Multi-Objective Optimization and Multi-Criteria Decision Making For FDM Using Evolutionary Approaches

Nikhil Padhye, Subodh Kalia and Kalyanmoy Deb
npdhye@gmail.com, subodh@iitk.ac.in, deb@iitk.ac.in
Department of Mechanical Engineering
Indian Institute of Technology Kanpur, Kanpur-208016, U.P., India

KanGAL Report 2009007

December 24, 2009

Abstract

In this paper, we describe a systematic multi-objective problem solving approach, simulataneosly minimizing two conflicting goals - average surface roughness '*Ra*' and build time '*T*', for object manufacturing in FDM process by usage of evolutionary algorithms. Popularly used multi-objective genetic algorithm NSGA-II and recently proposed multi-objective particle swarm optimization (MOPSO) algorithms, are employed for the optimization purposes. Statistically significant performance measures are employed to compare the two algorithms and means to arrive at approximate Pareto-optimal fronts are also suggested. To refine the solutions obtained by the optimizers, a mutation driven hill climbing local search is also proposed. Several suggestions and three new proposals pertaining to the issue of decision making in presence of trade-off solutions are also made. The overall procedure is integrated into a *MORPE* - Multi-objective Rapid Prototyping Engine. Several sample objects are considered for simulation to demonstrate the working of *MORPE*. Finally, a careful study of optimal build directions for several components considered indicates a trend, providing an insight into the FDM processes and can be considered useful for various practical RP applications.

Multi-objective Optimization, Decision Making, Genetic Algorithms, Particle Swarm Optimization and FDM rapid prototyping process.

1 Introduction

Rapid prototyping (RP) or layered manufacturing refers to processes in which a component is fabricated by layer-by-layer deposition of material from 3D computer assisted design models. It is an emerging technology which is becoming increasingly important in the market today. RP is playing an important role in reducing the time required for new product development and lowering development costs and thus, many companies are realizing the benefits of producing prototypes quickly and easily.

Today there exist multiple RP techniques. Common examples of RP techniques are Fused Deposition Method (FDM), Stereolithography (SLA), Selective Laser Sintering (SLS), Laminated Object Manufacturing (LOM), 3D printing and Direct Metal Deposition (DMD). With the advent of these technologies, it is now possible to fabricate physical prototypes directly from CAD models for checking the feasibility of design concept and prototype verification.

To create a physical object using RP cycle, first creation of geometric model by a Computer Aided Design (CAD) modeler is done. This is followed by determination of suitable deposition orientation, slicing, generation of material deposition paths, part deposition and post-processing operations. Many of these steps can be done automatically by the RP machine, but usually part deposition orientation is selected by the user. Part orientation has significant affect on build time and surface quality [2]. For some

RP methods it also affects the support structure. Since it is usually desired to manufacture components with low surface roughnesses (quantified as *Ra* in this paper) and build times (denoted by T), a mechanism to automatically determine orientation is desired. Such a study is conducted in this paper which performs the search for optimal build orientations, to simultaneosly minimimize *Ra* and T , with respect to FDM process using nature inspired algorithms. It turns out that minimization of considered objectives is conflicting in nature which leads to set of trade-off solutions with varying *Ra* and T. Further, in presence of such trade-off points the issue of decision making, selecting one orientation from a set of available orientations demands an addressing. It turns out that post optimal analysis of these trade-off solutions for various objects provides a deeper insight leads to discovery of new knowledge from optimization.

The entire procedure is automated using a developed software - *Multi-objective Rapid Prototyping Engine* ($MORPE$) to achieve the aforementioned task. The software tool is developed for FDM system and is easily modifiable for other RP techniques. $MORPE$ incorporates two evolutionary algorithms, NSGA-II and MOPSO, for optimization purposes, variable slicing module to carry out the slicing of solid model and subsequent of computation of (*Ra*, T) at any given orientation, and inbuilt tools like attainment surface estimator and hypervolume calculator to arrive at results of statistical importance. This tool is made freely down loadable from http://home.iitk.ac.in/∼npadhye and shall serve as useful resource for entire RP community.

The rest of the paper is organized as follows: Section 2 reviews various optimization studies carried out with respect to build orientations in LM literature.Section 3 sets up the multi-objective problem formulation for FDM process. In section 4, a systematic approach to address the multi-objective optimization task has been proposed. This section firstly discusses the variable slicing procedure utilized in this study. Secondly, two popular multi-objective evolutionary optimizers (NSGA-II and MOPSO) are described. Thirdly, introduction of statistically comparable performance measures, *Hypervolume Indicator* and *Attainment Surface Approximator*, is made. Finally, the section describes a proposed mutation driven hill climbing local search using achievement scalarizing function (ASF) for refinement of solutions obtained by evolutionary optimizers. In section 6 a series of simulations are carried out to validate and investigate the proposed approach. In section 7 results are compared and conclusions are drawn. This section also provides insight into the decision making issue and innovative design principles are deciphered via. post optimal analysis. Finally, conclusions are made in section 8.

2 RELATED WORKS

Choice of build orientation for part fabrication in LM manufacturing has been an active area of research for more than a decade. Broadly speaking goals are to minimize fabrication time (or cost) and maximize part accuracy. Usually these goals depend on the build orientation in accordance with the characteristics of the specific LM technology involved. The objective function formulation of such goals has been widely researched in past too. The measure employed for quantifying build time (or cost) is usually the number of layers [3, 7, 12, 14, 26, 31] or, the part height when layer thickness is constant [2, 13, 16]. For LM technologies that require support structures during fabrication, the estimated support structure volume has also been applied as time cost criterion [2, 10]. Post-processing time is another important cost factor that gets directly affected by the orientation choice; hence it has been considered also as a criterion for the orientation selection [13].

To account for the fabrication quality several indicators have been suggested: estimated surface roughness [7, 26, 30], weighted average surface roughness [4, 5], and total area of surfaces with estimated roughness above a certain limit [1]. Additionally, various criteria related with known sources of dimensional inaccuracies such as volumetric error [19, 20], the process planning or stair stepping error [3, 12, 30], and, for SL the trapped volume error [32], have been proposed. Other quality resembling measures, which have been proposed are, the total overhang area [10, 31], the stability of the part structure during fabrication [12, 31], and perceived mechanical strength [29].

Once the measure to quantify time or cost (first objective) and surface quality (second objective) are decided, a systematic search procedure is required to discover favorable orientations - which optimize the considered objectives simultaneously. Since determination of these objectives at different orientations often involves substantial computational cost (i.e. rotation of CAD model and subsequent slicing) employment of efficient optimization algorithms is desired. Further depending on part shape or models for criteria employed, objective functions may exhibit discontinuity rendering most gradient based methods ineffective. Evolutionary algorithms, like genetic algorithms(GAs), have established themselves as potential candidates in addressing challenges posed by real world problems including multi-objective optimization tasks [8]. A brief review of past works related to optimization tasks is in order as follows. Previous studies in LM literature employing genetic algorithms (GAs) for build orientation optimization have mostly considered either single objective study or combination of multiple objectives into one using weighted approach. In [5] optimal build directions were explored using genetic algorithm for different rapid prototyping processes. Two goals, average weighted surface roughness (AWSR) and build times, are combined to form a single objective function, considered for minimization. In [1] single objective genetic algorithm was employed to determine optimal fabrication directions for LM processes so as to minimize the required post-machining region (RPMR) in LM (as post-machining is often required to improve the surface quality). Here, the authors developed an expression of the distribution of surface roughness and relation between the RPMR and fabrication direction. In [6] build orientations for parts fabricated with stereolithography are derived for optimizing build time, surface roughness and post-processing times using single objective weighted approach. Other studies in literature that have also employed single objective weighted approach are [10, 14, 30, 31].

For the optimization of a single criterion, like the part height, the average cusp height, or the total post-processing area, specific algorithms have been proposed in previous studies [1, 18, 16]. In [7, 16] authors select orientations from a list of pre-selected set (determined by ranking of objectives and thus, allocating importance). Such a pre-selection mechanism or minimization of weighted single objective function as previously stated have well-known deficiencies and optimality of the solutions cannot be guaranteed [8]. However, more recently suitable multi-objective optimization approaches using genetic algorithms, i.e. simultaneously minimizing or maximizing several goals, have been studied for different LM processes [17, 24, 25, 26, 27]. Similar attempts to optimize multiple goals in this direction have been made [11, 33, 34, 35].

Despite such studies, a systematic application of nature inspired heuristics coherently addressing multi-objective optimization, decision-making and knowledge discovery through optimization is still missing. To address the existing shortcomings, we have chosen FDM process for which optimal build orientations are determined and post-optimal analysis is carried out. tlm

3 MULTI-OBJECTIVE PROBLEM

Without loss of generality, we assume that the goal is to minimize m functions $f_1, \ldots, f_m$ of n-dimensional decision variables ϕ. A decision vector $\phi_1 \in S$ is called Pareto-optimal if there is no other decision vector $\phi_2 \in S$ that dominates it. Any vector ϕ_1 is said to dominate ϕ_2 if ϕ_1 is not worse than ϕ_2 in all of the objectives and it is strictly better than ϕ_2 in at least one objective. In case two solutions ϕ_1 and ϕ_2 do not dominate each other, we say that they are indifferent to each other or are non-dominated with respect to each other. To solve such problems, algorithms which can find a well distributed set of trade-off and well converged set of solutions with least computational expense are desired.

In current study the objectives of interest are average surface roughness Ra and total build time T. Thus, the following bi-objective optimization problem with two variables is set up: ϕ=θ_x, θ_y can be set up.

Minimize f_1 = Ra(ϕ)

Minimize f_2 = T(ϕ)

subject to:
$0 \leq \theta_x \leq 180$,
$0 \leq \theta_y \leq 180$.

The problem variables are θ_x and θ_y representing the rotations from an initial configuration about some reference XYZ Cartesian coordinate system.

Figures 2(a) and 2(b) describe the rotation scheme stated here by considering rotation of a facet or planar triangle (CAD model represented in form of facets can be rotated by rotation of all facets).

