Blackbox optimization and surrogate models for machining free-form surfaces

Mahfoud Herraz, Jean-Max Redonnet, Mohammed Sbihi, Marcel Mongeau

Blackbox optimization and surrogate models for machining free-form surfaces

Mahfoud Herraz[a], Jean-Max Redonnet[a], Mohammed Sbihi[b] and Marcel Mongeau[b]

[a] Université Paul Sabatier and Institut Clément Ader, Université de Toulouse, France
[b] ENAC, Université de Toulouse, France

Computers & Industrial Engineering, vol. 177, pp 109029, march 2023
DOI:10.1016/j.cie.2023.109029

Abstract

This paper introduces an optimization model for machining free-form surfaces. It involves one categorical decision variable and continuous decision variables. Its objective function is partially separable. It is composed of two blackboxes: a clustering blackbox that outputs a partition of the surface into K zones, and K independent blackboxes, each of which outputs a machining time for a zone. This blackbox optimization problem is solved with the state-of-the-art software NOMAD. In order to improve the performance of the optimization process, we propose several surrogates of the machining-time blackboxes. Some of these surrogates are simple numerical approximations of the machining time, while one proposed surrogate is analytical, cheap to evaluate and exact for zones that are rectangles. Numerical experiments on two benchmark test surfaces show that our methodology outperforms other approaches from the literature. Although performances are strongly dependent on the topology of the test surfaces, the gains in machining time can go up to 40%.

Keywords: blackbox optimization; clustering; surrogate model; free-form surfaces; principal component analysis; toolpath planning

Contents

1 Introduction

Milling is an industrial manufacturing process where a rotating cutter is moved into a work piece and removes matter on its way. As a result, the envelope of all the successive positions of the cutter is removed from the initial part. When the tip of the cutter is the working area of the cutter, this process is called end-milling.

In the field of manufacturing, all non prismatic nor revolution surfaces are called *free-form surfaces*. To manufacture a part defined by free-form surfaces, a computer-driven machine, *i.e.* a CNC (Computer Numerical Control) machine, is needed. The CNC machines commonly encountered in industry are 3- or 5-axis machines, according to the number of degrees of freedom they provide. On a 3-axis machine, only pure translation of the cutter is allowed, while 5-axis machines permit some rotations of the

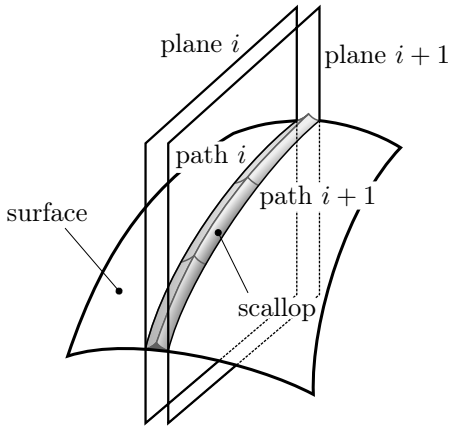
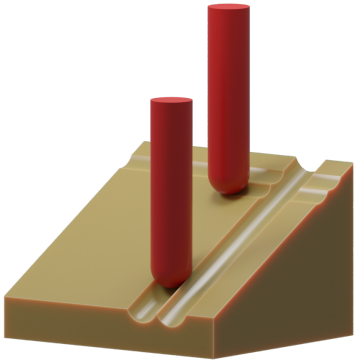
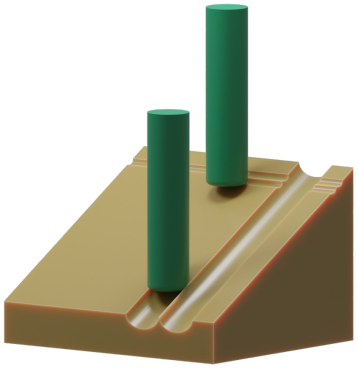


Figure 1: The parallel-plane strategy.



