



A Build-Time Estimator for Additive Manufactured Objects

Luca Di Angelo^(✉), Paolo Di Stefano^(✉),
and Emanuele Guardiani^(✉)

Department of Industrial and Information Engineering and Economics,
University of L'Aquila, L'Aquila, Italy
{luca.diangelo, paolo.distefano}@univaq.it,
emanuele.guardiani@graduate.univaq.it

Abstract. Additive Manufacturing is a very time consuming technology. An estimation of the build time is fundamental to:

- Evaluate the production cost in budgeting process.
- Make use of optimization methods, which use as parameter the build time, for determining optimal build direction.

In both these cases a fast and valid build time estimator, which can work with a few input data deducible from geometric model, is required.

In the proposed paper a reliable parametric-based method to determine the build time for additive manufactured objects is provided. The implemented method is based on a back-propagation artificial neural network, which gives the possibility to implement the complex functions that elapse some driving build-time factors and the build time. The neural network training is based on data provided by a properly developed analyzer of the list of commands given to AM machines, which performs an analytical estimation of the build time. The implementation of the proposed methodology is illustrated and some comparisons between the real and estimated build-time are provided, then the results are critically analyzed.

Keywords: Build time estimation · Adaptive model · Process planning · Additive Manufacturing

1 Introduction

Additive Manufacturing (AM) is a technology for fabricating functional prototypes and small batches of part. It allows to realize components characterized from a complex shape and provides important advantages related to the cost saving, compared with other manufacturing processes, and the reduction of the Time to Market [1, 2]. But AM is also a very time consuming technology. Due to the limitations of the technologies currently in use the speeds of the growing process need to be limited for maintaining a good quality of the manufactured part. Consequently AM is generally applied only to small batches of parts. In this sense is very important to provide an estimation of the build time before the part is realized, in order to evaluate the production costs.

This information is essential for the sellers during the budgeting process for providing the part cost to the customer and also for planning the production. Indeed many optimization methods for AM have been suggested during the last years [3–12], whose intents are to increase the profits and also the quality of the manufactured objects. In these methods, whose final output is the optimal build direction of the interested part, the build time is generally required as input. Consequently, a reliable way for evaluating this parameter is necessary.

Two possibilities are offered by the literature to estimate the build time: by parametric-based methods [13–21] or by detail analysis-based methods, such as [24]. In parametric-based methods the build time is expressed as a function of the driving build-time factors. These last can be evaluated starting from the geometrical model and before defining the manufacturing process, which can be very useful for the budgeting process. The parametric-based methods are fast and easy-to-use but, on the other side, their accuracy is poor. In detail-analysis based method instead the estimation of the build time is very accurate but they commonly require a full geometrical information of the interested part, which could not be available in some situations.

In this paper the use of a parametric-based method is illustrated. Most of the driving factors of the build time are identified and analyzed. Then these factors are used as input of an artificial neural network, which allows to identify the correlation between the driving build-time factors and the build time. A similar work was proposed by the Authors in a previous activity research, but it evidenced some limitations related to the calculation of some driving factors due to new typologies of hatching that have been introduced during this recent years (Fig. 1). Moreover network learning was very slow since, not being available an accurate tool for estimating the build time, the real value of this last was required. Both limitations have been passed thanks to the use of the new methodologies which will be afterwards described.

2 Proposed Method

In parametric-based methods the build time t_f is expressed as

$$t_f = t_f(T, DTF_1, DTF_2, \dots, DTF_n) \quad (1)$$

The parameter T is representative of the technology in use, while DTF_i identifies the i -th driving build-time factor. A very difficult step is represented by the individuation of these factors. In order to avoid cross-correlation the driving build-time factors should be independent each other. Moreover, to make the method general-purpose, they should be independent from the specific technology used for manufacturing the part.

In the proposed method the use of the following five driving build-time factors is suggested:

- $\frac{V_{mat}}{L}$ Volume of part/Layer thickness
- p_{mat} Total length of part's perimeter
- n_{r-mat} Number of repositioning required for part's hatching

$\frac{V_{sup}}{L}$ Volume of supports/Layer thickness
 n_{r-sup} Number of repositioning required for supports hatching

In Table 1 each factor is compared with the most widespread AM technologies to establish which parameters should be used for a specified process.