Computation of Ra and T has been borrowed from [24, 25, 26], where for FDM surface roughness is a function of slice thickness (t) and build angle θ, as shown in figure 1, and is computed for each layer as follows: If build angle θ is between 0-70 degrees:

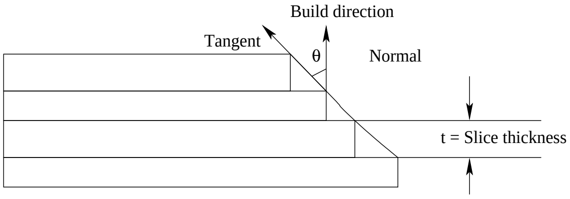


Figure 1: Showing how Ra is computed for a layer based on θ.

$$Ra(\mu m) = K \times \frac{t(mm)}{cos\theta}, \tag{1}$$

Where K lies in (69.28 - 72.36). For build angle = 90 degree i.e. for a horizontal surface

$$Ra(\mu m) = 117.6 \times t(mm), \tag{2}$$

If build angle is greater than 70 degrees and less than 90 degrees:

$$Ra(\mu m) = \frac{1}{20}[90Ra_{70^\circ} - 70Ra_{90^\circ} + \theta(Ra_{90^\circ} - Ra_{70^\circ})], \tag{3}$$

where Ra_{70} and Ra_{90} are surface roughnesses at 70° and 90° build angles. For the sake of simplicity we have ignored the Ra contribution due to support structures unlike done in the original proposals. Finally the average part surface roughness is calculated as:

$$Ra(\mu m) = \frac{\Sigma Rai}{total\ number\ of\ slices}, \tag{4}$$

where Ra_i is the roughness of the i^{th} trapezium. The build time (T) for a component is equal to the sum of build times of individual layers.

$$T_{build} = \sum_{i=1}^{N_{layer}} t_{layer_i} + N_{layer} \times t_{zmove} + \frac{N_{layer}}{k} \times t_{wipe}, \tag{5}$$

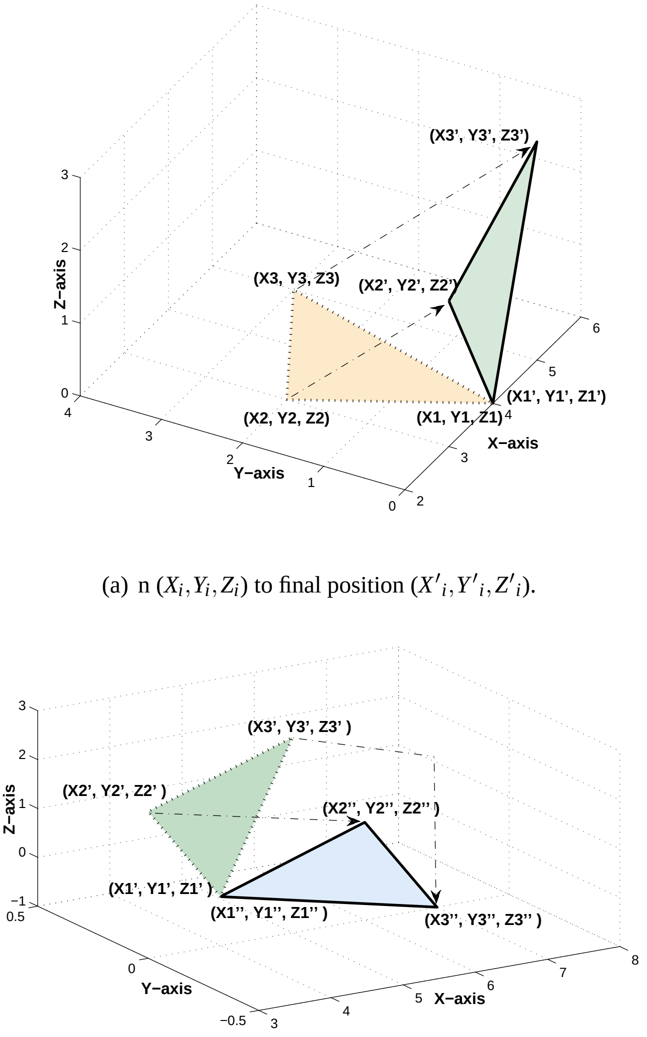
Here t_{wipe} is machine specific time to wipe the nozzle and time to build i^{th} layer is itself sum of times to lay the part t_{part_i}, support structure t_{supp_i} and table movement t_{movei}.

$$t_{layer_i} = t_{part_i} + t_{supp_i} + t_{movei}, \tag{6}$$

Where t_{part_i} for i^{th} layer is computed as follows:

$$t_{part_i} = \frac{A_{si}}{r_w} \times v. \tag{7}$$

Here, $\frac{A_{si}}{r_w}$ is the material contained area in i^{th} layer, r_w is the road width and v is the nozzle speed. It is assumed that due to support structure build time is negligibly affected as taken in [24]. The machine specific parameters have been taken for Stratasys FDM 1650 system installed at IIT Kanpur equipped for prototyping with ABS plastic. It is worthwhile to mention that optimal build orientation directly depends on the model employed for computation of Ra and T, thus more realistic and accurate model is favourable. However; the focus in this study is to work with a reasonable model and demonstrate the multi-objective optimization problem solving and decision making approach, elucidating its principles. Since material laying deposition is assumed to be along Z-axis, the rotation about Z-axis is invariant for the computation of objectives , hence, only X-axis and Y-axis rotations are considered.



(a) n (X_i, Y_i, Z_i) to final position (X'_i, Y'_i, Z'_i).

(b) Rotation of the facet about Y axis by 90° with initial position (X'_i, Y'_i, Z'_i) to final position (X''_i, Y''_i, Z''_i)

Figure 2: (a) Rotation of the facet about X axis by 90° with initial position (X'_i, Y'_i, Z'_i) to final position $(X''_i, Y''_i, Z''_i$. (b) Rotation of the facet about Y axis by 90° with initial position (X'_i, Y'_i, Z'_i) to final position $(X''_i, Y''_i, Z''_i$.

4 PROPOSED APPROACH

The overall procedure is carried out by MORPE which comprises of following modules: a) Adaptive slicing procedure b) Multi-objective optimizers - NSGA-II and MOPSO c) Performance comparison tools - Hypervolume Indicator and Attainment Surface Approximator d) Local Search Procedure e) Decision Making Tools. Figure 3 portrays the working of MORPE. For MORPE, adaptive slicing procedure has been developed in Matlab version R2007a. The optimization routines and performance comparison measures are developed in C (gcc version 4.3.2) language. Matlab code is compiled using MCR (matlab compiler runtime) version 7.6 and integrated with optimization engine. The experiments reported in this study have been carried out on Intel single core 2.9 GHz, RAM-1.0 GB, Hard disk-80GB, OS-Linux-Ubuntu-9.04, Computer architecture-32 bit. The codes developed in this paper can be obtained from following url http://home.iitk.ac.in/∼npadhye.

4.1 Adaptive Slicing

In past adaptive slicing procedure has been adopted to improve the surface quality and accuracy in LM processes. The adaptive procedure developed in this study is borrowed from [28]. Its salient features are discussed next.

The basic function of any slicing module is to generate two dimensional slices from a three dimensional tessellated model. The input to the slicing engine is a STL file of the solid model under consideration. The STL file comprises of coordinates of the triangular facets and their normals and entire solid model is represented by its constituent facets. A triangular facet comprises of three points each of which is associated with (x,y,z) coordinate. For following discussion we assume that z-axis denotes the vertical direction (direction of material deposition) and z_{min} and z_{max} denote the lowest and highest z-coordinates of any solid model, respectively.

For efficient slicing procedure an effective facet processing technique is employed: firstly, facets are grouped into facet groups (based on same z_{min}) and then into sub-facet-groups (based on same z_{min} and z_{max}). Next, slicing planes are considered at intervals from z_{min} to z_{max}. Since, facets have already been grouped and sub-grouped, as stated before, intersection of slicing planes with facets can be found efficiently, saving considerable amount of computational overhead. At each new slicing plane a check for a new feature (based on feature recognition rules) is carried out. In case a new feature is detected a series of slicing planes is considered at small intervals so that feature informations are well captured.

If user has defined an upper limit on *Ra*, then surface roughness is calculated at every slicing step for two adjacent slices and if *Ra* exceeds the specified bound, position for the upper slicing planes are recomputed so that *Ra* stays within the bounds.

In case of constant height slicing procedure, feature detection mechanism and bound-check on *Ra* are omitted. Here, the slicing planes are considered at specified interval height. Both the adaptive slicing and constant height slicing procedures can be downloaded from http//:home.iitk.ac.in/∼npadhye. Figures 5(a) and 5(b) compare constant and adaptive height slicing procedures for a cuboid oriented at an angle of 45 degree about X-axis. In constant slicing interval height of 1.5 units is considered. In adaptive slicing a maximum height interval of 1.5 units is allowed and bound on *Ra* for each layer is set to 40.0 units. As observed, in adaptive case slice thicknesses are automatically calculated while keeping *Ra* bounded.

4.2 Evolutionary Optimizers

Although there exist several multi-objective evolutionary algorithms (MOEAs) in literature, popularly used genetic algorithm based NSGA-II and particle swarm based MOPSO optimizers have been utilised in this study. In the following paragraphs we briefly describe the working and salient features of these algorithms.

MOPSO: Particle swarm optimization (PSO) is now a well established optimization technique in variety of contexts. PSO is a population based technique, similar in some respects to other evolutionary algorithms, except that potential solutions (particles) move rather than evolve through the search space. PSO consists of several candidate solutions called particles each of which has a position and velocity, and experiences linear spring-like attractions towards two attractors:

1) the best position attained by that particle so far (particle attractor or personal best (pbest);

2) the best of the particle attractors in a certain neighbourhood (neighbourhood attractor or global best global best).

More recently, PSO has successfully been extended to multi-objective optimization problems and such methods are called Multi-objective Particle Swarm Optimization (MOPSO). Its simple implementation, population based approach, success in handling continuous search spaces and notions of individual position and velocity are major reasons for its popularity. PSO works with a population of individuals each of which is subjected to movement in direction of *‘Pbest’* - position corresponding to best fitness attained by an individual and a *‘Gbest’* - position of best fitness individual in the entire population. In each generation or cycle (‘t’), every individual is associated with a position vector ($\bar{\phi_t}$) and a velocity vector ($\bar{v_t}$). The size of these vectors is equal to the size of the vector ϕ_t. The position and velocity of each individual is updated according to following equations:

$$\bar{v_t} = w\bar{v_t} + c_1\bar{r_1}\cdot(\bar{p}_{g,t} - \bar{\phi_t}) + c_2\bar{r_2}\cdot(G\bar{B}est_t - \bar{\phi_t})$$

$$\bar{\phi_{t+1}} = \bar{v_t} + \bar{\phi_t}$$

$$\bar{\phi_{t+1}} = \bar{\phi_t} + \bar{\delta}$$

Above are position and velocity update equations. The term w is known as inertia weight and, c_1 and c_2 are known as learning factors. In our procedure w has been chosen as 0.5, c_1 and c_2 are both taken to be 1.0. Once velocities and positions have been updated, a random perturbation, donated by $\bar{\delta}$, is added to an individual’s position based on some probability. This is known as ‘turbulence factor’ and is analogous to ”mutation” employed in Genetic Algorithms. Goal of ”turbulence” is to preserve diversity in the population.

The MOPSO utilised in this study has been borrowed from [22, 23]. At the start of optimization, for all N particles positions (ϕ) are initialised randomly and velocities (v) are set to zero. At the onset pbest for each particle is assigned as the the particle itself. The current MOPSO maintains an external archive of non-dominated solutions of the population which is updated after every generation. This global archive is empty in the beginning and can store a maximum number of non-dominated solutions which is specified at the start. In case the number of non-dominated solutions exceed the maximum size of the archive, in any generation, clustering is invoked to restore the archive size. For each particle in the population a personal archive, also called *pbest archive* is maintained. The pbest archive contains the most recent non-dominated positions that particle has encountered while searching the space. Such additional archiving scheme for the particles is often found to be extremely effective.

In every generation, each particle is assigned two guides pbest and gbest from its pbest archive and swarms global archive. The way in which these guides are allocated has a great impact on algorithms performance. Several methods for guide selections exist. In this study, ‘NWtd.’ and ‘Dom.’ methods for personal best selection and global best selection have been chosen . For more details on guide selection in MOPSO reader is referred to [21, 23]. Maximum number of generations is set as the termination criterion.

NSGA-II: Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) is one of the most popularly used GA for multi-objective optimization. Several salient features like elite preservation and explicit diversity preserving mechanisms ensure its good convergence and diversity. Brief description of NSGA-II procedure is described here, for further details reader is referred to [8, 9]: Offspring population (size N) is created using parent population (size N) by usual genetic operators - selection, crossover and mutation. The created child population is combined with parent population, to form combined population of size 2N, and then a non-dominated sorting is carried out to classify the entire population into several non-dominated fronts. The new population (size N) is then filled by the members of combined population belonging to different non-dominated levels or starting from first level. Since all members of combined population cannot be accommodated in new population - several non-dominated fronts have to discarded. Since all members of last front entering the new population may not be accommodated, only few members (corresponding to number of available slots) are selected from the last front based on the crowding distance technique. Binary tournament selection, SBX, and polynomial mutation operators are used for NSGA-II.

