# Determining the optimal build directions in layered manufacturing

A. SANATI NEZHAD, M. VATANI, F. BARAZANDEH, A.R RAHIMI

Mechanical Department Amirkabir University of Technology 424 Hafez Ave, Tehran IRAN amirsanatinejhad@gmail.com

*Abstract*: - Determining the optimal build directions is one of the most critical factors in RP processes because it affects on the build time, support structure, surface quality as well as the cost. The previous methods could handle simple parts or limited objective functions. Moreover the methods with multiple objective functions, had abstracted them into a single fitness function in which the characteristics of individual objectives could not be reflected properly. In the present work a new algorithm is presented to determine the optimal build-up directions of the part in Stereolithography systems. The algorithm can help RP users select among the optimal build-up directions. The optimization is done using Multi-Objective Genetic Algorithm (MOGA). The method proposed here handles build time, support volume and surface finish as objective functions individually. At each GA step, the surface finish is achieved applying adaptive layer thickness method and optimization algorithm finds the optimal build directions with minimum build time and volume support. The algorithm is developed by MATLAB. To evaluate the algorithm, several sample parts are checked.

Key-Words: - Stereolithography- Optimal Build Direction- Multi Objective Optimization - Genetic Algorithm

# **1** Introduction

Build direction is a vital factor in Rapid Prototyping (RP) affecting surface finish, build time, the complexity of support structure, as well as the cost. However, determining the best part direction is not always easy. One direction may result in the desired surface finish with a long build time. In other words, satisfying one objective may adversely affect other objects.

There are several studies on the best part orientation to satisfy different objectives, like build time, support complexity, surface finish and accuracy. Lan et al. find the part orientation for the best surface finish based on the minimum sum of areas of stepped surfaces [1]. Build time was assessed by the height of the part in the deposition direction. Support structure was evaluated by determining the supported points. Optimal orientation for one of the objectives at a time was determined from the list of pre-selected directions. Cheng et al. applied a multiple objective approach to determine the optimal part orientation applying build time and part accuracy as objectives [2]. The primary objective was Part accuracy. The secondary objective was to minimize the build time. It was achieved by reducing the number of slices. The method used the CAD model directly to reach the most accurate part with minimum build time. The main disadvantage of direct slicing is the capability among various CAD systems. It can only be used for a specific kind of CAD software and machine, and is not applicable to any other CAD combinations. This fact shows that STL-based method is still the commonly used method in slicing the part.

Masood's approach was to find the best part orientation by minimizing volumetric error [3]. In his method, difference in the volume of the part deposited using constant layer thickness method is minimized for selecting an appropriate direction. This method includes a primary volume approach, which assumes a complex part to be constructed from a combination of primitive volumes. The simplest build time reduction method was to count the number of layers among some user defined directions developed by Alexander [4]. He determined suitable part direction for better part accuracy and lower cost. The main source of part inaccuracy considered in his work was due to the stair stepping effect and is measured based on cusp height. Masood also presented multi objective Genetic Algorithm for part orientation utilizing minimum build time and average surface roughness [5]. Xu et al. used Genetic Algorithms (GA) to find the optimal part orientation among preselected orientations [6]. Weighted sum of build time, accuracy and stability of the part is considered as a criterion. Build time was calculated by the ratio of number of adaptive slices to largest possible number of slices. Accuracy was estimated by ratio of overhang area to the total surface area and stability was estimated by considering the penalty approach.

Hong's algorithm minimizes the build time and part cost by employing adaptive variable layer thickness [7]. Frank and Fadel proposed an expert system considering surface finish and build time as the objectives [8]. They emphasized that surface part quality should be selected as the primary objective. Build time was considered as second priority because rapid prototyping process is used as a faster fabrication method compared to traditional prototyping techniques. Support structure had a least preference among the objectives. The surface finish was assigned as the rules. The user then selects two geometric features (such as a hole, a round surface, a thin structure, or overhang). The expert system determines the direction with suitable surface finish and support structure for the two selected features.

Two recent attempts have been made to find out optimum build direction rather than finding out suitable build direction among pre-selected directions. Thrimurtullu et al. determined optimal part orientation in Fused Deposition Modeling (FDM) by considering weighted sum of average part surface roughness [9]. Build time has chosen as an objective function to minimize. They used real coded Genetic Algorithm for optimization. Pandey [10] has published a review paper to compare the various approaches for part orientation.

