To appear in the *International Journal of Production Research*
Vol. 00, No. 00, 00 Month 20XX, 1–21

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

Luiz J.P. Araújo[ab*] , Ajit Panesar[cd] , Ender Özcan[a] , Jason Atkin[a] , Martin Baumers[d] and Ian Ashcroft[d]

[a]*Automated Scheduling, Optimisation and Planning Group School of Computer Science, University of Nottingham, Nottingham NG8 1BB, UK;* [b]*Innopolis University, Innopolis 420500, Russia;* [c]*Department of Aeronautics, Imperial College London;* [d]*Centre for Additive Manufacturing, Faculty of Engineering, University of Nottingham, Nottingham NG7 2RD, UK*

(*v5.0 released June 2015*)

The adoption of Additive Manufacturing (AM) technology requires the efficient utilisation of the available build volumes to minimise production times and costs. Three-dimensional algorithms, particularly the Deepest Bottom-Left-Fill (DBLF) heuristic, have been extensively used to tackle the problem of packing arbitrary 3D geometries within the AM sector. A particularly common method applied to more realistic packing problems is the combination of DBLF and metaheuristics such as Genetic Algorithms (GAs). Through a series of experiments, this paper experimentally investigates the practical aspects, and comparative performance of different DBLF based methods including a brute force algorithm and GA combined with DBLF for AM build volume packing. The insights into the relationship between algorithm efficiency (in terms of volume utilisation), simulation runtime, and practical requirements, in particular geometry rotation constraints are investigated. In addition to providing an increased comprehension of the practical aspects of applying DBLF algorithms in the AM context, this study confirms the limitations of traditional DBLF and the requirements for more flexible and intelligent placement strategies while experimentally demonstrating that higher degrees of freedom for part rotation contribute to small improvements in volume density. The resulting additional computational effort discourages this strategy, however.

Keywords: build volume packing; benchmarking; additive manufacturing; 3D printing; deepest bottom-left-fill

1. Introduction

Additive Manufacturing (AM) technology, also known as 3D Printing, has emerged as a promising prospect for mainstream manufacturing (Petrovic et al. 2011; Conner et al. 2014). AM offers the opportunity to realise high-performance parts exhibiting complex geometries which may be more challenging or in some cases impossible to manufacture using conventional manufacturing processes. Over the recent years there has been significant attention on the utilisation of computational methods to progress different strands of AM research (Zhang, Yao, and Li 2019). Examples include: identifying optimal designs that incorporate support structure requirements (Langelaar 2016), efficient methods for AM designs (Wu, Clausen, and Sigmund 2017; Panesar et al. 2017b), generative design approaches for functionally graded cellular materials (Daynes et al. 2017; Panesar et al. 2017a), simulations to better understand and improve the processing conditions (Francois et al. 2017), novel frameworks that allow for the realisation of structures with embedded system componentry (Panesar et al. 2017a), and the efficient utilisation of the available build volume (i.e. packing), which is a relevant feature from a cost perspective (Baumers et al. 2017a).

*Corresponding author. Email: psxlja@exmail.nottingham.ac.uk

In particular, the utilisation of the available build volume will enable reduced per-unit production time and cost (Baumers et al. 2017a) which, according to models of technology diffusion, is likely to lead to increased technology adoption (Stoneman 2001). Such process modelling is aided by the layer-by-layer deposition process of currently available AM technology, allowing the precise estimation of material utilisation, energy consumption, build time and production costs (Baumers et al. 2013, 2017a). Among the available AM technology variants (Gibson, Rosen, and Stucker 2010) Laser Sintering (LS) serves as a good starting point for investigating the packing performance of algorithms in the context of real-world AM parts as it does not require support structures, avoiding the addition of extra process-imposed constraints, and allows fully three-dimensional build volume configurations.

Research from other communities can aid in addressing this AM problem. The Operations Research (OR) community study a number of combinatorial optimisation problems. The engineering problem of maximising the utilisation of a constrained n-dimensional space can be considered an instance of a problem type referred to as Cutting and Packing (C&P) problem in the OR literature. These problems occur in several other industrial applications such as shipping and packaging, where the aim is to maximise the utilisation of some limited space. Despite their seemingly straightforward definition, C&P problems are NP-Hard (Fowler, Paterson, and Tanimoto 1981). In practical terms, this means that the solution time to guarantee an optimal solution is likely to increase exponentially with the size of the input, making real-world problems intractable. This is especially true when more complex shapes are required to be arranged within the build volume. Wäscher, Haußner, and Schumann (2007) distinguish objects between *regular* (e.g. rectangles, boxes, cylinders, spheres) and *irregular* (or non-regular) with the latter being the more adequate to describe objects typically produced by AM processes. The optimisation problem under consideration is known as the three-dimensional irregular packing problem (3DIP).

C&P problems are often addressed by approximation methods that provide "good-enough" solutions in a reasonable time (Stoyan et al. 2004). Most of the approaches for solving 3DIP problems applied to the AM sector utilise variations of the bottom-left (BL) heuristic (Ikonen et al. 1997; Canellidis et al. 2006; Gogate and Pande 2008; Canellidis, Giannatsis, and Dedoussis 2009, 2013; Araújo et al. 2018). BL is a placement strategy firstly proposed by Art (1966) for solving two-dimensional packing problems and subsequently adapted to iteratively fill the spaces between the previously packed items, a variant commonly refereed to as bottom-left-fill (BLF). Finally, the latter method was extended to 3D in the form of the deepest bottom-left-fill (DBLF) heuristic (Karabulut and İnceoğlu 2004).

Existing solutions for 3DIP within usually integrate meta-heuristics to placement policies to find packing sequences that result in better volume utilisation. Examples of meta-heuristics used in the packing domain include tabu search (Lodi, Martello, and Vigo 2004; Gendreau et al. 2008; Crainic, Perboli, and Tadei 2009), guided local search (Voudouris, Tsang, and Alsheddy 2010) and algorithms (GA), which have been the predominant approach for three-dimensional problems (Ikonen et al. 1997). Concerning GA, however, there has been no comprehensive investigation on the effects that different degrees of freedom for rotation allowance have on the packing performance from different DBLF methods, which can be assessed by observing the resulting algorithm runtime and volume density.

This study focuses on the use of the DBLF heuristic with GA as this has been the most extensively approach for packing problems applied to AM (Ikonen et al. 1997; Gogate and Pande 2008; Canellidis et al. 2006; Baumers et al. 2013). This paper investigates the practicality of the DBLF algorithm by (i) discussing the main DBLF based approaches, their strengths and weaknesses within the remit of AM; (ii) presenting a study of the trade-offs in performance, in terms of computational runtime and packing efficiency (z-height), and (iii) experimentally demonstrating the effects that variations in constraints and input parameters for DBLF have on packing performance.

This paper is structured as follows. Firstly, the terminology and the main DBLF based packing approaches reported in the literature are reviewed including a brief discussion of the common workflow structure shared by these algorithms. Secondly, the methodology for implementing the considered three DBLF algorithms as well as the specifications for utilised instances and the design of a series of experiments are presented. Subsequently, the results and findings from the experiments are discussed together with some insights for AM practitioners. This study concludes by presenting some recommendations for the utilisation of DBLF methods and future research efforts in this field.

2. Approaches: terminology, data structures and workflow

As outlined in the introduction, the build volume packing problem identified in the AM-related engineering literature is viewed as a Cutting and Packing (C&P) problem in the field of of OR and the terminologies used in both fields differ. To facilitate the understanding of research across both areas, the following subsections introduce the key elements of this problem based on the usual terminology, and describe the DBLF algorithm in detail.