4.3 Performance Comparisons

Due to stochastic nature of evolutionary approaches, it is difficult to conclude anything about performance from just one simulation run. To eliminate the random effects and gather results of statistical significance, we perform multiple (11) runs of both the algorithms corresponding to different initial seeds. Two performance measures commonly used in EA literature have been employed in this study are described as follows:

Attainment Surfaces: Multiple runs, corresponding to different initial seeds, of an evolutionary algorithm usually result in multiple non-dominated set. Thus, to deduce overall performance an approximation of best non-dominated set, also referred to as 1^{st} (0%) attainment surface, is computed from available non-dominated sets. Since non-dominated can be visualised easily in two and three dimensions, such a method provides good insight into algorithms performance. The computation of attainment surfaces is done by using attainment surface package described in [15].

Hypervolume indicator: Hypervolume is a measurement which takes into account the diversity as well as the convergence of the solutions [36]. Hypervolume represents the sum of the areas enclosed within the hypercubes formed by the points on the non-dominated front and a chosen reference point. For minimisation type problems a higher value of hypervolume is desirable, as it is indicative of better spread and convergence of solutions. Figure 6(a) illustrates the hypervolume computation for a set of non-dominated points with respect to a reference point ’R’. It should be noted that contribution to hypervolume is only made by points which are dominated by the reference point. All points not dominated by the reference point have zero contribution to the hypervolume. In this study we have computed average hypervolume curves over several generations for study and comparisons purposes. Although, hypervolume computation is dependent on choice of reference point, yet it is regarded as a good measure and can be employed for higher number of objectives as well.

4.4 Local Search Method

Typically for any practical multi-objective optimization problem location of true Pareto-optimal solutions is unknown. Although, MOEAs provide a good means to reach approximate or close to Pareto-optimal solutions, often further improvement on obtained solutions is possible by conducting *local search*. Local search usually considers an already found non-dominated solution and tries to improve it by utilising a construction of single objective function.

In this study we construct an achievement scalarizing function (ASF), a single objective function, and consider its minimization. Following describes ASF scheme:

Consider such a starting point **y** (having objective vector **f(y)** and **z**=**f(y)**), then ASF = :

$$\min_{x \in S \subset \mathbb{R}^n} \max_{i=1}^{M} \frac{f_i(x)-z_i}{f_i^{max}-f_i^{min}} + \rho \sum_{j=1}^{M} \frac{f_j(x)-z_j}{f_j^{max}-f_j^{min}}$$

Where **z=f(y)** is usually referred to as a reference point for local search, and f_i^{max} and f_i^{min} are minimum objective values of the 'best non-dominated' set. By this minimization, **ASFs** solutions are projected on the Pareto-front and convergence can be guaranteed.

Although various single objective optimization techniques could be applied for minimizing ASF, but due to discontinuous nature of objective functions gradient based methods are not applicable. We employed SQP (Sequential Quadratic Programming) based local search for this purpose and no improvement was found. To overcome this problem a mutation driven or hill climbing strategy, is proposed for conducting local search. Table 1 describes the hill climbing approach. To conduct local search a maximum number of trials (MaxTrials) are pre-set to limit the number of function evaluations. Then, with equal probability, problem variables θ_x and θ_y are perturbed according to Gaussian distribution (mean 0.0 and standard deviation σ_i). Standard deviation (σ_i) for Gaussian distribution is varied linearly from 10.0 to 1.0 over the iterations. Such a local search enables to explore wider regions in the starting and becomes more focussed towards the end. If ASF value at newly created orientations is lowered, then the perturbations in θ_x and θ_y are accepted. The whole procedure is continued till termination criteria is met. In this study MaxTrials is set at 1500.

Table 1: Mutation Driven Hill Climbing Local Search

```
For i = 1 to MaxTrials
  r = RandomDouble(0,1)
    If (r$\leq$0.5)
      $\theta_x^{New} = \theta_x + N(0,\sigma_i{}^2)$
    else
      $\theta_y^{New} = \theta_y + N(0,\sigma_i{}^2)$
    end
    If($ASF(\theta_x^{New},\theta_y^{New}) \leq ASF(\theta_x,\theta_y)$)
  Update($\theta_x$, $\theta_y$)
  Update($\sigma_i$)
    end
End
```

5 Decision Making

When a set of trade-off solutions is obtained from a multi-objective optimization exercise, a decision point needs to be chosen to proceed further. This is often a non-trivial task for an operator and certain guidelines are necessary. To address this task, we introduce three decision making techniques, namely - '*Reference Point Method*', '*Marginal Utility Method*' and '*L_2 Metric Method*'. The first method requires an 'aspiration point', described later, as an input from the user. However; remaining two methods do not require any user input to arrive at the decision choice. The description of these methods follows next.

Reference Point Method: Here it is assumed that the designer has some pre-decided preference (or aspiration) for an operating point with which he/she is likely to settle. The goal is to find a solution which is better than the aspiration of the designer. Thus, it is also called an aspiration point method. To carry out the search we allocate this aspiration point as the reference point for ASF scheme (described in section 4.4), and evaluate ASF for all points on the Pareto-optimal front. The Pareto-optimal solution which corresponds to lowest ASF value, with respect to reference point, is selected. For illustration purposes, following three aspiration points are considered in this study.

Asp_1= ($\frac{Ra_{min}+Ra_{max}}{2}$, $\frac{T_{min}+T_{max}}{2}$),

Asp_2= ($\frac{Ra_{min}+Ra_{max}}{2}$, T_{max}),

Asp_3= (Ra_{max}, $\frac{T_{min}+T_{max}}{2}$)

The corresponding decision choices obtained on the Pareto-front will be indicated as P_1, P_2 and P_3. Asp_1, for example, implies that user is willing to accept an available point in proximity of the mean of best and worst obtained (Ra, T) values. In case of convex Pareto-optimal decision choice dominates the aspiration point, whereas in case of concave set decision choice gets dominated by the aspiration point.

Marginal Utility Method: This approach also does not require any prior information from the user and searches for a Pareto-optimal solution which shows least affinity towards any of its neighbours in objective space. To compute the affinity towards the neighbourhood, consider three non-dominated points P_1, P_0 and P_2, s.t. ($Ra_1 \leq Ra_0 \leq Ra_2$) and ($T_1 \geq T_0 \geq T_2$) and we are interested in evaluating the affinity at the middle point P_0. P_1 and P_2 lie in the neighbourhood of P_0 and are selected as follows: consider k points, $P_{0,m}$ m = 1 to k, nearest to P_0, with $Ra_{0,m} \leq Ra_0$. Then centroid of all $P_{0,m}$s is computed and a point out of $P_{0,m}$s, which is closest to the centroid, is selected as P_1. For selecting P_2, same exercise is repeated, but this time considering points s.t. $Ra_{0,m}$s are greater than Ra_0.

Once P_1 and P_2 are computed for P_0, *affinity function* (AF), is calculated as :

AF_{P_0}=max(W1,W2); W1= $\frac{Ra_{P_0}-Ra_{P_1}}{T_{P_1}-T_{P_0}}$ and W2= $\frac{Ra_{P_2}-Ra_{P_0}}{T_{P_0}-T_{P_2}}$.

For each point in the non-dominated set, except for k extreme points at both ends, AF is computed and the solution with minimum AF is assigned as decision choice. This solution is argued to posses least affinity towards any of its neighbours. In this study value of k is taken equal to 6. The value of k decides the resolution of the proximity in which we are interested to compute the affinity function. Decision point point by this method is usually a ‘knee point’.‘Knee points’ are often of great practical importance as they denote a coordinate on Pareto-front where increase (decrease) in one objective is very large compared to decrease (or increase) in other objective.

L_2**-metric:** This is a straight-forward method to select one solution out of many non-dominated solutions without requiring any information from user. Firstly, each objective is normalised in [0.0, 1.0]. Then an *’ideal point’* is constructed, which is origin in case of normalised space, and taken as the reference point. Euclidean distance (L_2) of each point in the non-dominated set is calculated from the reference point and the solution with smallest Euclidean distance is finally selected.

6 Experiments

In this section a series of simulations are performed on various solid models to demonstrate the working of *MORPE*. The major goals of this study are:

1. Compare the performances of MOPSO and NSGA-II by computing hypervolume over generations, and draw conclusions on their convergence and diversity preservation characteristics.

2. Approximate the Pareto-optimal set by computing 1^{st} (0%) attainment surface from 11 runs of each

Table 2: Parameter Setting for Evolutionary Algorithms

General Parameters:
Population size 40
Generations 80
Runs 11

Other NSGA-II Parameters:
Crossover probability 0.9
Mutation probability 0.5
Crossover Index 10
Mutation Index 20

Other MOPSO Parameters:
Turbulence Factor 0.25
pBest Archive Size 3
Archive Size 200

MOPSO and NSGA-II. This provides means to draw several conclusions based on shape of Pareto-optimal set.

3. Fine tune the best joint non-dominated of MOPSO and NSGA-II by carrying out proposed*’Local Search’*, and find truly (or close to) Pareto-optimal solutions.

4. Analyse the trade-off solutions, while validating the overall procedure, and focus on extreme solutions to identify similarities.

5. Demonstrate the working of *’Decision Making Methods’* and highlight their significance in proposing the build orientations.

6. Draw out practical guidelines for designer through careful post-optimal analysis.

Firstly, basic geometrical solid models, like Cuboid, Cuboidal Pyramid, Prism and Pyramid are considered. The objective function evaluation is comparatively less intensive (computationally speaking) for these simple models as they are made up of flat and lesser number of surfaces. Results arrived here provide preliminary conclusions regarding MOPSO and NSGA-II performances, and also validate the working of *MORPE*. Next, more complicated solid models (with time consuming function evaluations) - Pentagon Bar, Cylinder, Pie, Diamond and Connector are considered for the bi-objective optimization. For this set of objects only a single run of MOPSO and NSGA-II is performed and the *’best joint non-dominated set is computed’* - non-dominated sets from MOPSO and NSGA-II are merged and non-dominated sorting is carried out to find non-dominated solutions in the combined set.

For all solid models along with the non-dominated sets, orientations corresponding to minimum *Ra* and minimum T are plotted. It should be noted that minimum *Ra* and minimum T solutions are arrived after searching the joint set of solutions obtained by MOPSO and NSGA-II. Decision choice based on L_2 *metric* for each solid object is also highlighted. Similarities amongst extreme solutions of different solid models provides valuable information to arrive at basic optimal building thumb rules. Several practical aspects and design considerations are addressed through careful analysis of trade-off fronts.

7 Results and Discussions

In general, it is difficult to predict an optimal build direction for any solid model. Major difficulty arises due to non-linear nature and non-differentiable expressions for surface roughness. This is the major reason for using an optimization procedure. For the minimum time orientations least number of layers are desired. Since layer thicknesses vary according to the adaptive slicing procedure described earlier, an orientation with minimum length in build direction may contain more layers as compared to some other direction. Hence the total number of slices depends on slice thicknesses which in turn solely depends on the local geometry of the solid model.

An orientation in which faces are inclined with respect to vertical requires a support structure and is likely to result in higher surface roughness. However, one should note that due to adaptive slicing procedure local surface roughness is limited and appearance of support structure is counteracted by thinner slices, thus increasing the number of slices. The thinner slices are also associated with smaller strip areas, and since local roughness is weighted with the strip areas the overall surface roughness tends to decrease. For an optimal solution these two opposing factors are balanced. This also explains why a trade-off exists between the build time and surface roughness.

Figures 6, 8, 9 and 10 show the hypervolume curves, 1^{st} (0 %) attainment surfaces of MOPSO and NSGA-II from 11 runs, extreme solutions corresponding to minimum T and minimum Ra orientations and L_2 metric based decision choice. For each shown orientation the decision variables and (Ra, T) are stated.