(a) ball-end cutter



(b) toroidal cutter

Figure 2: Ball-end cutter *vs* toroidal cutter.

cutter. In the context of free-form surfaces machining, so-called 3+2-axis machining (*i.e.* 3-axis machining plus two rotation axes for positioning the part between two milling phases) is often preferred to 5-axis machining. Indeed, rotation axes dynamics often bring out slowdowns when continuously used [1]. This paper is therefore dedicated to both 3-axis machining and 3+2-axis machining.

From a purely geometric point of view, the key point is: whatever the path the cutter is taking, it cannot hug perfectly the definition surface at each point. The remaining matter, called scallop, is removed later, often manually (see Figure 1). Actually, the scallop height is the quality criterion of the whole operation; while, for productivity reasons, the objective is to machine the surface as fast as possible. Yet, because of machinability considerations, the speed of the cutter must be fixed to an acceptable value. The only way to decrease machining time is therefore to improve the cutter trajectories. This is the toolpath planning phase, and it is a wide topic of research in the manufacturing field.

End-milling of free-form surfaces on multi-axis CNC machines is complex, and it involves expensive operations in the production of many high-value parts, such as molds and stamping dies. Currently, in the industry, these operations are still carried out using reliable and proven, but far from optimal, toolpath planning strategies. The parallel-plane strategy (Figure 1) is the most commonly adopted strategy for industrial manufacturing, as it ensures that the entire surface is covered and it is easy to implement.

Using this strategy, toolpaths are defined by the intersection of the surface to be machined with a set of parallel planes covering the whole surface. Practically, each toolpath is a sequence of interpolation points calculated using surface-surface intersection algorithms, such as those described in [2].

Ball-end mills, *i.e.* cutters whose tip is hemispherical, are the most widely used cutters in industry. Many sources ([3] for instance) point out that toroidal cutters, *i.e.* cutters whose tip is rounded as a quarter torus, may however provide better results when machining along the steepest-slope direction. Unfortunately, they also can provide worse results when machining along a direction perpendicular to the steepest-slope direction. Figure 2 illustrates this point on a plane for the sake of clarity, but the same applies on a free-form surface (on this figure the same depth of cut is used for both ball-end and toroidal machinings along a given direction).

Since the steepest-slope direction may vary much across a free-form surface, partitioning this surface into zones (each of which will then be machined along an appropriate direction) is a good approach to improve efficiency of the toroidal cutter.

The present paper deals with optimization of toroidal cutter tool path planning in 3-axis and 3+2-axis machining contexts, using a parallel-plane strategy applied to dedicated zones on a free-form surface. It is structured as follows. Section 2 presents the required background on surface partitioning, toolpath planning optimization, and derivative-free and blackbox optimization. We propose an optimization formulation of the toolpath planning problem for miling free-form surfaces in Section 3.

Section 4 introduces and evaluates surrogate functions to improve the blackbox optimization process. Computational results are presented in Section 5, while Section 6 is devoted to comparison with previous work. Section 7 briefly illustrates the application of our methodology to a real-world industrial part. Finally, Section 8 draws conclusions and presents perspectives for future work.

2 Background

This section presents the required background on surface partitioning, toolpath planning optimization, and derivative-free and blackbox optimization. Only the most relevant papers in relation to our work are presented here. For a broad review of the state of the art in CNC machining of free-form surfaces, see [4] or [5] and references therein.

2.1 Surface partitioning

Finding an optimal surface partitioning for 3-axis machining is discussed for instance in [6]. The idea is to use a clustering approach to partition the surface based on a metric specifically dedicated to free-form surface clustering. Other works on surface partitioning include heuristic-based methods inspired by the Vehicle Routing Problem [7], an approach based on Adaptive Multi-Agent Systems [8], and an approach based on the concept of efficiency intervals (set of directions where toroidal cutter is better than ball-end one) is proposed by [9]. In [10], surface zones are generated based on a vector field of locally-optimal machining directions. A similar approach is adopted in [11] for flank milling. The objective of surface partitioning approaches is to find zones with small variations of the steepest-slope direction. This increases the effective radius of the cutter leading to maximal step-over distances, and minimizes thereby the toolpath length, and as a consequence the machining time is expected to decrease.

