces to move between paired bins to further improve bin assignment.

Stage 6, the Bin Addition and Random Reassignment (BARR) strategy opens a new empty bin and moves pieces selected randomly from existing bins into the new bin. The remaining pieces in each disrupted bin are re-oriented and repacked to reduce its cost.

Figure 6.1 provides a schematic summary of key steps in each of the six stages; where G denotes a set of geometries with a corresponding set of quantities Q; P denotes a set of pieces, where each piece p has a corresponding area a_p, height h_p and support structure volume s_p; and B is a set of bins in the solution, which is updated at each stage of the procedure. A detailed description of each stage is provided in the remainder of this section. As referenced in Figure 6.1, Section 6.2.1 provides a detailed methodology for SHR; Section 6.2.2 describes SSVR; Section 6.2.3 describes PHR; Section 6.2.4 describes PSVR; and Section 6.2.5 describes BARR. As indicted in Figure 6.1, each stage uses a local Tabu search to update pieces, as discussed in Section 6.3. Whenever pieces are updated a new 2DIBPP must be solved; to do this the algorithm discussed in Section 3.5 is used.

6.2.1 Single Bin Build Height Reduction (SHR)

A typical initial solution can be expected to contain at least one "weak" partially full bin, as demonstrated in Figure 6.2. Unlike in many traditional cutting and packing (C&P) problem applications, such as the ceramic tile and sheet metal industries, minimising the utilised space inside the weak bin does not offer any benefit in the context of SLM, since the wasted machine space cannot be recycled after the build. Instead, the build orientation of each piece in Figure 6.2a can be changed in a way that reduces the build cost of the bin, as shown in figures 6.2b and 6.2c. Modifying the build orientation of the two pieces will change their shape and likely increase their area, however, the chances of successfully repacking these two pieces into a single bin are high given the very low bin utilisation. The SHR strategy exploits this scenario.

As shown in Figure 6.1, the set of bins $B = \{b_1, \ldots, b_M\}$ given by the initial solution is sorted by non-increasing utilisation and only bins with $U_i < u_{weak}$ are

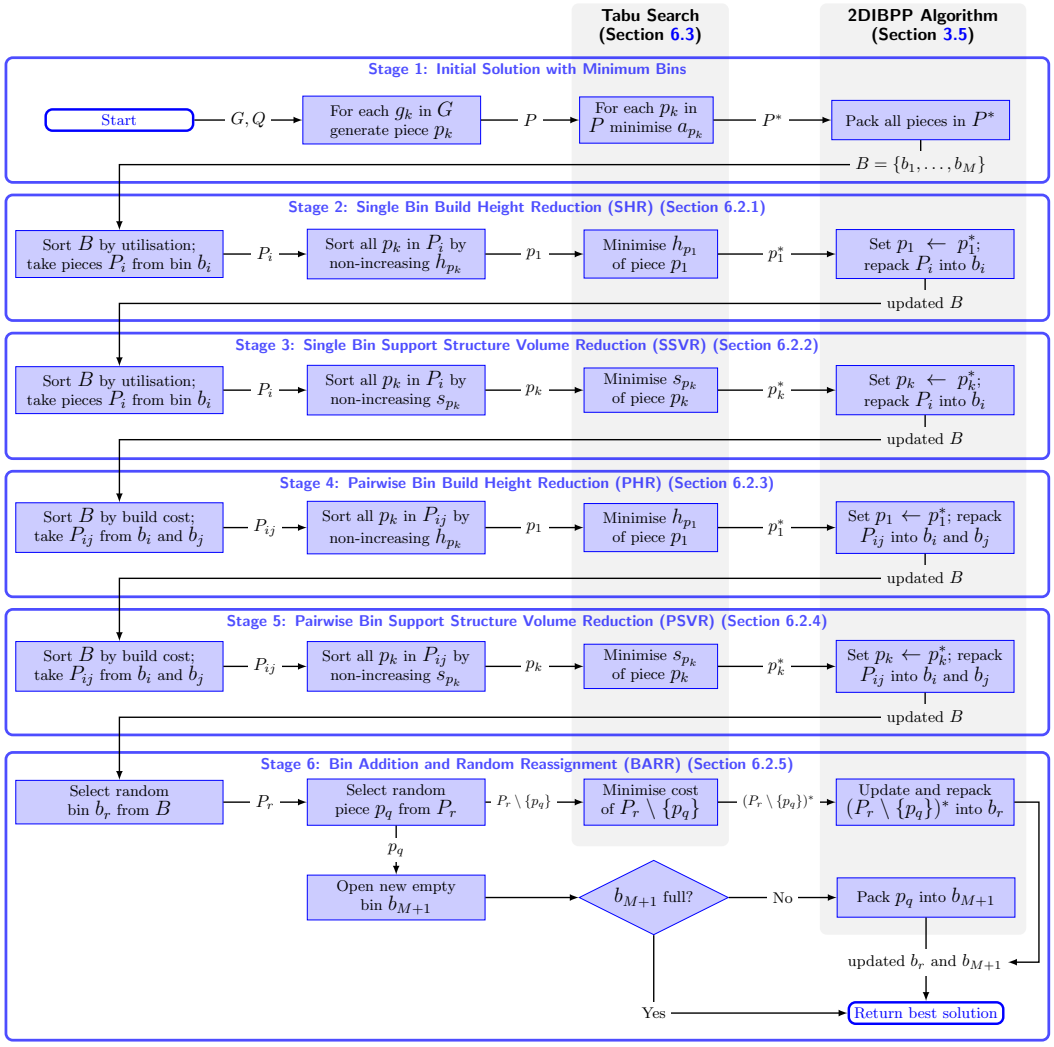
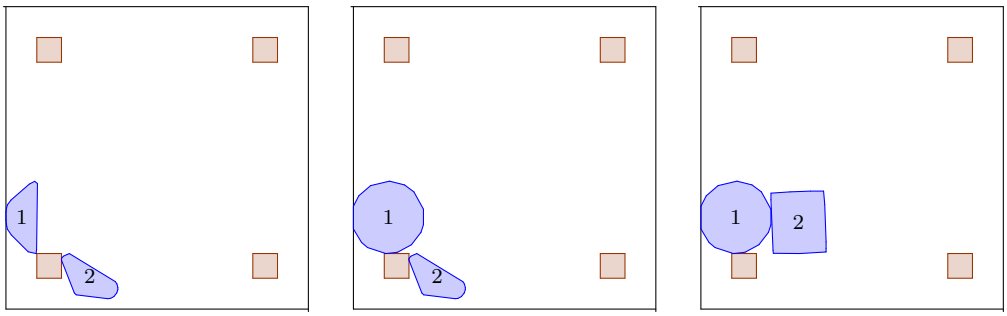


FIGURE 6.1: Schematic overview of the ITSP; for the sake of readability, only a single iteration of each stage is shown. Key steps in each stage are shown in blue boxes, while grey boxes indicate which steps use the Tabu search and the 2DIBPP algorithm in each stage, respectively.

considered, where u_{weak} denotes the threshold for weak bins. For each bin b_i, the set of pieces $P_i = \{1, 2, \ldots, n\}$ is sorted by non-increasing height. The tallest piece p_1 is selected and a local Tabu search is performed to minimise the height of p_1, where the resulting piece is denoted by p_1^*. If the height of p_1^* is lower than the height of p_1, P is updated, such that $p_1 \leftarrow p_1^*$, and repacked using the 2DIBPP algorithm. Since the number of bin in P is kept fixed during this stage, the bin assignment in the 2DIBPP is skipped and the problem is reduced to a single-bin packing problem.

If b_i contains multiple instances of p_1, those instances must also be updated. For this reason P_i contains only unique pieces, and has a corresponding set of quantities $Q_i = \{1, 2, \ldots, n\}$, thus, p_1 only needs to be updated once in P_i. However, if a bin other than b_i contains an instance of p_1 that bin must also be updated, so an additional check is performed on all bins in $B \setminus b_i$[2].

[2]Set B excluding b_i.



(A) $U_2 = 0.06$ (B) $U_2 = 0.06$ (C) $U_2 = 0.11$

FIGURE 6.2: Initial solution produced by the first stage in the ITSP, with average bin height of 116.5 mm. Improved solution produced by SHR, with average bin height reduced to 103.5 mm.

If all bins containing p_1 are repacked successfully, the solution is updated and the process repeats, until no further height reduction can be achieved in b_i. If the SHR strategy fails to achieve a height improvement, either during the Tabu search or during repacking, the strategy will move on to the next bin in B. The strategy terminates when all bins $b \in B$ with $U_b < u_{weak}$ have been addressed. It should be noted that any solution which successfully reduces build height is accepted by the SHR strategy, so it is possible for the current solution cost to temporarily increase during this stage in the ITSP.

6.2.2 Single Bin Support Structure Volume Reduction (SSVR)

SSVR follows a very similar procedure to SHR; the only differences being the iteration of pieces in P_i, and the Tabu search objective, where height is replaced with support volume structure. Because the build height of b_i is determined by the tallest piece in P_i, each iteration of SHR sorts P_i by non-decreasing height and only applies the TS to the first piece in P_i, denoted by p_1. In contrast, the support structure volume of each piece can be reduced independently of other pieces. Hence, the SSVR strategy sorts P_i only once, in order of non-decreasing support structure volume, and iterates through each piece p_k, until $k = n$, where n is the length of P_i. Similarly to SHR, the SSVR strategy accepts any solution which reduces the support structure volume, including solutions with increased build cost.

6.2.3 Pairwise Bin Build Height Reduction (PHR)

In addition to improving individual pieces in each bin, the bin assignment can also be improved. To do this the PHR and PSVR stages address bins in pairs. As before, the former addresses the height and the latter addresses the support structure volume of pieces. Unlike SHR and SSVR, PHR and PSVR address every possible pair of bins regardless of their utilisation. By pairing strong bins with weaker ones and by allowing the pieces to move between them, the PHR and PSVR strategies aim

to further utilise empty spaces within the weak bins and increase the freedom of movement inside strong bins.

As shown in Figure 6.1, the first step in PHR sorts the current set of bins B by non-increasing build cost. The second step takes a set of pieces $P_{ij} = P_i \cup P_j$ from a pair of bins b_i and b_j in B, where $i < j$. The third step orders P_{ij} by non-increasing height and the fourth step performs a Tabu search to minimise the height of the tallest piece p_1. If this search is successful, the resulting piece p_1^* is accepted, so that $p_1 \leftarrow p_1^*$ in P_{ij}, and the final step in PHR solves a 2DIBPP for P_{ij}. Additionally, if any bin b_a in $B \setminus b_i, b_j$ contains a duplicate of p_1, that bin must also be repacked, by solving a Knapsack problem for the updated set of pieces P_a. If the resulting Knapsack solution fails to place all pieces in P_a, the whole solution is rejected.

Finally, if P_{ij} is successfully packed into a maximum of two bins b_i^* and b_j^*, where the combined cost of b_i^* and b_j^* is lower than the cost of b_i and b_j, the new solution is accepted and B is updated. Following this, P_{ij} is sorted by non-increasing height again and the process repeats until no further cost improvement can be made to b_i and b_j; at which point PHR updates $i \leftarrow i + 1$ and $j \leftarrow j + 1$, and the above steps are repeated for the next pair of bins. PHR terminates when all pairs in B have been permuted.

6.2.4 Pairwise Bin Support Structure Volume Reduction (PSVR)

The PSVR strategy closely mirrors the steps in PHR, substituting support structure volume for height as the Tabu search objective. The main difference between these two strategies is that PSVR sorts P_{ij} by non-increasing support structure volume, once for every pair of bins, then performs a Tabu search to minimise the support structure volume of every piece p_k in P_{ij}, as opposed to only the tallest piece p_1 as done in PHR. The reason for this is explained in Section 6.2.2. After every successful Tabu search, all bins containing p_k must be repacked and the updated P_{ij} is repacked into a maximum of two bins. If the repacked solution is both feasible and lower in cost than the current solution, this solution is accepted as the current solution and B is updated.

6.2.5 Bin Addition and Random Reassignment (BARR)

Each of the four strategies described above tries to reduce the cost of the solution without increasing the overall number of bins. In contrast, this final stage of the ITSP explores the possibility of reducing the overall cost of the solution at the expense of adding an extra bin.

Following the improvement of individual bins in SHR and SSVR, and the bin assignment improvement in PHR and PSVR, it is going to be increasingly difficult for the ITSP procedure to find further feasible movements in the solution. Thus, the solution must be disrupted slightly in order to create addition free space in the current set of bins B. The BARR strategy does this by introducing a new empty

bin, denoted by b_{M+1}, and by reassigning pieces from the original bins in B to b_{M+1}. This leaves additional free space in the disrupted bins, which can be used to further improve the build orientations of remaining pieces.

To encourage solution space exploration, pieces to be moved are selected randomly. Thus, to select a piece for reassignment, the BARR strategy selects a random bin $b_r \in B$, then a random piece $p_q \in P_r$, where P_r is a set of pieces corresponding to b_r. To avoid re-visiting previously attempted pieces, p_q is stored in a "Tabu list" (which is unrelated to the actual Tabu search). If p_q is placed into b_{M+1} successfully, the overall cost of the remaining set of pieces in b_r is minimised using the local Tabu search. This is done iteratively for each piece in $P_r \setminus \{p_q\}$, using the change in cost to drive the search. For clarity, the details of this search are explained separately in Section 6.3.

If the Tabu search is successful, the next step is to repack the updated pieces, denoted by $(P_r \setminus \{p_q\})^*$, into b_r. If the packing is unsuccessful, a piece $p^* \in (P_r \setminus \{p_q\})^*$ with the smallest reduction in cost is reverted back to its original build orientation, such that $p^* \leftarrow p$, and the packing is attempted again. This process is repeated until all pieces are successfully packed or until all pieces have been reverted to their original state. In the latter case, the solution yields no cost improvement and is therefore rejected. The procedure terminates either when a maximum number of no-improvement iterations is reached or when b_{M+1} is full – whichever occurs first.

6.3 Local Tabu Search

The Tabu search heuristic has already been introduced and discussed in detail in Section 4.2. In addition to its simplicity and its ability to escape local minima, it is also more greedy than other popular heuristics (e.g. Simulated Annealing), since it must always move to the best available neighbour of the current solution. Furthermore, by adding visited solutions to the Tabu list, the TS is able to explore the solution space more effectively in a short amount of time, than a heuristic which is allowed to return to visited solutions.

As indicated in Section 6.2, each stage in the ITSP is required to perform a local search for numerous pieces in reasonable solution time, making TS a suitable local search heuristic due to its greedy nature. The local TS search used in the ITSP, has the same basic steps and local neighbourhood structure as those used in Chapter 4. For clarity, the updated pseudo-code is shown in Figure 6.3; where $\mathcal{E}$ denotes the local neighbourhood step size; L_{size} denotes the size of the Tabu list; and $I_{n_{max}}$ denotes the maximum number of consecutive non-improvement iterations. As outlined above, the attribute $Obj(p)$ used to evaluate piece p depend varies with each ITSP stage:

- For the initial solution $Obj(p) \leftarrow a_p$, where a_p is the area of p;
- For SHR and PHR $Obj(p) \leftarrow h_p$, where h_p is the height of p;

- For SSVR and PSVR $Obj(p) \leftarrow s_p$, where s_p is the support structure volume of p;
- Finally, for BARR $Obj(p) \leftarrow \Delta c_b$, where Δc_b is the change in cost of bin b, which contains p.

Input: $p_{in}, \mathcal{E}, L_{size}, I_{n_{max}}$
Result: p_b
$p_c \leftarrow p_{in}$;
$p_b \leftarrow p_c$;
$I_n \leftarrow \varnothing$;
$TabuList \leftarrow \emptyset$;
while $I_n < I_{n_{max}}$ **do**
  Construct neighbourhood $P^N(p_c)$ for p_c using step size $\mathcal{E}$;
  Return best solution $p' \in P^N(p_c)$;
  if $TabuList.Size > L_{size}$ **then**
    Remove oldest entry in $TabuList$;
  end
  Add p_c to $TabuList$;
  Set $p_c \leftarrow p'$;
  if $Obj(p_b) > Obj(p_c)$ **then**
    $I_n \leftarrow \varnothing$;
    $p_b \leftarrow p_c$;
  end
  else
    $I_n \leftarrow I_n + 1$;
  end
end

FIGURE 6.3: Pseudo-code of local TS used in the ITSP, where p_{in}, p_c and p_b denote the initial, current and best solution, respectively; and $Obj(p)$ denotes the piece attribute used to evaluate piece p.

To reduce the overall cost of bin b, the TS must consider the set of bin pieces P_b collectively. Prior to the search, the pieces in P_b are sorted by non-increasing height. Starting from the tallest piece, the algorithm attempts to improve the build orientation of each consecutive piece $p \in P$, with the objective of minimising the negative cost difference Δc_b of bin b. To calculate Δc Equation 6.6 is used at each iteration.

$$\Delta c_b = c_1 \Delta h + c_2 \Delta s \tag{6.6}$$

$$\text{where} \quad \Delta h = h_k - h_{in} \quad \text{and} \quad \Delta s = s_k - s_{in}$$

where Δh denotes the change in build height, Δs denotes the change in support structure volume, and c_1 and c_2 are their respective cost coefficients; h_{in} and s_{in} are the build height and support structure volume, respectively, corresponding to the initial piece in the local TS; while h_k and s_k are the build height and support structure, respectively, corresponding to the k^{th} iteration of the local TS. It follows that all pieces in P_b must be updated and sorted by non-increasing height after each

iteration of the TS. As can be seen from the above equations, negative values of Δc indicate a cost improvement, while positive values indicate a cost increase.

Because all pieces in P_b are updated simultaneously, the increase in area of each piece $p \in P_b$ must be moderated, so as to maximise the likelihood of b being repacked successfully after P_b is updated. Therefore, only a limited increase in area is permitted for each $p \in P_b$ as shown in line 16 of Figure 6.4, where a_{max} is an additional hyperparameter. The pseudo-code of this updated Tabu search is shown in Figure 6.4.

```
   Input: P_b, E, L_size, I_{n_max}, a_max
   Result: Updated P_b
 1 Sort P_b by non-increasing height;
 2 foreach p ∈ P_b do
 3     Set TabuList ← ∅;
 4     Set n ← ∅;
 5     Set current bin height h_b to height of tallest piece in P_b;
 6     Set initial area a_in ← a_p, where a_p is the area of p;
 7     Set current piece p_c ← p;
 8     while I_n < I_{n_max} do
 9         if TabuList.Size > L_size then
10             Remove oldest entry in TabuList;
11         end
12         Add p_c to TabuList;
13         Construct neighbourhood P^N(p_c) for p_c using step size E;
14         Return best neighbour p' ∈ P^N(p_c) with minimum Δc_p;
15         Set p_c ← p';
16         if (a_{p_c} − a_in)/a_in ≤ a_max then
17             if Δc_{p_c} < 0 then
18                 I_n ← ∅;
19                 Replace p with p_c in P_b and sort P_b by non-increasing height;
20                 Update current bin height h_b to height of tallest piece in P_b;
21             end
22             else
23                 I_n ← I_n + 1;
24             end
25         end
26         else
27             Exit while loop;
28         end
29     end
30 end
```

FIGURE 6.4: Pseudo-code of local TS used in the BARR strategy, where P_b denotes the set of pieces belonging to bin b and p_c denotes the current piece.