Based on hypervolume curves and attainment surfaces MOPSO performs better than NSGA-II, for Cuboid. In all hypervolume curves MOPSO shows fatser convergence behaviour in initial few generations, as hypervolume rises rapidly, but eventually settles at values lesser than NSGA-II hypervolume values, except for Cuboid where MOPSO performs better than NSGA-II both in terms of hypervolume and 1^{st} (0%) attainment surfaces. Hypervolume trends of MOPSO indicate *premature convergence* - a well known drawback in particle swarm optimization (PSO). According to the authors pre-mature convergence of MOPSO, in this case, highlights the absence of potential global guides due to discontinuities present in the objective space. Presence of discontinuity slows the march towards the true Pareto-optimal solutions. For Cuboidal Pyramid, Pyramid and Bipyramid NSGA-II clearly performs better as indicated by hypervolume curves. From attainment surfaces it can be observed that major regions of 1^{st} (0%) attainment surface of MOPSO are dominated by NSGA-II attainment surface. However; extreme solutions corresponding to minimum T or minimum Ra were sometimes found better by MOPSO.

For these four objects the minimum time orientation occurs when minimum dimension occurs along Z-direction (build orientation). Planar surfaces of in these solid objects result in flat orientations for minimum T. It can be argued that particularly in absence of curved surfaces slicing occurs with uniform thickness and minimum length dimension leads to least number of slices. Minimum Ra orientation is not easy to guess and occurs at titled configuration, requiring support structure.

For Cuboid, decision choice based on *'Reference Point Method'* and *'Marginal Utility'* are also shown in Figures 6(h) and 6(i), respectively. Three solutions are found by *'Aspiration Method'* corresponding to three aspiration points. *'Marginal Utility Method'* discovers the *knee point*, the point with least affinity. At any such *'knee point'* there is a large trade-off in one objective for marginal trade-off in other objective. According to Figure 6(i) at the *knee point* decrease in Ra is accompanied by a large increase in T. Thus, user would ideally like to work at the marked knee position. For all four solid models L_2 metric based configuration is also shown and lies spatially in between the two extreme configuration.

Overall it can be concluded that NSGA-II shows an acceptable better performance and application of another evolutionary optimizer MOPSO may be advantageous in providing some better solutions. The fact that attainment surfaces of MOPSO and NSGA-II have similar spread and distribution further builds our confidence on obtained solutions and their proximity to true Pareto optimal solutions.

Next, we consider more complicated solid models - Pentagon-Bar, Cylinder, Pie shape, Diamond and Connector in Figures 11-17. Since function evaluations for these solid models is computationally intensive, a single run of MOPSO and NSGA-II is considered. Non-dominated sets from NSGA-II and MOPSO runs are combined followed by global non-dominated sorting and *Pareto Front* solutions are shown. For various solid models NSGA-II solutions were found to dominate many MOPSO solutions.

As before, along with Pareto Front solutions L_2 metric based decision choice is plotted for all objects. For Pentagon-Bar the minimum T solution is almost horizontal (but not exactly flat) and minimum *Ra* is tilted. On further investigation it was found that for Pentagon-Bar slightly perturbed orientation from horizontal (minimum T solution), corresponds to least number of layers (31.0). While, perfectly horizontal configuration leads to slightly more number of layers (33.0). According to authors such a behaviour is shown due to adaptive slicing which results in more number of layers in perfectly horizontal configuration.

For Cylinder it was found that minimum T orientation was titled and required support structure. Whereas, minimum *Ra* orientation is a lying-flat position. Such extreme orientations are contrary to those obtained earlier; where minimum T orientation was flat and minimum *Ra* titled. It is believed that such nature of extreme orientation arises due to curved surface on Cylinder in conjunction with adaptive slicing.

For Pie minimum T is flat with minimum length along build direction. Minimum *Ra* orientation stands straight and requires support structure only in the lower half. It is clearly observable that in minimum *Ra* orientation strip areas of slices is less compared to horizontal flat. The overall effect leads to lowering of weighted surface roughness sum and thus yielding minimum *Ra* orientation.

For Diamond minimum T orientation is one where its conical curved surface area lies flat horizontally, which also leads to minimum dimension along build direction. Due to large number of faces in Diamond, variable slice thicknesses occur almost in any direction and thus only preferred direction is possibly one with minimum height along Z-axis. The minimum *Ra*occurs at titled orientation, with conical surface pointed upwards and requiring little support.

For Connector model, flat orientation corresponds to minimum T and minimum *Ra* is slightly tilted from vertical. It is observed trend that for thin models minimum T orientation occurs with smallest dimension along build direction. It should be noted that contribution of holed spaces to surfaces roughness is negligible compared to outer area surface roughness.

For Prism (Figure 17(a)) and Sharp (Figure 17(b)) both MOPSO and NSGA-II were applied and trade-off solutions were discovered. However; based on manual neighbourhood local search, solutions (shown in these figures) were found and these solutions dominated the obtained non-dominated set. Thus, for these two objects in reality there exist one solution which is minimum in *Ra* and T both. Due to discontinuity in objective space MOEAs failed to identify such solution in given number of function evaluations. Thus, it is recommended that often manual investigation and local-search should be carried out to fine-tune solutions which are often predictable.

8 Conclusions

In this paper, a systematic approach has been presented to derive the optimal build orientations, simultaneously minimizing surface roughness *Ra* and build time T, for FDM process. To address the multi-objective optimization task two popularly used evolutionary approaches Non-dominated Sorting Genetic Algorithm *NSGA-II* and Multi-objective Particle Swarm Optimization *'MOPSO'* have been applied. The performance comparisons of these two optimizers is carried out by evaluating *'hypervolume'* metric and NSGA-II was found to perform better. Attainment surfaces are also computed to provide an approximation of Pareto Optimal set. Employment of two optimizers is found useful in identifying overall best non-dominated set, particularly extreme solutions which were often found by only one optimizer. To further refine the non-dominated solutions obtained from MOEAs a mutation driven hill climbing strategy based on *'Achievement Scalarizing Function'* is proposed to conduct local search. Local search is found effective in bringing solutions closer to true Pareto-optimal solutions. Three methods addressing the issue of *'Decision Making'* are introduced to aid the user in choosing a solution once Pareto-optimal solutions are found. Post-optimal analysis of several objects considered in this study indicates a trend, particularly amongst the extreme solutions found. Thus, objects can be categorised into families based on resemblance in optimal build directions.

9 Acknowledgements

First author sincerely thanks Dr. N.V. Reddy and Dr. P.M. Pandey for introducing the problem and showing consistent support.

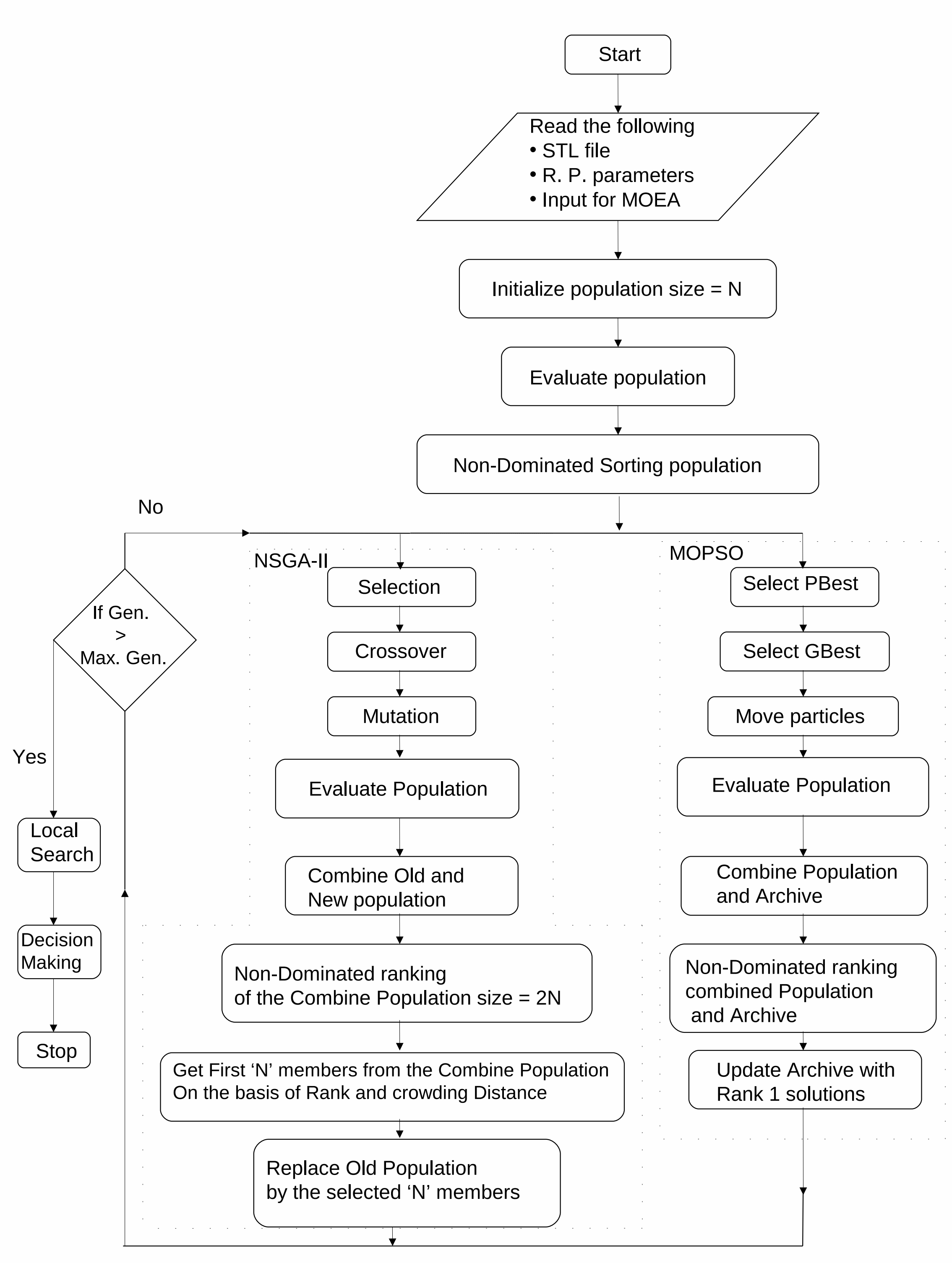

Start
Read the following
• STL file
• R. P. parameters
• Input for MOEA
Initialize population size = N
Evaluate population
Non-Dominated Sorting population
No
NSGA-II
Selection
Crossover
Mutation
Evaluate Population
Combine Old and New population
Non-Dominated ranking of the Combine Population size = 2N
Get First 'N' members from the Combine Population On the basis of Rank and crowding Distance
Replace Old Population by the selected 'N' members
MOPSO
Select PBest
Select GBest
Move particles
Evaluate Population
Combine Population and Archive
Non-Dominated ranking combined Population and Archive
Update Archive with Rank 1 solutions
If Gen. > Max. Gen.
Yes
Local Search
Decision Making
Stop