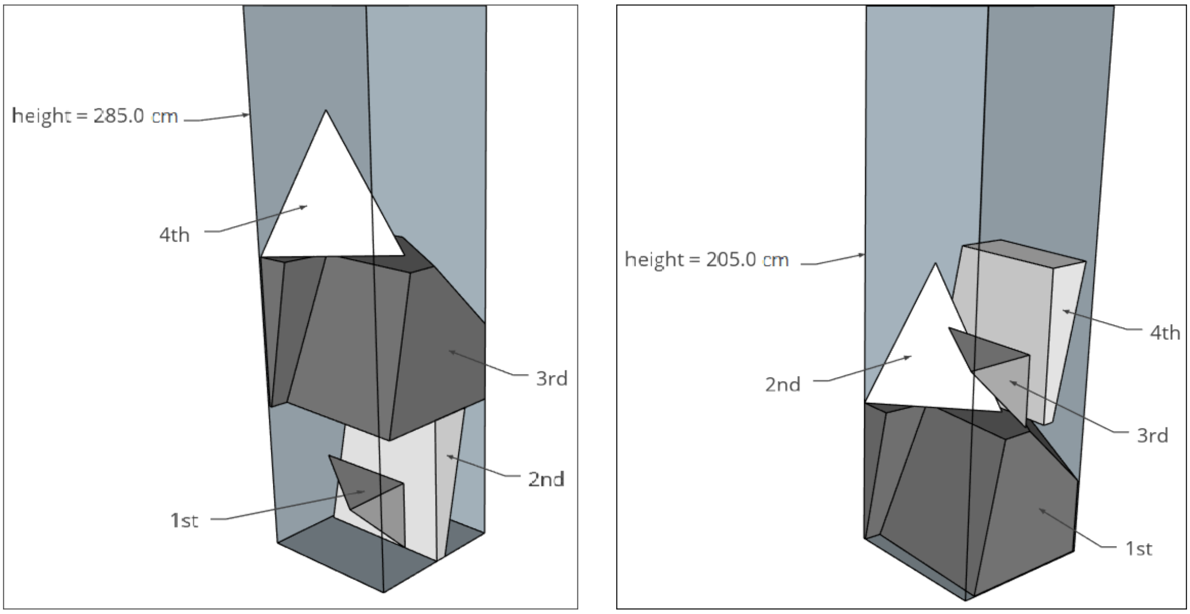
2.1 *Cutting and Packing Problems in Operational Research*

C&P problems are multidisciplinary problems that have appeared in the OR literature since the 1960s (Art 1966) in several forms, with varying objectives and constraints. Surveys and typologies for classifying C&P problems have been presented in the past by Dyckhoff (1990), Wäscher, Haußner, and Schumann (2007), and Araújo et al. (2018), with each new version attempting to address issues identified in the prior art. While a variety of definitions of the common OR terminology have been suggested in the literature, this paper will use the nomenclature introduced by Araújo et al. (2018) in conjunction with an adapted terminology familiar to the AM community.

According to Dyckhoff (1990, page 148), C&P problems "in the narrow sense are characterised by large objects defined as the empty useful space of ... containers and bins". In this study, the term 'container' is utilised to refer to the cuboidal volume within AM systems in which part geometry is built up. Dyckhoff (1990) also stated that "the principal aspect of container loading concerns the geometric combination of small items to packing patterns which can be assigned to containers". Applied to the AM context, the term 'small items' reflects the digital 3D geometries provided in a computer-aided design (CAD) format for manufacturing. Here, the STereoLithography (STL) file format is an industry standard employed to represent parts in a modelling environment. Further, Araújo et al. (2018) distinguished between the physical manufactured product as 'part' (or item) and its data structure representation as 'model' (also 'shape').

The maximisation of the build volume utilisation is an objective that can arise in different forms in a layer manufacturing process (Araújo et al. 2018). For example, Baumers et al. (2013) demonstrated that the build volume height is a variable that affects the estimated production cost and process energy consumption. Such optimisation problems are is known in the OR literature as the Three-dimensional Strip Packing (3DSP) problems (Wäscher, Haußner, and Schumann 2007). In other cases, C&P problems aim to minimise of the number of containers to accommodate the a range of demanded items, which can be translated as the minimum number of build operations in AM. This problem is commonly referred to and the Three-dimensional Bin Packing (3DBP) problem (Wäscher, Haußner, and Schumann 2007).

It should also be noted that C&P problems have in common with other combinatorial optimisation problems that they are characterised by a large search spaces, disallowing exhaustive investigation of each candidate solution (Jakobs 1996; Falkenauer and Delchambre 1992). Hence, most of the solution methods for packing problems are heuristic in nature. Therefore, this study includes an empirical analysis of the most common packing approaches that have been applied to AM and aims to identify the strengths and weaknesses of each, as well to provide recommendations for future implementations of such methods.



(a) Resulting build volume height: 285.0 cm (b) Resulting build volume height: 205.0 cm

Figure 1. Application of the DBLF heuristic to a group of parts considering two different orders.

2.2 *Bottom-Left and Deepest Bottom-Left-Fill algorithms*

The intuition behind the bottom-left (BL) heuristic is that each of the two-dimensional objects should be arranged onto as near as possible to the bottom-left corner of the available two-dimensional space (Art 1966). Items are considered one at a time, and the order of consideration can often affect the efficiency of the algorithm. Each item is then placed as low as possible ('bottom') in the container, and as far to the left as possible on that lowest layer ('left').

The Deepest Bottom-Left-Fill (DBLF) algorithm is an extension of the BL heuristic to 3D, which considers the deepest positions in the container, packing the item as close to the bottom and left (in that order) on the deepest level as possible (Art 1966). The performance of such methods, in terms of the volume density achieved, depends on a number of factors including: the number of parts, the sequence they are processed in and the allowable part orientations (Bennell and Oliveira 2009). Figure 1 illustrates how different build volume heights, and hence volume densities, result from changing the sequence of parts placed by a DBLF algorithm.

There are two common research directions pursued by the OR community to improve the efficiency of BL and DBLF methods. The first direction aims to improve the placement strategy by minimising the gaps that appear during the packing process (Albano and Sapuppo 1980; Błażewicz, Hawryluk, and Walkowiak 1993). Methods like the no-fit polygon approach (Art 1966) often involve advanced geometric and topologic calculations for detecting and resolving overlap between the geometries. Simpler alternatives, on the other hand, tackle this by considering small perturbations in position and orientations for the parts (Knight, Jaeger, and Nagel 1993; Möbius et al. 2001). The second direction is related to the choice of the sequence in which shapes are considered, or the criteria for sorting the parts. For example, Art (1966) sorted the parts in increasing order of widths due to the characteristics of the addressed problem, which required the minimisation of the width of the utilised area. Other sorting criteria include, for example, decreasing order of a part feature (e.g. height, volume) as implemented by the so-called 'Bottom-Left Decreasing' (BLD) algorithms (Coffman Jr et al. 1980; Lesh et al. 2005).