6.4 Implementation

The ITSP is implemented in C# using the Visual Studio 2012 development environment, and parallelised where appropriate. To solve the MIP problem for the 2DIBPP the commercial solver Gurobi (version 6.5.2) is used. Tests were performed on a i7-3820 four core machine, with 2.70 GHz maximum frequency, eight logical processors and 16GB of RAM.

To ensure good performance of the ITSP, a number of parameters needed to be tuned. Firstly, a utilisation of 0.85 was considered to be the threshold between strong and weak bins. Secondly, the acceptance condition for each new build orientation was set to a minimum reduction of 1 mm when improving build height (i.e. 50 layers), and minimum reduction of 100 mm^3 for support structure volume (i.e. just under a gram of material). Movements below these thresholds were not considered worthwhile since they would yield negligible cost improvements. In the SHR and SSVR stages, solutions are allowed to temporarily increase the cost up to a limit of 1%, in the hopes of finding a better solution by delaying convergence. Finally, the maximum piece area increase in the Tabu search is constrained to 30% during the BARR stage. This value was selected after constraint values of 15% and 20% proved to be too stringent in the preliminary results.

The Tabu search parameters are perhaps the most crucial in determining the success of this procedure. Namely, these are the size of the Tabu list; the step size in each local neighbourhood; the maximum area constraint where applicable; and the number of no-improvement iterations before termination.

6.4.1 Tabu Search Parameters

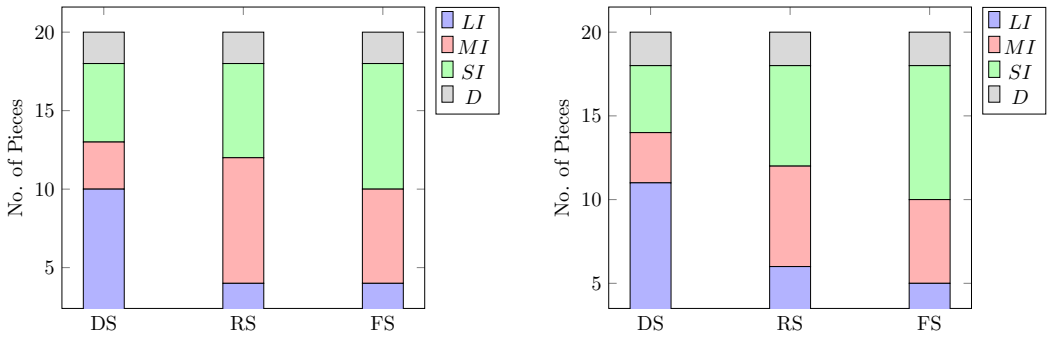
As can be seen, the ITSP is designed to iterate between modifying build orientations and repacking numerous times throughout the search, where many repacking attempts can be expected to fail. Therefore, the aim should be to try out many promising build orientations relatively quickly, rather than spending a long time on searching for a very good build orientation that may ultimately lead to an infeasible bin packing solution. Following this logic, the TS step size and termination condition are selected to encourage the local search to find "good enough" build orientations quickly, rather than attempting a more prolonged search for a better build orientation.

Thus, the following three strategies were considered: Decreasing Step (DS), Random Step (RS) and Fixed Step (FS). At each iteration of the Tabu search, RS sets the step size to a random integer value between 1° and 30°, while FS keeps the step size fixed at 5° throughout the search. The DS strategy starts with an initial step of 15°, which is then multiplied by a reduction factor of 0.85 at each iteration. In all cases the search is terminated after eight iterations without improvement and the Tabu list stores 10 most recent build orientations.

Following the results of Chapter 4, the rationale behind the DS strategy is to quickly cover the solution space with a relatively large initial step and compare a number of potential local minima, before converging on the best local minimum with a decreasing step size. This is similar to the strategy discussed in Section 4.4.2, expect that the DS strategy does not wait for an initial convergence, instead, the step size is kept constant for the first three iterations before applying the reduction factor. This prevents the search from getting stuck if the initial solution happens to fall in a local minimum.

Test Pieces	Objective Improvement (%)			No. of Iterations			Solution Time (s)		
	DS	RS	FS	DS	RS	FS	DS	RS	FS
Aero housing 1	11.8	11.8	12.7	12	20	12	72	93	74
Aero housing 2	27.4	28.4	14.4	26	25	12	57	27	27
Alcoa bracket	41.2	52.4	48.1	11	24	32	37	61	107
Bearing block	52.9	51.2	51.5	32	14	12	24	15	10
Combustor plate 1	45.6	41.2	41.8	13	19	15	23	60	25
Combustor plate 2	85.8	84.4	84.4	11	13	11	312	326	340
Control arm	56.8	58.9	62.8	11	17	12	5	7	6
Engine block	0.9	0	1.3	16	8	9	4	2	2
GE bracket	26.5	21.6	20.0	25	16	26	378	458	441
Gear	0	0	0	8	8	8	0.4	0.6	0.4
Impeller	71.0	69.0	69.2	14	11	19	4,989	7,900	6,325
RC Jet bracket	34.7	32.7	31.6	18	19	28	1,996	3,325	3,112
RC Jet NGV ring	53.6	52.6	49.0	10	10	12	92	193	69
Seal segment	70.6	76.7	72.3	10	37	12	69	148	84
Sector	82.5	80.0	79.5	10	11	19	11	48	19
Swivel hinge	0	0	0	8	8	8	2	2	2
Turbine blade	40.7	40.8	40.2	12	20	19	245	607	367
Turbine wheel (Baumers)	0	0	0.6	8	8	10	23	41	31
Turbine wheel (turbocharger)	67.5	66.3	28.7	26	15	14	264	456	173
ULA bracket	50.2	39.7	13.4	20	18	25	105	105	116
Average	**41.0**	**40.4**	**36.1**	**15.5**	**17.2**	**15.7**	**576**	**901**	**759**

TABLE 6.1: Results for DS, RS and FS tested on 20 pieces.



(A) Objective improvement. (B) Number of iterations.

FIGURE 6.5: Comparison of DS, RS and FS performance, based on objective improvement and number of iterations to final solution.

The three strategies were tested on 20 different pieces with the objective of reducing the area of each piece. Table 6.1 presents the results using three measures of performance: objective improvement, which is the difference in piece area between the initial and best solutions; number of iterations before termination; and the total solution time for each piece. Both DS and RS outperform FS in terms of objective improvement, while FS and DS outperform RS in terms of iterations and solution time.

In general, DS outperforms both RS and FS for the majority of tested pieces. This is demonstrated by the two graphs shown in Figure 6.5. For each tested piece, the three strategies were ranked as 'Largest Improvement' (LI), 'Medium Improvement' (MI) and 'Smallest Improvement' (SI) based on the objective improvement, as shown in Figure 6.5a; and similarly ranked based on the number of iterations to final solution, as shown in Figure 6.5b. Two out of the 20 tested pieces ('Gear' and 'Swivel hinge') were ranked as 'Draw' (D), as all three strategies yielded no improvement and terminated after eight iterations. Based on these results, we employ the DS strategy

in the Tabu search throughout the ITSP.

6.4.2 Test Data

The bin geometry and dimensions are the same as in other experiments, and can be found in Figure 3.17, with bin width W and length L set to 240 mm, and height H set to 275 mm. As before, the indirect cost rate is set to 26.64 £/h, while the material cost calculation is based on CoCr, a high-performance alloy commonly used in gas turbine applications, with a density of 8.3 g cm^{-3} and cost rate of 237.95 £/kg^3. Similarly, the regression coefficients used in the build time model are based on CoCr alloy data, which is shown in Appendix A. All parameters settings are summarised in Table 6.2. It is assumed that all parts are built directly on the SLM build plate without any base support structure, and support structure density is set to 100% (i.e. assumed fully solid).

Parameter	Value	Units	Definition
c_1	26.64	£/h	Indirect cost coefficient
c_2	237.95	£/kg^3	Material cost rate
ρ	8.3	g cm^{-3}	Co-Cr density produced by SLM
β_1	0.5	h	SLM constant time coefficient
β_2	1.16×10^{-1}	h mm^{-1}	SLM recoat time coefficient
β_3	2.04×10^{-4}	h mm^{-3}	SLM part exposure time coefficient
β_4	8.33×10^{-5}	h mm^{-3}	SLM support structure time coefficient
α_{sup}	45.0	°	Minimum unsupported build angle
ρ_{sup}	100	%	Density fraction for lattice support structure
d_0	10.0	mm	Minimum distance between pieces in 2DIBPP
L	50.0	mm	Maximum edge length in 2D piece polygon

TABLE 6.2: Summary of cost coefficients and modelling parameters.

To test the ITSP, a total of 27 instances have been generated manually using the geometries listed in Appendix D. Out of 68 geometries, 24 were taken from literature, 41 were taken from GrabCAD, an online open-source community for sharing 3D geometry files, and three geometries were provided by Rolls-Royce plc. Additionally, three of the components taken from GrabCAD were specifically designed for SLM and were produced by three real-case design challenges sponsored by Alcoa, General Electric (GE) and United Launch Alliance (ULA), respectively. The generated instances are shown in Table 6.3.

The first six instances in Table 6.3 vary the number of pieces – where 'SI', 'MI' and 'LI' are used to denote small, medium and large instances, respectively. Geometries were selected randomly to generate two instances of each size: SI1 and SI2 containing 10 geometries; MI1 and MI2 containing 25 geometries; and LI1 and LI2 containing 40 geometries. To limit computational expense, the remaining instances are fixed at a size of 25 geometries; this is also deemed a reasonable value for a low volume on-demand production scenario.

Instances containing geometries of low (LAV1), medium (MAV1) and high (HAV1) average volume were generated. Based on the available 68 test geometries, these instances are designed to approximately correspond to 4,500 mm^3 for LAV1, 20,000

Instance	N	D	Avg. Vol. (mm^3)	Vol. Std. Dev. (mm^3)	Avrg. VR	Avrg. SA (mm^2)	Avrg. MLR	Avrg. no. of Facets
SI1	10	0	37,078	32,232	21,870	0.19	0.22	46,893
SI2	10	0	60,412	81,256	22,665	0.24	0.28	27,198
MI1	25	0	35,551	35,232	21,372	0.19	0.22	55,921
MI2	25	0	42,980	56,650	24,099	0.20	0.27	42,290
LI1	40	0	32,687	49,707	21,257	0.21	0.23	56,657
LI2	40	0	38,652	48,822	20,992	0.20	0.25	51,622
LAV1	25	0	4,791	3,912	7,413	0.22	0.13	39,344
LAV2	25	5	4,794	3,819	7,686	0.22	0.12	41,084
LAV3	25	12	4,500	4,064	7,658	0.24	0.13	51,138
MAV1	25	0	20,290	8,205	15,822	0.22	0.24	56,218
MAV2	25	5	19,938	8,275	15,517	0.23	0.24	57,756
MMV3	25	12	20,475	6,932	15,528	0.21	0.23	56,311
HAV1	25	0	70,152	50,441	36,265	0.18	0.32	101,244
HAV2	25	5	70,709	49,632	36,877	0.18	0.29	97,053
HAV3	25	12	70,412	49,159	36,331	0.18	0.29	71,158
LSDV	25	3	17,283	6,073	15,735	0.21	0.23	35,338
MSDV	25	3	29,718	21,747	20,692	0.18	0.22	96,728
HSDV	25	4	47,536	65,326	20,718	0.25	0.20	54,903
LVR	25	4	27,746	26,368	28,078	0.07	0.28	71,167
MVR	25	4	39,306	34,097	20,413	0.16	0.23	115,934
HVR	25	4	57,222	90,786	16,928	0.29	0.20	51,093
LMLR	25	4	7,922	9,679	5,624	0.31	0.09	32,722
MMLR	25	4	34,837	27,383	20,585	0.14	0.22	104,884
HMLR	25	4	48,226	55,593	30,169	0.13	0.33	124,617
LSA	25	3	6,786	8,707	4,250	0.31	0.14	19,561
MSA	25	4	19,487	15,310	15,107	0.15	0.23	50,277
HSA	25	3	63,952	53,593	39,780	0.12	0.31	82,144

TABLE 6.3: ITSP test instances. N is the total number of geometries and D is the number of duplicate geometries. SA is the surface area, VR is the volume ratio and MLR is the maximum length ratio.

mm^3 for MAV1 and 70,000 mm^3 for HAV1. Duplicate geometries were also considered by generating two more instances of each type, with 20% (LAV2, MAV2, HAV2) and 50% (LAV3, MAV3, HAV3) of the total geometries being replaced with duplicates, respectively, while keeping all other variables at approximately the same values. Realistically, a few duplicates can be expected in a typical batch of parts, hence, this value is set to 12-16% of the total geometries for all remaining instances.

In addition to the size of pieces, the distribution of piece size has an equally direct effect on the success of the packing solution. For this reason, instances with low (LSDV), medium (MSDV) and high (HSDV) standard deviation of volume are also presented in Table 6.3.

The LSA, MSA and HSA instances correspond to low, medium and high average surface area of geometries, respectively, which has been suggested to have a correlation with the build time of SLM (Rickenbacher et al., 2013; Zhang et al., 2015).

Finally, the concavity and compactness of geometries can determine how well the pieces might fit together inside the bin, as well as how much support structure may be required. To quantify these two properties, two additional parameters have been introduced, namely, maximum length ratio (MLR) and volume ratio (VR). MLR is measured by taking the longest vector that fits inside the geometry and dividing its magnitude by the longest vector which fits inside the machine build envelope (i.e. a diagonal line between the front-bottom-left corner and the back-top-right corner of the bin). The VR is a ratio of the geometry volume divided by the volume of its bounding box. Instances of low, medium and large values were generated for both MLR and VR resulting in LMLR, MMLR and HMLR and LVR, MVR and HVR, respectively.

6.5 Computational Results

6.5.1 Analysis of ITSP Strategies

To analyse the effectiveness of the ITSP the cost improvement for each of the five stages in the procedure is compared relative to the initial solution. For clarity, the cost improvement, denoted by CI, is defined by Equation 6.7 for the overall ITSP, and Equation 6.8 for each stage in the procedure.

$$CI = S_i - \frac{S_{min}}{S_i} \tag{6.7}$$

where S_i is the cost of the initial solution and S_{min} is the cost of the best solution at the end of the ITSP.

$$CI = S_i^* - \frac{S_{min}^*}{S_i} \tag{6.8}$$

where S_i is the cost of the initial solution in the ITSP; S_i^* is the cost at the beginning of each stage; and S_{min}^* is the cost of the best solution found during that stage. It should be noted that a negative value of CI indicates an increase in the overall solution cost (i.e. negative improvement), and vice versa. Table 6.4 shows the average, best and worst CI produced by each stage and the overall ITSP, for the 27 tested instances; while Figure 6.6 shows the distribution of CI for each stage. As indicated by Table 6.4, SHR yielded mixed results, producing the single largest improvement (25.1% for the LSA instance) out of all the stages, but failing to reduce the cost for more than half of the tested instances, two of which resulted in a slight cost increase of 0.2%. On the other hand, SSVR produced the best results on average, followed closely by PSVR, as can be seen in Figure 6.6. The PHR and BARR stages yielded the least cost improvement, averaging 0.4% and 1.5%, respectively, and resulted in no improvement for majority of instances.

	Average CI (%)	Best CI (%)	Worst CI (%)	Average T (s)
SHR	2.6	25.1	-0.2	994
SSVR	6.7	22.6	0.2	7,758
PHR	0.4	4.0	0.0	668
PSVR	4.9	22.1	0.0	10,910
BARR	1.5	8.9	0.0	5,750
ITSP	16.0	31.0	4.0	29,092

TABLE 6.4: Summary of computational results showing average, best and worst cost improvement and average solution time for each stage in the procedure, as well as the overall ITSP. Cost improvement and solution time are denoted by CI and T, respectively.

There are a number of possible reasons why SHR and PHR performed worse than SSVR and PSVR. Firstly, this could be a natural outcome of the objective function, which is driven by support structure volume both through build time and material cost, while the build height parameter only influences the former. Consequently, the weight of support structure is likely to be higher than the weight of build height in the cost function, and since these two objectives are often conflicting, reducing build height may result in a higher overall cost, as in the case of the worst solution

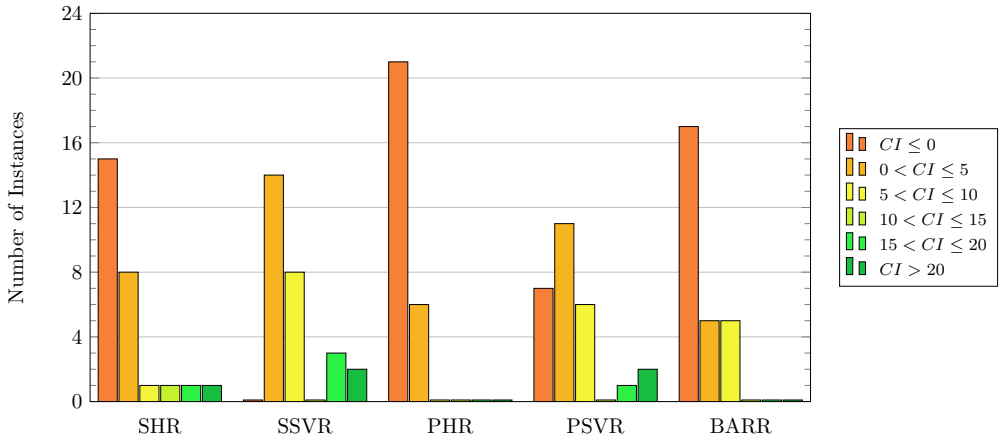


FIGURE 6.6: Histogram of cost improvement shown for each stage in the ITSP. Cost improvement is denoted by CI (%).

in Table 6.4. This is further supported by Figure 6.7, which compares the average variation in height (h_R), support structure volume (s_R), bin utilisation (u_R), and cost (c_R) during each stage in the ITSP. It can be observed that the variations in h_R are much smaller than s_R, with the cost reduction trend closely following that of s_R. In fact the h_R gradient shows a slight increase during SSVR and PSVR, indicating that cost improvements were gained by reducing support structure at the expense of build height. The dominance of support structure volume is also likely due to the fact that support structure density has been kept at 100% for these experiments, as indicated in Table 6.2; this value can be easily changed by the user.