Thus far, none of the developed algorithms are capable of handling multi-objective optimization independently. Furthermore, in most algorithms, the user had to assign some pre-selected orientations, among which the algorithm could determine the best one. In addition, they were only valid for a few categories of parts. Most of the proposed methods did not take advantage of adaptive layer thickness to save build time.

This paper presents an Optimized Pareto Based Part Orientation (OPBPO) algorithm to find out a series of the optimal part orientations considering the build time, support volume and surface finish, both simultaneously and independently. The surface finish criteria are achieved based on an adaptive slicing method to decrease build time with desired surface roughness. Build time and support volume are optimized as the objective functions in Pareto Based optimization using the Multi-Objective Genetic Algorithm (MOGA) method in MATLAB toolbox. To evaluate the algorithm, several sample parts are checked.

# 2 Adaptive Slicing

One method to save the build time under desired surface finish is to employ the variable slicing thickness method. The layer thickness depends on the geometry of the part. Jamieson and Hacker [11] proposed the direct slicing of CAD models without extracting the STL file. They first sliced the part into layers with maximum identical thickness of 0.2 to 0.25 mm. The layers are then bisected continuously until the desired surface accuracy is achieved. Dolenc and Makela proposed the variable slicing thickness method and introduced the cusp height tolerance concept [12]. The layer thickness at a layer is computed based on the cusp height, which is always less than the maximum user-specified cusp height. Cormier et al. used variable cusp heights for different locations of the part to reduce production time [13]. Sabourin et al. proposed an adaptive slicing method for layered manufacturing [14]. The CAD model is first sliced with constant layer thickness. Each layer is then re-sliced uniformly as needed to achieve the desired surface finish. Applying this method of slicing showed reduction in build time by approximately 50% without reducing surface finish. Weiyin et al. presented slicing algorithms that works based on a non-uniform rational B-spline surface model [15]. The slicing algorithm is developed to obtain an accurate and smooth part surface. A selective hatching method is applied to reduce the build time by solidifying the kernel regions of a part with the maximum allowable thickness while solidifying the skin areas with adaptive thin layers to obtain the required surface accuracy. Lee and Choi presented an adaptive slicing scheme to improve the accuracy and speed of computing [16]. They used sampling of points on a sliced contour to find the optimal point which decides the next slice thickness. The concept of character line is introduced by them which can further increase the speed of computing.

Substantial amount of work has been done to reduce the build time by pre specifying cusp height through adaptive slicing, but most of these adaptive slicing procedures use the concept of cusp height tolerance, developed by Dolenc and Makela [12]. Dutta and Kulkarni used cusp height concept using CAD model directly. The vertical normal curvature is introduced by them to calculate the slice thickness [17]. The cusp height concept is also employed in OPBPO to calculate layer thickness. The maximum allowable cusp height (Ra) is received from user (Fig. 1). Ra is in fact an indication of maximum allowable surface roughness. The smaller the cusp height, the finer the surface finish is.



Fig. 1 Thickness computation based on cusp height

To calculate the thickness for each layer  $(t_i)$ , the facets intersecting the layer slicing plane are determined. Then the angle between the normal vectors of these facets with working plane (XY plane, Fig. 1) is detected  $(\theta_i)$ . The layer thickness is then obtained from Equation (1).

$$t_i = \frac{R_a}{\sin \theta_{\min}} \tag{1}$$

Where  $\theta_{\min} = \min(\theta_i)$ .

In Stereolithography,  $t_i$  must lay between 0.05 to 0.2 mm (allowable thickness). If the result of Equation (1) is beyond this limit,  $t_i$  is replaced with 0.05 or 0.2 (whichever is closer).