When applied to AM, BL based techniques present a similar top-level structure of tasks, which

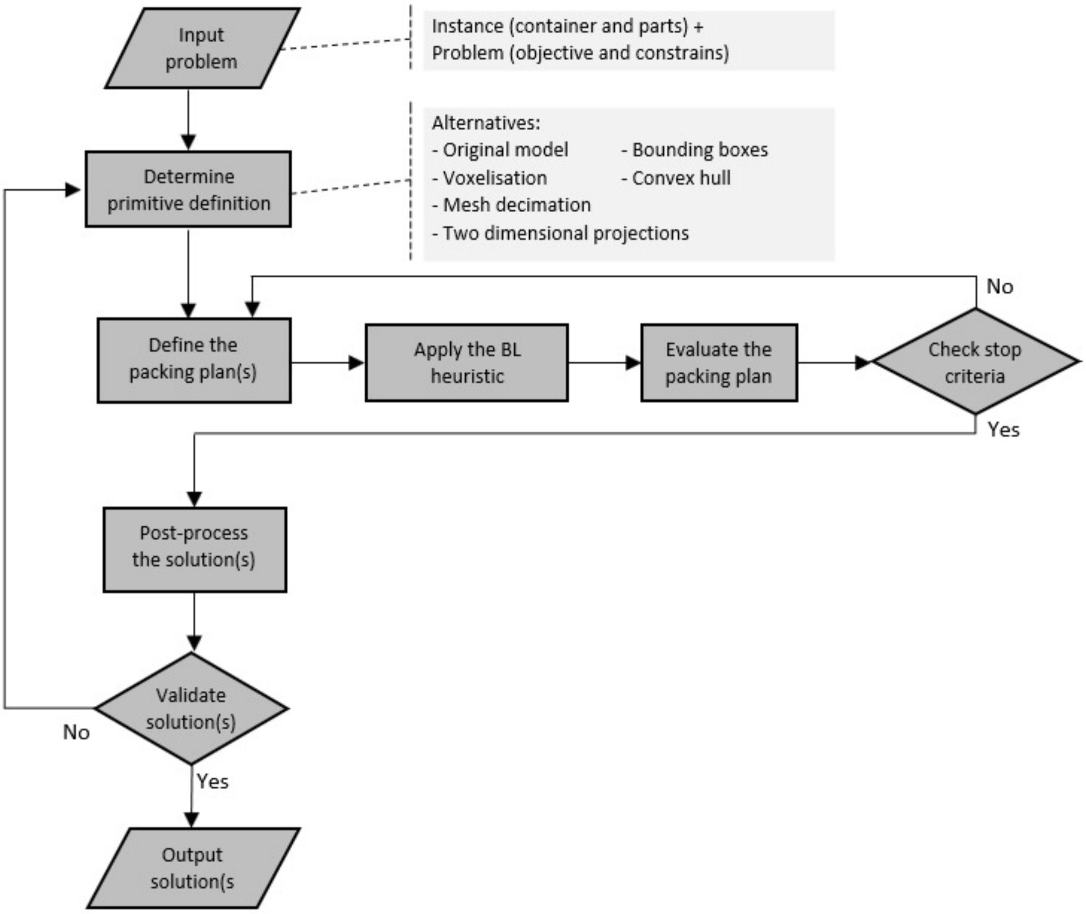
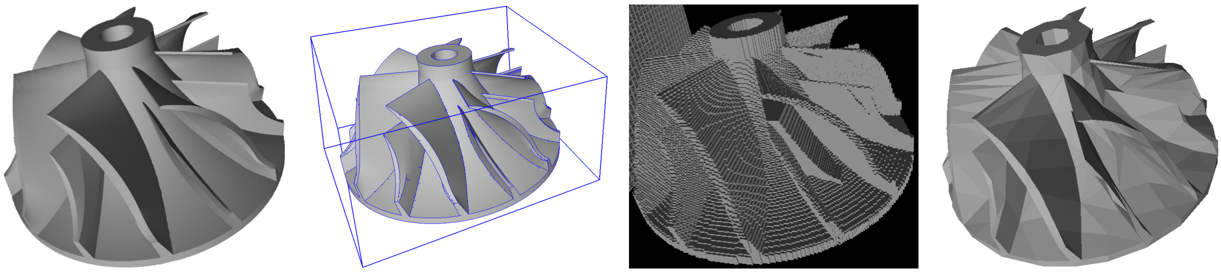


Figure 2. Workflow of practical BL and DBLF based packing algorithm.

can be illustrated by the workflow in Figure 2 and is discussed in detail below.

An important aspect of BL approaches, including DBLF, when addressing arbitrary geometries is the adopted boundary representation (i.e. how the solids are represented) (Cagan, Shimada, and Yin 2002; Stroud 2006). In the original implementation of the BL procedure (Art 1966), for example, non-convex 2D polygons were simplified to convex envelopes to allow more efficient execution of the algorithm. The majority of existing 3DIP approaches employs either the original STL model as presented in Figure 3(a) (Stoyan et al. 2004; Gogate and Pande 2008); or their respective orthogonal bounding boxes (Canellidis et al. 2006; Canellidis, Giannatsis, and Dedoussis 2013) as illustrated in Figure 3(b). Although the latter representation requires lower computational effort, it is more likely to result in inefficient packing due to underutilised spaces particularly when considering non-convex geometries. More sophisticated alternatives for primitive definition include 2D horizontal projections (Canellidis et al. 2006; Canellidis, Giannatsis, and Dedoussis 2013), voxel discretisation (Min 2004) as shown in Figure 3(c). This reduces to the minimum convex hull (Jia et al. 2007) and mesh decimation (Garland and Heckbert 1997), as presented in Figure 3(d). A simplified representation for complex geometries is desirable as it reduces the computational effort when processing several instances or a large number of parts (Cagan, Degentesh, and Yin 1998).

It can also be observed that the general algorithm shown in Figure 2 requires the selection of one or more packing plans before proceeding with the BL or DBLF heuristic. Most of DBLF approaches for 3DIP integrates a search algorithm to asses the large search space of possible sequences and orientations for the parts. In the recent decades, however, several authors have systematically utilised metaheuristics to find promising packing patterns for such problems, mainly



(a) Original model (b) Bounding box (c) Voxelised model (d) Decimated model

Figure 3. Possible primitive definitions for a given part.

in AM-targeted applications. Metaheuristics are combinatorial optimisation techniques which are not restricted to a particular problem domain but are rather designed to be amenable to different complex combinatorial problems and provide good enough solutions in reasonable time (Hertz and Widmer 2003). Examples of metaheuristics seen in the AM literature include Simulated Annealing (Cagan, Shimada, and Yin 2002) and Genetic Algorithms (Ikonen et al. 1997; Hur et al. 2001; Gogate and Pande 2008; Canellidis, Giannatsis, and Dedoussis 2013).

Returning to the workflow shown in Figure 2, the general BL method validates whether the solution satisfies the termination criteria or a new sequence is selected to be processed by the packing heuristic. The termination criteria for the described loop is often determined by the top layer search algorithm (e.g. maximum number of evaluations or total runtime).

BL and DBLF algorithms have been the most widely used computational tools to address practical packing problems in the context of AM applications (Araújo et al. 2018). The following sections focus, therefore, on the strengths and weaknesses of three of the most common approaches for such algorithms: brute force, DBLF Decreasing and the integration of GAs. The objective is to generate experimental evidence that can support better decision making regarding the choice of packing algorithm and parameters in future applications.

3. Introducing DBLF approaches: brute force search, DBLF Decreasing and Genetic Algorithm with DBLF

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

3.1 *Brute force search*

Brute force algorithms exhaustively search through all possible candidates to find globally optimum solutions. Therefore, their algorithmic computational cost can be gauged by assessing the size of the search space (Schaeffer et al. 1993). In the context of DBLF algorithms, the cardinality of the search space is the number of possible packing plans for the given input and depends on the number of parts and allowable orientations per part. Consider, for instance, the problem of packing a set of 'n' parts where part orientations arising from angle increments of 'θ' are allowed. The attainable 'b' angular states about the x, y and z-axes that give rise to a part orientation are captured by function 1. For example, 90° increments result in 4 possible angular states $(0°, 90°, 180°, 270°)$.