2.2 Toopath planning optimization

Toolpath planning is a complex problem and few authors have approached it in its globality with an optimization approach. The authors of [12] are addressing this problem by building an objective function that attempts to ensure all requirements for obtaining acceptable milling trajectories. The objective function is in fact a weighted sum of conflicting criteria that neglects the real objective for practitioners, which is to minimize machining duration. Several authors nevertheless propose optimization formulations to improve some particular aspects of the overall toolpath planning problem. For instance [13] uses a genetic algorithm to find a parametric cutter path curve of the form $v = A_0 + A_1 u + A_2 u^2 + A_3 u^3$, that minimizes the toolpath length using isoscallop strategy in 3-axis machining, where the coefficients $A_0, \ldots, A_3$ are the optimization variables. A traveling salesman problem formulation involving precedence constraints is proposed in [14] to minimize the non-productive time (so-called *airtime*) by optimally connecting different toolpath segments. Again, a traveling salesman problem formulation is used by [15] together with a minimum spanning tree in view of minimizing cutting forces, with imposed passing points. Two toolpath planning optimization procedures are presented in [10]. The first one defines a toolpath based on a curvilinear grid generation approach, defined by minimizing the kinematics error in 5-axis machining. The second one relies on space filling curves algorithms to fill an isoparametric grid according to the machining strip calculated for a flat-end cutter. CNC machine-tool kinematics is also a field of interest: [16] introduces permissible *jerk* (derivative of the acceleration) as a constraint in an optimization model of the feedrate under velocity and acceleration constraints to generate minimum-time trajectories.

Free-form surface machining-time cannot be expressed as a closed-form formula. It requires a complete milling simulation that includes an accurate evaluation of the elapsed time. This simulation relies on complex algorithms and produces a non-smooth objective function that is subject to noise, and that features numerous spurious local minima. That is why most of the proposed optimization algorithms are either heuristics, or address partially the machining-time minimization problem.

2.3 Derivative-free and blackbox optimization

Since the 1990's, due to strong demand from industry, optimization methods with proofs of convergence have been developped to address problems involving objective and constraint functions given by costly simulations and/or without any gradient information available. Introductions to blackbox and derivative-free optimization algorithms are given in [17, 18].

In this paper, we shall use the blackbox optimization software NOMAD [19, 20] to solve the machining time minimization problem. NOMAD is considered as the best performer for the type of problem we consider here [21]. It is a C++ implementation of the MADS algorithm, an iterative method that proposes a new point where to evaluate the objective function (and the constraints) at the next iteration. An iteration of MADS algorithm involves two steps: a search step and a poll step. The search step attempts to escape from local optima; any heuristic can be used in the search step in order to find a better point. This step is to be defined by the user. If the search step is not successful, then the poll step is performed. It selects a set of directions that rely on a mesh constructed from a positive spanning set of the search space. These directions are used to define new points. If a better point

is found, the mesh is coarsened ; otherwise, it is refined. The convergence of MADS is mathematically guaranteed by the poll step. Details about the MADS algorithm can be found in [22].

NOMAD allows the user to provide a so-called *surrogate* (cheaper-to-evaluate models) of the costly objective and constraints functions in order to improve the optimization computing time. It is either a simplified physical model of the blackbox functions, or an approximation model obtained by evaluating the blackbox function at a given number of points then by interpolating or smoothing the values. Two types of surrogates can be envisaged: static and dynamic. Static surrogates are provided by the user and are not updated during the optimization, but in return they allow introduction of business knowledge about the application into the surrogate. Dynamic surrogates are in contrast constructed gradually from the evaluations available made in previous iterations. A surrogate can also be used to re-order the poll points, before they are evaluated by the blackbox. The use of surrogates may reduce considerably the number of blackbox evaluations and, sometimes, may help to find a better solution (than the one found without surrogate). NOMAD provides a library of dynamic surrogates. For more details, see [23] which presents a survey on surrogate optimization.