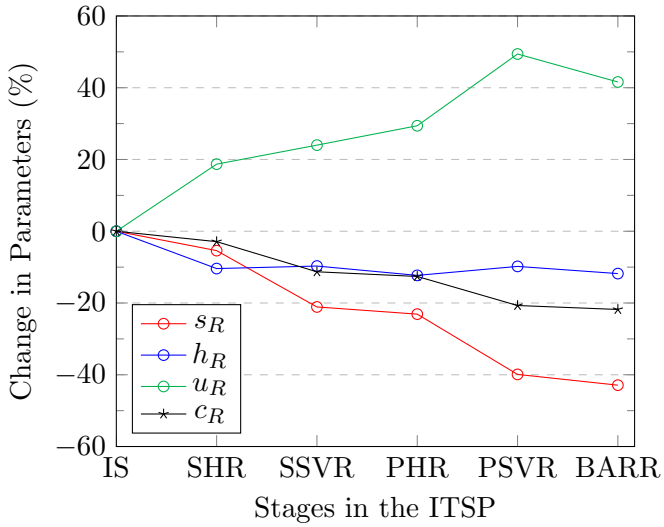
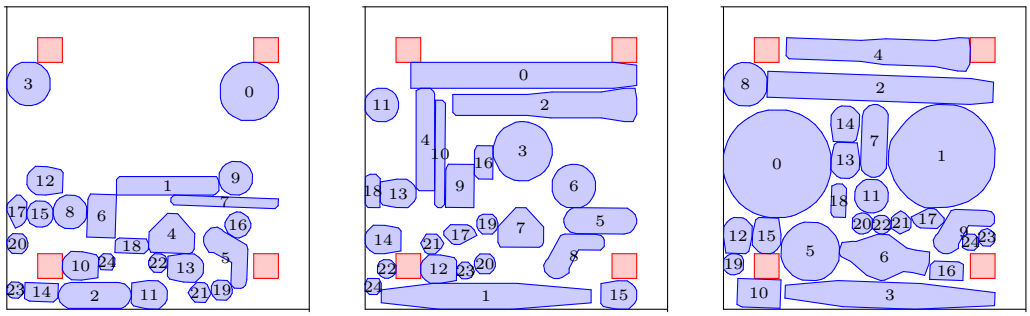


FIGURE 6.7: A comparison of the change in total support structure volume s_R, average build height of bins h_R, average bin utilisation u_R and total build cost c_R through the different stages of the ITSP, relative to the initial solution (IS). The values of s_R, h_R, u_R and c_R are calculated as a percentage difference between the initial solution and the best solution at the end of each stage, averaged across all test instances.

Another reason is that the SHR and PHR stages fail more often than SSVR and PSVR. This is partly due to the inverse correlation between the height and base

area of the geometries; for example, when a tall prism with a small cross-section is moved from an upright position to resting on its side, the height of the piece decreases while piece area is increased. On the other hand, the relationship between support structure volume and piece area is weaker, since support structure is more dependent on the complexity and overhanging surface area of the geometry, than its aspect ratio. Furthermore, since build height can only be reduced by repositioning the tallest piece in the bin, SHR and PHR are only guaranteed as many iterations as there are bins (or pairs of bins) in the solution. Since most test instances in this paper were packed into a maximum of four bins, the SHR often failed after only four iterations, or less. In contrast, the SVR and PSVR stages iterate through every piece in the solution, regardless of the previous iteration – thus having greater chances of success. The upside of this is that both SHR and PHR terminated relatively quickly compared to SSVR and PSVR, as shown in Table 6.4.

Finally, the cost reduction yielded by SHR and PHR, is not dependent on the height of the tallest piece, but rather on the height difference between the two tallest pieces; this is in contrast to SSVR and PSVR, where pieces are independent of each other. This explains why SHR occasionally produces significant cost improvements for instances such as LSA, shown in Figure 6.8, which consists of mostly small geometries and a few larger rod-shaped geometries, while failing to produce a large enough height decrease in most other instances.



(A) 1st iteration. $U = 0.14$, $H = 177.8$mm.

(B) 4th iteration. $U = 0.25$, $H = 82.3$mm.

(C) 7th iteration. $U = 0.41$, $H = 65.4$mm.

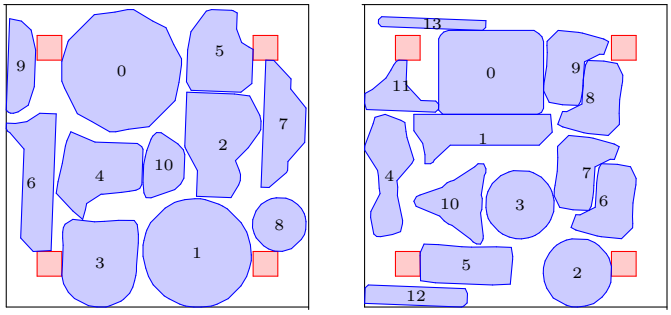
FIGURE 6.8: Solution steps for the LSA instance; (a) shows the initial solution, (b) shows an intermediate SHR solution and (c) show the final best solution produced by SHR. U and H denote bin utilisation (based on uninflated pieces) and build height, respectively.

Similarly, the poor results produced by PHR could be explained by a large increase in piece area after the Tabu search, and consequent failure of the 2DIBPP algorithm to find a feasible solution; Figure 6.7 shows that only a moderate increase in bin utilisation was achieved after PHR, compared to the subsequent PSVR stage.

As described in Section 6.2.3, PHR addresses bins in pairs in order to further utilise empty bin space in the solution. However, a successful iteration only guarantees a reduction in the height of the taller of the two bins; the height of the second bin is determined arbitrarily by the bin assignment during repacking. Thus, the PHR stage may be improved by considering the build height of the second bin as a constraint

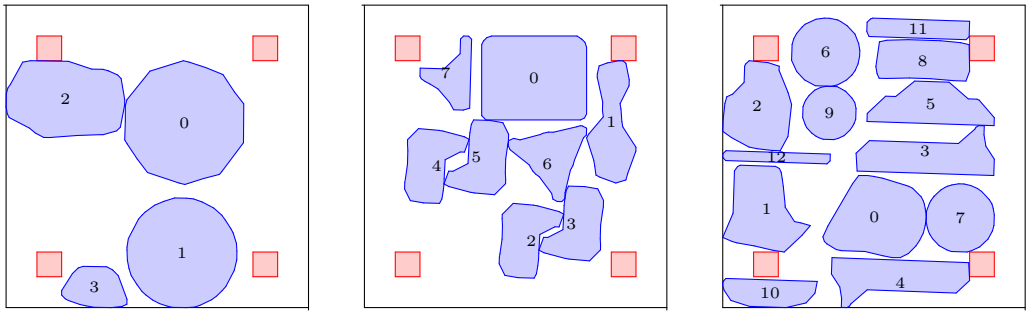
in the 1D bin packing model and the assignment mending strategy in the 2DIBPP algorithm.

BARR is quite different to the other four stages in the ITSP. The purpose of this final stage in the procedure is to jump to an unexplored area in the solution space by adding an extra bin to the more-or-less converged solution at the end of PSVR. For 18 test cases no cost improvement was found and the number of bins remained the same, while in the other ten instances a cost reduction of up to 8.9% was gained by adding an extra bin, increasing the average number of bins from 2.2 to 2.6, as shown in Table 6.5. At this point it could be argued that in cases where BARR produced only a negligible cost improvement (e.g. $< 1\%$) the solution should be reverted to the best found solution at the end of PSVR, as the very small cost improvement does not justify the use of an additional bin. The best improvement (8.9%) produced by BARR was for the MMLR instance, which is shown in Figure 6.9 and Figure 6.10.



(A) Bin 1 (B) Bin 2

FIGURE 6.9: Solution for the MMLR test instance at the end of PSVR.



(A) Bin 1 (B) Bin 2 (C) Bin 3

FIGURE 6.10: Solution for the MMLR test instance at the end of BARR.

6.5.2 Benchmarking

Due to the novelty of the problem, the ITSP procedure cannot be reproduced exactly by any currently existing method or software; instead, the initial solution of the ITSP is benchmarked against the commercial software Magics (version 20.03). The benchmark results were produced in the following two steps, keeping manual intervention

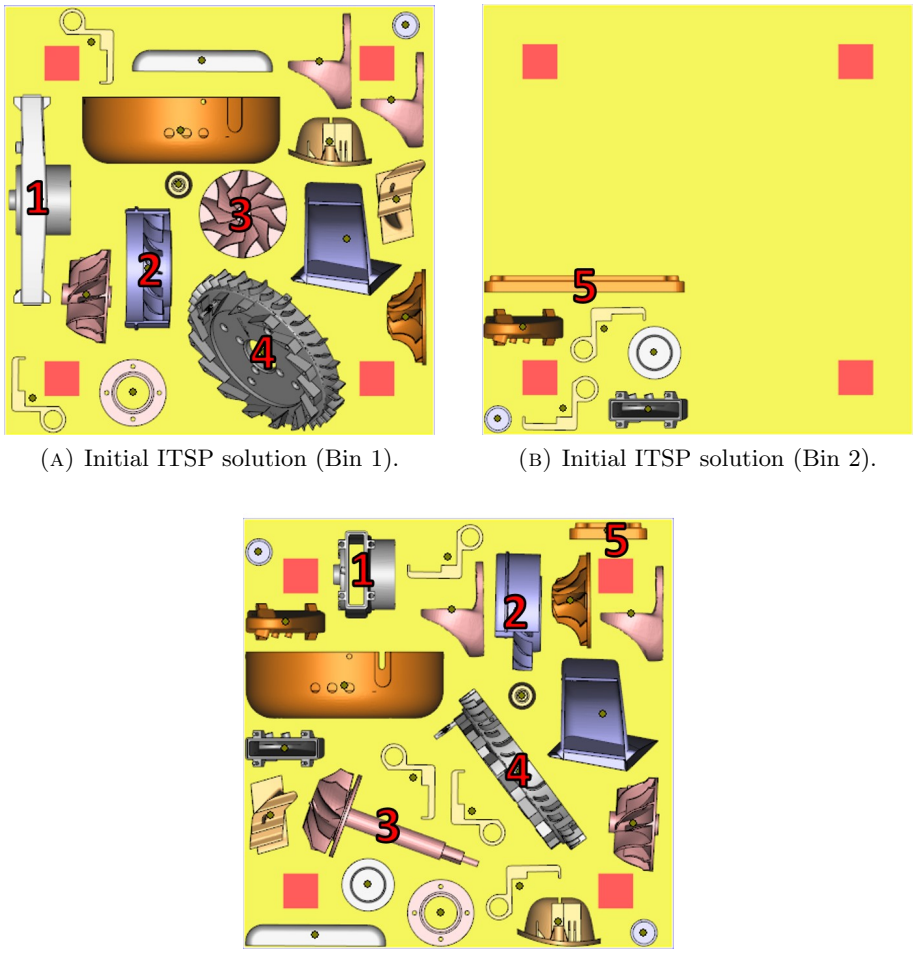
to a minimum. Firstly, the build orientation of each geometry was optimised for minimum projected 2D area (i.e. piece area); secondly, all parts were placed into the minimum number of bins while keeping the build orientations fixed. As in the ITSP, a minimum distance of 5mm was maintained between all packed parts.

	Initial ITSP			Final ITSP			Magics			D_I (%)	D_F (%)
	C (£)	T (h)	M	C (£)	T (h)	M	C (£)	T (h)	M		
SI1	4,893.5	124.4	1	4,142.8	109.5	1	6,203.6	154.4	1	-21.1	-33.2
SI2	6,310.8	167.7	1	6,056.5	166.7	2	7,302.1	187.4	1	-13.6	-17.1
MI1	13,994.0	345.3	2	12,759.2	325.2	3	13,969.6	348.0	2	0.2	-8.7
MI2	15,434.2	386.6	2	13,718.7	357.0	2	14,407.6	366.6	2	7.1	-4.8
LI1	19,548.3	493.9	3	18,182.3	470.7	4	18,863.6	476.4	3	3.6	-3.6
LI2	21,481.3	541.9	3	19,092.2	493.6	3	22,005.2	554.5	3	-2.4	-13.2
LAV1	2,668.6	65.4	1	1,955.3	50.8	1	2,543.0	63.0	1	4.9	-23.1
LAV2	2,529.1	61.8	1	1,951.3	51.3	2	2,645.6	64.2	1	-4.4	-26.2
LAV3	2,798.1	71.4	2	2,116.6	56.5	2	3,090.3	73.5	1	-9.5	-31.5
MAV1	9,051.6	227.8	2	6,716.8	179.6	2	8,340.7	207.6	2	8.5	-19.5
MAV2	8,657.4	220.2	2	6,322.1	167.2	2	7,833.6	194.1	1	10.5	-19.3
MAV3	6,621.9	178.6	2	5,826.5	158.8	2	7,564.9	196.6	2	-12.5	-23.0
HAV1	25,595.6	648.0	4	22,293.0	587.7	5	24,289.6	614.2	3	5.4	-8.2
HAV2	25,561.2	643.9	4	22,922.1	590.9	4	25,209.6	633.3	3	1.4	-9.1
HAV3	25,525.3	639.1	4	22,382.9	577.8	4	22,908.0	577.7	3	11.4	-2.3
LSDV	7,303.2	186.1	2	5,501.0	147.9	2	7,234.9	177.7	1	0.9	-24.0
MSDV	10,474.9	267.3	2	9,468.3	249.2	3	10,134.1	260.7	2	3.4	-6.6
HSDV	16,730.5	419.6	2	14,794.0	387.3	3	16,647.6	420.4	2	0.5	-11.1
LVR	14,615.6	360.7	3	10,579.9	287.0	4	13,020.0	322.0	3	12.3	-18.7
MVR	13,376.5	338.4	2	11,479.7	304.2	3	12,190.7	315.1	2	9.7	-5.8
HVR	18,827.2	472.4	2	17,188.2	435.9	2	17,690.7	446.7	2	6.4	-2.8
LMLR	2,415.9	63.1	1	1,975.3	53.8	1	2,462.7	64.1	1	-1.9	-19.8
MMLR	13,039.3	320.2	2	10,927.3	284.6	3	11,735.6	295.0	2	11.1	-6.9
HMLR	17,551.5	443.0	3	16,412.9	420.5	4	16,609.6	418.4	3	5.7	-1.2
LSA	2,559.0	70.4	1	1,766.6	48.5	1	2,418.2	67.6	1	5.8	-26.9
MSA	7,746.6	195.7	2	5,971.2	159.2	2	7,426.6	189.8	2	4.3	-19.6
HSA	25,095.8	628.3	4	22,642.9	577.1	4	24,390.3	603.0	3	2.9	-7.2
Average	**12,607.7**	**317.8**	**2.2**	**10,931.3**	**285.1**	**2.6**	**12,190.3**	**307.1**	**2.0**	**1.9**	**-14.6**

TABLE 6.5: Comparison of ITSP initial and final solutions with Magics results. C denotes solution cost, M denotes the number of bins, H denotes average build height and SV denotes the total support structure volume.

Table 6.5 provides a comparison of Magics and ITSP results for the 27 test instances. It can be seen that the average initial cost of ITSP was marginally higher than the Magics benchmark, while the final ITSP solution outperformed the benchmark by an average of 14.5%. The difference between the benchmark and initial ITSP solution can be attributed to two factors; firstly, the support structure volume is higher than the benchmark by an average of 14%; secondly, seven of the 27 test cases produced one more bin than the benchmark, resulting in the average difference of 0.2 bins seen in Table 6.5.

As can be seen in Figure 6.11, the Tabu search was not able to find the minimum area solution for every piece, as in some cases it got stuck in a local minimum early on. Furthermore, in order to solve the problem in reasonable time, the Tabu search was allowed a maximum run time of 5 min per piece; as a result it was occasionally terminated early, usually in the case of large complex geometries (e.g. $\geq 500,000$ STL facets), such as piece 5 in Figure 6.11a. This explains why some ITSP initial solutions had more bins, and accounts for some of the difference in support structure volume.



(A) Initial ITSP solution (Bin 1).

(B) Initial ITSP solution (Bin 2).

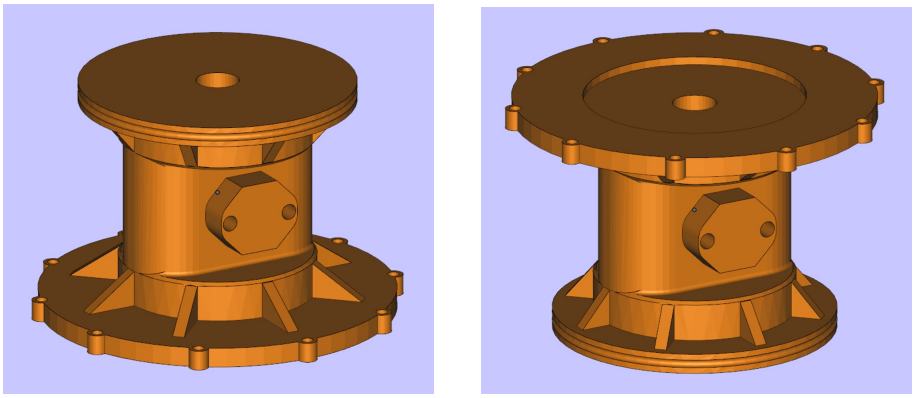
(C) Benchmark solution.

FIGURE 6.11: Comparisons of the benchmark and initial ITSP solution for the MAV2 test instance. Pieces where the Tabu search performed worse than the benchmark are numbered in red.

It should be noted that, since there is no strong correlation between the support structure volume and piece area, the poor Tabu search solutions can sometimes work in favour of the ITSP, as in the case of SI1 and SI2, where the ITSP incidentally outperformed the benchmark due to lower support structure, while the number of bins remained the same due to the small size of the instances.

Moreover, two build orientations with the same optimal piece area and height, can result in different volume of support structure, as shown in Figure 6.12. Although support structure and height are already used to break tie in the piece area Tabu search, the current heuristic is most likely to converge on whichever position is closest to the initial build orientation. The above shortcomings can be improved by tuning hyperparameters in the local TS (e.g. Tabu list size) to encourage further solution space exploration, however, this is likely to significantly increase solution time. Since the overall solution time of the ITSP is already quite long, averaging 7.8 h, the focus of the local TS should be on a fast solution rather than a very good solution is discussed in Section 6.4.

As described in Section 3.4, the intersection checking performed in the support structure volume calculation, is by far the most time-consuming and computationally expensive aspect of the Tabu search, accounting for more than 80% of the time taken to generate each build orientation. Thus, improving the efficiency of this method or replacing it with an alternative, for example, a machine learning approach based on support structures designed by an expert user or Magics, could significantly improve the solution time of the ITSP (although other challenges would have to be considered, such as acquiring sufficient training data).



(A) Build orientation 1 (B) Build orientation 2

FIGURE 6.12: An example of two build orientations, which result in the same height, 60 mm, and piece area, 6699 mm^2, with support structure volume of 246 134 mm^3 and 105 645 mm^3 for (a) and (b), respectively.

6.6 Conclusion

This chapter addresses the combined problem of multi-part build orientation, bin assignment and irregular bin packing in the context of SLM. The proposed ITSP heuristic is driven by build cost and is aimed at the MBP scenario, which is helpful for dealing with HVLV demands that often arise in industry. The ITSP consists of six different stages; in each stage a local Tabu search is used to find or modify the build orientation of individual parts and the resulting 2DIBPP is solved using the method of Martinez-Sykora et al. (2017), discussed in Section 3.5. Each stage explores a different area of the solution space by varying the bin assignment of pieces and by addressing a different aspect of the cost objective function in the TS; namely, the number of bins, the build height of each bin and the total support structure volume.