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

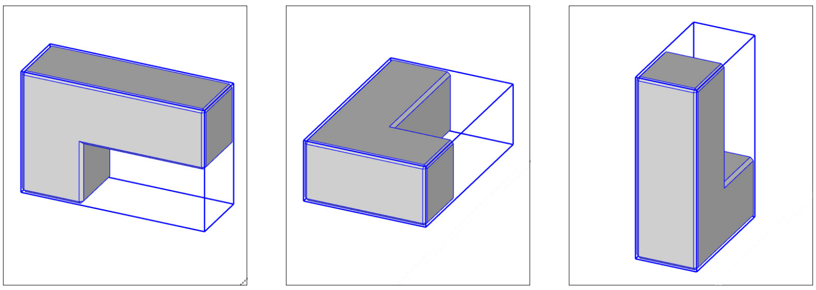
The cardinality of the solution space can, therefore, be estimated by Equation 2, which reflects the runtime of the brute force algorithm based on n and b.

$$f(n, b) = n!\ b^{3n} \tag{2}$$

The brute-force complete search method yields the best packing sequence with the minimum volume height after investigating each candidate in the search space. For inputs of size n and a constant b, it requires $O(n!\ b^n)$ time, which is not practical even for minimal values of b. Hence, solving 3DIP problems using brute-force search is not and amenable strategy for problems containing a large number of demanded parts as in most of the real-world instances. Nevertheless, the inclusion of this approach in the conducted experiment aims to demonstrate some of the limitations of search algorithms integrated into the DBLF heuristic.

3.2 *Deepest Bottom-Left-Fill Decreasing*

Deepest bottom-left-fill decreasing (DBLFD) is a strategy to manage the large search space of packing plans by sorting the parts in decreasing order of a particular numeric feature. It extends the standard DBLF by adding a preliminary task in which parts are sorted in decreasing order of volume; the sequence of parts is then processed using DBLF. Another particularity of the implemented approach is that three different orientations per part, which are shown in Figure 4, are tested while the standard DBLF uses only the native (i.e. original) orientation. This mechanism mitigates the effect of poor native orientations achieved during the part design and explores a more extensive area of the search space (see equation 2).



(a) Original model and first possible orientation

(b) Second possible orientation for the given part

(c) Third possible orientation for the part

Figure 4. Three orientations per part, as in the implemented DBLF.

3.3 *A Genetic Algorithm combined with DBLF*

Ikonen et al. (1998) used the acronym GARP to refer to methods that integrate genetic algorithms (GAs) with the DBLF heuristic for rapid prototyping, which has been one of the dominant strategies for applying 3DIP in AM (Ikonen et al. 1996, 1997, 1998; Hur et al. 2001; Canellidis et al.

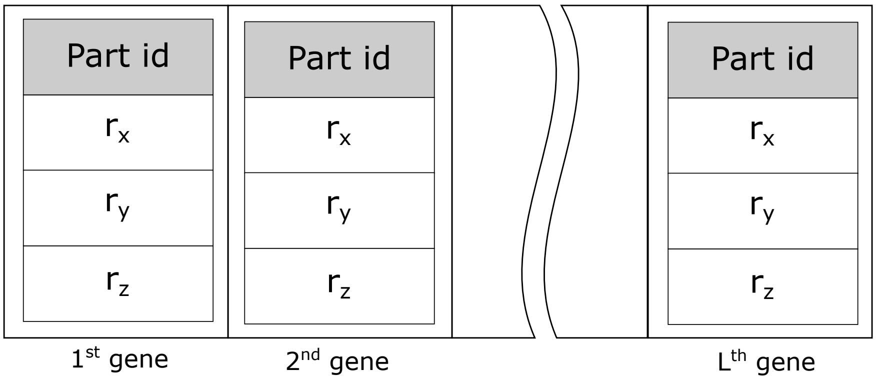


Figure 5. Chromosome in the current GA implementation.

2006; Canellidis, Giannatsis, and Dedoussis 2009, 2013; Araújo et al. 2015). GAs are evolutionary algorithms (population-based metaheuristics) that mimic the process of evolving a population throughout the selection and combination of the fittest individuals (Holland 1992).

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

Parameter tuning is highly recommended when applying metaheuristics to any problem domain, although this has not been done for 3D packing studies in AM (Grefenstette 1986). Approaches using GA within this context have explored only a restricted range of parameter values, which have been selected in an arbitrary manner or using limited rationalisation (see Table 6). The GA components that have been used in such studies, which are also employed in the experiments shown in section 4, are explained below.

Chromosome: A chromosome is an array of genes with a length (L) that is equal to the number of parts (n). Each gene is a 4-tuple comprised of the id of the part and the rotations around the x, y and z-axis (r_x, r_y and r_z respectively), as illustrated in Figure 5.

Crossover: Crossover is responsible for transmitting genetic characteristics from good individuals (parents) to the next generation (Holland 1992). In GARP, order-1 crossover is often used since this operator is suitable for combining individuals that are represented as non-binary chromosomes and is commonly used for permutations (Poon and Carter 1995). An aspect to consider is the probability that crossover is applied and the strategy for selecting parents. Based on the parameters used in reported GARP solutions and suggestions in the literature, the following values are used for crossover probability (CP): 0.5, 0.75 and 1.0 (Srinivas and Patnaik 1994).

Selection: Individuals with higher fitness values have higher probability of being select to generate new individuals. Three parent selection schemes appear in GARP solutions and are used in section 4 (Goldberg and Deb 1991):

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

- Tournament: selects the best individual from a randomly selected group of individuals, typically comprised of 2 or 4 elements

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

- Insert (Sivanandam and Deepa 2007): the gene is moved to a random position in the chromosome
- Creep (Sivanandam and Deepa 2007): different values are assigned to the r_x, r_y and r_z of the gene
- Insert and creep: the gene values for r_x, r_y and r_z are changed, and then the gene is randomly moved to a different position in the chromosome. This scheme is used in the experiments shown in section 4.

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

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

$$f = \frac{H^*}{H + PEN} \tag{3}$$

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

Replacement strategy: This is the strategy to replace individuals of the current population by the offspring. Two replacement strategies are used the experiments presented in section 4:

- Generational replacement with elitism: (i) copies the best two individuals of the current population to the pool of individuals which will constitute the population of the next generation, (ii) successively combines individuals, and (iii) copying the offspring to next generation until its population is complete
- Steady state: instead of generating a new population, the best two individuals among the two selected parents and the offspring are copied back to the population

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

- the best individual is above the acceptable threshold, i.e., its fitness is within the interval $[0.99, 1]$ (Safe et al. 2004)
- the algorithm ends if there is no improvement of the best individual after 1,400 (calculated after preliminary tests) consecutive evaluations

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

Parameter sets: A parameter set is comprised of the definition of the population size, CP, the parent selection scheme, MP and the replacement strategy. Table 1 summarises the parameters and values that have been tested in the existing GARP solutions and that are employed in the experiments shown in section 4.

Table 1. Parameter settings of the GA implemented in this study.

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

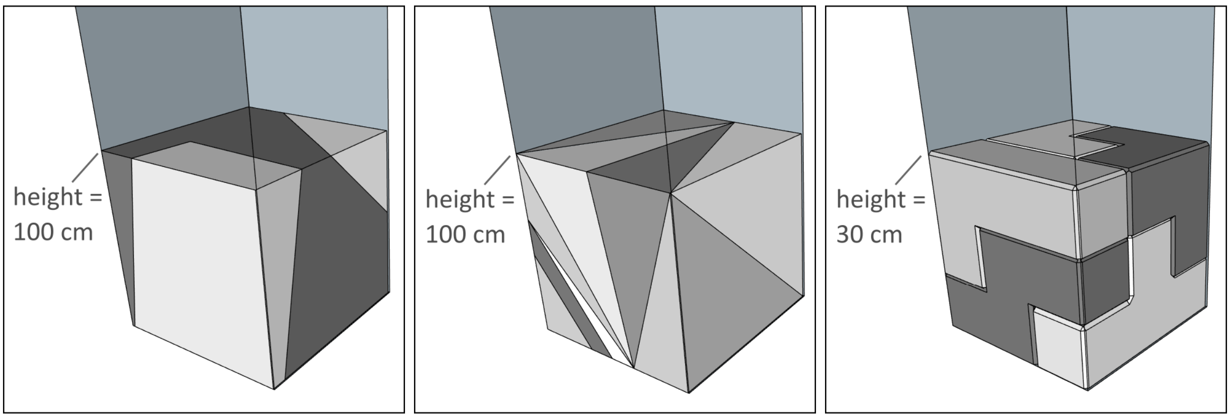
[a] L: Length of chromosome; [b] Tournament of 2 individuals; [c] Tournament of 4 individuals

4. Experimental design

This section follows on from the description for three DBLF approaches in Section 3, and focuses on a series of experiments to compare their algorithmic performances. Also presented here are the benchmark problem instances and their properties, evaluation methods and targeted observations.