The chosen factors can be evaluated relatively easily starting from the geometrical model of the part. Moreover they are quite independent between them, as it was demonstrated in [20].

The driving build-time factors are evaluated by a method implemented into a Matlab script, which process a.stl representation of the object.

The volume of the part V_{mat} is computed using the methodology proposed by [22].

Table 1. Correlation between the driving build-time factors and typically commercial rapid prototyping technologies.

Driving build-time factors	Technologies							
	FDM	SLA	SLS	LOM	3DP	MJM	EBM	SLM
<u>MATERIAL</u>								
$\frac{V_{mat}}{L}$	✓	✓	✓	✓	✓	✓	✓	✓
P_{mat}	✓	✓	✓	✓			✓	✓
n_{r-mat}	✓	✓	✓		✓	✓	✓	✓
<u>SUPPORT</u>								
$\frac{V_{sup}}{L}$	✓	✓		✓	✓	✓		
n_{r-sup}	✓	✓		✓	✓	✓		

Some technologies, as evidenced in Table 1, require a deposition of the layer contour. A very accurate estimation of the total layers' contour is provided by the formula (2) suggested by [23], in which the perimeter is calculated as the “projection” of the triangular facets of the geometrical model divided by the layer distance.

$$P_{mat} = \frac{\sum_{j=1}^{n_T} S_j \sqrt{1 - (\mathbf{z} \cdot \mathbf{n}_j)^2}}{L} \tag{2}$$

Where:

- n_T Number of triangular facets of geometrical model
- S_j Area of j -th triangular facet
- \mathbf{z} Normal vector of building direction
- \mathbf{n}_j Normal vector of j -th triangular facet

Another parameter which significantly affects the build time is related to the number of repositioning of the deposition tool. During each repositioning an acceleration and a jerk phase are required, causing an increasing of the build time [24]. The number of repositioning is strictly related with the typology of hatching used. In Fig. 1 some of the most common hatching structures are reported.

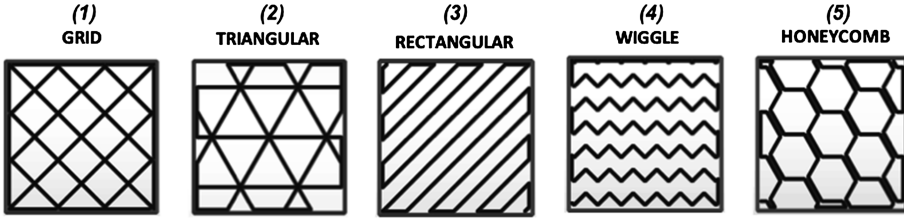


Fig. 1. Most in use typologies of hatching for AM.

For the structures 1, 2 and 3, in order to evaluate the number of repositioning n_{r-mat} , an extension of the methodology already in use for the rectangular hatching is proposed [21].

$$n_{r-mat} = \sum_{k=1}^2 \left(\frac{\left(\sum_{j=1}^{n_T} S_j \sqrt{1 - (\mathbf{z} \cdot \mathbf{n}_j)^2} \right)}{H_{mat} L} |\boldsymbol{\tau}_{k,c} \cdot \mathbf{n}_{j,xy}| \right)$$

Where:

- H_{mat} Hatching distance
- $\boldsymbol{\tau}_{k,c}$ Normal vector referred to hatching direction
- $\mathbf{n}_{j,xy}$ Projection onto the stratification plane of the normal vector of j -th triangular facet

The phase θ of the vector $\boldsymbol{\tau}_{k,c}$ assumes different values depending on hatching typology, in accordance with Table 2.

Table 2. Relationship between k , phase θ and hatching typology.