### 3 Multi Objective Optimization by GA

In multiple-objective problems, the objectives usually conflict with each other, preventing simultaneous optimization of each one individually. GA is a meta-heuristic tool that solves optimization problems. There are two general methods for multiple objective optimizations. One method is to combine the individual objectives into a single function. The main problem is to select the appropriate weight for each objective. In practice, it can be very difficult to precisely select these weights. An objective function can also be utilized as a constraint in the fitness function. In addition, in both cases, an optimization method produces a single solution rather than a set of solutions that can be examined for trade-offs. The second method is to determine Pareto optimal solutions. A Pareto optimal set is a series of solutions that are not dominated by one another. From one Pareto solution to another, there might be a loss of one objective at the same time that the other is enhanced. Pareto optimal solutions are often preferred to single solution. Consider а minimization problem with K objectives and n decision variable ( $\vec{x} = \{x_1, \dots, x_n\}$ ). The optimization algorithm finds p series of vectors  $\vec{z}_i(\vec{x}_i) =$  $\{z_{i1}(\vec{x}_i),\ldots,z_{ik}(\vec{x}_i)\}\$  (1<i<p), that minimizes a part of the K objective functions [18]. Each  $\vec{x}_i$  is a pareto optimum solution and X={  $\vec{x}_i$ , 1 < i < p} is the matrix of non dominated solutions.

In a minimization case, solution  $\vec{x}$  is said to dominate solution  $\vec{y}$ , if each element of answer  $\vec{z}(\vec{x})$ is less than the corresponding element of  $\vec{z}(\vec{y})$ . A solution is said to be Pareto optimal if it is not dominated by any other solutions in the solution space. In other words, a Pareto optimal solution cannot be improved further without criticizing at least one objective function.

The first multi-objective GA, called Vector Evaluated GA (VEGA), was proposed by Schaffer [19]. Afterwards. several multi-objective evolutionary algorithms (MOEAs) were developed including Multi-objective Genetic Algorithm (MOGA) [20], Niched Pareto Genetic Algorithm (NPGA) [21], Non dominated Sorting Genetic Algorithm (NSGA) [22]. These cited GA are wellknown and credible algorithms used in many applications. Their performance was tested in several comparative studies [23]. Generally, multiobjective GA differs based on its fitness assignment procedure, elitism, or diversification approaches.

MOGA was the first multi-objective GA that explicitly used Pareto-based ranking and niching techniques together to search the true Pareto front while maintaining diversity in the population [18]. Therefore, it is a good method to demonstrate how Pareto based ranking and fitness sharing can be integrated in a multi-objective GA to find true Pareto fronts.

# **4 OPBPO Algorithm**

As mentioned earlier, in OPBPO, surface roughness, build time and support volume are selected as the main objective functions. In this algorithm the surface are achieved using an adaptive slicing method to decrease build time with desired surface roughness. Build time and support volume are optimized as the objective functions applying the Multi-Objective Genetic Algorithm (MOGA) method. The algorithm finds a series of nondominate  $(\theta_x, \theta_y)$  to minimize build time and support volume under the desired surface finish. Fig. 2 shows the block diagram of OPBPO. OPBPO first reads the STL file. The allowable thickness limits, Ra, and h<sub>s</sub> (hatching space) are entered next. The new generation is then produced based on different part orientation  $(\theta_x, \theta_y)$  where  $\theta_x \epsilon [0,360]$ and  $\theta_y \epsilon [0,180]$  (as Kim [24] has mentioned as well). For each  $(\theta_x, \theta_y)$ , a transfer matrix is produced to orient the part. Because  $(\theta_x, \theta_y)$  covers all possible orientations,  $\theta_z$  does not require consideration (Fig. 3).



**X** Fig.3 Rotating of part around x and y axis

In any generation, for all members (orientations), the surface finish is satisfied utilizing adaptive slicing method mentioned in Equation (1). The build time and the support volume are then computed for each orientation. The build time is composed of the time to build the part itself, the time to make the supports and layers' recoating time. In the next step, the dominant individuals are selected. The procedure continues until the stop condition is met. The stop condition is assigned by the operator. It can be the number of generations or a difference smaller than a user defined number in fitness value between two consequent generations. The output is a group of orientations with minimum build time and support satisfying the desired surface finish. The algorithm to calculate the part (itself) build time is presented in Fig. 4.



Fig. 4 Build time calculation Algorithm

After applying the adaptive slicing method, the number of the layers and the thickness of every layer are determined. These data are used as input parameters to calculate the build time.

In an STL file, every triangular facet is described by three vertices and a normal unit vector. As shown in Fig. 5, assume P1(x1,y1, z1), P2(x2, y2, z2) and P3(x3, y3, z3) as three vertices of a triangle. The sample slice plane with Z = z[i] intersects edges P1P2 and P2P3, coordinate of the intersection points are  $P_k[i]$  and  $P_{k+1}[i]$ . In this algorithm, for every layer, all intersecting facets are found. The intersection points (Fig. 5) are then stored in a matrix. This matrix includes all the contours points (internal+ external).