4.1 *Problem instances used in the experiments*

Three problem instances were generated to investigate the packing performance of DBLF-based approaches. The first two instances, Cutcube1 with four parts and Cutcube2 with 11 parts, are generated by slicing a three-dimensional cube using bounded regions of non-orthogonal planes. The third instance is a Somacube (Peter-Orth 1985) comprised of seven polycube parts. Figure 6 illustrates the problem instances arranged in their best packing solution, which is known.



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

Figure 6. The number of parts and optimal packing configuration for the three instances.

The instances[1] shown in Figure 6 are comprised of relatively few parts: 4, 11 and 7 for the Cutcube1, Cutcube2 and Somacube, respectively. The parts considered here pose a greater challenge from a packing perspective than the bounding box approximations. While they do not represent AM parts designed for industrial applications, they approximate the levels of complexity found in reality. Using instances with intermediate levels of complexity and known optimal packing arrangements is useful for comparing the effectiveness of DBLF-based approaches for packing AM parts in reasonable runtime. Underlying this approach, of course, is the acknowledgement that obtaining the optimal packing for a random set of AM parts, as pursued in some studies, is an unfathomable task. Therefore, having known optimal packing configurations is a distinct advantage for benchmarking (Szykman and Cagan 1995). The experiments for the present study also use the original STL models for the parts to prevent the loss of information that would result from using alternative primitive representations.

4.2 *Experiments*

In this section, we describe our experiments comparing the three DBLF approaches shown in section 3 and the effects that different algorithm parameters have on performance regarding volume utilisation and runtime.

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

Parameter tuning for the GA approach. Studies presenting GARP techniques often fail to analyse how different values for GA parameters affect packing performance. Moreover, limited rationalisation is given regarding the choice of these parameters, which is unfortunate since parameter tuning is an essential task while applying heuristics to any problem domain (Smit and Eiben 2009). In this experiment, 96 parameter sets were generated from combinations of the GA parameters shown in Table 1. For each parameter set, GA was executed 35 times to ascertain the average runtime and obtained build volume height when solving the Cutcube2 instance with angle increment of 90°. As a result, the parameter set that maximises the average fitness withing reasonable runtime is selected for further testing.

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

5. Results and Discussion

5.1 *Testing for brute force search*

As explained previously, 50 packing plans were randomly generated for each instance-θ, except in the case of Cutcube1-0 as its search space is comprised of 24 elements. Table 2 shows the mean runtime per instance and part, as well as the number in non-convex parts in each instance.

The results confirm the intuition that the runtime per instance depends on the quantity and complexity of the parts processed. The runtime to solve Cutcube1 is lower than Cutcube2 due to a lower number of parts. On the other hand, the mean runtime per part of Cutcube1 is higher, mainly because of the presence of one non-convex part. The Somacube instance, which has the most

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

Table 2. Estimated runtime for the brute force per instance-angle increment.

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

non-convex parts, had the highest mean runtime for both instance and parts. The mean runtime per instance and equation 2 enables an estimation of the time necessary to investigate each element in the search space.

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

5.2 *Parameter tuning for GA-based approach*

The mean runtime and build volume height are used to compare the 96 different sets of parameter configurations generated from combinations of GA parameter values as shown in Table 1. First, each attribute was analysed separately with respect to its effects on the performance (see Table 3).

Table 3. Comparing the mean and standard deviation of GA parameters separately.

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

It can be observed from Table 3 that GA parameters do not strongly affect the mean runtime and mean build volume height, except for CP. Increasing the value of CP from 0.5 to 1.0 results in a noticeable improvement ($> 10\%$) in build volume z-height at a modest increase in computational

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

cost. It is of note that generational GA with elitism produced marginally better results compared to steady state implementation. Figure 7 shows that build volume height decreased as the time spent on the search increased, which was expected. The first cluster of parameter sets with mean times of less than 65 seconds is entirely comprised of configurations with CP equal to 0.5, while the group of parameters between 65 and 105 seconds is predominantly comprised of configurations with CP equal to 0.75 (33 out of 42 parameter sets). The graph stresses the two parameter sets that resulted in highest mean fitness: 35 and 47. Table 4 presents the attributes and the mean fitness obtained by the parameter sets 35, 47 and 33, which is formed by the best individual parameters extracted from Table 3.

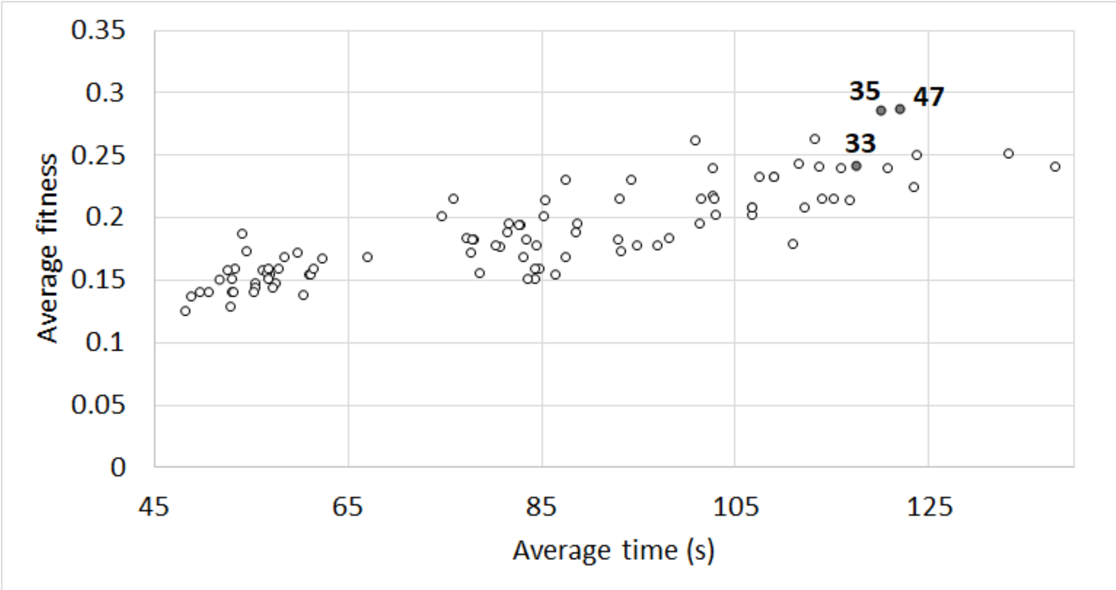


Figure 7. Scatter graph summarising the mean fitness and runtime for each parameter set.

Table 4. Comparing three selected parameter sets (33, 35 and 47).

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

The three parameter configurations (sets) shown in Table 4 share the same values for the probability of crossover, size of population and replacement strategy. The parameter configuration identified by the ID 35 was chosen for further experiments, since the tournament-2 selection method yielded better mean results than the tournament-4 method (see Table 3).