The ITSP was tested on 27 different test instances, consisting of various geometries ranging in size and complexity. The two most successful stages in the ITSP, SSVR and PSVR, were driven by reducing support structure volume in the Tabu search, and produced average cost improvements of 6.7% and 4.9%, respectively; while the build height-driven stages, SHR and PHR, produced average cost improvements of 2.6% and 0.4%, respectively. The BARR stage produced non-trivial improvements of 2.4-8.9% for seven of the 27 test cases. On average, the initial bin utilisation was increased

by over 40% at the end of the ITSP (after a slight reduction in the BARR stage) indicating that the proposed procedure was quite successful in effectively utilising machine space. The relative success of the BARR stage suggests that increasing the number of bins can improve the cost, even for relatively small instances, provided that the additional machine space is utilised to improve the build orientation of parts. Additionally, using more bins allows the parts to be built in parallel; this presents an interesting trade-off between build cost, time and the number of machines, which could be addressed as part of future work.

The initial solution of the ITSP produced a marginally higher average cost and number of bins, as well as a considerably longer solution time when compared to a commercial benchmark; thus, a number of possible future improvements to the ITSP are proposed in Section 6.5. Despite its long running time, the final ITSP solution yielded a cost improvement over the benchmark in all 27 test cases, with an average of 14.5% and ranging up to 33.2%. Furthermore, the ITSP does not require any supervision once it is running. Most importantly, the cost improvements provided by the ITSP demonstrate the benefit of combining build orientation and bin packing into a single problem, instead of addressing these two parameters in separate stages as done in literature and in current commercial software.

Chapter 7

Conclusions and Future Work

Process planning is a key step of product development and has a significant effect on the cost, lead time and quality of manufactured parts. While the close link between build orientation and bin packing in AM has been acknowledge in literature, no previous works have been found to provide a satisfactory solution to the combined problem of these two parameters. As discussed in Chapter 2, many models have been proposed for optimising the build orientation, with the most popular criteria including surface roughness, the presence of support structure and build time. Surprisingly few works have addressed cost, and only Alexander et al. (1998) was found to consider the cost of post-processing; no works have been found to address the full cost of SLM as a function of build orientation and bin packing.

Thus, Chapter 3 proposes a framework for optimising the build orientation and bin packing of parts simultaneously in SLM, as a function of total production cost. Since the framework is aimed at the production of end-use parts, the desired surface finish of the part is treated as a hard constraint (as opposed to an optimisation criterion), thus, the surface finish produced by SLM is treated as a post-processing cost, rather than a competing objective. It should also be noted that the surface finishing cost is strongly dependent on the choice of finishing method; as discussed in Section 3.3, surface finishing should only be included in the optimisation if a local finishing method is used, since using a global method makes this aspect of cost independent of the build orientation.

Chapter 3 also considers lean manufacturing principles in the context of SLM, thus, proposing two alternative production scenarios – Mixed Batch Production (MBP) and Identical Batch Production (IBP). The former scenario is most suitable for meeting High Variety Low Volume (HVLV) demands, which are commonly encountered in the aerospace industry, in particular, in the context of spare parts. The latter scenario is suitable for the production of identical parts to meet steady and predictable, medium-to-large demands. The presence of identical safety-critical parts is also considered.

Chapter 4 proposes a Tabu search strategy for optimising the build orientation and bin packing of parts in the IBP scenario. Due to the relatively high computational expense of the bin packing problem, the Tabu search is coupled with an area-based approximation strategy, which reduced the solution time by as much as 50% when

applied to three different test geometries. The presence of symmetries, which is a common occurrence in many industrial parts, is also discussed in detail in this chapter and a generic method for dealing with these symmetries is proposed.

Chapter 5 extends the Tabu search to address the multi-objective problem of cost and residual stresses predicted via FEA. Due to the computational expense of the FEA, it is only solved for a limited number of initial sampling solutions and potential non-dominated solutions found by the Tabu search, which are based on approximations of the residual stress solution space. Two alternative approximation strategies are considered and compared – nearest-neighbour (NNTS) and linear interpolation (LITS). As expected, the more naive NNTS approach produced a less diverse Pareto set with weaker solutions that LITS. This is mainly due to the fact that NNTS only uses cost to drive movements, since it cannot introduce new values of stress into the pool of known solutions. The results produced by both strategies demonstrate that residual stresses can be reduced significantly with the right build orientation. The results also indicate that the best build orientation may not necessarily be the most obvious one, even for an experienced SLM user, thus, it is suggested that the full range of continuous rotations should be allowed in this optimisation problem.

Finally, Chapter 6 addresses the problem of build orientation and bin packing in the context of MBP. Because this problem deals with multiple heterogeneous parts, it is much more complex than the IBP version as three sub-problems must be solved simultaneously; namely, the build orientation of multiple parts and the 2DIBPP, which consists of bin assignment and 2D bin packing. To address all three problems in a coupled way an Iterative Tabu Search Procedure (ITSP) is developed, which consists of six stages. The aim of the different stages is to explore different neighbourhoods in the highly complex solution space of this problem; the first stage focuses on establishing the minimum possible number of bins in the solution; the second and third stages attempt to iteratively improve each bin in the resulting solution; the fourth and fifth stages attempt to further improve bins, as well as the bin assignment (i.e. by creating additional room for further improving build orientations); finally, the last stage of the ITSP introduces an additional empty bin into the solution, allowing further freedom to move and rotate parts and, thus, to further reduce the overall cost.

The ITSP has been tested on 27 manually created instances with different geometric proprieties, and was benchmarked against a commercial decision-support software. The benchmark results demonstrate the validity of the proposed approach, since solutions produced by ITSP were on average 14.5% cheaper than those produced by the commercial tool. These results also confirm that solving the two problems of build orientation and 2DIBPP in a coupled way, can indeed produce a significantly cheaper solution than solving the two problems independently as done throughout the literature.

7.1 Future Work

The proposed framework can be improved in a number of ways. Firstly, modelling 2D pieces with closed concavities can potentially lead to better bin packing solutions, by allowing tiny pieces to be placed inside large concavities, as discussed in Section 3.4.3. This only applies to specific geometries in specific build orientations (for instance, a large hollow cylinder built upright) and assumes the presence of tiny and very large geometries in the batch.

As mentioned in Section 3.1.3, the proposed framework assumes the horizontal orientation of parts to have no effect on build quality, allowing for full free rotation of 2D pieces within the 2D bin packing problem. In reality this is not the case as the angle of the powder-spreading rake has a significant effect on the build quality and risk of build failure of parts, particularly with respect to delicate features and thin walls. Therefore, future work should consider the rake angle as an additional optimisation criterion or build quality constraint when rotating and placing parts in the bin packing problem.

The post-processing cost model could also be made more realistic by considering the different available post-processing methods and their applicability to different production scenarios. Incorporating process- and part-specific rules into the framework (e.g. in the form of a decision tree, as done by Gordon et al. (2016)) could be particularly helpful, as this would not only improve the accuracy of the cost optimisation model, but would also allow the user to compare the cost and suitability of different post-processing methods. The build time modelling accuracy could also be improved with additional data. A database of build times for different materials and SLM machines would be particularly useful in this regard.

As discussed in Chapter 6, the solution time of the ITSP could be improved by reducing the computational time of the support structure volume, since this is the most time-consuming aspect of the model. It is suggested that a machine learning approach such as an artificial neural network algorithm, could be used to develop a faster prediction model of support structure volume based on part geometry parameters, such as part height, supported area and the number of protruding features. However, such a model would require a substantial amount of data and is likely to introduce some error into the model. A major limitation of this concept is the current lack of standardised design rules for support structure, which is another area that requires further research. The ITSP could also be extended to consider post-processing cost and residual stresses, although the latter would significantly increase the solution and pre-processing time of the model, and would require further automation of the model presented in Chapter 5.

To the best of the author's knowledge, experimental data directly linking residual stresses and the density of defects in a part is currently lacking and thus could be a useful contribution of future research. Such data could be used to develop an explicit probabilistic model linking part quality or build failure to the build orientation

through residual stresses. It could also be used to develop more informative residual stress criteria, as suggested in Chapter 5. For instance, the data could be used to provide a threshold stress for a given material and SLM machine, and a new objective could be introduced which minimises the volume fraction of the part exceeding the threshold stress; alternatively, a constraint could be developed where the allowable volume fraction and the threshold stress must be determined from experimental data. Moreover, if the risk of build failure or poor build quality can be quantified, it could also be translated into an ill-conditioned cost (Baumers and Holweg, 2016), in the form of rework or scrap (Dodd, 2015).

Finally, while unit cost is a critical decision factor in production, the production time is equality important, as delays lead to penalties and loss of business. Therefore, it would be particularly useful to extend the proposed framework to consider the multi-objective optimisation of unit cost and production time. Since production time is not considered in this thesis, it is possible to assume an infinite number of available SLM machines, as discussed in Chapter 3. Conversely, it is impossible to produce a meaningful optimisation solution for production time without considering the number of available machines. Moreover, in a realistic production scenario deadlines and different levels of priority can be expected for different part orders. This presents an interesting and very challenging multi-disciplinary optimisation problem, as it combines the engineering problem of SLM process planning with 2DIBPP and scheduling, which are two common problems studied by the Operational Research community.

Appendix A

Build Time Regression Models

Two linear regression models of build time were created, using the data provided in tables A.1 and A.2[1], respectively; where n denotes the number of parts in the build; t_r denotes total recoating time; t_v denotes total part exposure time; t_s denotes total support structure exposure time; v_{tot} and s_{tot} denote total part volume and support structure volume, respectively; and h denotes build height. Table A.1 shows a total of seven builds, which contained only identical parts and were built on an EOS M270 machine using Ti64 powder. Table A.2 shows six builds produced on the same machine using CoCr powder. The two sets of results were produced using a constant layer thickness of 30 µm for Ti64 and 20 µm for CoCr.

Build	n	v_{tot} (cm^3)	s_{tot} (cm^3)	h (mm)	t_r (h)	t_v (h)	t_s (h)
1	1	287.4	301.1	177.0	18.5	59.5	22.7
2	1	30.8	0.8	11.1	1.1	3.9	0.0
3	2	61.2	41.4	78.2	7.7	9.3	2.9
4	1	32.0	0.7	17.8	1.8	9.9	0.1
5	6	176.2	4.9	73.8	6.0	19.4	0.3
6	1	57.3	1.0	91.2	10.0	11.1	0.1
7	1	33.0	2.2	12.5	2.2	8.5	0.2

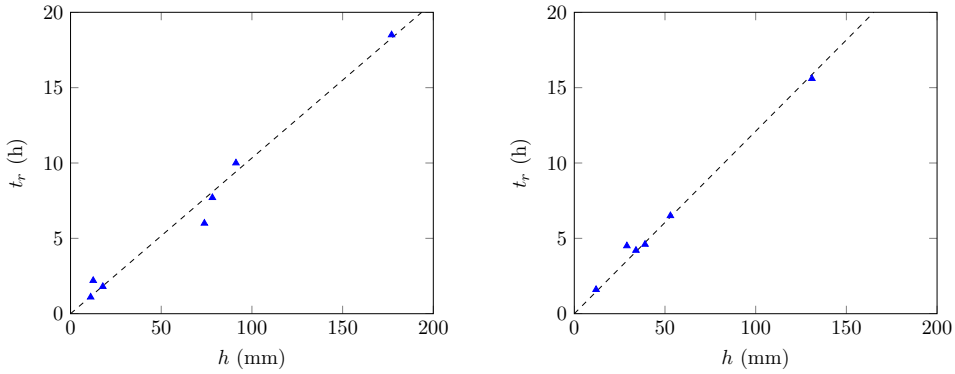
TABLE A.1: Regression model input data for Ti64 alloy.

Build	n	v_{tot} (cm^3)	s_{tot} (cm^3)	h (mm)	t_r (h)	t_v (h)	t_s (h)
1	81	67.1	0	12.4	1.6	0	1.6
2	3	99.2	145.6	34.1	23.2	20.9	4.2
3	2	80.8	394.7	39.0	18.5	33.7	4.6
4	3	595.6	0	53.0	119.9	0	6.5
5	6	63.7	173.4	28.8	14.4	8.6	4.5
6	1	167.2	119.9	131.0	34.9	13.2	15.6

TABLE A.2: Regression model input data for CoCr alloy.

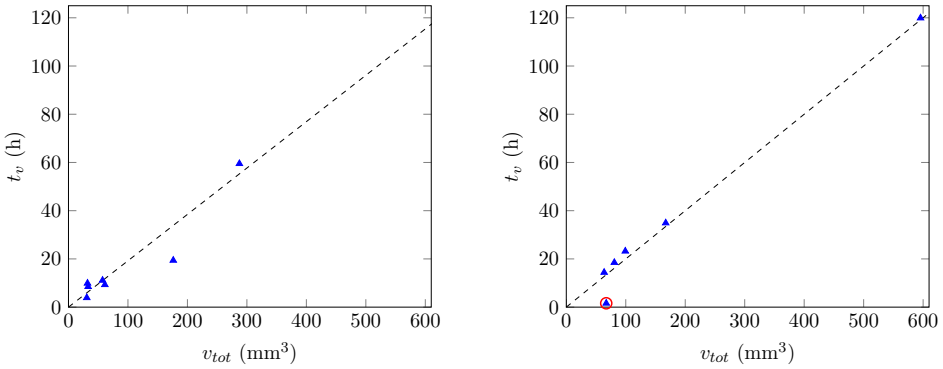
The coefficients of h, v_{tot} and s_{tot} are determined via the linear fit models shown in Figure A.1, Figure A.2 and Figure A.3, respectively.

[1] All data was provided by Rolls Royce plc.



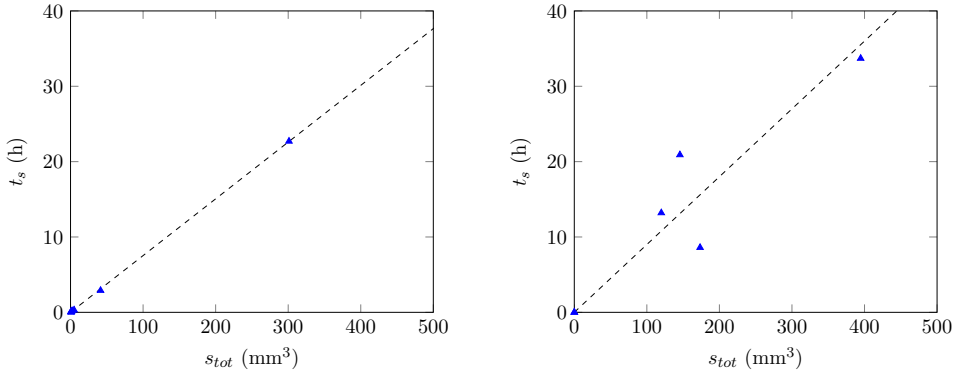
(A) Ti64 (B) CoCr

FIGURE A.1: Recoat time vs. build height.



(A) Ti64 (B) CoCr

FIGURE A.2: Part exposure time vs. part volume.



(A) Ti64 (B) CoCr

FIGURE A.3: Support structure exposure time vs. support structure volume.

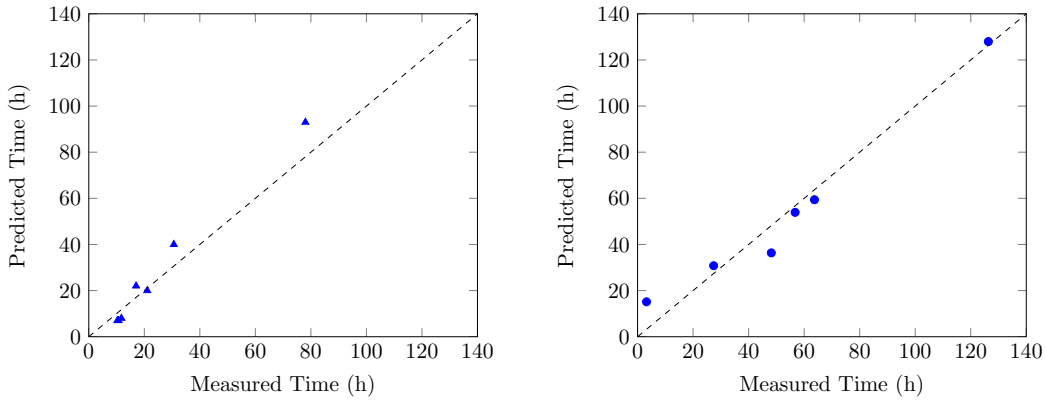
The resulting model parameters are summarised in Table A.3. It can be seen that the above models do not account for part cool-down, extraction and machine set-up time, since the intercept of each linear function is set to zero. These aspects of build time are highly variable and are difficult to predict accurately; however, it

is suggested that β_1 should be set to a value between 0.5 h and 3 h (Baumers et al., 2017), to reduce the risk of under-predicting the build time and to encourage the model to reduce the number of bins in the optimisation search. Due to the limited amount of available data, the same data sets are used to both train and validate each model. The resulting maximum errors (ME) and root mean square errors (RMSE) are shown in Table A.3.

A particularly high error of 78.5% was obtained for Build 1 in the CoCr data set, as a result of the outlier highlighted in red in Figure A.2b. As can be seen, the total part exposure time is much lower than expected in this case, resulting in an over-prediction. However, since the total part volume v_{tot} is constant and has no effect on the build orientation or bin packing solutions, it can be assumed that any error in predicting the part exposure time t_v will have a negligible effect on the optimisation models proposed in this thesis. Overall, a clear linear fit can be observed between the parameters in figures A.1 - A.3; the correlation in Figure A.3b is slightly weaker; however, no definitive conclusions can be made without more data. Thus, the proposed build time regression models are considered acceptable for the purpose of demonstrating the optimisation models in chapters 4 - 6, but larger data sets should be obtained and used to properly validate this approach as part of future research.

Parameters	Ti64	CoCr	Units
β_2	1.03×10^{-1}	1.16×10^{-1}	$\mathrm{h\,mm^{-1}}$
β_3	1.81×10^{-4}	2.04×10^{-4}	$\mathrm{h\,mm^{-3}}$
β_4	7.50×10^{-5}	8.33×10^{-5}	$\mathrm{h\,mm^{-3}}$
R^2	99.3	96.1	%
ME	20.4	78.5	%
RMSE	7.3	18.0	h

TABLE A.3: Regression model coefficients and errors shown for the two models.



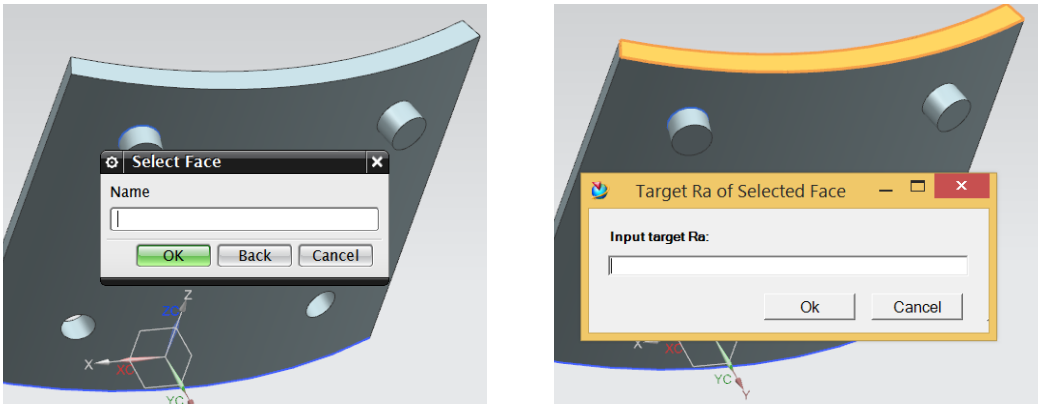
(A) Ti64 (B) CoCr

FIGURE A.4: Measured vs. predicted total build time calculated for the training data.