Hatching typology	$k = 1$	$k = 2$
Grid (1)	$\theta(\boldsymbol{\tau}_{1,c}) = \tau$	$\theta(\boldsymbol{\tau}_{2,c}) = \tau + \frac{\pi}{2}$
Triangular (2)	$\theta(\boldsymbol{\tau}_{1,c}) = \tau$	$\theta(\boldsymbol{\tau}_{2,c}) = \tau + \frac{\pi}{3}$
Rectangular (3)	$\theta(\boldsymbol{\tau}_{1,c}) = \tau$	$\boldsymbol{\tau}_{2,c} = \mathbf{0}$

For the structures 4 and 5 instead, the number of repositioning is not proportional to the external perimeter p_{mat} but rather to the hatching area A_i of the i -th layer. In that case the expressions reported in Fig. 2 are used for evaluating n_{r-mat} :

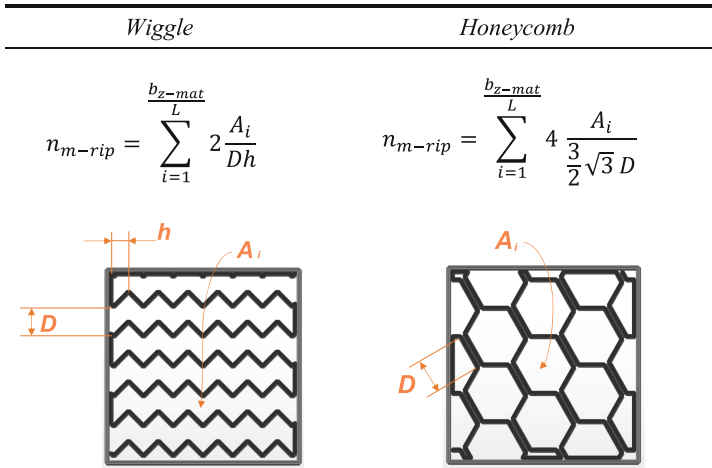


Fig. 2. Formulation of for wiggle and honeycomb hatching.

A formulation for determining the same parameters with reference to the supports should now be provided. But no information about the support structures is contained into the geometrical model and, moreover, these last are strictly dependent on user choices and technology in use. So a strategy, based on voxelization, has been implemented into a Matlab script for determining the geometrical information referred to the supports [24]. The total volume of supports V_{sup} is provided by the sum of the volume of the voxel elements:

$$V_{sup} = \sum_{i=1}^{N_v} V_{v,i}$$

Where:

N_v Number of voxel elements

$V_{v,i}$ Volume of the i -th voxel element

While the total number of repositioning n_{r-sup} required for the supports is obtained considering the total perimeter p_{sup} of the supporting structure.

$$p_{sup} = N_v \left(4 \frac{d^2}{L} \right)$$

Where:

d Grid dimension of voxel element

Then the same methodology proposed for the material can be used, depending on the kind of hatching chosen.

Now a relationship between the proposed driving build-time factors and the build time needs to be founded. Generally speaking, the function (1) is very complex and unknown. So an Artificial Neural Network (ANN) is used to meet the request. Those mathematical instruments represent a powerful solution for generating a complete representation of a domain not entirely known.

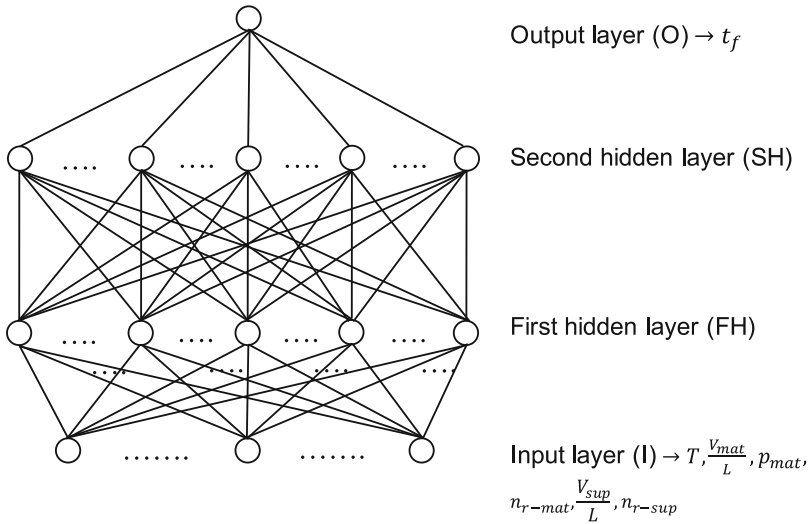


Fig. 3. Configuration of the artificial neural network.