5.3 *Comparing DBLFD and GA based packing methods*

The mean fitness and mean runtime are reported in Figure 8, together with the result from DBLFD. The data obtained from this experiment reinforces the need to adopt simpler primitive representations when aiming to minimise computational effort.

Although the Somacube instance is comprised of fewer parts than Cutcube2 (7 and 11, respectively), the higher complexity, i.e., non-convexity of parts, resulted in considerable higher runtime. While the mean runtimes for all Cutcube1-θ and Cutcube2-θ are less than 140 seconds, all the Somacube-θ configurations have mean runtime over 2,000 seconds. Interestingly, for most of the

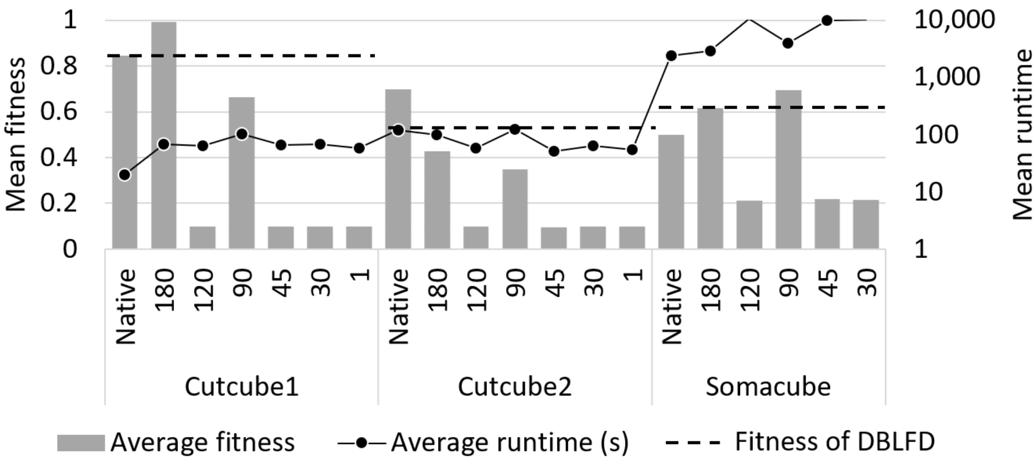


Figure 8. Mean fitness and runtime obtained for each instance-θ.

instance-*theta* pairs (except for Cutcube2-0, Somacube-180 and Somacube-90), the DBLFD heuristic produced more competitive results than the GA approach. This finding suggests that the former method is a viable alternative, as it simpler to implement and has a more reasonable runtime than the popular GA approach with angle increments less than 90°. Therefore, future research on packing algorithms using such a strategy should integrate efficient search algorithms into DBLF with more degree of freedom for part rotation; otherwise, a simple DBLFD could be employed.

The effects of part orientation on mean fitness and the importance of 'good' initial orientation of the parts can also be observed in Figure 8. For configurations with non-orthogonal rotation ($\theta = 120°$ or $\theta < 90°$), initial populations were comprised of several packing plans that resulted in non-packable parts. This was due to horizontal projection of the parts exceeding the bounds of the container. This observation indicates that local search methods should be run before proceeding with the packing process to ensure that each part fits within the container (Canellidis et al. 2006; Canellidis, Giannatsis, and Dedoussis 2009, 2013).

5.4 *Discussion on the overall results and packing approaches for real-world problems*

This study provides evidence that shape complexity affects computational runtime to a greater extent than the number of parts. As shown in the first experiment, instances with a smaller number of parts exhibiting higher degrees of non-convexity require greater computational effort than those with higher quantity requirements but simple geometries. For example, the mean runtime for solving a Somacube instance is greater than in Cutcube2, despite its fewer number of parts (see Table 2). This implies that high-resolution voxelised representations (Min 2004), which bear a resemblance the non-convex polycubes contained in the Somacube instance, are likely to result in higher runtimes compared to convex hull envelopes. Therefore, the results underline the importance of the adoption of simple boundary representations in the early stages of packing solution development. This is of special significance in AM, since the adoption of AM is often justified by the necessity to manage highly complex or non-convex product geometries (Baumers et al. 2017b).

Another observation that can be made from these experiments concerns the limitations of DBLF in obtaining configurations with no waste of space between objects (when such configurations exist) due to the fixed order in which geometric operations (translation and rotation) are performed. For example, the use of brute force search for solving Cutcube1-0 fails in achieving a 'no-waste' configuration despite the small number of parts, which have been designed to be perfectly assembled. As shown in Table 5, the best build volume height achieved by DBLF (depicted in Figure 9) was

Table 5. Results of the 24 packing plans of Cutcube1 with the native orientation of parts.

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

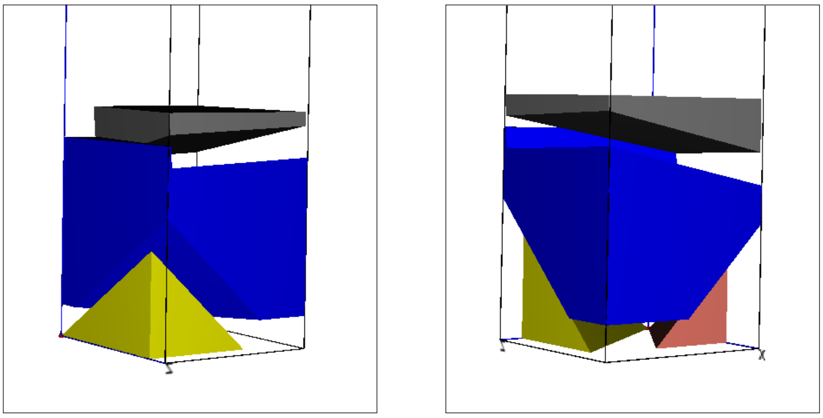


Figure 9. The best out of 24 possible packing plans of Cutcube1-0 with no rotation allowed.

118.32mm. This demonstrates a need for placement heuristics that are more flexible regarding the identification, translation and rotation of the processing part. Advantageous strategies would be to temporarily allow infeasible states and to adopt more efficient methods of detecting eventual geometry overlaps.

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

- Simplifying part representations can lead to quicker determination of satisfactory build configurations. This is essential in some practical situations, such as when cost estimates or price quotations are required instantaneously.
- Similarly to other combinatorial optimisation problems, parameter tuning can heavily influence the entire functioning and performance regarding runtime and volume utilisation of packing approaches. However, most of GARP techniques in the literature omit this critical preprocessing stage and use arbitrary values for parameters and rotation. The incorporation of such task into the AM workflow can, therefore, improve the packing outcome and thereby reduce manufacturing costs in AM (Ruffo and Hague 2007; Baumers et al. 2017a).
- As shown in the experiment comparing GA to DBLFD, higher degrees of freedom for part rotation does not necessarily result in a gain of performance concerning volume density for packing algorithms. Instead, it occurs in additional computational effort due to the exponential increase of the search space. The use of orthogonal rotation, therefore, seems a reasonable approach for this domain.
- Recent research has shown that the problem of build volume packing cannot be divorced from the problem of machine scheduling in practice (Li, Kucukkoc, and Zhang 2017; Khajavi et al. 2018). As evident from the literature on control systems for flexible manufacturing systems

(Pannequin, Morel, and Thomas 2009), dealing with problems in a decomposed way leads to significant performance loss and results in high opportunity costs. Therefore, integrated heuristic approaches such as GA only pose a partial solution of the AM workflow problem faced in reality.

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

6. Conclusion

With the increased interest in AM to create end-use parts, the efficient utilisation of machines is an imminent challenge. This work provides valuable insight into the shortcomings, challenges and practical aspects of computational solutions when addressing realistic packing problems in AM. This work differs from previous studies on 3DIP algorithms in the sense it contains a rationalisation regarding the configuration for the DBLF heuristic and the parameter tuning for the GA based packing method, an approach which has been the cornerstone of packing approaches in AM.