Figure 3: Flowchart suggesting the working of developed engine

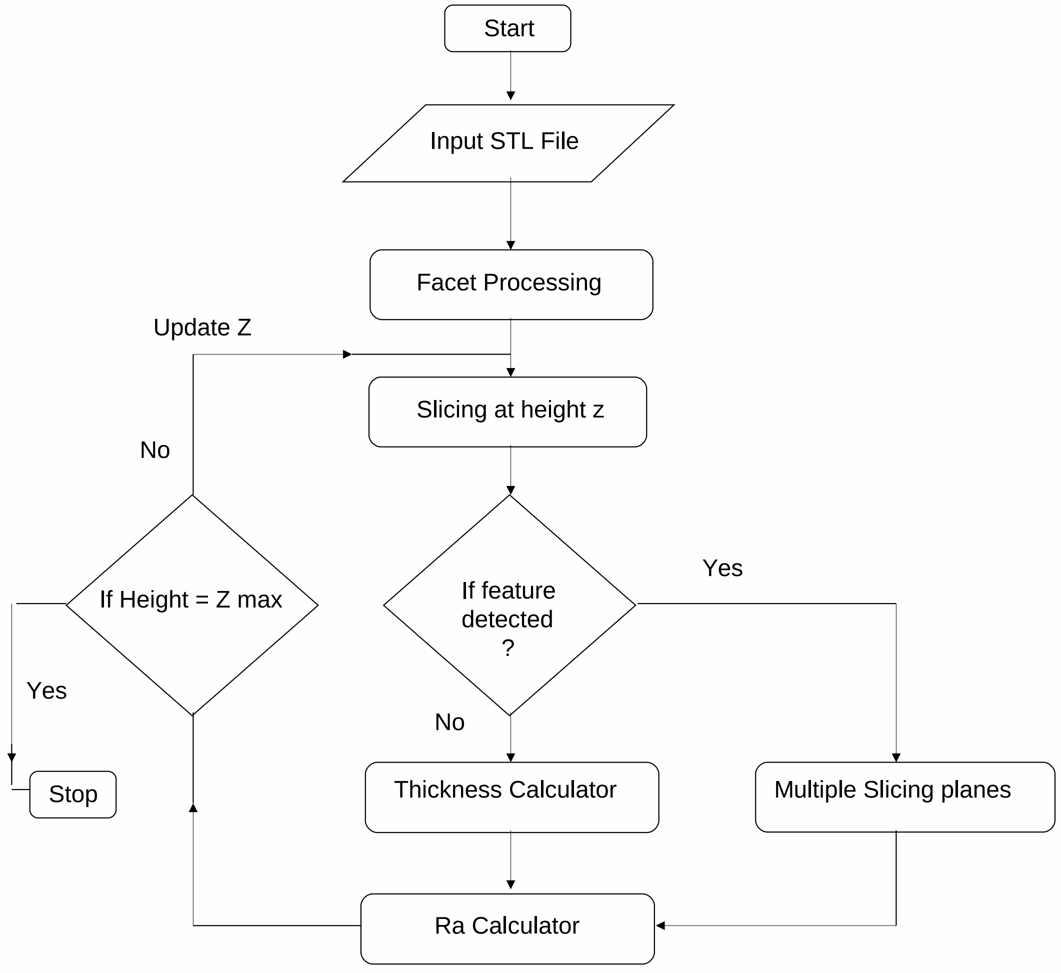


Figure 4: Flowchart for Slicing procedure

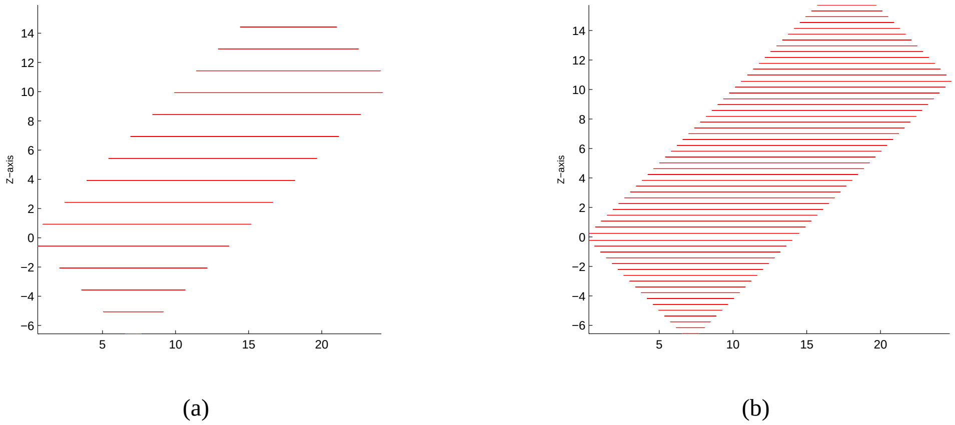
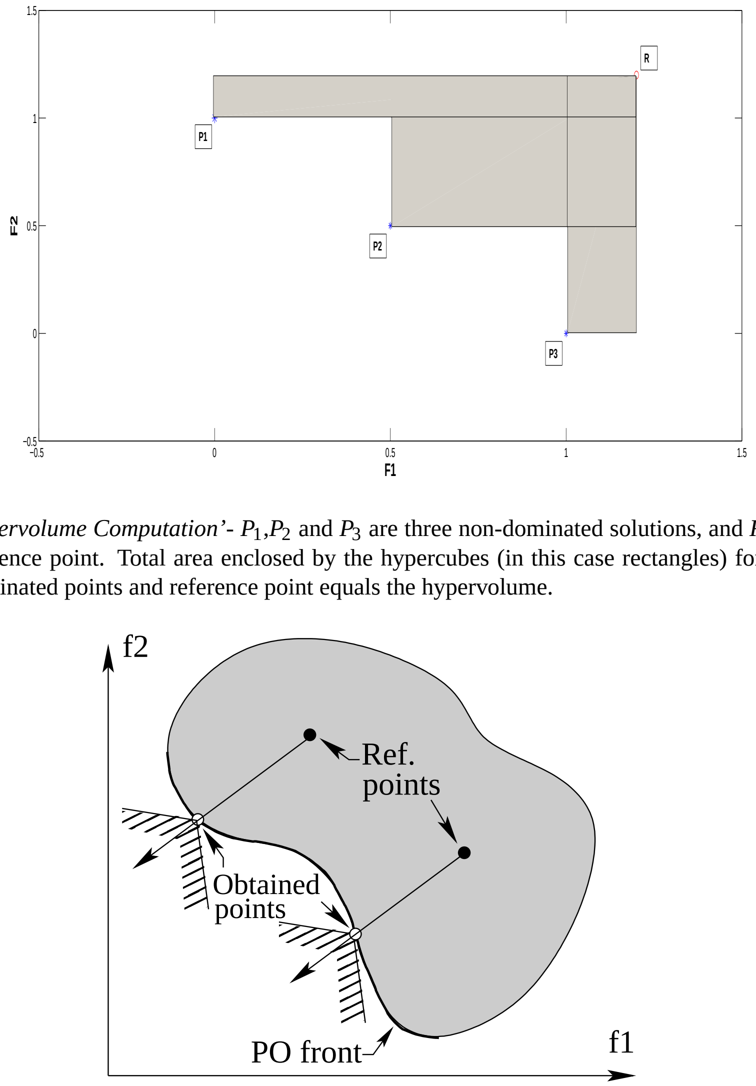
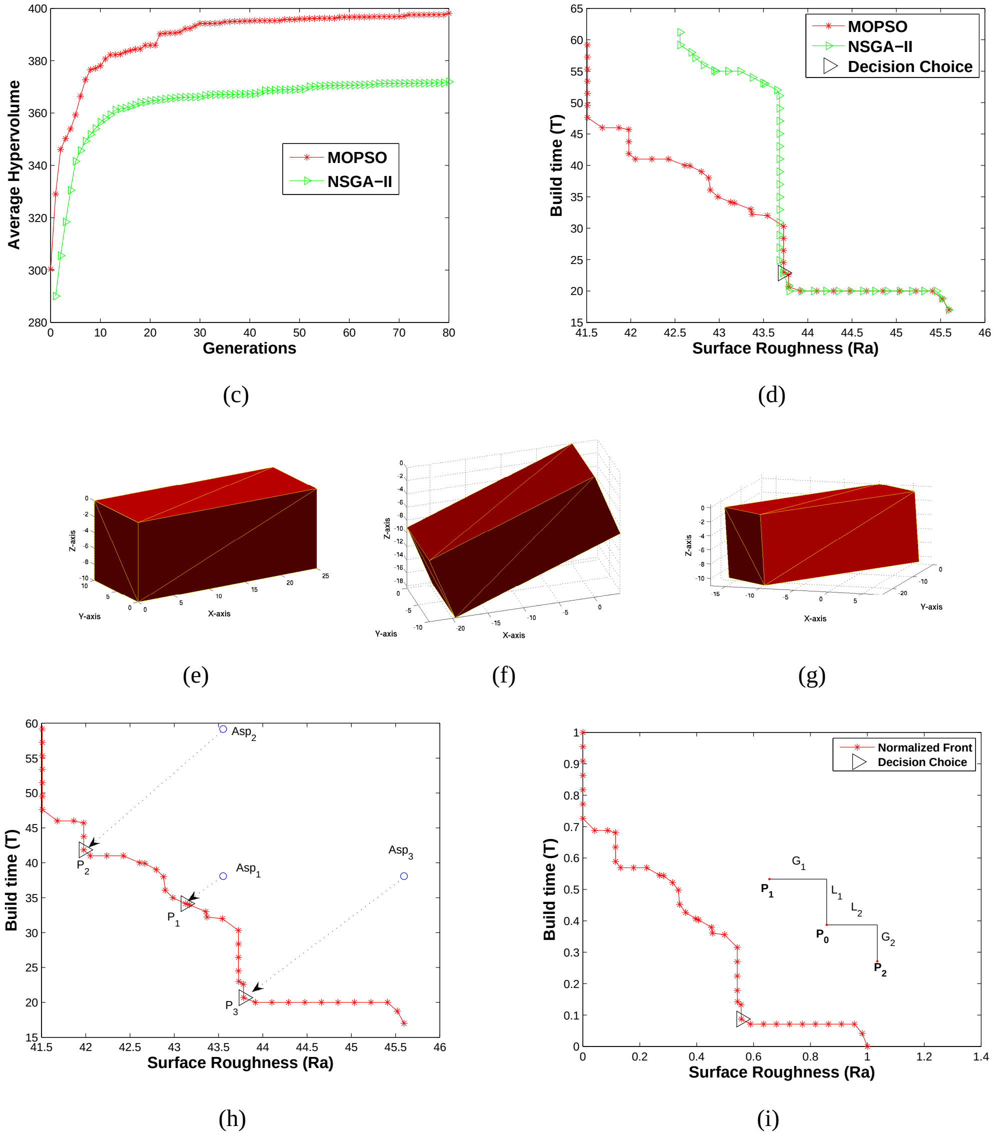


Figure 5: (a) Constant Height Slicing Procedure. $\overline{Ra}$ for object turns out to be 61.67 units. (b) Ra Adaptive Slicing Procedure. $\overline{Ra}$ for object turns out to be 45.12 units.



(a) *'Hypervolume Computation'*- P_1,P_2 and P_3 are three non-dominated solutions, and R is chosen reference point. Total area enclosed by the hypercubes (in this case rectangles) formed by non-dominated points and reference point equals the hypervolume.

(b) Achievement Scalarization based *Local Search*



(c)

(d)

(e)

(f)

(g)

(h)

(i)

Figure 6: (a) Average hypervolume curves for Cuboid with reference point (50, 75.0). (b) 1st Attainment surface for Cuboid. (c) Min. T orientation for Cuboid (θ_x,θ_y)= (0.0°, 90.0°), (Ra, T)= (45.58, 17.0). (d) Min. Ra orientation for Cuboid (θ_x,θ_y)= (180.0°, 68.62°), (Ra, T)= (41.51, 46.0). (e) L_2-metric Decision choice orientation for Cuboid (θ_x,θ_y)= (133.4°, 86.4°), (Ra, T)= (43.725, 23.0). (f) ‘Aspiration Point Method’ based on ’*ASF*’. (g) ‘Marginal Utility Method’ in Normalised Space.