Appendix B

Input Target Surface Roughness via CAD

As discussed in Chapter 3, an additional preliminary step is required in order to calculate the build orientation-driven post-processing cost, in which the user must specify local values of the target surface roughness, Ra_t, for different local surfaces and features. For the purpose of this step, a custom application has been developed using NXOpen version 9.0, which is a CAD API in Siemens NX. The application asks the user a number of simple questions via the NX graphical user interface (GUI). Firstly, the user is asked to provide a baseline value of Ra_t for all surfaces in the geometry file. Secondly, the user may select individual faces in the CAD file and provide a local value of Ra_t for that specific surface, as shown in Figure B.1. This function is placed in side a while-loop, allowing the user to iterate through as many surfaces as needed. Finally, the user may terminate the process once all target roughness values have been specified.



(A) Select a face. (B) Input target Ra for selected face.

FIGURE B.1: Defining local target Ra of individual faces in NX GUI.

Following this, the application will output a modified STL file, which contains the target surface roughness of each facet f, denoted by Ra_{t_f}, in addition to the usual information about the facet normal vector $\hat{n}_f$ and its vertices v_{1_f}, v_{2_f} and v_{3_f}. Figure B.2 provides a schematic representation of the resulting modified STL file.

	solid				
$f_1 \rightarrow$	facet normal	$\hat{n}_x$	$\hat{n}_y$	$\hat{n}_z$	Ra_t
	outer loop				
	vertex	v_{1_x}	v_{1_y}	v_{1_z}	
	vertex	v_{2_x}	v_{1_y}	v_{1_z}	
	vertex	v_{3_x}	v_{1_y}	v_{1_z}	
	endloop				
	endfacet				
$f_2 \rightarrow$	facet normal	$\hat{n}_x$	$\hat{n}_y$	$\hat{n}_z$	Ra_t
	outer loop				
	vertex	v_{1_x}	v_{1_y}	v_{1_z}	
	vertex	v_{2_x}	v_{1_y}	v_{1_z}	
	vertex	v_{3_x}	v_{1_y}	v_{1_z}	
	endloop				
	endfacet				
$f_3 \rightarrow$	facet normal	$\hat{n}_x$	$\hat{n}_y$	$\hat{n}_z$	Ra_t
	...				
	endsolid				

FIGURE B.2: Schematic description of the modified STL file, where Ra_t is included in addition to the normal and vertex 3D coordinates of each facet f.

Appendix C

Surrogate Modelling Approach

Several attempts were made to create a Kriging surrogate model relating residual stresses to the part build orientation in SLM. This work was carried out using OPTIMAT (version 2), which is a modelling and optimisation tool developed at the University of Southampton. All models were created using the test geometry shown in Figure 5.8 and the DOE points shown in Figure 5.9, both of which can be found in Section 5.5.

The first version of this surrogate model was created using the mean tensile residual stress criterion, $\bar{\sigma}$, as described in Chapter 5. The resulting response surface is shown in Figure C.1; as can be seen the model is over-smoothed, resulting in a maximum error of 55.3% (with respect to the range of stress values for the whole DOE) and a very low R^2 value of 0.2.

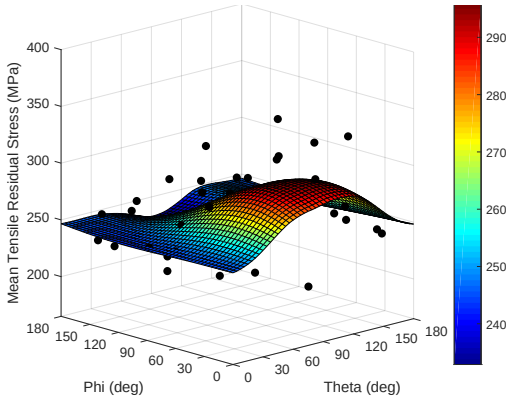
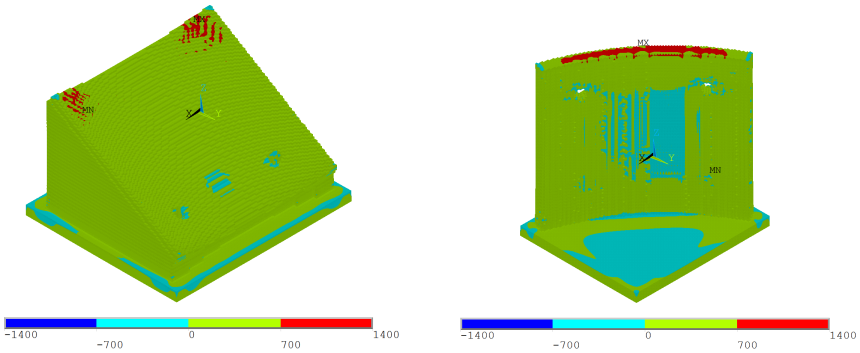


FIGURE C.1: Response surface for mean tensile residual stress, $\bar{\sigma}$, with respect to build orientation.

The use of a mean stress criterion was the assumed cause of this over-smoothing, since $\bar{\sigma}$ only captures the magnitude but not the location of stresses in the geometry. For example, if the geometry is rotated from one build orientation to another (e.g. from p_1 to p_2 as shown in Figure C.2), a local high tensile stress spot can "move" from one region of the geometry to another, and thus can result in two very different residual stress profiles and build orientations having similar values of $\bar{\sigma}$. This results in an apparent lack of strong correlation in the DOE data set, leading to the Kriging model producing an over-smoothed response surface, as can be seen in Figure C.1.



(A) Build orientation p_1 (45^o, 0^o). (B) Build orientation p_2 (0^o, 45^o).

FIGURE C.2: Comparison of two build orientations p_1 and p_2, with relatively similar stress magnitudes, but very different residual stress profiles and local high-stress regions.

C.1 Geometry Sectioning Approach

Based on the above assumption an alternative approach was considered, where the geometry was divided into sections, as shown in Figure C.3, and a separate surrogate model was created for each section, resulting in a total of ten response surfaces. As can be seen in Figure C.3 this sectioning method ignores some regions of the test geometry – this is considered acceptable as the named sections capture those regions where spots of high tensile stresses are most likely to occur (in different build orientations). It should also be noted that some of the named sections are geometric reflections of each other (e.g. *EDGE1* and *EDGE3*), meaning that their corresponding response surfaces can also be expected to show reflectional symmetry with respect to one another.

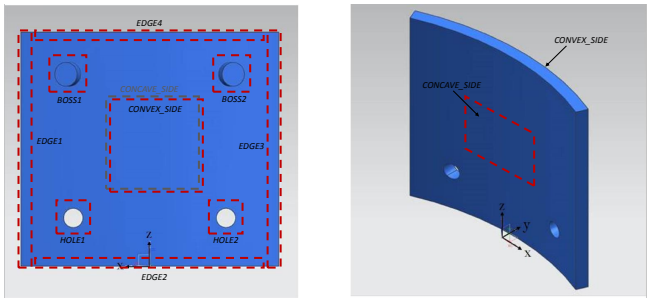


FIGURE C.3: Sectioned test geometry, where each section is outlined by a red dashed box. The *CONVEX_SIDE* and *CONCAVE_SIDE* sections include only visible mesh elements on the surface of the geometry.

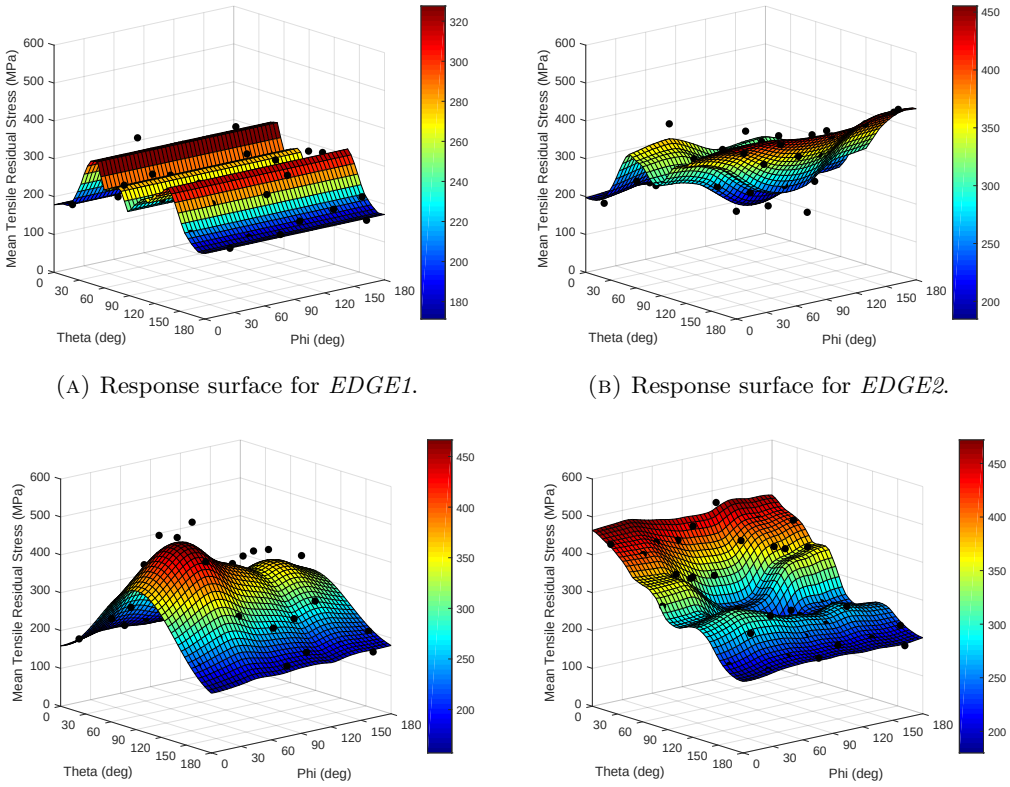
The resulting response surfaces are shown in Figure C.4 and Figure C.5, while the errors and regression coefficients can be seen in Table C.1. As can be seen, the *CONCAVE_SIDE*, *CONVEX_SIDE* and the hole sections produced particularly low-accuracy response surfaces with very high regression, while the edge and boss sections produced slightly lower errors with some acceptable levels of regressions (e.g. R^2 of 0.8). Nevertheless, in all 10 cases the maximum errors exceed 30% and RMSE

values exceed 10% (with respect to the range of stress values for the whole DOE), so these models are considered to be of insufficient quality. It is also interesting to note the absence of any significant reflectional symmetry between the different response surfaces (e.g. between *BOSS1* and *BOSS2*), which were expected based on the test part symmetry as mentioned above.

Section name	R^2	RMSE (%)	Max Error (%)
EDGE1	0.8	12.2	38.4
EDGE2	0.8	14.0	46.7
EDGE3	0.8	12.9	42.7
EDGE4	0.8	11.5	38.9
CONCAVE_SIDE	0.1	21.9	51.6
CONVEX_SIDE	0.6	13.1	42.2
BOSS1	0.8	10.8	31.1
BOSS2	0.6	10.4	30.2
HOLE1	0.1	20.2	68.9
HOLE2	0.3	13.6	47.8

TABLE C.1: Regression coefficients and surface fitting errors shown for each named section of the test geometry.

Similarly to the previous surrogate model for the whole test geometry, the high errors and regression produced by the geometry sectioning approach are thought to be the result of averaging of tensile stresses without consideration of the location and distribution of stresses within each section.



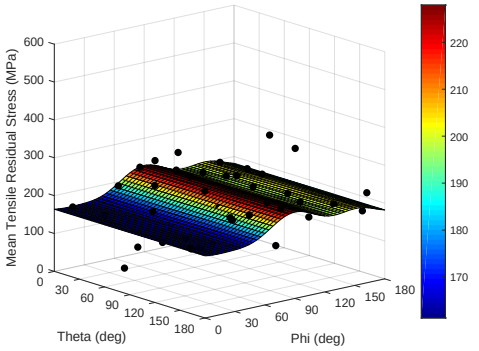
(A) Response surface for *EDGE1*.

(B) Response surface for *EDGE2*.

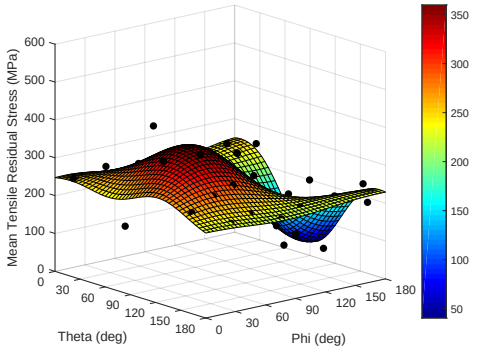
(C) Response surface for *EDGE3*.

(D) Response surface for *EDGE4*.

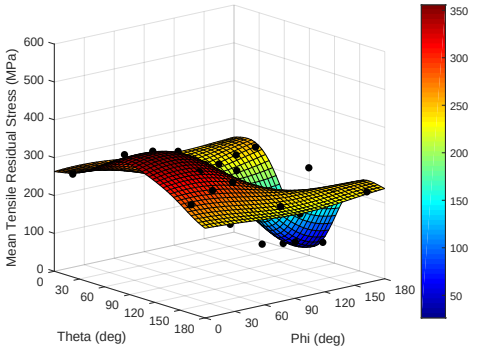
FIGURE C.4: Response surfaces shown for each section.



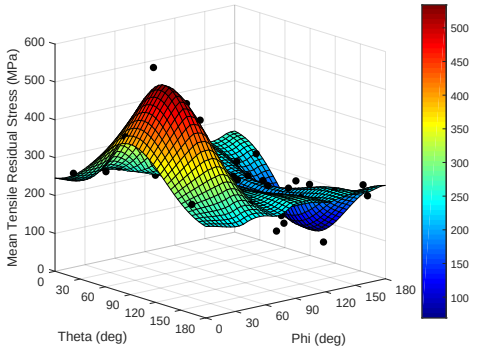
(A) Response surface for *CONCAVE_SIDE*.



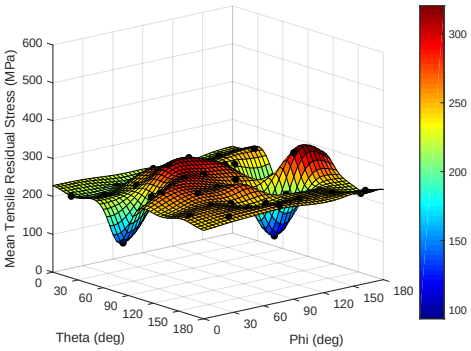
(B) Response surface for *CONVEX_SIDE*.



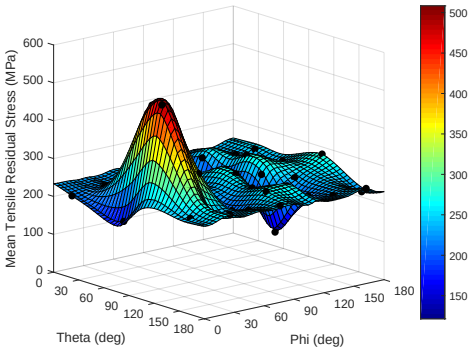
(C) Response surface for *BOSS1*.



(D) Response surface for *BOSS2*.



(E) Response surface for *HOLE1*.



(F) Response surface for *HOLE2*.

FIGURE C.5: Response surfaces shown for each section (continued).

C.2 High-Stress Points Approach

Following the above results, a third approach was considered. For each of the 50 build orientations in the DOE, the location of the maximum tensile stress element in the part mesh was recorded. The recorded location was then fixed and the stress value in that location on the part geometry was then recorded for the remaining 49 points. This process was repeated for each point in the DOE, resulting in a total of 50 recorded "high-stress points" on the part as shown in Figure C.6.

It can be seen from Figure C.6 a number of the high-stress points lie very close to each other on the part and thus can be expected to produce very similar response

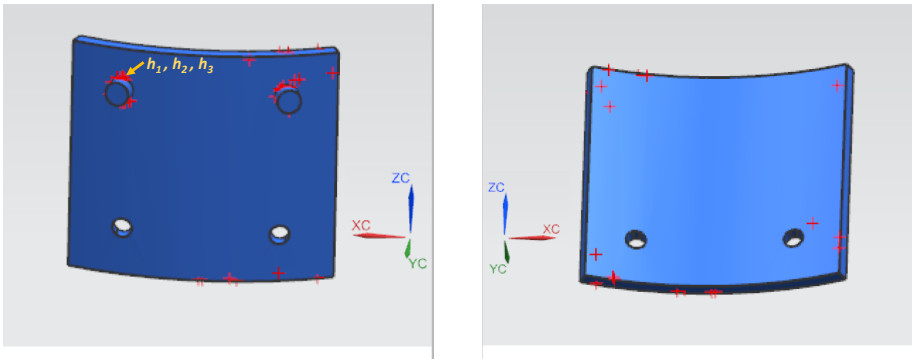
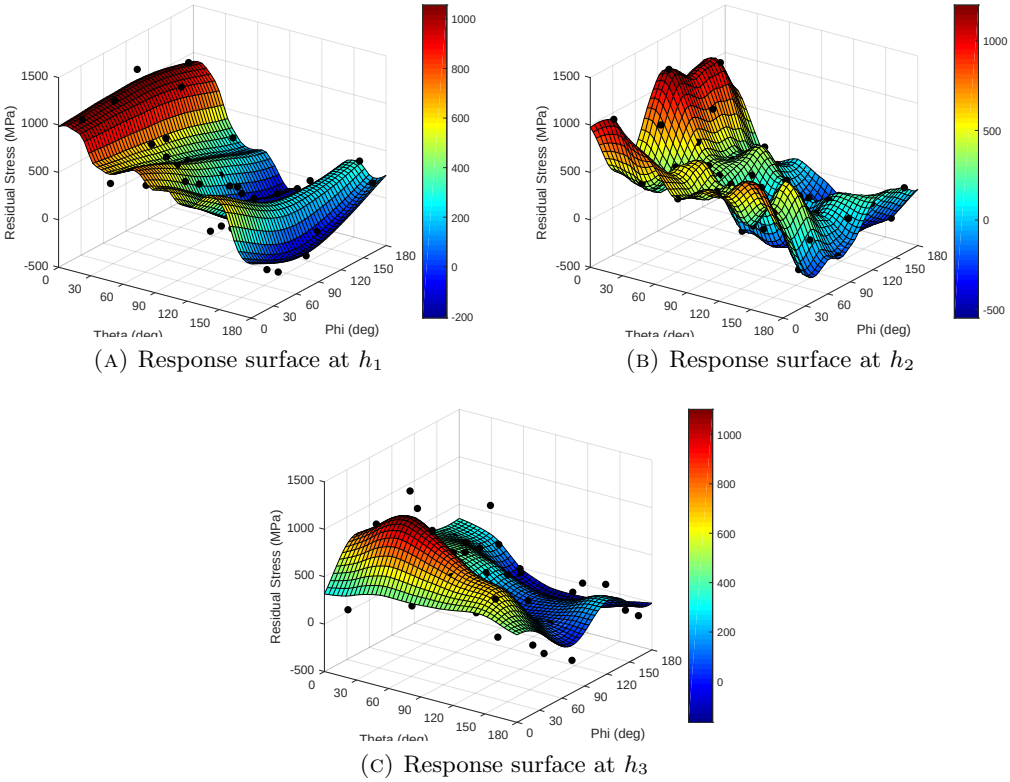


FIGURE C.6: Locations of 50 high stress mesh elements (recorded for each build orientation in the DOE) marked in red on the test geometry.

surfaces; moreover, evaluating 50 surrogate models is not computationally practical, so it is suggested that the number of high-stress points should be reduced to no more than 15. This can be done using a cluster analysis method (e.g. k-means clustering) to represent several high-stress points shown in Figure C.6 (for instance, the clusters of points around the bosses can be replaced with two or three points around each boss of the test geometry).