It is concluded that the significance of implementing a simpler (enveloping shape) primitive representation far out-weighs algorithmic choice when aiming to achieve good-enough packing in a reasonable runtime. This study also exposes a core weakness of the DBLF heuristic, i.e. being unable to obtain an optimal packing when dealing with non-convex parts, indicating the need for a more flexible placement heuristic.

Further work is required to establish the relationship between part complexity and predicted runtime as an important question for real time packing tools required by the AM community. The comparison of DBLF based algorithms with other placement policies and meta-heuristics are timely as they can expose the opportunities for improving the runtime performance of commercial applications.

Acknowledgement

This work was supported by the Engineering and Physical Sciences Research Council [grant number EP/N010280/1]. The authors would also like to thank CNPq (Brazilian Council for Research and Development), process 248602/2013-6 for sponsoring Luiz J.P. Araújo.

References

Albano, Antonio, and Giuseppe Sapuppo. 1980. “Optimal allocation of two-dimensional irregular shapes using heuristic search methods.” *Systems, Man and Cybernetics, IEEE Transactions on* 10 (5): 242–248.

Araújo, Luiz Jonata Pires de, Ender Özcan, J A D Jason Atkin, Martin Baumers, Christopher Tuck, and Richard J M Hague. 2015. “Toward better build volume packing in additive manufacturing: classification of existing problems and benchmarks.” *Proceedings of the Solid Freeform Fabrication Symposium* 401–410.

Araújo, Luiz Jonatã Pires de, Ender Özcan, Jason A. D. Atkin, and Martin Baumers. 2018. “Analysis of irregular three-dimensional packing problems in additive manufacturing: a new taxonomy and dataset.” *International Journal of Production Research* 0 (0): 1–15.

Art, Richard Carl. 1966. “An approach to the two dimensional irregular cutting stock problem.” PhD diss., Massachusetts Institute of Technology.

Baumers, Martin, Luca Beltrametti, Angelo Gasparre, and Richard Hague. 2017a. "Informing additive manufacturing technology adoption: total cost and the impact of capacity utilisation." *International Journal of Production Research* 55 (23): 6957–6970.

Baumers, Martin, Chris Tuck, Ricky Wildman, Ian Ashcroft, and Richard Hague. 2017b. "Shape complexity and process energy consumption in electron beam melting: A case of something for nothing in additive manufacturing?" *Journal of industrial Ecology* 21 (S1): S157–S167.

Baumers, Martin, Chris Tuck, Ricky Wildman, Ian Ashcroft, Emma Rosamond, and Richard Hague. 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.

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

Błażewicz, J, P Hawryluk, and Rafal Walkowiak. 1993. "Using a tabu search approach for solving the two-dimensional irregular cutting problem." *Annals of Operations Research* 41 (4): 313–325.

Cagan, Jonathan, Drew Degentesh, and Su Yin. 1998. "A simulated annealing-based algorithm using hierarchical models for general three-dimensional component layout." *Computer-aided design* 30 (10): 781–790.

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

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

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

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

Coffman Jr, Edward G, Michael R Garey, David S Johnson, and Robert Endre Tarjan. 1980. "Performance bounds for level-oriented two-dimensional packing algorithms." *SIAM Journal on Computing* 9 (4): 808–826.

Conner, Brett P., Guha P. Manogharan, Ashley N. Martof, Lauren M. Rodomsky, Caitlyn M. Rodomsky, Dakesha C. Jordan, and James W. Limperos. 2014. "Making sense of 3-D printing: Creating a map of additive manufacturing products and services." *Additive Manufacturing* 1: 64–76.

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

Daynes, Stephen, Stefanie Feih, Wen Feng Lu, and Jun Wei. 2017. "Optimisation of functionally graded lattice structures using isostatic lines." *Materials and Design* 127 (February): 215–223.

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

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

Falkenauer, Emanuel, and Alain Delchambre. 1992. "A genetic algorithm for bin packing and line balancing." In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, 1186–1192. IEEE.

Fowler, Robert J., Michael S. Paterson, and Steven L. Tanimoto. 1981. "Optimal packing and covering in the plane are NP-complete." *Information processing letters* 12 (3): 133–137.

Francois, M. M., A. Sun, W. E. King, N. J. Henson, D. Tourret, C. A. Bronkhorst, N. N. Carlson, et al. 2017. "Modeling of additive manufacturing processes for metals: Challenges and opportunities." *Current Opinion in Solid State and Materials Science* 21 (4): 198–206.

Garland, Michael, and Paul S. Heckbert. 1997. "Surface simplification using quadric error metrics." In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 209–216. ACM Press/Addison-Wesley Publishing Co.

Gendreau, Michel, Manuel Iori, Gilbert Laporte, and Silvaro Martello. 2008. "A Tabu Search heuristic for the vehicle routing problem with two-dimensional loading constraints." *Networks* 51 (1): 4–18.

Gibson, I, D Rosen, and B Stucker. 2010. "Additive manufacturing technologies, 3D printing, rapid prototyping, and direct digital manufacturing, Springer." *New York Heidelberg Dordrecht London* .

Gogate, A. S., and S. S. Pande. 2008. "Intelligent layout planning for rapid prototyping." *International*

Journal of Production Research 46 (20): 5607–5631.
Goldberg, David E, and Kalyanmoy Deb. 1991. "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms." *Foundations of Genetic Algorithms* 1: 69–93.
Grefenstette, John J. 1986. "Optimization of control parameters for genetic algorithms." *IEEE Transactions on systems, man, and cybernetics* 16 (1): 122–128.
Hague, Richard, I Campbell, and Phill Dickens. 2003. "Implications on design of rapid manufacturing." *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 217 (1): 25–30.
Hertz, Alain, and Marino Widmer. 2003. "Guidelines for the use of meta-heuristics in combinatorial optimization." *European Journal of Operational Research* 151 (2): 247–252.
Holland, John.H. 1992. "Adaption in Natural and Artificial Systems." *Ann Arbor MI: The University of Michigan Press* 211.
Hur, Sung Min, Kyung Hyun Choi, Seok Hee Lee, and Pok Keun Chang. 2001. "Determination of fabricating orientation and packing in SLS process." *Journal of Materials Processing Technology* 112 (2-3): 236–243.
Ikonen, Ilkka, William E Biles, Anup Kumar, and Rammohan K Ragade. 1996. "Concept for a genetic algorithm for packing three dimensional objects of complex shape." .
Ikonen, Ilkka, William E Biles, Anup Kumar, John C Wissel, and Rammohan K Ragade. 1997. "A Genetic Algorithm for Packing Three-Dimensional Non-Convex Objects Having Cavities and Holes." In *Proceedings of the 7th International Conference on Genetic Algorithms*, 591–598.
Ikonen, Ilkka, William E Biles, James E Lewis, Anup Kumar, and Rammohan K Ragade. 1998. "GARP : Genetic algorithm for part packing in a rapid prototyping machine." *Intelligent Systems in Design and Manufacturing* 3517: 54–63.
Jakobs, Stefan. 1996. "On genetic algorithms for the packing of polygons." *European journal of operational research* 88 (1): 165–181.
Jia, X., M. Gan, R. A. Williams, and D. Rhodes. 2007. "Validation of a digital packing algorithm in predicting powder packing densities." *Powder Technology* 174 (1-2): 10–13.
Karabulut, Korhan, and Mustafa Murat İnceoğlu. 2004. "A hybrid genetic algorithm for packing in 3d with deepest bottom left with fill method." In *International Conference on Advances in Information Systems*, 441–450. Springer.
Khajavi, Siavash H., Martin Baumers, Jan Holmström, Ender Özcan, Jason Atkin, Warren Jackson, and Wenwen Li. 2018. "To kit or not to kit: Analysing the value of model-based kitting for additive manufacturing." *Computers in Industry* 98: 100–117.
Knight, James B., H. M. Jaeger, and Sidney R. Nagel. 1993. "Vibration-induced size separation in granular media: The convection connection." *Physical Review Letters* 70 (24): 3728–3731.
Langelaar, Matthijs. 2016. "Topology optimization of 3D self-supporting structures for additive manufacturing." *Additive Manufacturing* 12: 60–70.
Lesh, Neal, Joe Marks, Adam McMahon, and Michael Mitzenmacher. 2005. "New heuristic and interactive approaches to 2D rectangular strip packing." *Journal of Experimental Algorithmics (JEA)* 10: 1–2.
Li, Q, I Kucukkoc, and D Z Zhang. 2017. "Production planning in additive manufacturing and 3D printing." *Computers and Operations Research* 83: 1339–1351.
Lodi, Andrea, Silvano Martello, and Daniele Vigo. 2004. "TSpack: A Unified Tabu Search Code for Multi-Dimensional Bin Packing Problems." *Annals of Operations Research* 131 (1-4): 203–213.
Min, Patrick. 2004. "Binvox, a 3d mesh voxelizer." .
Möbius, Matthias E, Benjamin E Lauderdale, Sidney R Nagel, and Heinrich M Jaeger. 2001. "Brazil-nut effect: Size separation of granular particles." *Nature* 414 (6861): 270.
Panesar, Ajit, Ian Ashcroft, David Brackett, Ricky Wildman, and Richard Hague. 2017a. "Design framework for multifunctional additive manufacturing: Coupled optimization strategy for structures with embedded functional systems." *Additive Manufacturing* 16: 98–106.
Panesar, Ajit, David Brackett, Ian Ashcroft, Ricky Wildman, and Richard Hague. 2017b. "Hierarchical remeshing strategies with mesh mapping for topology optimisation." *International Journal for Numerical Methods in Engineering* 111 (7): 676–700.
Pannequin, Rémi, Gérard Morel, and André Thomas. 2009. "The performance of product-driven manufacturing control: An emulation-based benchmarking study." *Computers in Industry* 60 (3): 195–203.
Peko, Ivan, Andrej Bašić, and Amanda Aljinović. 2018. "Computer aided design, additive manufacturing and 3D product scanning." In *Inovativno pametno poduzeće*, Fakultet elektrotehnike, strojarstva i