A neural network, being able to reproduce non-linear functions, provides the possibility of considering all the complex dependencies which elapse between the driving build-time factors, offering, for example, the opportunity to predict the effects of the acceleration and jerk phases. Moreover a neural network is characterized by an adaptive capability; if the estimation provided by the network for a prototype results to be unsatisfactory, this last can be added to the set of samples used for training the network. In this way the accuracy of results increases over time.

For our purpose, a typical back-propagation artificial network with two hidden layer (Fig. 3) has been used. The parameter T , which identifies the AM technology in use, is provided in input to the network in addition to the driving build-time factors. That parameter should work as a logical value and it is chosen between the positive integer numbers depending on the technology in use ($T_{FDM} = 1, T_{SLA} = 2, \dots, T_{SLM} = 8$). In order to train the network, the real build time of the test samples is required. In [21] the learning process was very slow because, not being disposable a reliable way for evaluating this quantity, each test case required to be physically realized. Therefore, in order to maintain the method economically and temporally sustainable, few samples could be provided for network learning. For example no other build direction except the optimal one was used and this represented a significant limitation of the network's ability to generalize the results.

In the proposed method, by the use of a detail-analysis based method [24], now an extremely accurate evaluation of the build time, based on the use of GCode, is applied for evaluating very precisely the real build time. The use of this methodology requires that some specific information concerning the manufacturing process should be provided, for example the magnitude of accelerations and jerks, or the infill percentage of part and support. The necessity to know this information represents a limit of the method itself in comparison with the proposed one, but the use of it allows us to obtain very quickly an evaluation of the build time. In this way many test cases, using different orientations or shapes, can be used as reference for the learning process, increasing the method accuracy. In Fig. 4, the workflow followed for obtaining the real build time of the samples used for network learning is illustrated.

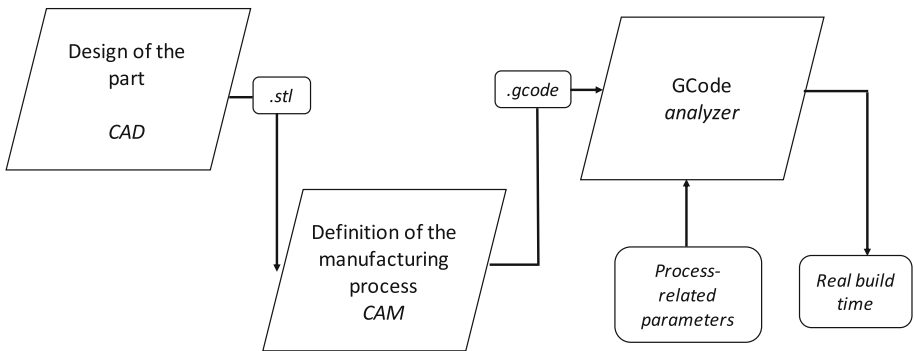


Fig. 4. Workflow for obtaining the build time using the proposed detail-analysis based method.

3 The Results

In order to train the ANN, seventy-seven test cases, which are characterized from different shape, dimension, geometrical and topological complexity, have been used taking as reference the FDM technology. Some of these test cases have been evaluated more than once by changing building direction and/or scaling its geometry. Therefore the ANN was trained by using ninety training instances.

A first experiment (A) was conducted for identifying the best number of nodes to use for the two hidden layers of ANN. As evidenced by [25], an optimal number of nodes can be founded only through an empirical investigation and it is affected by the complexity of the problem. If the number of nodes is too low, the accuracy of the neural network will be low too. Otherwise if the number of nodes is increased beyond a threshold level, then the method will lose its capability to generalize the results.