(A) Response surface at h_1

(B) Response surface at h_2

(C) Response surface at h_3

FIGURE C.7: Comparison of response surfaces of nearby high-stress points h_1, h_2 and h_3.

However, before applying a cluster analysis method, the response surfaces of three closely positioned high-stress points, denoted by h_1, h_2 and h_3, were compared for consistency; these response surfaces are shown in Figure C.7 and their positions are

indicated in Figure C.6. As can be seen, the three high-stress points, which all lie within a maximum distance of 1.3 mm of one another, have very different response surfaces. Moreover, all three response surfaces have values of R^2 ranging between 0.51 and 0.69; RMSE ranging between 14% and 15% and maximum errors ranging between 33.7% and 46.2%. Consequently, the produced response surfaces were considered too inaccurate and no further work was done in this direction. Additionally, the significant variation within each response surface (i.e. large number of peaks and valleys) and the inconsistencies between very closely positioned high-stress points indicate that there may be a high level of noise in the FEA solutions.

C.3 FEA Noise Study

Following the above results, the FEA model was studied more closely to find the source of the apparent noise. FEA convergence was found to be sufficient with negligible variations in stress ($< 1\%$) between multiple runs of the same mesh. Additionally, the variation of stress for a fixed point in the geometry was studied over a very small range of rotations, where $\theta \in (0^o, 4^o)$ and $\phi = 0^o$. Figure C.8 shows the first (S_1), second (S_2) and third (S_3) principal stresses, as well as stress intensity S_{int} for the fixed point over the range of rotations. As can be seen there are large and arguably unrealistic variations in all three components of principal stress. Furthermore, the stress intensity, which is expected to smooth out the noise, still indicates relatively large variations, even over 0.5 degree increments.

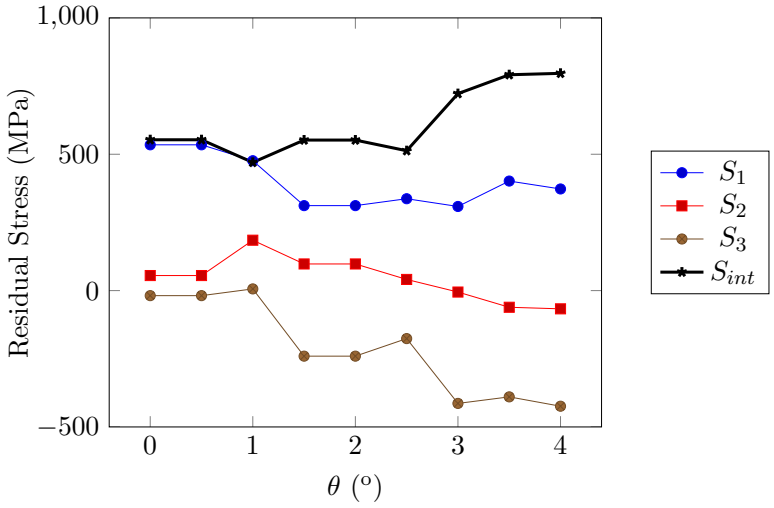


FIGURE C.8: Change in principal stresses and overall stress intensity over a range of $\theta \in (0^o, 4^o)$.

As discussed in Section 5.3, one of the simplifying assumptions in the FEA is the grouping of layers in order to make the model tractable at the expense of some loss in accuracy. Section 5.3 also explains that a voxel mesh is most likely to produce a successfully converged FEA solution while producing the least accurate representation of the part geometry, particularly at the surface where most tensile stress spots are

expected to occur. Furthermore, due to the layer-by-layer nature of the FEA the geometry is re-meshed for each new build orientation (as opposed to rotating the same mesh, which would result in irregular layers). This means that the location of each high-stress point is not fixed between different build orientation, but rather, this location is approximated as the centre point of the nearest element in each mesh. Therefore, it is expected that the FEA and geometric errors can be reduced significantly by reducing both the layer thickness and mesh element size; however, this is also expected to substantially increase the solution time. For this reason, it was decided to maintain the current mesh size of 0.5 mm and layer thickness of 1 mm.

Because of the high local variation of stresses in the mesh, the above high-stress point approach was not pursued further. Moreover, while it is possible to manually tune the level of data-fitting and regression of the Kriging models this was also avoided for two reasons; firstly, such manual tuning can lead to numerical problems (e.g. ill-conditioned matrices); secondly, these problems can be avoided by substituting Kriging with another less sophisticated approach (e.g. linear interpolation via triangulation), which can still provide a sufficient surrogate model for optimisation as demonstrated in Chapter 5.

Appendix D

Test Geometries

All test geometries used in Chapter 6 are listed in Table D.1 and Table D.2, where V denotes the geometry volume; SA denotes the surface area; VR and MLR denote the volume ratio and the maximum length ratio, respectively; and F denotes the number of facets in the STL file. Where applicable, source links are provided in the footnotes below. Additionally, images corresponding to each numbered geometry in Table D.1 can be found in figures D.1 and D.2.

Group & Source	*No.*	*Geometry Name*	*V (mm^3)*	*SA (mm^2)*	*VR*	*MLR*	*F*
RC Jet Assembly	1	Tube	19,870	16,184	0.14	0.25	247,318
(GrabCAD[1])	2	Axle	18,750	6,078	0.44	0.39	1,296
	3	Bearing	1,920	1,489	0.57	0.05	2,516
	4	Bracket	6,900	14,739	0.02	0.39	308,318
	5	Compressor bearing	2,770	2,504	0.32	0.07	34,424
	6	Compressor housing	48,970	30,367	0.11	0.27	77,780
	7	RC compressor wheel	8,760	11,461	0.13	0.13	15,640
	8	Diffuser	34,770	31,326	0.09	0.25	554,636
	9	Exhaust inner cone	3,280	9,492	0.05	0.11	44,816
	10	Exhaust nozzle	8,240	28,327	0.02	0.21	71,812
	11	Front bolt	880	710	0.47	0.04	1,852
	12	Front casing	11,960	40,358	0.02	0.27	127,916
	13	Jet Turbine bearing	305	435	0.42	0.03	1,374
	14	NGV ring	10,520	17,836	0.08	0.17	60,350
	15	Rear bolt	477	504	0.56	0.03	1,012
	16	Rear casing	57,230	82,533	0.04	0.35	170,028
	17	Spark plug	569	487	0.33	0.05	1,750
	18	Spring	767	2,131	0.16	0.06	192,536
	19	Starter	56,830	9,866	0.20	0.23	620,764
	20	Turbine casing	4,633	4,792	0.23	0.19	9,328
	21	RC turbine wheel	15,560	14,763	0.06	0.24	9,386
Turbocharger Assembly	22	Clamp	9,780	7,977	0.15	0.19	8,218
(GrabCAD[2])	23	Compressor flow	69,234	43,022	0.15	0.31	22,236
	24	TC compressor wheel	8,761	11,464	0.13	0.13	23,316
	25	TC housing	121,353	33,752	0.23	0.24	9,336
	26	Socket & cap screw	20,529	5,668	0.46	0.13	3,100
	27	Turbine flow	81,678	55,912	0.11	0.34	16,548
	28	TC turbine wheel	15,664	14,769	0.06	0.24	12,578

TABLE D.1: Test geometries and their properties.

[1] https://grabcad.com/library/rc-jet-engine-3
[2] https://grabcad.com/library/turbocharger-24

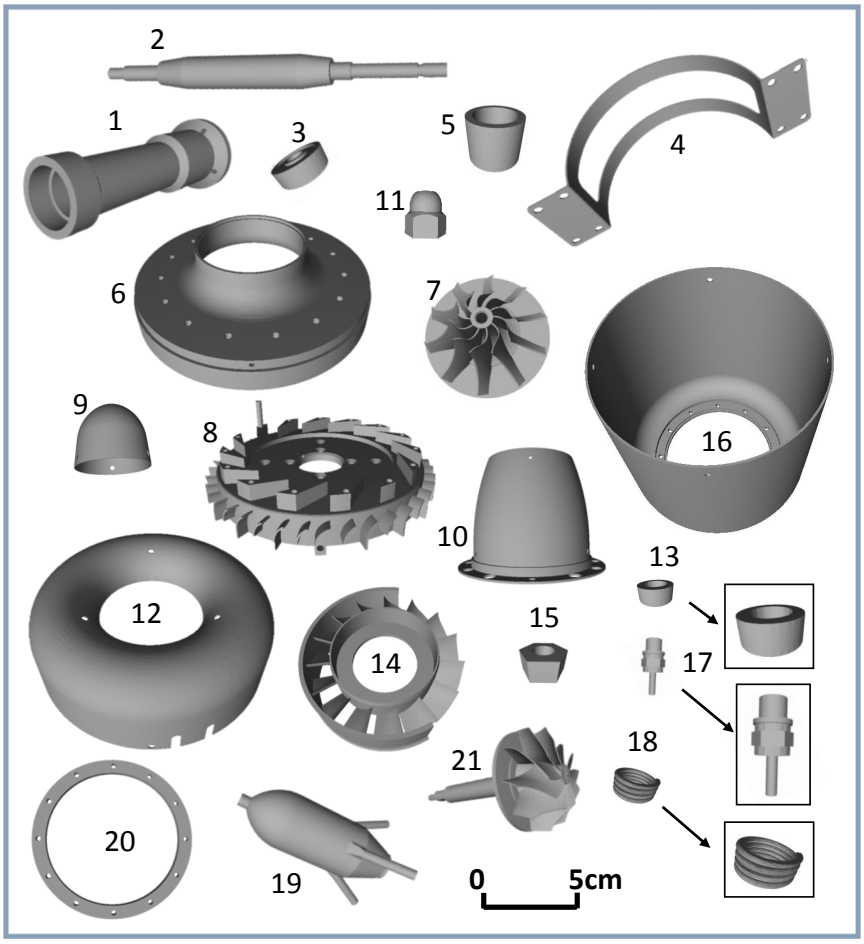


FIGURE D.1: Test geometries in the RC Jet Assembly (taken from GrabCAD).

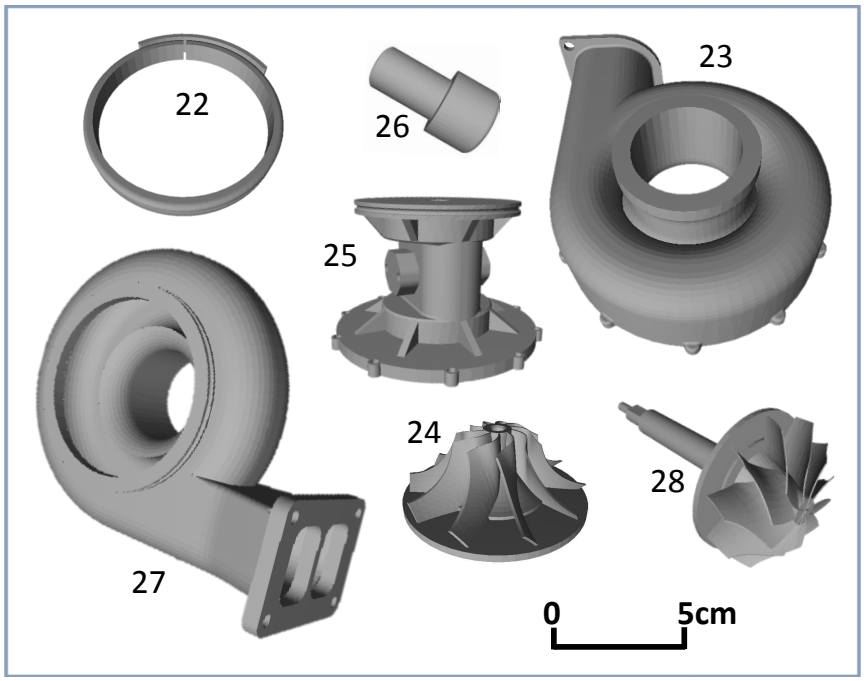


FIGURE D.2: Test geometries in the Turbocharger Assembly (taken from GrabCAD).

Group & Source	*Geometry Name*	*V* (mm^3)	*SA* (mm^2)	*VR*	*MLR*	*F*
Araujo et al. (2015)	Bearing block	96,646	28,939	0.19	0.32	12,846
	Belt link	16,595	8,057	0.55	0.15	16,148
	End cap	1,766	2,217	0.23	0.09	26,892
	Turbine	20,618	11,625	0.25	0.13	107,806
	Venturi tube	960	1,042	0.40	0.07	59,294
Canellidis et al. (2006, 2009)	Caliper	51,601	23,483	0.17	0.21	43,154
	Card slot cover	11,759	14,710	0.09	0.23	12,498
	Cover	19,162	17,478	0.25	0.26	12,300
	Distributor	7,149	13,189	0.10	0.14	4,028
	Engine block	38,595	28,112	0.18	0.21	4,924
	Flip-top	2,073	4,427	0.07	0.15	6,552
	Gear	540	659	0.39	0.04	1,420
	Handle	28,250	29,844	0.08	0.27	9,698
	Impeller	80,661	40,349	0.17	0.28	474,222
	Insect trap	5,631	16,474	0.07	0.17	50,910
	Jawbone	73,270	22,190	0.08	0.31	100,948
	Phone cover	53,704	57,385	0.08	0.45	30,004
	Pipe support	2,296	2,959	0.24	0.09	620
	Sector	90,050	52,234	0.10	0.50	14,574
	Swivel hinge	10,952	7,647	0.22	0.15	4,860
	Washing machine arm (centre section)	28,552	29,441	0.14	0.30	17,718
	Washing machine arm (left section)	20,550	20,624	0.17	0.30	10,394
	Washing machine arm (right section)	20,551	20,624	0.17	0.30	10,370
Gearbox Assembly (GrabCAD[3])	Cover plate	27,134	13,880	0.38	0.28	19,972
	GB housing	282,507	69,052	0.28	0.42	66,658
	Offset shaft	31,520	8,798	0.78	0.42	6,596
	Plate lug	78,908	20,952	0.28	0.23	19,294
	Worm gear	37,698	14,996	0.39	0.15	16,262
	Worm gear shaft	23,995	7,243	0.26	0.34	5,502
Rolls Royce	Combustor section 1	46,455	44,154	0.06	0.43	20,648
	Combustor section 2	64,000	48,494	0.15	0.53	8,186
	Seal segment	14,544	25,798	0.14	0.25	51,376
Miscellaneous Parts (GrabCAD)	Aero-bearing bracket[4]	40,806	20,037	0.12	0.30	32,686
	Control arm	90,512	33,471	0.13	0.42	7,654
	Fuel injector[5]	8,077	7,644	0.13	0.16	39,272
	GE bracket[6]	76,857	78,004	0.06	0.44	84,468
	Housing1[7]	40,840	26,450	0.13	0.21	48,586
	Housing2[8]	65,620	31,605	0.14	0.24	25,702
	Space bracket[9]	30,540	13,682	0.14	0.25	75,982
	Turbine blade[10]	13,216	10,478	0.11	0.22	31,230

TABLE D.2: Test geometries and their properties (continued).

[3] https://grabcad.com/library/gear-box-assembly-1
[4] https://grabcad.com/challenges/airplane-bearing-bracket-challenge
[5] https://grabcad.com/library/fuel-injector--2
[6] https://grabcad.com/challenges/ge-jet-engine-bracket-challenge
[7] https://grabcad.com/library/crankcase--4
[8] https://grabcad.com/library/aero-engine-housing/details?folder_id=52721
[9] https://grabcad.com/challenges/3-2-1-liftoff-ula-rocket-hardware-challenge
[10] https://grabcad.com/library/turbine-blade--4

References

3T RPD (2018). Metal Additive Manufacturing Studies. Autodesk Within - heat exchanger challenges designers. Accessed May 24, 2018. *http://www.3trpd.co.uk/portfolio/within-heat-exchanger-challenges-designers/gallery/metal-additive-manufacturing-case-studies/*.

Aboulkhair, N. T., Everitt, N. M., Ashcroft, I., and Tuck, C. (2014). Reducing porosity in AlSi10Mg parts processed by selective laser melting. *Additive Manufacturing*, 1-4:77–86.

Agarwala, M., Bourell, D., Beaman, J., Marcus, H., and Barlow, J. (1995). Direct selective laser sintering of metals. *Rapid Prototyping Journal*, 1(1):26–36.

Ahn, D., Kim, H., and Lee, S. (2007). Fabrication direction optimization to minimize post-machining in layered manufacturing. *International Journal of Machine Tools and Manufacture*, 47(3-4):593–606.

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

Alfaify, A. Y., Hughes, J., and Ridgway, K. (2018). Critical evaluation of the pulsed selective laser melting process when fabricating Ti64 parts using a range of particle size distributions. *Additive Manufacturing*, 19:197–204.

Ali, H., Ghadbeigi, H., and Mumtaz, K. (2018). Effect of scanning strategies on residual stress and mechanical properties of Selective Laser Melted Ti6Al4V. *Materials Science and Engineering A*, 712:175–187.

Ali, H., Ma, L., Ghadbeigi, H., and Mumtaz, K. (2017). In-situ residual stress reduction, martensitic decomposition and mechanical properties enhancement through high temperature powder bed pre-heating of Selective Laser Melted Ti-6Al-4V. *Materials Science and Engineering A*, 695:211–220.

Allaire, G., Dapogny, C., Estevez, R., Faure, A., and Michailidis, G. (2017). Structural optimization under overhang constraints imposed by additive manufacturing technologies. *Journal of Computational Physics*, 351:295–328.

Amato, K. N., Gaytan, S. M., Murr, L. E., Martinez, E., Shindo, P. W., Hernandez, J., Collins, S., and Medina, F. (2012). Microstructures and mechanical behavior of Inconel 718 fabricated by selective laser melting. *Acta Materialia*, 60(5):2229–2239.