Fig. 5 Intersection points for a sample facet

The internal and external contours at each layer are distinguished using the method proposed by Choi in [25]. He employed the parent-and-son relationship algorithm to separate internal and external contours (Fig. 6). All contours are assumed to be closed and non-intersecting.



Fig. 6 Parent-and-son relationship among contours [15]

To define the parent-and-son relationship, contours  $C_x$  and  $C_y$  are considered.  $C_y$  is the parent of  $C_x$ , if an arbitrary point of  $C_x$  is inside  $C_y$ . For each contour, the number of parents is calculated. If the number of parents is odd, the contour is internal. Otherwise it is an external contour. Once internal and external contours are assigned, the layer is hatched. The total

travel length of laser for both contours and hatches is then calculated. With access to the data of the contours (coordinates of intersection points), the total travel length to solidify the layer's contours is calculated. To determine the total travel length to hatch a layer, the cross section area for all layer's contours is calculated. In this case, the area for internal contours is negative and for external contours is considered positive. Algebraic sum of areas of all contours is the cross section area of the layer. On the other hand, the surrounding area of the inner contours is subtracted from the surrounding area of outer contours. By dividing the cross section area by  $h_s$  parameter, the travel length by laser to hatch the layer is estimated. Due to differences existing in the layers' thickness (as a result of adaptive layer thickness), the laser speed should be calculated independently for each layer. Dividing the total travel length by laser speed, the layer build time is obtained. The part build time is the sum of the individual layers' build time and recoating time, as well as the time to build the supports. The time to build the support structure can be computed similar to the method used to build the layers.

The support structure is used for different situations. The most common one is to support the surfaces of the part not to permit to warp or sag when elevator moves up and down in the resin. The overhanging surfaces on which material is solidified continuously need support structures. In this case, support structures prevent the overhanging surface from dropping. Support structure is also used when the part becomes unstable during the solidification. Support structure includes not only the surface area of the support which is in contact with the part, but also the surface area, which have contact with the base of the elevator.

structures The support are fabricated simultaneously with the original part. After solidification, the support structures must be removed. As this process is often done manually, the more support structures, the more time for the finishing operations. For complicated parts, the removal operations may be difficult and time consuming and also can reduce the accuracy of the part. It is therefore important to minimize the number of support structures. Depending on the part orientation on the machine different amounts of support will be required. The influence of support structure can be considered by two factors: overhanging area and volume [26]. The overhanging area is the area of the surfaces where supports are located. Such surfaces have poor surface finish. The volume of supports affects the build time and the cost of the prototype.

OPBPO output is a series of orientations with minimum build time and support volume satisfying the desired surface finish. Figures 7, 9 and 12 are the outputs for some sample cases.

A question that may arise here is why do authors think Pareto methods like OPBPO are preferred to traditional GAs, where all objective functions are abstracted to a single fitness function? To answer this question, it should be mentioned that parameters such as build time and support volume are not congruent objective functions. Therefore, adding them in a single fitness function cannot show the roll of each objective function properly to evaluate the optimum part orientation. In addition, the output of traditional GAs is one unique optimum direction. OPBPO applies the build time and support volume as independent objective functions to provide a group of optimum directions. Because a family of optimum solutions is presented, the user can select among the solutions.

### **5** Case Studies

To find out the functionality of OPBPO, the algorithm is run for several cases. Fig. 7 illustrates the solid model and STL model of a perfume box. The minimum and maximum allowable layer thickness are 0.05 mm and 0.2 mm, Ra is 0.1 mm,  $h_s = 0.25$  mm and the laser power is chosen to be 50 mw. MATLAB toolbox uses MOGA method as the multiple objective evolutionary method to solve optimization problems. The MOGA parameters are listed in Table 1.

Parameters	description	
Population Ssize	20	
Number of Generation	20	
Crossover Fraction	0.8	
Migration Fraction	0.2	
Elite Count	2	
Scaling Function	Rank Method	
Population Type	Double Vector	

Table 1 MOGA parameters in OPBPO for case studies

Table 2 shows OPBPO non-dominant solutions data including part directions build time and support volume. Fig. 8 is the plot of non-dominant solutions.