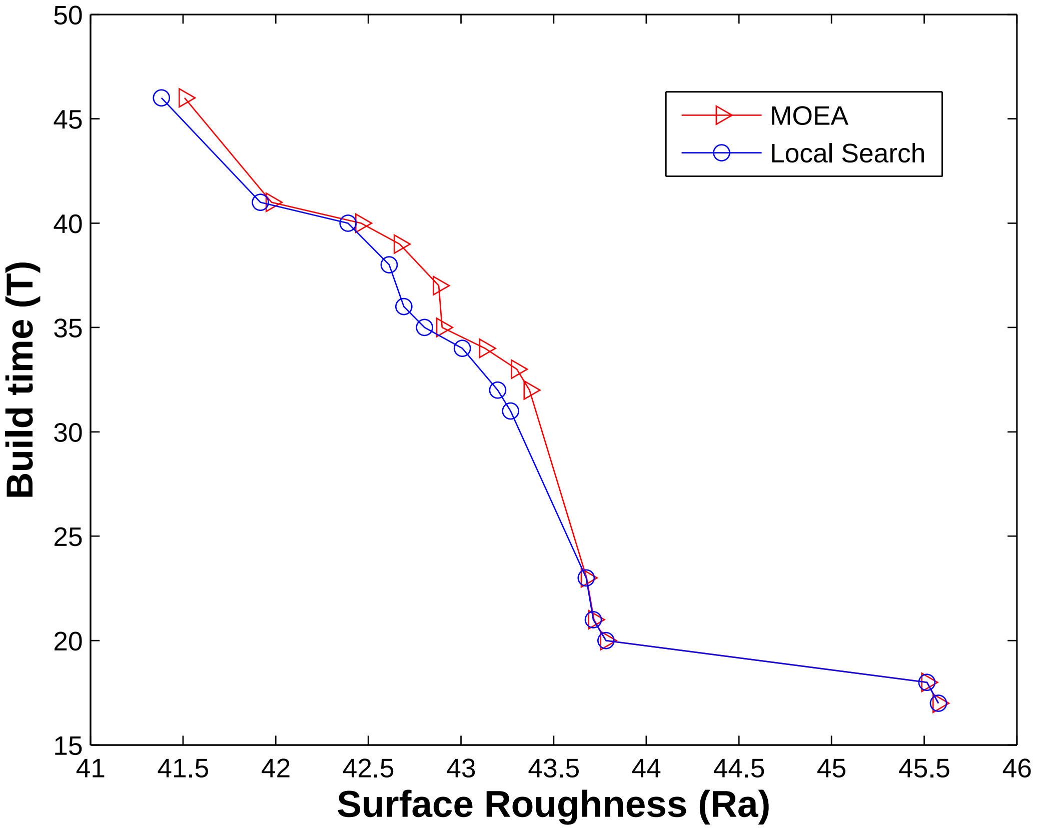
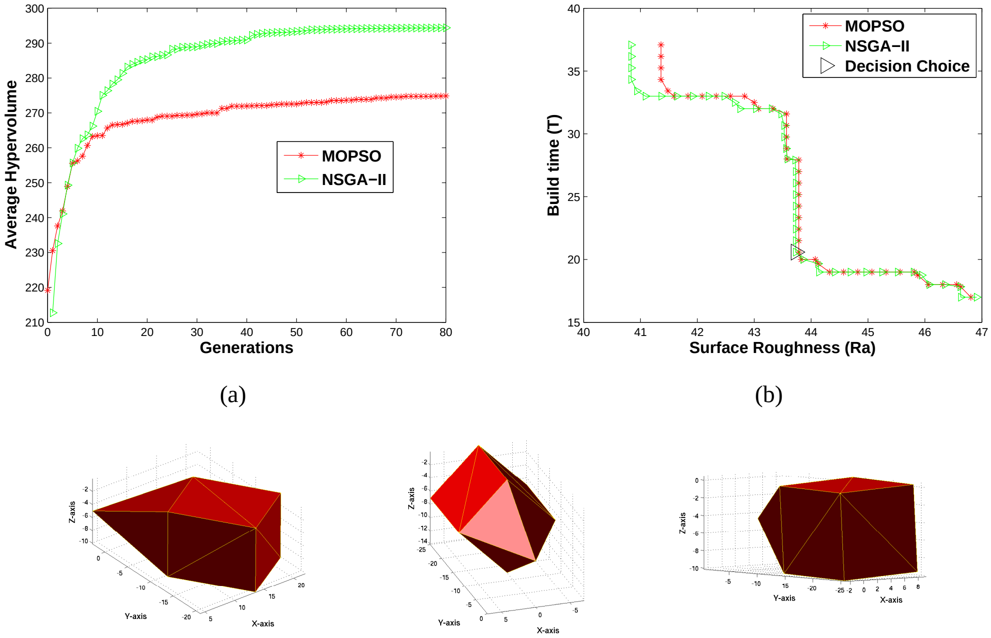


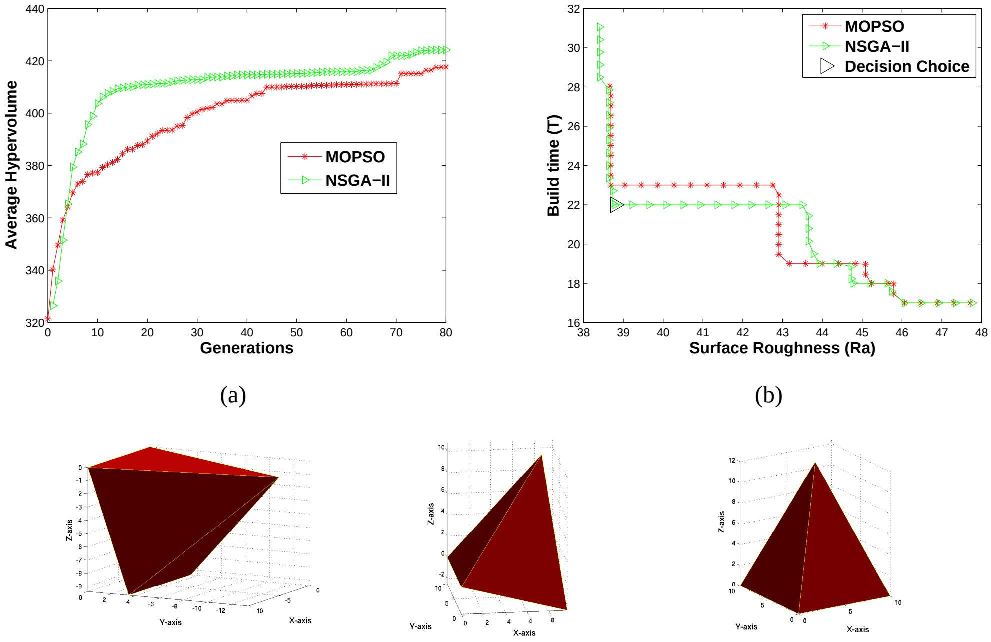
Figure 7: ‘Hill Climbing Local Search’ using ASF



(a) (b)

(c) (d) (e)

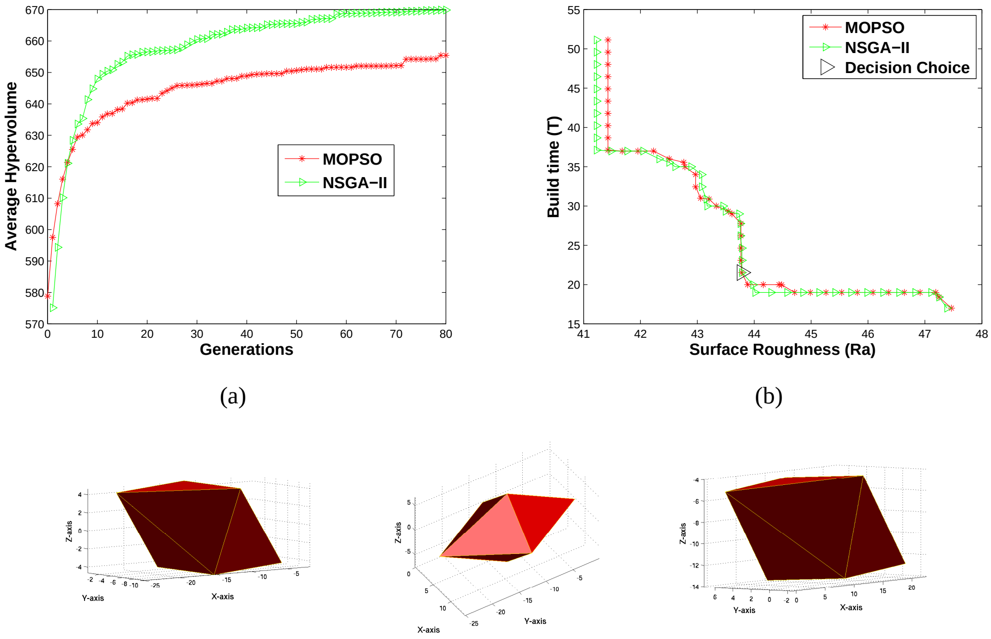
Figure 8: (a) Average hypervolume curves for Cuboidal-Pyramid with reference point (50.0, 55.0). (b) 1st Attainment Surface for Cuboidal-Pyramid. (c) Min. T orientation for Cuboidal-Pyramid (θ_x,θ_y)= (57.07°, 90.36°), (Ra, T)= (46.78, 17.0). (d) Min. Ra orientation for Cuboidal-Pyramid (θ_x,θ_y)= (90.0°, 135.029°), (Ra, T)= (40.84, 34.0). (e) L_2-metric Decision Choice orientation Cuboidal Pyramid for (θ_x,θ_y)= (96.1°, 86.43°), (Ra, T)= (43.725, 20.0).



(a) (b)

(c) (d) (e)

Figure 9: (a) Average hypervolume curves Pyramid with reference point (55.0,45.0). (b) 1st Attainment Surface for Pyramid. (c) Min. T orientation for Pyramid (θ_x, θ_y)= (110.0°, 180.0°), (Ra, T)= (47.68, 17.0). (d) Min. Ra orientation for Pyramid (θ_x, θ_y)= (0.0°, 14.64°), (Ra, T)= (38.40, 28.0). (e) L_2-metric Decision Choice orientation for Pyramid (θ_x, θ_y)= (0.0°, 0.0°), (Ra, T)= (38.8, 22.0).