Six different ANN configurations were trained, using, respectively, 25, 50, 65, 75, 85 and 100 nodes. Then the samples reported in Fig. 5 were submitted to the ANNs to verify their capability to evaluate the build time. These samples belong to the seventy-seven models used for training ANNs, but they were rotated and/or scaled and then submitted to the neural network. The estimation error, shown in Fig. 6, evidence that the optimal case is provided when 75 nodes are used for each hidden layer. In that configuration the area of the polygonal figure that corresponds to 75 nodes is minimum.

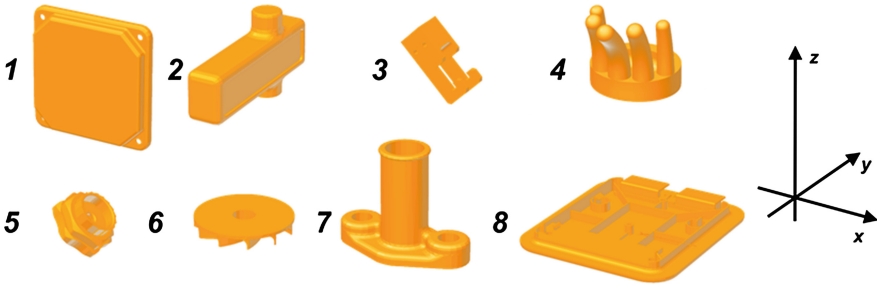


Fig. 5. Validation samples used for experiment (A).

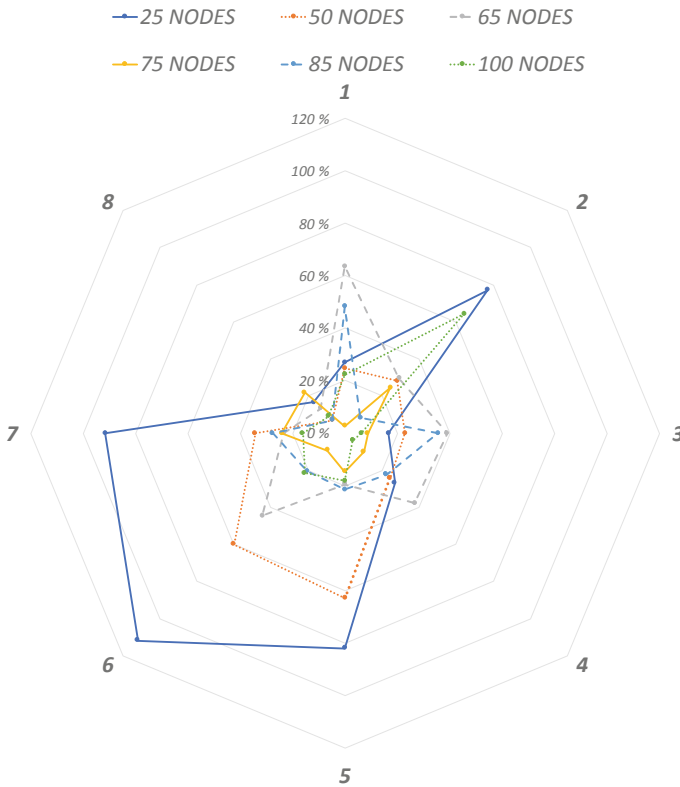


Fig. 6. Relationship between number of nodes and percentage error in build time estimation of experiment (A) for the eight proposed samples.

In order to analyze the behavior of the ANN in regard to new geometrical models, not used in the training phase, another experiment (B) was conducted. Six samples, reported in Fig. 7, were analyzed and their build time estimated. The results, reported in Table 3, evidenced a very satisfying estimation of the build time when 100 nodes are used.

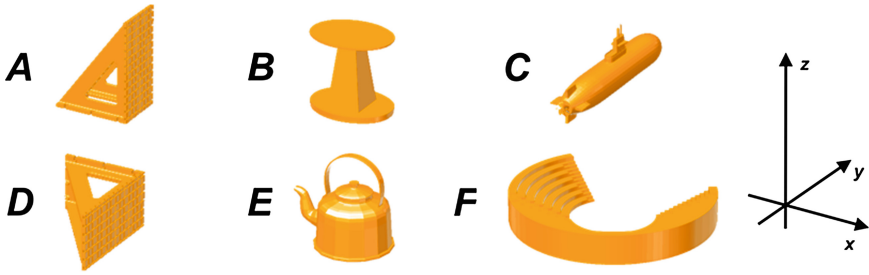


Fig. 7. Validation samples used for experiment (B).