brodogradnje.
Peter-Orth, Christoph. 1985. "All solutions of the Soma cube puzzle." *Discrete mathematics* 57 (1-2): 105–121.
Petrovic, Vojislav, Juan Vicente Haro Gonzalez, Olga Jord Ferrando, Javier Delgado Gordillo, Jose Ramn Blasco Puchades, and Luis Portols Grian. 2011. "Additive layered manufacturing: sectors of industrial application shown through case studies." *International Journal of Production Research* 49 (4): 1061–1079.
Poon, P. W., and J. N. Carter. 1995. "Genetic algorithm crossover operators for ordering applications." *Computers and Operations Research* 22 (1): 135–147.
Ravindran, Ashwin. 2003. "An octree based genetic algorithm for three-dimensional packing of irregular parts." PhD diss.
Royall, Richard M. 1986. "The effect of sample size on the meaning of significance tests." *The American Statistician* 40 (4): 313–315.
Ruffo, M, and R Hague. 2007. "Cost estimation for rapid manufacturing - simultaneous production of mixed components using laser sintering." *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 221 (11): 1585–1591.
Safe, Mart\'\in, Jessica Carballido, Ignacio Ponzoni, and Nélida Brignole. 2004. "On stopping criteria for genetic algorithms." In *Brazilian Symposium on Artificial Intelligence*, 405–413. Springer.
Schaeffer, Jonathan, Paul Lu, Duane Szafron, and Robert Lake. 1993. "A re-examination of brute-force search." In *Proceedings of the AAAI Fall Symposium on Games: Planning and Learning*, 51–58.
Sivanandam, S N, and S N Deepa. 2007. *Introduction to genetic algorithms.* Springer Science & Business Media.
Smit, Selmar K, and Agoston E Eiben. 2009. "Comparing parameter tuning methods for evolutionary algorithms." In *2009 IEEE congress on evolutionary computation*, 399–406. IEEE.
Srinivas, M., and L. M. Patnaik. 1994. "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms." *IEEE Transactions on Systems, Man and Cybernetics* 24 (4): 656–667.
Stoneman, Paul. 2001. *The economics of technological diffusion.* Wiley-Blackwell.
Stoyan, Y. G., N.I. Gil, A. Pankratov, and G. Scheithauer. 2004. "Packing non-convex polytopes into a parallelepiped." *MATH-NM-06-2004* (June).
Stroud, Ian. 2006. *Boundary representation modelling techniques.* Springer Science & Business Media.
Szykman, S, and J Cagan. 1995. "A simulated annealing-based approach to three-dimensional component packing." *... .Asmedigitalcollection.Asme.Org* 117: 308.
Voudouris, Christos, Edward P.K. Tsang, and Abdullah Alsheddy. 2010. "Guided Local Search." *Springer* 321–361.
Wäscher, Gerhard, Heike Haußner, and Holger Schumann. 2007. "An improved typology of cutting and packing problems." *European journal of operational research* 183 (3): 1109–1130.
Wu, Jun, Anders Clausen, and Ole Sigmund. 2017. "Minimum compliance topology optimization of shellinfill composites for additive manufacturing." *Computer Methods in Applied Mechanics and Engineering* 326: 358–375.
Zhang, Jianming, Xifan Yao, and Yun Li. 2019. "Improved evolutionary algorithm for parallel batch processing machine scheduling in additive manufacturing." *International Journal of Production Research* 1–20.

Appendix

Reference	Problem	# of runs	Elitism	Replacement strategy	Population	Crossover	CP[a] (%)	Selection scheme	Mutation	MP[b] (%)	Rotation
Ikonen et al. (1996)	3DSP[c] and min. overlap	NI[d]	NI[d]	Generational	NI[d]	OX1[e]	100	Ranking	Swap	0.5 - 5	45
Ikonen et al. (1997)	3DSP[c] and min. overlap	5	NI[d]	Generational	50	OX1[e]	90	Roulette wheel	Swap	1 - 20	45
Ikonen et al. (1998)	3DSP[c] and min. overlap	1	NI[d]	Generational	100	OX1[e]	80	Roulette wheel	Swap	7 - 13	45
Hur et al. (2001)	3DSP[c]	NI[d]	NI[d]	Generational	NI[d]	OX1[e]	NI[d]	NI[d]	Swap	NI[d]	NI[d]
Ravindran (2003)	3DBP[f]	NI[d]	NI[d]	Generational	75	OX1[e]	70	Ranking	Non-uniform (angles)	12	90
Canellidis et al. (2006)	2D Knapsack	NI[d]	NI[d]	NI[d]	50	SJX[g]	NI[d]	Roulette wheel	Swap; rotate parts in 90	75	90
Gogate and Pande (2008)	3DSP[c]	1	Yes	Generational	3 * # of parts	Single point	75	Roulette wheel	Complementary	20	45
(Canellidis, Giannatsis, and Dedoussis 2013)	2D Knapsack	NI[d]	NI[d]	Generational with weak replacement	50	SJX[g]	100	Roulette wheel	Swap	1.25	90

[a] Crossover probability; [b] Mutation probability; [c] Three-dimensional Strip Packing; [d] Not informed;
[e] Order 1 Crossover (Falkenauer and Bouffouix 1991); [f] Three-dimensional Bin Packing; [g] (Jakobs 1996)

Table 6. Reported characteristics of the GA implementations for AM in the 3DIP literature.