(a)

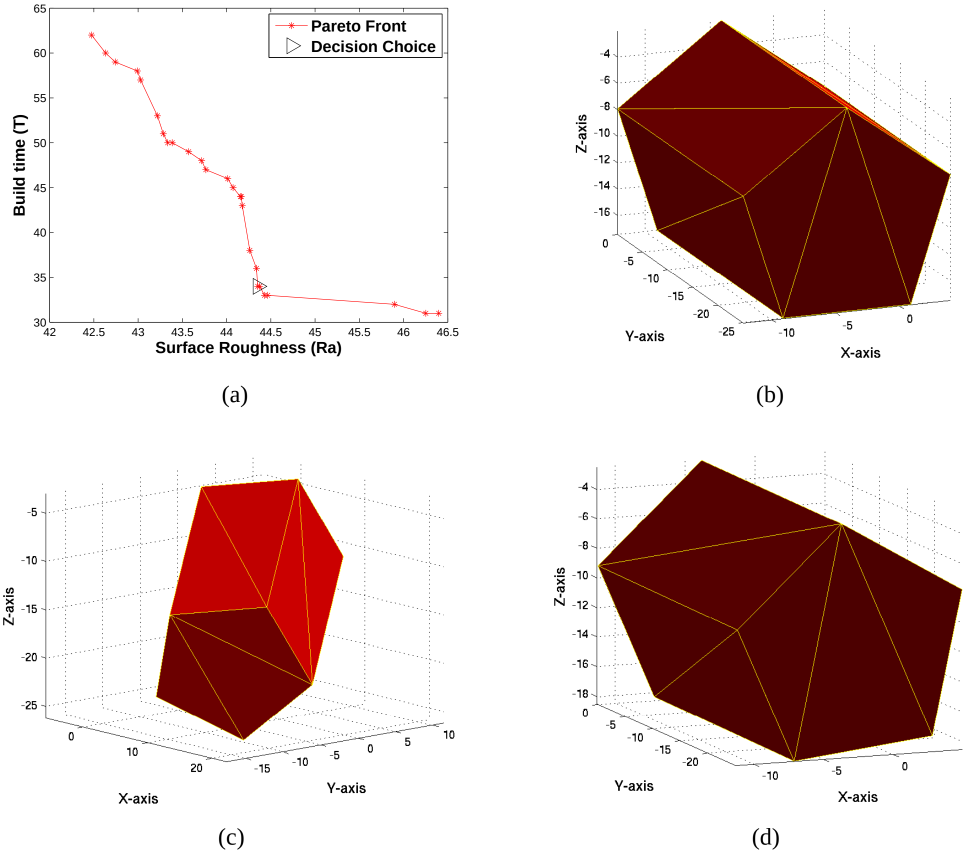
(b)

(c)

(d)

(e)

Figure 10: (a) Average Hypervolume Curves Bi-Pyramid with reference point (55.0, 70.0). (b) 1_{st} Attainment Surface for Bi-Pyramid. (c) Min. T orientation for Bi-Pyramid (θ_x, θ_y)= (179.98°, 111.79°), (Ra, T)= (47.36, 17.0). (d) Min. Ra orientation for Bi-Pyramid (θ_x, θ_y)= (91.25°, 46.39°), (Ra, T)= (41.23,37.0). (e) L_2-metric Decision Choice orientation for Bipyramid (θ_x, θ_y)= (11.93°, 108.78°), (Ra, T)= (43.8, 21.0).



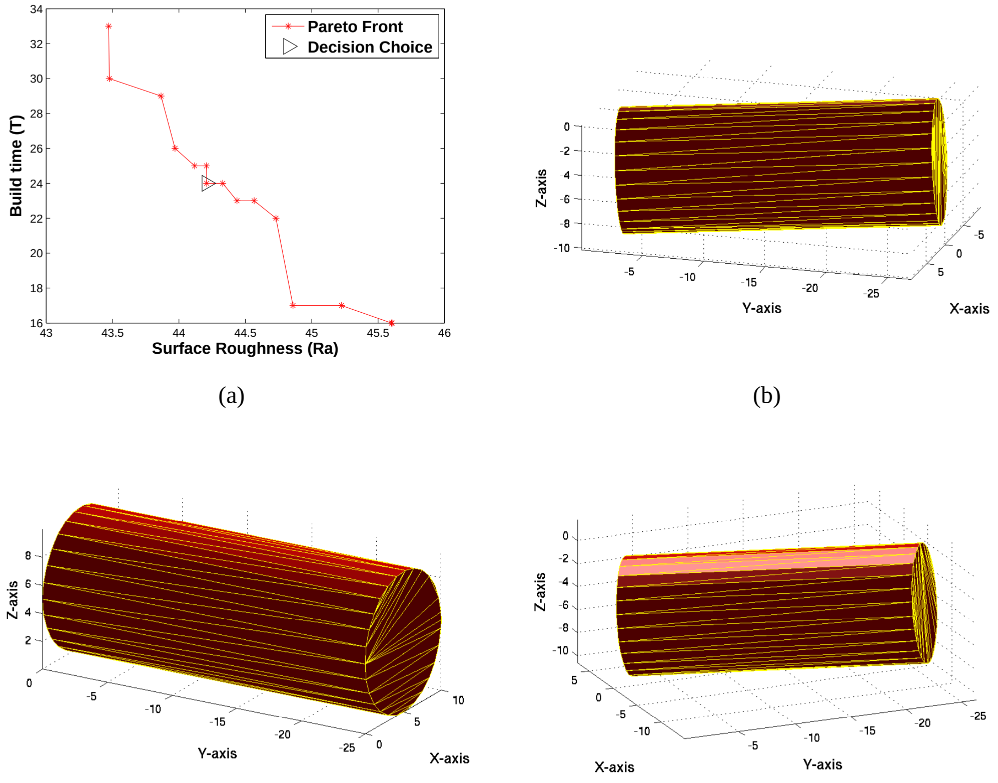
(a)

(b)

(c)

(d)

Figure 11: (a) Trade-off front for Pentagon-Bar. (b) Min. T orientation Pentagon-Bar (θ_x, θ_y)= (90.1°, 153.5°), (Ra, T)= (46.25, 31.0). (c) Min. Ra orientation (θ_x, θ_y)= (44.7°, 117.89°), (Ra, T)= (42.47, 62.0). (d) L_2-metric decision choice orientation for Pentagon Bar (θ_x, θ_y)= (89.34°, 147.46°), (Ra, T)=(44.35, 35.0).



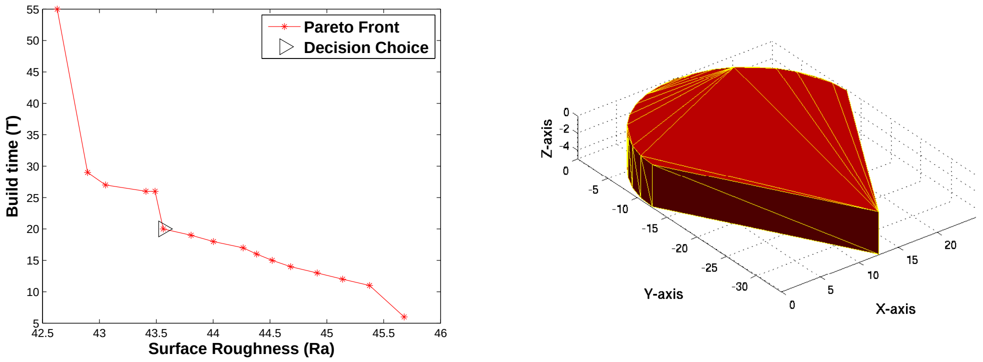
(a)

(b)

(c)

(d)

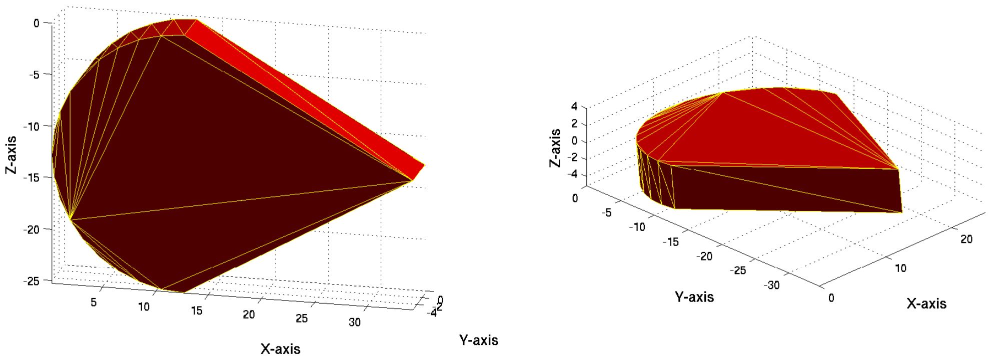
Figure 12: (a) Trade-off front for Cylinder. (b) Min. T orientation Cylinder (θ_x, θ_y)= (113.43°, 88.27°), (Ra, T)= (45.6, 16.0). (c) Min. Ra orientation (θ_x, θ_y)= (90.0°, 0.0°), (Ra, T)= (43.31,19.0). (d) L_2-metric decision choice orientation for Cylinder (θ_x, θ_y)= (124.49°, 100.5°), (Ra, T)= (44.25, 24.0).



(a)

(b)

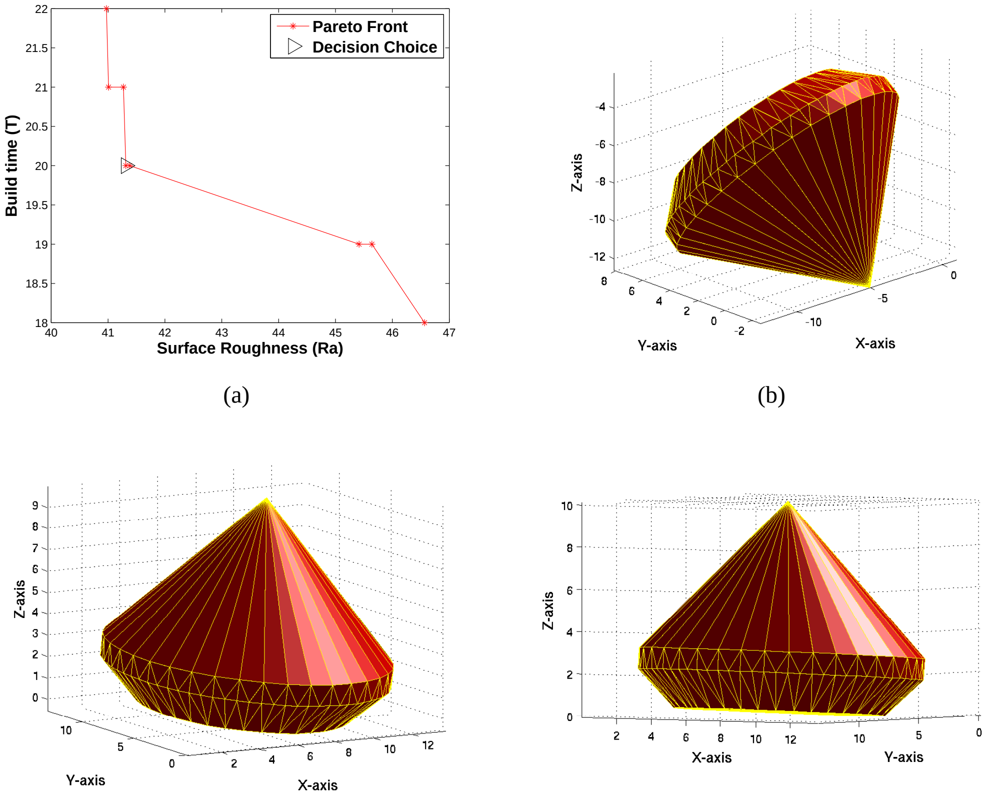
Figure 13: (a) Trade-off front for Pie. (b) Min. T orientation for Pie (θ_x, θ_y)= (180.0°, 0.0°), (Ra, T)= (45.8, 6.0).



(a)

(b)

Figure 14: (a) Min. *Ra* orientation for Pie (θ_x, θ_y)= (90.0°, 90.0°), (*Ra*, *T*)= (42.39, 54.0). (b) L_2-metric decision choice orientation for Pie (θ_x, θ_y)= (172.8°, 0.72°), (*Ra*, *T*)= (43.51, 20.0).