Table 3. Results of experiment (B).

Model	75 NODES		100 NODES		REAL BUILD TIME
	Estimation [min]	Error	Estimation [min]	Error	[min]
A	123	18%	106	3%	104
B	241	39%	159	8%	173
C	239	146%	89	8%	97
D	131	3%	131	3%	135
E	553	15%	423	12%	479
F	490	36%	748	2%	762

The difference between the behavior of the neural network towards already-trained samples, in which only the building direction or the size have been modified, and new samples has been attributed to a mathematical instability of ANN. When too many neural connections are present into the hidden layers, small variations of the driving build-time factors cause a significant change of the build time. In the case of experiment (A) many of the driving build-time factors does not vary a lot when the build direction changes. For example one the most significant driving factors, such as the volume, is not affected by build direction. This does not occur in the experiment (B) when new geometrical models are analyzed. In that experiment an increasing number of neurons seems to improve the accuracy of the network. This is demonstrated by the fact that, when 75 nodes are used, the average error for the build time estimation is equal to 43% of the real build time, while for the case of 100 nodes the average error is only 6%.

4 Conclusions

In this paper a parametric-based approach for evaluating the build time for AM technologies has been presented. A parametric estimation of this quantity could be very useful for a fast and privacy-aimed evaluation of the costs required for manufacturing an object. Respect to the published methods by Authors, in this paper the following new elements are introduced:

- Among eight original driving build-time factors, only the five with a low correlation each other have been considered. A new parameter T has been introduced in formula (1) which considers the AM technology in use.
- A new formulation of the driving build-time factors is proposed in order to take into account the different typologies of hatching for AM.
- A fast analytical procedure was used for determination of the real build time.

In order to approximate the build time, the very complex and non-linear function of the previously defined driving build-time factors, an ANN with two hidden layers is used.

With a first experimentation the optimal number of neurons for the two hidden layers was defined. The net training was performed with ninety test cases opportunely chosen.

The obtained results show a good performance with a mean error of 6% in the build time estimation for objects for which the ANN has never been trained.

In a further work may be interesting to use an ANN with more neural connections which may lead to an even better estimation of the build time. Furthermore the availability of a reliable detailed-analysis based method for evaluate the build time could be exploit for improving even more the quality of the test cases. For example, as suggested by the results of experiment (A), a better specialization of the ANN for different build directions of a same object is required.

References

1. Rajenthirakumar, D., Jagadeesh, K.A.: Analysis of interaction between geometry and efficiency of impeller pump using rapid prototyping. *Int. J. Adv. Manuf. Technol.* **44**, 890–899 (2009)
2. Vinodh, S., Devadasan, S.R., Maheshkumar, S., Aravindakshan, M., Arumugam, M., Balakrishnan, K.: Agile product development through prototyping technologies: an examination pump-manufacturing company. *Int. J. Adv. Manuf. Technol.* **46**(5–8), 663–679 (2010)
3. Alexander, P., Allen, S., Dutta, D.: Part orientation and build cost determination in layered manufacturing. *Comput. Aided Des.* **30**(5), 343–356 (1998)
4. Byun, H.S., Lee, K.H.: Determination of optimal build direction in rapid prototyping with variable slicing. *Int. J. Adv. Manuf. Technol.* **28**(3–4), 307–313 (2006)
5. Byun, H.S., Lee, K.H.: Determination of the optimal build direction for different rapid prototyping processes using multi-criterion decision making. *Robot. Comput. Integr. Manuf.* **22**(1), 69–80 (2006)
6. Armillotta, A., Cavallaro, M., Minnella, S.: A tool for computer-aided orientation selection in additive manufacturing processes. In: *Proceedings of the 6th International Conference on Advanced Research in Virtual and Rapid Prototyping* (2013)
7. Thrimurthulu, K.P.P.M., Pandey, P.M., Reddy, N.V.: Optimum part deposition orientation in fused deposition modeling. *Int. J. Mach. Tools Manuf.* **44**(6), 585–594 (2004)
8. Canellidis, V., Giannatsis, J., Dedoussis, V.: Genetic-algorithm-based multi-objective optimization of the build orientation in stereolithography. *Int. J. Adv. Manuf. Technol.* **45**(7–8), 714–730 (2009)