Fig. 7 Solid and STL model of the electronic part

Directions	θ <sub>x</sub> (Degree)	θ <sub>y</sub> (Degree)	Build Time (hr)	Support Volume (cm <sup>3</sup> )
1	164.33	59.24	3.799	10.436
2	115.98	102.15	4.006	7.834
3	115.98	99.34	4.022	7.275
4	132.72	82.50	4.457	5.663
5	128.73	82.15	4.411	6.713
6	165.49	66.440	3.896	9.898
7	126.79	80.97	4.296	6.990

Table 2 OPBPO non-dominant Pareto optimum solutions of perfume box



Fig. 8 Build time and support volume for non dominant directions of perfume box

The OPBPO proposes optimal orientations as Pareto front solutions (directions). These directions are not dominated by each other but are the best direction among all other directions in orientation space.

The next case study is the body of mouse. The CAD model and STL view of the mouse is seen in

Fig. 9. Optimal directions are seen in Table 3. Comparing the first and second solutions in Table 3, the build time for the first solution is more than the second one, but the support volume is less. However all of them are non-dominant solutions. This is the strong point of OPBPO. The user can choose the proper orientation based on his/her needs. In Fig. 10, the build time and support volume for the optimal directions are depicted.



Fig. 9 CAD and STL model of the body of mouse

Directions	θ <sub>x</sub> (Degree)	θ <sub>y</sub> (Degree)	Build Time (hr)	Support Volume (cm <sup>3</sup> )
1	0	180	2.4651	32.0392
2	196.31	90.34	4.1410	13.931
3	219.95	90.45	4.2561	3.3319
4	217.1420	90.45	4.2513	3.3729
5	143.20	99.66	4.7515	2.4551
6	185.61	89.01	4.1044	21.0537

Table 3 Pareto solutions of the mouse



Fig. 10 Build time and support volume for non dominant directions of the body of mouse

The other case study is an electronic part  $(200 \times 80 \times 30 \text{ mm})$ . Its CAD and STL model have been shown in Fig. 11. In Table 4 and Fig. 12 are the non-dominant solutions. The input parameters are similar to the first case.



Fig. 11 STL model of the electronic part

Directions	$\theta_x$ (Degree)	θ <sub>y</sub> (Degree)	Build Time (hr)	Support Volume (cm <sup>3</sup> )
1	95.25	89.95	20.84	35.60
2	181.14	89.93	20.70	38.25
3	94.03	89.935	20.83	36.13
4	102.13	89.93	20.95	30.14
5	316.18	89.673	23.02	25.5
6	102.13	89.93	20.95	30.14
7	101.46	89.952	20.92	31.84

Table 4 Pareto solutions of the electronic part



Fig. 12 Build time and support volume for non dominant directions of the electronic part

Skipping the small variations of  $\theta_y$  in Table 4, due to calculation errors, the optimum solutions do not depend on  $\theta_x$ . To clarify the result, the initial direction of the part along with seven optimum directions in Table 4 are pictured in Fig. 13. As the figure shows, when  $\theta_y = 90$ , changing  $\theta_x$  will not have any effect on the build time and support volume. This is because the cross section of the part is identical for all sections parallel to xy plane. The diversity of answers for  $\theta_x$  also proves this result. In fact the number of non-dominant answers here are infinite. This case study can be considered as a sample part that future shows the ability of OPBPO method to find the unique optimal build-up direction on the machine.



Fig. 13 The initial and optimum directions for the electronic part

### **6** Conclusions

OPBPO is proposed to compute the optimum part orientations with minimum build time and support

volume as well as desired surface finish. It employs adaptive slicing method along with Pareto based multi objective GA to handle the parts. OPBPO provides a series of best part orientations while optimizing objective functions both simultaneously and independently.

OPBPO functions quite well for exceptional parts, where there is a unique optimum direction satisfying both minimum support volume and build time. This promising feature is illustrated in the electronic part in Fig. 13.

Previous studies have shown that surface finish is more important in comparison to build time and support volume. It has a direct effect on die quality, produced through RP. Build time and support volume mainly affect cost. OPBPO considers surface finish a priority via the adaptive slicing method. The build time and support volume do not have a conceptual relationship to be considered in a unique fitness function. OPBPO can evaluate them independently as the second priority items.

The algorithm was run for different parts. The method has been shown to work for simple parts such as rectangles and cylinders. The results are also impressive for complex parts.

#### References:

[1] P.T. Lan, S.Y. Chou, L.L. Chen, D. Gemmill, Determining fabrication orientation for rapid prototyping with Stereolithography apparatus, *Computer Aided Design*, 1997.