Araujo, L. L., Ozcan, E., Atkin, J., Baumers, M., Tuck, C., Hague, R. J., Özcan, E., Atkin, J., Baumers, M., Tuck, C., and Hague, R. J. (2015). Toward Better Build Volume Packing In Additive Manufacturing: Classification Of Existing Problems And Benchmarks. pages 401–410, Austin, Texas. University of Nottingham.

Art, R. (1966). An approach to the two dimensional irregular cutting stock problem. Technical report.

Atzeni, E. and Salmi, A. (2012). Economics of additive manufacturing for end-usable metal parts. *International Journal of Advanced Manufacturing Technology*, 62(9-12):1147–1155.

Basu, S. (2012). Tabu Search Implementation on Traveling Salesman Problem and Its Variations: A Literature Survey. *American Journal of Operations Research*, 02(02):163–173.

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

Baumers, M., Dickens, P., Tuck, C., and Hague, R. (2016). The cost of additive manufacturing: Machine productivity, economies of scale and technology-push. *Technological Forecasting and Social Change*, 102:193–201.

Baumers, M. and Holweg, M. (2016). Cost Impact of the Risk of Build Failure in Laser Sintering. *Proceedings of the 26th Annual International Solid Freeform Fabrication Symposium*, 1910:1910–1921.

Baumers, M., Tuck, C., Bourell, D. L., Sreenivasan, R., and Hague, R. (2011). Sustainability of additive manufacturing: measuring the energy consumption of the laser sintering process. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 225(12):2228–2239.

Baumers, M., Tuck, C., Wildman, R., Ashcroft, I., Rosamond, E., and Hague, R. (2013). 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.

Baykasoglu, A. (2006). Applying multiple objective tabu search to continuous optimization problems with a simple neighbourhood strategy. *International Journal for Numerical Methods in Engineering*, 65(3):406–424.

Bennell, J. A. and Oliveira, J. F. (2008). The geometry of nesting problems: A tutorial. *European Journal of Operational Research*, 184(2):397–415.

Berumen, S., Bechmann, F., Lindner, S., Kruth, J.-P., and Craeghs, T. (2010). Quality control of laser- and powder bed-based Additive Manufacturing (AM) technologies. *Physics Procedia*, 5:617–622.

Buchbinder, D., Meiners, W., Pirch, N., Wissenbach, K., and Schrage, J. (2014). Investigation on reducing distortion by preheating during manufacture of aluminum components using selective laser melting. *Journal of Laser Applications*, 26(1):012004.

Burke, E., Hellier, R., Kendall, G., and Whitwell, G. (2006). A New Bottom-Left-Fill Heuristic Algorithm for the Two-Dimensional Irregular Packing Problem. *Operations Research*, 54(3):587–601.

Burke, E. K., Hellier, R. S., Kendall, G., and Whitwell, G. (2007). Complete and robust no-fit polygon generation for the irregular stock cutting problem. *European Journal of Operational Research*, 179(1):27–49.

Byun, H. S. and Lee, K. H. (2006). Determination of the optimal build direction for different rapid prototyping processes using multi-criterion decision making. *Robotics and Computer-Integrated Manufacturing*, 22(1):69–80.

Cain, V., Thijs, L., Van Humbeeck, J., Van Hooreweder, B., and Knutsen, R. (2015). Crack propagation and fracture toughness of Ti6Al4V alloy produced by selective laser melting. *Additive Manufacturing*, 5:68–76.

Calignano, F. (2014). Design optimization of supports for overhanging structures in aluminum and titanium alloys by selective laser melting. *Materials & Design*, 64:203–213.

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

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

Canellidis, V., Giannatsis, J., and Dedoussis, V. (2010). Effective nesting of layer manufacturing fabricated parts using a genetic algorithm and a bottom-left ray casting procedure. pages 547–551.

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

Carter, L. N., Attallah, M. M., and Reed, R. C. (2012). Laser Powder Bed Fabrication of Nickel-Base Superalloys: Influence of Parameters; Characterisation, Quantification and Mitigation of Cracking. pages 577–586.

Carter, L. N., Martin, C., Withers, P. J., and Attallah, M. M. (2014). The influence of the laser scan strategy on grain structure and cracking behaviour in SLM powder-bed fabricated nickel superalloy. *Journal of Alloys and Compounds*, 615:338–347.

Cavalieri, S., Maccarrone, P., and Pinto, R. (2004). Parametric vs. neural network models for the estimation of production costs: A case study in the automotive industry. *International Journal of Production Economics*, 91(2):165–177.

Chen, C. C. and Sullivan, P. A. (1996). Predicting total build-time and the resultant cure depth of the 3D stereolithography process. *Rapid Prototyping Journal*, 2(4):27–40.

Cheng, W., Fuh, J., Nee, A., Wong, Y., Loh, H., and Miyazawa, T. (1995). Multi-objective optimization of part- building orientation in stereolithography. *Rapid Prototyping Journal*, 1(4):12–23.

Chlebus, E., Kuźnicka, B., Kurzynowski, T., and Dybała, B. (2011). Microstructure and mechanical behaviour of Ti-6Al-7Nb alloy produced by selective laser melting. *Materials Characterization*, 62(5):488–495.

Choi, S. H. and Samavedam, S. (2002). Modelling and optimisation of rapid prototyping. *Computers in Industry*, 47(1):39–53.

Chowdhury, S., Mhapsekar, K., and Anand, S. (2017). Part Build Orientation Optimization and Geometry Compensations for Additive Manufacturing Process. *Journal of Manufacturing Science and Engineering*, 140(3):031009.

Cloots, M., Spierings, A., and Wegener, K. (2013). Assessing new support minimizing strategies for the additive manufacturing technology SLM. pages 131–139.

Coffman, E. G., Galambos, G., Martello, S., and Vigo, D. (1999). Bin Packing Approximation Algorithms: Combinatorial Analysis. pages 151–207.

Cooper, D. E., Stanford, M., Kibble, K. A., and Gibbons, G. J. (2012). Additive Manufacturing for product improvement at Red Bull Technology. *Materials and Design*, 41:226–230.

Das, P., Chandran, R., Samant, R., and Anand, S. (2015). Optimum Part Build Orientation in Additive Manufacturing for Minimizing Part Errors and Support Structures. *Procedia Manufacturing*, 1:343–354.

Das, P., Mhapsekar, K., Chowdhury, S., Samant, R., and Anand, S. (2017). Selection of build orientation for optimal support structures and minimum part errors in additive manufacturing. *Computer-Aided Design and Applications*, 14:1–13.

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

Deckard, C. R. (1989). Method and apparatus for producing parts by selective sintering.

Delfs, P., Tows, M., and Schmid, H. J. (2016). Optimized build orientation of additive manufactured parts for improved surface quality and build time. *Additive Manufacturing*, 12:314–320.

Delgado, J., Ciurana, J., and Rodríguez, C. A. (2012). Influence of process parameters on part quality and mechanical properties for DMLS and SLM with iron-based materials. volume 60, pages 601–610.

Di Angelo, L. and Di Stefano, P. (2011). A neural network-based build time estimator for layer manufactured objects. *International Journal of Advanced Manufacturing Technology*, 57(1-4):215–224.

Dickinson, J. J. K. and Knopf, G. K. G. (1998). Serial packing of arbitrary 3D objects for optimizing layered manufacturing. *Intelligent Robots and Computer Vision XVII: Algorithms, Techniques, and Active Vision*, 3522(November):130–138.

Dodd, C. (2015). *A Paradigm to Maximise Performance and Profitability of Engineering Products in the Presence of Manufacturing Uncertainty*. PhD thesis, University of Southampton.

Dub, A. V., Beregovskiy, V. V., Tretyakov, E. V., and Shchurenkova, S. A. (2016). Experimental investigations of the nickel alloy laser melting parameters influence on porosity and surface roughness of the complex geometry products during the process of their three-dimensional formation. *Non-ferrous Metals*, (2):29–33.

Duckham, M., Kulik, L., Worboys, M., and Galton, A. (2008). Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern Recognition*, 41(10):3224–3236.

English, C. L., Tewari, S. K., and Abbott, D. H. (2010). An overview of ni base additive fabrication technologies for aerospace applications.

Farinia Group (2018). Industrial Production Systems for Metal Additive Manufacturing. Accessed June 6, 2018. *https://www.farinia.com/additive-manufacturing/3d-technique/metal-additive-manufacturing-production-systems*.

Fera, M., Fruggiero, F., Costabile, G., Lambiase, A., and Pham, D. T. (2017). A new mixed production cost allocation model for additive manufacturing (MiPro-CAMAM). *International Journal of Advanced Manufacturing Technology*, 92(9-12):4275–4291.

Ford, S. and Despeisse, M. (2016). Additive manufacturing and sustainability: an exploratory study of the advantages and challenges. *Journal of Cleaner Production*, 137:1573–1587.

Frank, D. and Fadel, G. (1995). Expert system-based selection of the preferred direction of build for rapid prototyping processes. *Journal of Intelligent Manufacturing*, 6(5):339–345.

Garrett, B. (2014). 3D printing: New economic paradigms and strategic shifts. *Global Policy*, 5(1):70–75.

Gaynor, A. T. and Guest, J. K. (2016). Topology optimization considering overhang constraints: Eliminating sacrificial support material in additive manufacturing through design. *Structural and Multidisciplinary Optimization*, 54(5):1157–1172.

Ghobbar, A. (2004). Forecasting Intermittent Demand for Aircraft Spare Parts: A Comparative Evaluation of Methods. *Journal of Aircraft*, 41(3):665–673.

Giannatsis, J., Dedoussis, V., and Laios, L. (2001). A study of the build-time estimation problem for Stereolithography systems. *Robotics and Computer-Integrated Manufacturing*, 17(4):295–304.

Gilmore, P. C. and Gomory, R. E. (1961). A Linear Programming Approach to the Cutting-Stock Problem. *Operations Research*, 9(6):849–859.

Glover, F. (1989). Tabu Search – Part I. *ORSA Journal on Computing*, 1(3):190–206.

Glover, F. (1990). Tabu Search – Part II. *ORSA Journal on Computing*, 2(1):4–32.

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

Goh, G. D., Agarwala, S., Goh, G. L., Dikshit, V., Sing, S. L., and Yeong, W. Y. (2017). Additive manufacturing in unmanned aerial vehicles (UAVs): Challenges and potential. *Aerospace Science and Technology*, 63:140–151.

Gong, H., Rafi, K., Gu, H., Starr, T., and Stucker, B. (2014). Analysis of defect generation in Ti6Al4V parts made using powder bed fusion additive manufacturing processes. *Additive Manufacturing*, 1:87–98.

Gordon, E. R., Shokrani, A., Flynn, J. M., Goguelin, S., Barclay, J., and Dhokia, V. (2016). A Surface Modification Decision Tree to Influence Design in Additive Manufacturing. *Sustainable Design and Manufacturing*, 52:423–434.

Gorsse, S., Hutchinson, C., Gouné, M., and Banerjee, R. (2017). Additive manufacturing of metals: a brief review of the characteristic microstructures and properties of steels, Ti6Al4V and high-entropy alloys. *Science and Technology of Advanced Materials*, 18(1):584–610.

Gusarov, A. V., Pavlov, M., and Smurov, I. (2011). Residual stresses at laser surface remelting and additive manufacturing. volume 12, pages 248–254.

Hajji, O., Brisset, S., and Brochet, P. (2004). A new tabu search method for optimization with continuous parameters. *IEEE Transactions on Magnetics*, 40(2):1184–1187.

Hajji, O., Brisset, S., and Brochet, P. (2005). A new tabu search method for continuous parameter optimization: application to design problems in electromagnetic. *European Transactions on Electrical Power*, 15(6):527–540.

Harrison, N. J., Todd, I., and Mumtaz, K. (2015). Reduction of micro-cracking in nickel superalloys processed by Selective Laser Melting: A fundamental alloy design approach. *Acta Materialia*, 94:59–68.

Hellemann, L., English, C., Borne, B., and Glover, R. (2003). Blisk Weld Repair.

Holmström, J., Partanen, J., Tuomi, J., and Walter, M. (2010). Rapid manufacturing in the spare parts supply chain. *Journal of Manufacturing Technology Management*, 21(No.6):687–697.

Hopkinson, N. and Dicknes, P. (2003). 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.

Horn, T. J. and Harrysson, O. L. (2012). Overview of current additive manufacturing technologies and selected applications. *Science Progress*, 95(3):255–282.

Hu, K., Jin, S., and Wang, C. C. (2015). Support slimming for single material based additive manufacturing. *CAD Computer Aided Design*, 65:1–10.

Huang, R., Riddle, M., Graziano, D., Warren, J., Das, S., Nimbalkar, S., Cresko, J., and Masanet, E. (2016). Energy and emissions saving potential of additive manufacturing: thecase of lightweight aircraft components. *Journal of Cleaner Production*, 135:1559–1570.

Hull, C. W. (1986). Apparatus for production of three-dimensional objects by stereolithography.

Hur, J. and Lee, K. (1998). The development of a CAD environment to determine the preferred build-up direction for layered manufacturing. *International Journal of Advanced Manufacturing Technology*, 14(4):247–254.

Hur, S.-M., Choi, K.-H., Lee, S.-H., and Chang, P.-K. (2001). Determination of fabricating orientation and packing in SLS process. *Journal of Materials Processing Technology*, 112(2-3):236–243.

Hussein, A., Hao, L., Yan, C., Everson, R., and Young, P. (2013). Advanced lattice support structures for metal additive manufacturing. *Journal of Materials Processing Technology*, 213(7):1019–1026.

Ikonen, I., Biles, W. E., Kumar, A., Ragade, R. K., and Wissel, J. C. (1997). A Genetic Algorithm for Packing Three-Dimensional Non-Convex Objects Having Cavities and Holes. *International Computer Games Association*, pages 591–598.

Imahori, S., Yagiura, M., and Nagamochi, H. (2006). Practical algorithms for two-dimensional packing. Technical Report March, The University of Tokyo.

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

Jaiswal, P., Patel, J., and Rai, R. (2018). Build orientation optimization for additive manufacturing of functionally graded material objects. *International Journal of Advanced Manufacturing Technology*, 96(1-4):223–235.

Jakobs, S. (1996). On genetic algorithms for the packing of polygons. *European Journal of Operational Research*, 88(1):165–181.

Järvinen, J.-P., Matilainen, V., Li, X., Piili, H., Salminen, A., Mäkelä, I., and Nyrhilä, O. (2014). Characterization of Effect of Support Structures in Laser Additive Manufacturing of Stainless Steel. *Physics Procedia*, 56:72–81.

Kalentics, N., Boillat, E., Peyre, P., Gorny, C., Kenel, C., Leinenbach, C., Jhabvala, J., and Logé, R. E. (2017). 3D Laser Shock Peening A new method for the 3D control of residual stresses in Selective Laser Melting. *Materials and Design*, 130:350–356.

Kamath, C., El-Dasher, B., Gallegos, G. F., King, W. E., and Sisto, A. (2014). Density of additively-manufactured, 316L SS parts using laser powder-bed fusion at powers up to 400 W. Technical Report 1-4, Lawrence Livermore National Laboratory (LLNL), Livermore, CA (United States).

Karimi, A., Nobahari, H., and Siarry, P. (2010). Continuous ant colony system and tabu search algorithms hybridized for global minimization of continuous multi-minima functions. *Computational Optimization and Applications*, 45(3):639–661.

Keller, N., Neugebauer, F., Xu, H., and Ploshikhin, V. (2013). Thermo-mechanical Simulation of Additive Layer Manufacturing of Titanium Aerospace structures. *LightMAT Conference*, 3(5).

Kellner, T. (2014). GE Reports. Fit to Print: New Plant Will Assemble World's First Passenger Jet Engine with 3-D Printed Fuel Nozzles, Next-Gen Materials. Accessed May 24, 2018. *http://www.gereports.com/post/80701924024/fit-to-print*.

Kempen, K., Thijs, L., Van Humbeeck, J., and Kruth, J.-P. (2015). Processing AlSi10Mg by selective laser melting: parameter optimisation and material characterisation. *Materials Science and Technology*, 31(8):917–923.

Kempen, K., Vrancken, B., Buls, S., Thijs, L., Van Humbeeck, J., and Kruth, J.-P. (2014). Selective Laser Melting of Crack-Free High Density M2 High Speed Steel Parts by Baseplate Preheating. *Journal of Manufacturing Science and Engineering*, 136(6):061026.

Khaing, M. W., Fuh, J. Y. H., and Lu, L. (2001). Direct metal laser sintering for rapid tooling: Processing and characterisation of EOS parts. *Journal of Materials Processing Technology*, 113(1-3):269–272.

Khajavi, S. H., Partanen, J., and Holmström, J. (2014). Additive manufacturing in the spare parts supply chain. *Computers in Industry*, 65(1):50–63.

Kok, Y., Tan, X. P., Wang, P., Nai, M. L., Loh, N. H., Liu, E., and Tor, S. B. (2018). Anisotropy and heterogeneity of microstructure and mechanical properties in metal additive manufacturing: A critical review. *Materials and Design*, 139:565–586.

Kruth, J.-P., Badrossamay, M., Yasa, E., Deckers, J., Thijs, L., and Van Humbeeck, J. (2010). Part and material properties in selective laser melting of metals. *16th International Symposium on Electromachining*, pages 1–12.

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

Langelaar, M. (2018). Combined optimization of part topology, support structure layout and build orientation for additive manufacturing. *Structural and Multidisciplinary Optimization*, 57:1–20.

Laureijs, R. E., Roca, J. B., Narra, S. P., Montgomery, C., Beuth, J. L., and Fuchs, E. R. H. (2017). Metal Additive Manufacturing: Cost Competitive Beyond Low Volumes. *Journal of Manufacturing Science and Engineering*, 139(8):081010.

Leary, M. (2016). Surface roughness optimisation for selective laser melting (SLM): Accommodating relevant and irrelevant surfaces. *Laser Additive Manufacturing: Materials, Design, Technologies, and Applications*, pages 99–118.

Leary, M., Merli, L., Torti, F., Mazur, M., and Brandt, M. (2014). Optimal topology for additive manufacture: A method for enabling additive manufacture of support-free optimal structures. *Materials and Design*, 63:678–690.

Lefky, C. S., Zucker, B., Wright, D., Nassar, A. R., Simpson, T. W., and Hildreth, O. J. (2017). Dissolvable Supports in Powder Bed Fusion-Printed Stainless Steel. *3D Printing and Additive Manufacturing*, 4(1):3–11.

Leuders, S., Thöne, M., Riemer, A., Niendorf, T., Tröster, T., Richard, H. A., and Maier, H. J. (2013). On the mechanical behaviour of titanium alloy TiAl6V4 manufactured by selective laser melting: Fatigue resistance and crack growth performance. *International Journal of Fatigue*, 48:300–307.

Levatti, H. U., Innocente, M. S., Morgan, H. D., Cherry, J., Lavery, N. P., Mehmood, S., Cameron, I., and Sienz, J. (2014). Computational Methodology for Optimal Design of Additive Layer Manufactured Turbine Bracket. *Sustainable Design and Manufacturing*, 1(1):641.