9. Singhal, S.K., Prashant, K.J., Pandey, P.M., Nagpal, A.K.: Optimum part deposition orientation for multiple objectives in SL and SLS prototyping. *Int. J. Prod. Res.* **47**(22), 6375–6396 (2009)
10. Phatak, A.M., Pande, S.S.: Optimum part orientation in rapid prototyping using genetic algorithm. *J. Manuf. Syst.* **31**(4), 395–402 (2012)
11. Brika, S.E., Zhao, Y.F., Brochu, M., Mezzetta, J.: Multi-objective build orientation optimization for powder bed fusion by laser. *J. Manuf. Sci. Eng.* **139**(11), 111011 (2017)
12. Jaiswal, P., Patel, J., Rai, R.: Build orientation optimization for additive manufacturing of functionally graded material objects. *Int. J. Adv. Manuf. Technol.* 1–13 (2018)
13. Kamash, T., Flynn, D.: Build time estimator for stereo-lithography machines—a preliminary report. *Prototype Express* **2**, 2 (1995)
14. Kechagias, J., Maropoulos, S., Karagiannis, S.: Process build-time estimator algorithm for laminated object manufacturing. *Rapid Prototyping J.* **10**(5), 297–304 (2004)
15. Nezhad, A.S., Vatani, M., Barazandeh, F., Rahimi, A.: Build time estimator for determining optimal part orientation. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **224**(12), 1905–1913 (2010)
16. Chen, C., Sullivan, P.: Predicting total build-time and the resultant cure depth of the 3D stereo-lithography process. *Rapid Prototyping J.* **2**(4), 27–40 (1996)
17. Ruffo, M., Tuck, C., Hague, R.: Empirical laser sintering time estimator for Duraform PA. *Int. J. Prod. Res.* **44**(23), 5131–5146 (2006)
18. Campbell, L., Combrinck, J., Barnard, D.L.: Stereo-lithography build time estimation based on volumetric calculations. *Rapid Prototyping J.* **14**(5), 271–279 (2008)
19. Munguía, J., Ciurana, J., Riba, C.: Neural-network-based model for build-time estimation in selective laser sintering. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **223**(8), 995–1003 (2009)
20. Zhang, Y., Bernard, A.: Generic build time estimation model for parts produced by SLS, in high value manufacturing. In: *Advanced Research in Virtual and Rapid Prototyping: Proceedings of the 6th International Conference on Advanced Research in Virtual and Rapid Prototyping*, Leiria. CRC Press (2010)
21. Di Angelo, L., Di Stefano, P.: A neural network-based build time estimator for layer manufactured objects. *Int. J. Adv. Manuf. Technol.* **57**(1), 215–224 (2011)
22. Zhang, C., Chen, T.: Efficient feature extraction for 2D/3D objects in mesh representation. In: *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, vol. 3, pp. 935–938 (2001)
23. Di Angelo, L., Di Stefano, P.: Parametric cost analysis for web-based e-commerce of layer manufactured objects. *Int. J. Prod. Res.* **48**(7), 2127–2140 (2010)
24. Di Angelo, L., Di Stefano, P., Guardiani, E.: A build time estimator for additive manufacturing. In: *Proceedings 2019 International Workshop on Metrology for Industry 4.0 and Internet of Things*, pp. 327–332, IEEE catalog number CFP19N49-USB (2019)
25. Sukthomya, W., Tannock, J.: The training of neural networks to model manufacturing processes. *J. Intell. Manuf.* **16**, 39–51 (2005)