[2] W. Cheng, JYH. Fuh, AYC. Nee, Multi objective optimization of part building orientation in Stereolithography, *Adv Manufacturing Technology*, 1995.

[3] S.H. Masood, W. Rattanawong, and P. Iovenitti, Part build orientations based on volumetric error in fused deposition modelling, *Adv Manufacturing Technology*, 2003.

[4] Alexander, and Allen, S. and Dutta, D., Part orientation and build cost determination in layered manufacturing, *Computer Aided Design*, 1998.

[5] S.H. Massod, W. Rattanawong, P. Iovenitti, A generic algorithm for part orientation system for complex parts in rapid prototyping, *Journal of Material Processing Technology*, 2003.

[6] F. Xu, Y. S. Wong, H. T. Loh, J. Y. H. Fuh, and T. Miyazawa, Optimal orientation with variable slicing in stereolithography, *Rapid Prototyping Journal*, Vol. 3, No. 3, 1997, pp 76-88.

[7] S. B. Hong, K.H. Lee, Determination of optimal build direction in rapid prototyping with variable slicing, *Journal of Adv Manufacturing Technology*, 2006.

[8] D. Frank, G. Fadel, Expert system-based selection of the preferred direction of build for rapid prototyping processes, *Journal of Intelligent Manufacturing*, 1995.

[9] K. Thrimurtullu, P. M. Pandey, N.V. Reddy, Optimal part deposition orientation in fused deposition modelling, *International Journal of Machine Tools and Manufacture*, Vol. 44, No. 6, 2003, pp 585-594

[10] P.M. Pandey, Part Deposition orientation studies in 3 layered manufacturing, *Journal of Materials Processing Technology*, 2006

[11] R. Jamieson, H. Hacker, Direct slicing of cad models for rapid prototyping. *Rapid Prototyping Journal*, 1995.

[12] A. Dolenc, I. Makela, Slicing procedure for layered manufacturing techniques, *Computer Aided Design*, 1994.

[13] D. Cormiert, K. Unnanon, E. Sanni, Specifying non-uniform cusp heights as a potential for adaptive slicing, *Rapid Prototyping Journal*, 2000.

[14] E. Sabourin, S.A. Houser, J.H. Bohn, Adaptive slicing using stepwise uniform refinement, *Rapid Prototyping Journal*, Vol.2, No.4, 1996, pp. 20–26.

[15] M. Weiyin, W. Chung, NURBS-based adaptive slicing for efficient rapid prototyping, *Computer Aided Design*, Vol.36, 2004, pp. 1309–1325

[16] K.H. Lee, K. Choi, Generating optimal sliced data for layered manufacturing, *International Journal of Advanced Manufacturing Technology*, Vol.16, 2000, pp. 277–285.

[17] P. Kulkarni, D. Dutta, An accurate slicing procedure for layered manufacturing, *Computer Aided Design*, 1996, PP. 683–697.

[18] JD. Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, *Proceedings of the international conference on genetic algorithm and their applications*, 1985.

[19] A. Abdullah Konak, D.W. Coit, A.E. Smith, Multi-objective optimization using genetic algorithms, *Reliability Engineering and System Safety*, 2006, PP. 992–1007.

[20] CM. Fonseca, PJ. Fleming, Multi-objective genetic algorithms, *IEEE colloquium on Genetic Algorithms for Control Systems Engineering*, 1993.

[21] J. Horn, N. Nafpliotis, DE. Goldberg, A niched Pareto genetic algorithm for multiobjective optimization, *IEEE world congress on computational intelligence*, 1994, Orlando, FL, USA.

[22] N. Srinivas, K. Deb, Multi-objective optimization using nondominated sorting in genetic algorithms, *J Evol Comput*, 1994.

[23] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms, *J of Evol Comput*, 2000.

[24] C. Kim, S.H, Lee, Reduction of post-processing for Stereolithography systems by fabricationdirection optimization, *Computer-Aided Design*, 2005.

[25] S.H. Choi, K.T. Kwok, Hierarchical Slice Contours for Layered Manufacturing, *Computers in industry*, 2002.

[26] J. Hur, K. Lee, The Development of a CAD Environment to Determine the Preferred Build-up Direction for Layered Manufacturing, *Int J Adv Manuf Technol*, Vol.14, 1998, pp. 247-254