Lewandowski, J. J. and Seifi, M. (2016). Metal Additive Manufacturing: A Review of Mechanical Properties. *Annual Review of Materials Research*, 46(1):151–186.

Liefvendahl, M. and Stocki, R. (2006). A study on algorithms for optimization of Latin hypercubes. *Journal of Statistical Planning and Inference*, 136(9):3231–3247.

Lindemann, C., Jahnke, U., Moi, M., and Koch, R. (2012). Analyzing product lifecycle costs for a better understanding of cost drivers in additive manufacturing. *International Solid Freeform Fabrication Symposium*, 23:177–188.

Lindmark, J. (2013). *No Fit Polygon problem : Developing a complete solution to the No Fit Polygon problem.* PhD thesis, Royal Institute of Technology, Sweden.

Liu, P., Huang, S. H., Mokasdar, A., Zhou, H., and Hou, L. (2014). The impact of additive manufacturing in the aircraft spare parts supply chain: Supply chain operation reference (scor) model based analysis. *Production Planning and Control*, 25(13-14):1169–1181.

López-Castro, J. D., Marchal, A., González, L., and Botana, J. (2017). Topological optimization and manufacturing by Direct Metal Laser Sintering of an aeronautical part in 15-5PH stainless steel. *Procedia Manufacturing*, 13:818–824.

Ma, D., Lin, F., and Chua, C. K. (2001). Rapid prototyping applications in medicine. Part 2: STL file generation and case studies. *International Journal of Advanced Manufacturing Technology*, 18(2):118–127.

Madison, J. D. and Aagesen, L. K. (2012). Quantitative characterization of porosity in laser welds of stainless steel. *Scripta Materialia*, 67(9):783–786.

Marimuthu, S., Triantaphyllou, A., Antar, M., Wimpenny, D., Morton, H., and Beard, M. (2015). Laser polishing of selective laser melted components. *International Journal of Machine Tools and Manufacture*, 95:97–104.

Martinez-Sykora, A., Alvarez-Valdes, R., Bennell, J. A., Ruiz, R., and Tamarit, J. M. (2017). Matheuristics for the irregular bin packing problem with free rotations. *European Journal of Operational Research*, 258(2):440–455.

Masood, S. H., Rattanawong, W., and Iovenitti, P. (2000). Part build orientations based on volumetric error in fused deposition modelling. *International Journal of Advanced Manufacturing Technology*, 16(3):162–168.

Masood, S. H., Rattanawong, W., and Iovenitti, P. (2003). A generic algorithm for a best part orientation system for complex parts in rapid prototyping. *Journal of Materials Processing Technology*, 139(1-3 SPEC):110–116.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 21(2):239.

Megahed, M., Mindt, H.-W., N'Dri, N., Duan, H., and Desmaison, O. (2016). Metal additive-manufacturing process and residual stress modeling. *Integrating Materials and Manufacturing Innovation*, 5(1):4.

Mellor, S., Hao, L., and Zhang, D. (2014). Additive manufacturing: A framework for implementation. *International Journal of Production Economics*, 149:194–201.

Mercelis, P. and Kruth, J.-P. (2006). Residual stresses in selective laser sintering and selective laser melting. *Rapid Prototyping Journal*, 12(5):254–265.

Mertens, R., Clijsters, S., Kempen, K., and Kruth, J.-P. (2014). Optimization of Scan Strategies in Selective Laser Melting of Aluminum Parts With Downfacing Areas. *Journal of Manufacturing Science and Engineering*, 136(6):061012.

Morgan, D., Agba, E., and Hill, C. (2017). Support Structure Development and Initial Results for Metal Powder Bed Fusion Additive Manufacturing. *Procedia Manufacturing*, 10:819–830.

Mumtaz, K. and Hopkinson, N. (2009). Top surface and side roughness of Inconel 625 parts processed using selective laser melting. *Rapid Prototyping Journal*, 15(2):96–103.

Mumtaz, K. and Hopkinson, N. (2010). Selective Laser Melting of thin wall parts using pulse shaping. *Journal of Materials Processing Technology*, 210(2):279–287.

Mumtaz, K., Hopkinson, N., and Vora, P. (2011). A Method to Eliminate Anchors/Supports from Directly Laser Melted Powder Bed Processes.

Munguía, J., Ciurana, J., and Riba, C. (2009). Neural-network-based model for build-time estimation in selective laser sintering. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 223(8):995–1003.

Murr, L. E., Gaytan, S. M., Ramirez, D. A., Martinez, E., Hernandez, J., Amato, K. N., Shindo, P. W., Medina, F. R., and Wicker, R. B. (2012). Metal Fabrication by Additive Manufacturing Using Laser and Electron Beam Melting Technologies.

Myrtveit, I. and Stensrud, E. (1999). A controlled experiment to assess the benefits of estimating with analogy and regression models. *IEEE Transactions on Software Engineering*, 25(4):510–525.

Niendorf, T., Leuders, S., Riemer, A., Richard, H. A., Tröster, T., and Schwarze, D. (2013). Highly Anisotropic Steel Processed by Selective Laser Melting. *Metallurgical and Materials Transactions B*, 44(4):794–796.

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

Olakanmi, E. O., Cochrane, R. F., and Dalgarno, K. W. (2015). A review on selective laser sintering/melting (SLS/SLM) of aluminium alloy powders: Processing, microstructure, and properties. *Progress in Materials Science*, 74:401–477.

Palumbo, N. and Granville, N. (2016). MMM6010 - Additive Layer Manufacturing (ALM) Process Modelling. *Material and Mechanical Methods, DNS190249*.

Pandey, P. M., Thrimurthulu, K., and Reddy, N. V. (2004). Optimal part deposition orientation in FDM by using a multicriteria genetic algorithm. *International Journal of Production Research*, 42(19):4069–4089.

Patterson, A. E., Messimer, S. L., and Farrington, P. A. (2017). Overhanging Features and the SLM/DMLS Residual Stresses Problem: Review and Future Research Need. *Technologies*, 5(2):15.

Peng, A. H., Ghasri-khouzani, M., Gong, S., Attardo, R., Ostiguy, P., Gatrell, B. A., Budzinski, J., Tomonto, C., Neidig, J., Shankar, M. R., Billo, R., Go, D. B., and Hoelzle, D. (2017). Optimization of Build Orientation for Minimum Thermal Distortion in DMLS Metallic Additive Manufacturing. Number 2, pages 820–835.

Poyraz, Ö., Yasa, E., Akbulut, G., A.Orhangü, and Pilatin, S. (2015). Investigation of support structures for direct metal laser sintering (DMLS) of IN625 parts. *Internation Solid Free Form Fabrication Symposium*, pages 560–574.

Protasov, C. E., Safronov, V. A., Kotoban, D. V., and Gusarov, A. V. (2016). Experimental study of residual stresses in metal parts obtained by selective laser melting. volume 83, pages 825–832.

Rafi, H. K., Starr, T. L., and Stucker, B. E. (2013). A comparison of the tensile, fatigue, and fracture behavior of Ti6Al4V and 15-5 PH stainless steel parts made by selective laser melting. *International Journal of Advanced Manufacturing Technology*, 69(5-8):1299–1309.

Rao, H., Giet, S., Yang, K., Wu, X., and Davies, C. H. (2016). The influence of processing parameters on aluminium alloy A357 manufactured by Selective Laser Melting. *Materials and Design*, 109:334–346.

Reed, N. (2010). *Development and evaluation of a design support system (EngD Thesis)*. PhD thesis, University of Southampton.

Rickenbacher, L., Spierings, A., and Wegener, K. (2013). An integrated cost-model for selective laser melting (SLM). *Rapid Prototyping Journal*, 19(3):208–214.

Riemer, A., Leuders, S., Thöne, M., Richard, H. A., Tröster, T., and Niendorf, T. (2014). On the fatigue crack growth behavior in 316L stainless steel manufactured by selective laser melting. *Engineering Fracture Mechanics*, 120:15–25.

Roberts, I. A. (2012). *Investigation of residual stresses in the laser melting of powders in additive layer manufacturing*. PhD thesis, University of Wolverhampton.

Ruffo, M., Tuck, C., and Hague, R. (2006). Cost estimation for rapid manufacturing - Simultaneous production of mixed components using laser sintering. *Proceedings of the Institution of Mechanical Engineers, Part B*, 221(11):1585–1591.

Ruffo, M., Tuck, C., and Hague, R. (2007). Make or buy analysis for rapid manufacturing. *Rapid Prototyping Journal*, 13(1):23–29.

Saha, B., Wanhill, R. J., Eswara Prasad, N., Gouda, G., and Tamilmani, K. (2013). Airworthiness Certification of Metallic Materials. In *Aluminum-Lithium Alloys: Processing, Properties, and Applications*, pages 537–554.

Saka, M. P., Hasançebi, O., and Geem, Z. W. (2016). Metaheuristics in structural optimization and discussions on harmony search algorithm. *Swarm and Evolutionary Computation*, 28:88–97.

Sames, W. J., List, F. A., Pannala, S., Dehoff, R. R., and Babu, S. S. (2016). The metallurgy and processing science of metal additive manufacturing. *International Materials Reviews*, 6608(March):1–46.

Savalani, M. M. and Pizarro, J. M. (2016). Effect of preheat and layer thickness on selective laser melting (SLM) of magnesium. *Rapid Prototyping Journal*, 22(1):115–122.

Schmid, M. and Levy, G. (2012). Quality Management and Estimation of Quality Costs for Additive Manufacturing with SLS. *Direct Digital Manufacturing Conference*.

Schröder, M., Falk, B., and Schmitt, R. (2015). Evaluation of cost structures of additive manufacturing processes using a new business model. *Procedia CIRP*, 30:311–316.

Schwerdt, J., Smid, M., Janardan, R., and Johnson, E. (2000). Protecting critical facets in layered manufacturing: Implementation and experimental results. *CAD Computer Aided Design*, 35(7):647–657.

Shewchuck, J. R. (2002). Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry*, 22:21–74.

Shields, M. D. and Zhang, J. (2016). The generalization of Latin hypercube sampling. *Reliability Engineering and System Safety*, 148:96–108.

Shifeng, W., Shuai, L., Qingsong, W., Yan, C., Sheng, Z., and Yusheng, S. (2014). Effect of molten pool boundaries on the mechanical properties of selective laser melting parts. *Journal of Materials Processing Technology*, 214(11):2660–2667.

Simonelli, M., Tse, Y. Y., and Tuck, C. (2014). Effect of the build orientation on the mechanical properties and fracture modes of SLM Ti6Al4V. *Materials Science and Engineering A*, 616:1–11.

Singhal, S., Jain, P. K., Pandey, P. M., and Nagpal, A. (2009). Optimum part deposition orientation for multiple objectives in SL and SLS prototyping. *International Journal of Production Research*, 47(22):6375–6396.

Snyder, J. C., Stimpson, C. K., Thole, K. A., and Mongillo, D. (2015). Build Direction Effects on Additively Manufactured Channels. volume 5A, page V05AT11A034.

Song, B., Zhao, X., Li, S., Han, C., Wei, Q., Wen, S., Liu, J., and Shi, Y. (2015). Differences in microstructure and properties between selective laser melting and traditional manufacturing for fabrication of metal parts: A review. *Frontiers of Mechanical Engineering*, 10(2):111–125.

Stoyan, Y., Scheithauer, G., Gil, N., and Romanova, T. (2004). Phi -functions for complex 2D-objects. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 2(1):69–84.

Strano, G., Hao, L., Everson, R. M., and Evans, K. E. (2011). Multi-objective optimization of selective laser sintering processes for surface quality and energy saving. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 225(9):1673–1682.

Strano, G., Hao, L., Everson, R. M., and Evans, K. E. (2013a). A new approach to the design and optimisation of support structures in additive manufacturing. *International Journal of Advanced Manufacturing Technology*, 66(9-12):1247–1254.

Strano, G., Hao, L., Everson, R. M., and Evans, K. E. (2013b). Surface roughness analysis, modelling and prediction in selective laser melting. *Journal of Materials Processing Technology*, 213(4):589–597.

Stuart, N. (2015). Rolls-Royce breaks additive record with printed Trent-XWB bearing. *The Engineer*.

Szykman, S. and Cagan, J. (1997). Constrained Three-Dimensional Component Layout Using Simulated Annealing. *Journal of Mechanical Design*, 119:28.

Tang, M. and Pistorius, P. C. (2017). Oxides, porosity and fatigue performance of AlSi10Mg parts produced by selective laser melting. *International Journal of Fatigue*, 94:192–201.

Thomas, D. (2009). *The Development of Design Rules for Selective Laser Melting*. PhD thesis, Cardiff University.

Thomas, D. (2016). Costs, benefits, and adoption of additive manufacturing: a supply chain perspective. *International Journal of Advanced Manufacturing Technology*, 85(5-8):1857–1876.

Thompson, D. C. and Crawford, R. H. (1997). Computational Quality Measures for Evaluation of Part Orientation in Freeform Fabrication. *Journal of Manufacturing Systems*, 16(4):273–289.

Townsend, A., Senin, N., Blunt, L., Leach, R. K., and Taylor, J. S. (2016). Surface texture metrology for metal additive manufacturing: a review. *Precision Engineering*, 46:34–47.

Triantaphyllou, A., Giusca, C. L., Macaulay, G. D., Roerig, F., Hoebel, M., Leach, R. K., Tomita, B., and Milne, K. A. (2015). Surface texture measurement for additive manufacturing. *Surface Topography: Metrology and Properties*, 3(2).

Ulu, E., Korkmaz, E., Yay, K., Burak Ozdoganlar, O., and Burak Kara, L. (2015). Enhancing the Structural Performance of Additively Manufactured Objects Through Build Orientation Optimization. *Journal of Mechanical Design*, 137(11):111410.

Vandenbroucke, B. and Kruth, J.-P. (2007). Selective laser melting of biocompatible metals for rapid manufacturing of medical parts. *Rapid Prototyping Journal*, 13(4):196–203.

Vayre, B., Vignat, F., and Villeneuve, F. (2012). Metallic additive manufacturing: state-of-the-art review and prospects. *Mechanics & Industry*, 13(2):89–96.

Vora, P., Martinez, R., Hopkinson, N., Todd, I., and Mumtaz, K. (2017). Customised Alloy Blends for In-Situ Al339 Alloy Formation Using Anchorless Selective Laser Melting. *Technologies*, 5(2):24.

Vora, P., Mumtaz, K., Todd, I., and Hopkinson, N. (2015). AlSi12 in-situ alloy formation and residual stress reduction using anchorless selective laser melting. *Additive Manufacturing*, 7:12–19.

Vrancken, B., Buls, S., Kruth, J.-P., and Van Humbeeck, J. (2016). Preheating of Selective Laser Melted Ti6Al4V : Microstructure and Residual Stress. pages 2–6. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Vrancken, B., Thijs, L., Kruth, J. P., and Van Humbeeck, J. (2012). Heat treatment of Ti6Al4V produced by Selective Laser Melting: Microstructure and mechanical properties. *Journal of Alloys and Compounds*, 541:177–185.

Waterman, P. J. (2017). Optimize your Additive Manufacturing Know-How. Accessed May 25, 2018. *http://www.digitaleng.news/de/optimize-your-additive-manufacturing-know-how/*.

Watts, D. M. and Hague, R. J. M. (2006). Exploiting the Design Freedom of RM. *17th Annual International Solid Freeform Fabrication Symposium*, (2003):656–667.

Webb, P., Asif, S., Hogger, S., Kosche, T., and Kiernan, P. (2015). Advanced flexible automation cell control for aerospace manufacturing. *Aircraft Engineering and Aerospace Technology*, 87(2):156–164.

Whitwell, G. (2004). *Novel heuristic and metaheuristic approaches to cutting and packing*. PhD thesis, University of Nottingham.

Williams, S. W., Martina, F., Addison, A. C., Ding, J., Pardal, G., and Colegrove, P. (2016). Wire + Arc Additive Manufacturing. *Materials Science and Technology*, 32(7):641–647.

WJW Waterjet GmbH (2018). Metal laser melting. Accessed June 6, 2018. *https://www.wjw.de/production/lasermelting.html*.

Wodziak, J. R., Fadel, G. M., and Kirschman, C. (1994). A Genetic Algorithm for Optimizing Multiple Part Placement to Reduce Build Time. *Proceedings of the Fifth International Conference on Rapid Prototyping.*, pages 201,210.

Wohlers, T. (2014). Wohlers Report 2014: 3D Printing and Additive Manufacturing State of the Industry Annual Worldwide Progress Report. Technical report, Wohlers Associates, Fort Collins, Colorado.

Wu, H., Gao, S., Wang, R., and Ding, M. (2017). A global approach to multi-axis swept mesh generation. volume 203, pages 414–426.

Wu, S., Kay, M., King, R., Vila-Parrish, A., and Warsing, D. (2014). Multi-objective optimization of 3D packing problem in additive manufacturing. *IIE Annual Conference and Expo 2014*, 2:1485–1494.

Xu, F., Loh, H., and Wong, Y. (1999). Considerations and selection of optimal orientation for different rapid prototyping systems. *Rapid Prototyping Journal*, 5(2):54–60.

Yadroitsava, I., Grewar, S., Hattingh, D., and Yadroitsev, I. (2015). Residual Stress in SLM Ti6Al4V Alloy Specimens. *Materials Science Forum*, 828-829:305–310.

Yang, S. and Zhao, Y. F. (2015). Additive manufacturing-enabled design theory and methodology: a critical review. *The International Journal of Advanced Manufacturing Technology*.

Yang, Y., Wu, J., Sun, X., Wu, J., and Zheng, C. (2013). A niched Pareto tabu search for multi-objective optimal design of groundwater remediation systems. *Journal of Hydrology*, 490:56–73.

Yasa, E. and Kruth, J. (2011). Application of Laser Re-Melting on Selective Laser Melting Parts. *Advances in Production Engineering & Management*, 6:259–270.

Ye, K. Q. (1998). Orthogonal Column Latin Hypercubes and Their Application in Computer Experiments. *Journal of the American Statistical Association*, 93(444):1430–1439.

Zaeh, M. F. and Branner, G. (2010). Investigations on residual stresses and deformations in selective laser melting. *Production Engineering*, 4(1):35–45.

Zhang, W. and Zhou, L. (2018). Topology optimization of self-supporting structures with polygon features for additive manufacturing. *Computer Methods in Applied Mechanics and Engineering*, 334:56–78.

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

Zhang, Y. and Bernard, A. (2014). Grouping parts for multiple parts production in additive manufacturing. *Procedia CIRP*, 17:308–313.

Zhang, Y., Bernard, A., Harik, R., and Karunakaran, K. P. (2017). Build orientation optimization for multi-part production in additive manufacturing. *Journal of Intelligent Manufacturing*, 28(6):1393–1407.

Zhang, Y., Bernard, A., Valenzuela, J. M., and Karunakaran, K. P. (2015). Fast adaptive modeling method for build time estimation in Additive Manufacturing. *CIRP Journal of Manufacturing Science and Technology*, 10:49–60.

Zhang, Y., Gupta, R. K., and Bernard, A. (2016). Two-dimensional placement optimization for multi-parts production in additive manufacturing. *Robotics and Computer-Integrated Manufacturing*, 38:102–117.