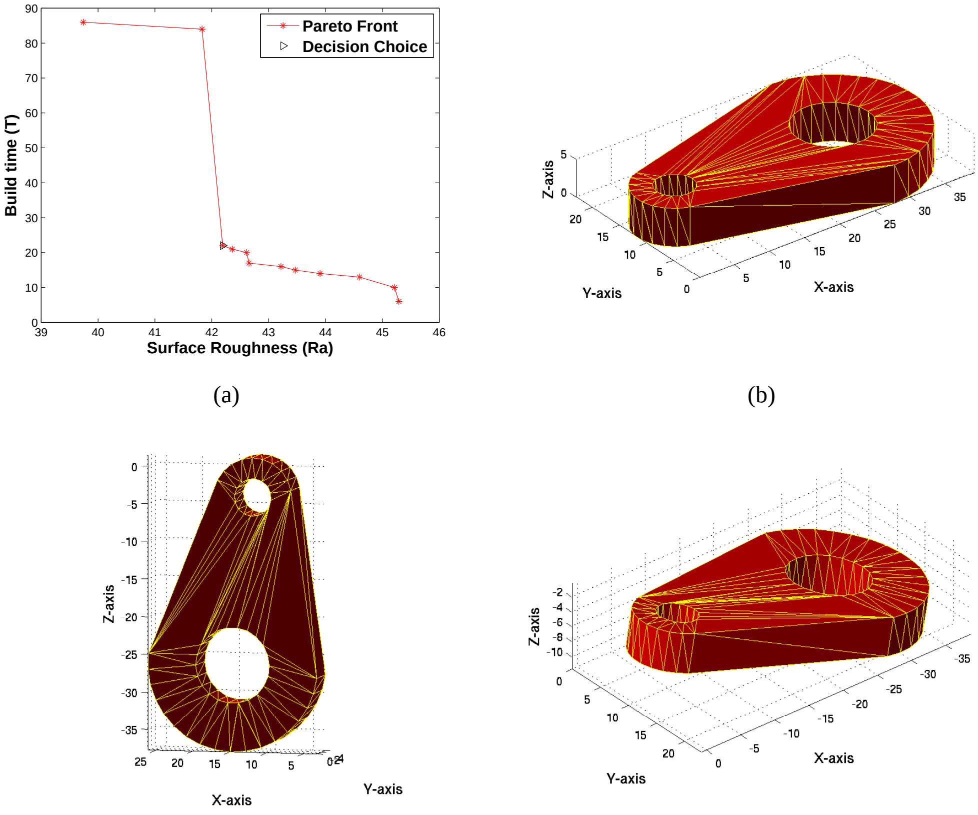
(a)

(b)

(c)

(d)

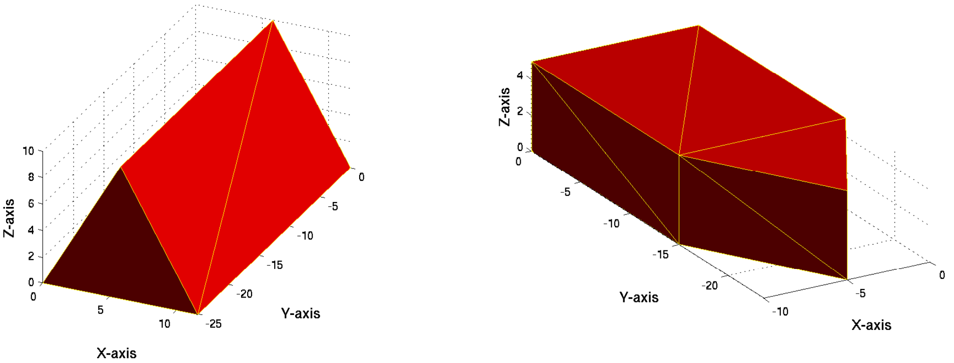
Figure 15: (a) Trade-off front for Diamond. (b) Min. *T* orientation for Diamond (θ_x, θ_y)= (46.41°, 171.82°), (*Ra*, *T*)= (46.56, 18.0). (c) Min. *Ra* orientation for Diamond (θ_x, θ_y)= (5.45°, 4.8°), (*Ra*, *T*)= (40.97, 22.0). (d) L_2-metric decision choice orientation for Diamond (θ_x, θ_y)= (2.40°, 1.12°), (*Ra*, *T*)=(41.3,20.0,).



(a) (b)

(c) (d)

Figure 16: (a) Trade-off front for Connector. (b) Min. T orientation for Connector(θ_x, θ_y)= (.17°, .106°), (Ra, T)= (45.23899, 6.0). (c) Min. Ra orientation for Connector (θ_x, θ_y)= (90.0°, 83.0°), (Ra, T)=(39.74, 86.0). (d) L_2-metric Decision Choice orientation for Connector (θ_x,θ_y)= (0.44°,170.28°), (Ra, T)=(42.20, 20.0). (e) L_2-metric decision choice orientation for Connector (θ_x,θ_y)= (0.44°,170.28°), (Ra, T)=(42.20, 20.0)



(a) (b)

Figure 17: (a) Min. T and Ra orientation for Prism (θ_x, θ_y)= (90.0°, 0.0°), (Ra, T)= (37.69, 20.0). (b) Min. T and Ra orientation for Sharp (θ_x, θ_y)= (180.0°, 180.0°), (Ra, T)=(40.36, 7.000).

References

[1] Daekeon Ahn, Hochan Kim, and Seokhee Lee. Fabrication direction optimization to minimize post-machining in layered manufacturing. *International Journal of Machine Tools and Manufacture*, 47(3-4):593 – 606, 2007.

[2] P. Alexander, S. Allen, and D. Dutta. Part orientation and build cost determination in layered manufacturing. *Computer Aided Design*, 30:343–356, 1998.

[3] M. Bagchi and A. Bagchi. Quantification of errors in rapid prototyoing processes and determination of preferred orientation of parts. *Transactions of the North American Manufacturing Research Institution/SME*, pages 319–323, 1995.

[4] Hong-Seok Byun and Kwan H. Lee. Determination of optimal build direction in rapid prototyping with variable slicing. *Internation Journal of Advanced Manufacturing Technology*, 28:307–313, 2006.

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

[6] V. Canellidis, J. Giannatsis, and V. Dedoussis. Genetic-algorithm-based multi-objective optimization of the build orientation in stereolithography. *The International Journal of Advanced Manufacturing Technology*, 2009.

[7] W. Cheng, J. Y. H. Fuh, , A. Y. C. Nee, Y. S. Wong, H. T. Loh, and T. Miyazawa. Multi-objective optimization of part-building orientation in stereolithography. *Rapid Prototyping Journal*, 1:22–33, 1995.

[8] K. Deb. *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons, Dordrecht, 2001.

[9] K. Deb, S. Agarwal, and T. Meyarvian. A fast and elitist multi-objective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[10] D.T. Dimov D.T. Pham and R.S. Gault. Part orientation in stereolithography. *International Journal of Advanced Manufacturing Technology*, 15:674–682, 1999.

[11] J. Hong, W. Wang, and Y. Tang. Part building orientation optimization method in stereolithography. *Chinese Journal of Mechanical Engineering (English Edition)*, 19(1):14–18, 2006.

[12] J. Hur and K. Lee. The development of a cad environment to determine the preferred build-up direction for layered manufacturing. *International Journal of Advanced Manufacturing Technology*, 14:247–254, 1998.

[13] S.M. Hur, K.H. Choi, S.H. Lee, and P.K. Chang. Determination of fabricating orientation and packing in sls process. *Material Process Technology*, 112(2-3):236–243, 2001.

[14] H.C. Kim and S.H. Lee. Reduction of post-processing for stereolithography systems by fabrication-direction optimization. *Computer Aided Design*, 37(7):711–725, 2005.

[15] J. Knowles. A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. In *IEEE Intelligent Systems Design and Applications (ISDA V)*, pages 552–557, 2005.

[16] P.-T. Lan, S.-Y. Chow, L.-L. Chen, and D. Gemmill. Determining fabrication orientations for rapid prototyping with stereolithography apparatus. *Computer Aided Design*, 29:53–62, 1997.

[17] J. A. Leitao, R. Everson, N. Sewell, and M. Jenkins. Multi-objective optimal positioning and packing for layered manufacturing. In *Proceedings of the 3rd International Conference on Advanced Research in Virtual and Rapid Prototyping: Virtual and Rapid Manufacturing Advanced Research Virtual and Rapid Prototyping*, pages 655–660, 2008.

[18] J. Majhi, R. Janardan, M. Smid, and P. Gupta. On some geometric optimization problems in layered manufacturing. *Computational Geometry*, 12(3-4):219–239, 1999.

[19] S. H. Masood and W. Rattanawong. A generic part orientation system based on volumetric error in rapid prototyping. *International Journal of Advanced Manufacturing Technology*, 19(3):209–216, 2000.

[20] S. H. Masood, W. Rattanawong, and P. Iovenitti. Part build orientations based on volumetric error in fused deposition modeling. *International Journal of Advanced Manufacturing Technology*, 19:162–168, 2000.

[21] N. Padhye. Comparision of archiving methods in mopso: Empirical study. In *GECCO ’09: Proceedings of the 2009 GECCO conference companion on Genetic and evolutionary computation*.

[22] N. Padhye. Topology optimization of compliant mechanism using multi-objective particle swarm optimization. In *GECCO ’08: Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation*, pages 1831–1834.

[23] N. Padhye, J. Juergen, and S. Mostaghim. Empirical comparison of mopso methods - guide selection and diversity preservation. In *Proceedings of Congress on Evolutionary Computation (CEC)*. IEEE, 2009.

[24] N. Padhye and S. Kalia. Rapid prototyping using evolutionary algorithms: Part 1. In *GECCO ’09: Proceedings of the 2009 GECCO conference companion on Genetic and evolutionary computation*.

[25] N. Padhye and S. Kalia. Rapid prototyping using evolutionary algorithms: Part 2. In *GECCO ’09: Proceedings of the 2009 GECCO conference companion on Genetic and evolutionary computation*, pages 2737–2740.

[26] P. M. Pandey, K. Thrimurthulu, and N. V. Reddy. Optimal part deposition orientation in fdm by using a multicriteria. *International Journal of Production Research*, 42 (19):4069–4089, 2004.

[27] P.M. Pandey, N. Venkata Reddy, and S.G. Dhande. Part deposition orientation studies in layered manufacturing. *Journal of Materials Processing Technology*, 185:125–131, 2007.

[28] K. Tata, G. Fadel, A. Bagchi, and N. Aziz. Efficient slicing for layered manufacturing. *Rapid Prototyping Journal*, 4(4):151–167, 1998.

[29] D.C. Thompson and R.H. Crawford. Optimizing part quality with orientation. In *Proceedings of the 6th SFF Symposium*, pages 362–368, 1995.

[30] K. Thrimurthuu, P.M. Pandey, and N. V. Reddy. Optimum part deposition orientation in fused deposition modeling. *Optimum part deposition orientation in fused deposition modeling. International Journal of Machine Tools and Manufacture*, 4:585–594, 2004.

[31] F. Xu, S.Y. Wong, T.H. Loh, HYJ Fuh, and T Miyazawa. Optimal orientation with variable slicing in stereolithography. *Rapid Prototyping Journal*, 3(3):76–88, 1997.

[32] A.B. Yew, C.C. Kai, and D. Zhaohui. Development of an advisory system for trapped material in rapid prototyping parts. *International Journal of Advanced Manufacturing Technology*, 16:733–738, 2000.

[33] L. Q. Zhang, D.-H. Xiang, M. Chen, and B.-X. Wang. Optimum design for rp deposition orientation by genetic algorithm. *Nanjing Hangkong Hangtian Daxue Xuebao/Journal of Nanjing University of Aeronautics and Astronautics 37 (SUPPL.)*, pages 134–136, 2005.

[34] J. Zhao. Determination of optimal build orientation based on satisfactory degree theory for rpt. In *Proceedings - Ninth International Conference on Computer Aided Design and Computer Graphics, CAD/CG 2005 2005, art. no. 1604640*, pages 225–230.

[35] J. Zhao, L. He, W. Liu, and H. Bian. Optimization of part-building orientation for rapid prototyping manufacturing. *Journal of Computer-Aided Design and Computer Graphics*, 18(3):456–463, 2006.

[36] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, ETH Zurich, Switzerland, 1999.