One contribution of this article is to propose (in Section 4) fast surrogates of the expensive machining-time blackbox.

3 Optimization formulation

We are given a free-form surface, S, parameterized by two coordinates, u and v. The objective is to machine S in minimal time under a given threshold constraint. The optimization approach proposed in this paper involves a black box that consists of two main steps: partitioning and machining. Partitioning the surface S into zones is done using unsupervised clustering algorithms applied on a set of N sample points, denoted $\mathbf{S_i}$, $i = 1, 2, \ldots, N$, built from an isoparametric meshing (Figure 3).

Some clustering algorithms, such as K-means, requires the user to specify the number of clusters, denoted K, which can thus be considered as a first optimization variable. An upper bound, K_{max}, of K can easily be defined since beyond a certain number of clusters, machining time increases, due to the time loss each time the tool finishes a zone and moves to the next one. This time loss must be taken into account otherwise the problem becomes trivial: the solution is obtained by considering as many clusters as there are sample points. Furthermore, clustering algorithms require in general a *feature vector* and a measure of dissimilarity, hereafter called (by misuse of language) the *metric*.

Following in [6], we choose the feature vector (u_i, v_i, s_i, θ_i), where (see Figure 4):

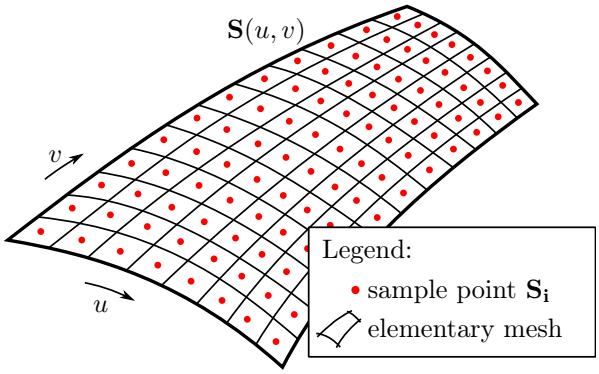


Figure 3: The sample points to be clustered.

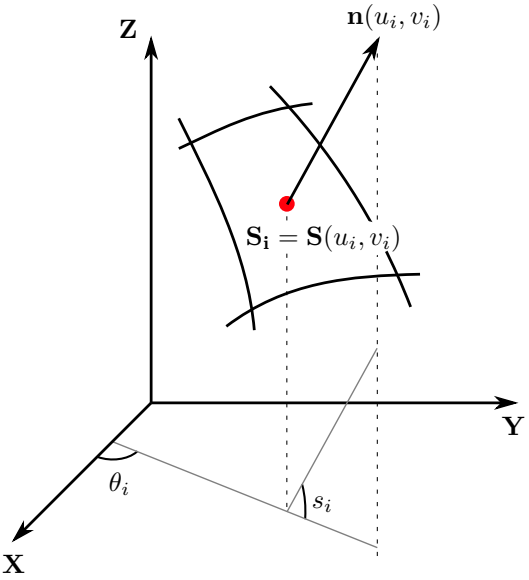


Figure 4: One sample point S_i of the surface and the four components (u_i, v_i, s_i and θ_i) of its feature vector.

- (u_i, v_i) is the parametric coordinates of the sample point $\mathbf{S_i}$;
- s_i is the steepest-slope angle, *i.e.* the angle between the vector $\mathbf{n}_{(u_i,v_i)}$, normal to the surface at $\mathbf{S_i}$, and the horizontal plane $(\mathbf{X}, \mathbf{Y})$;
- θ_i is the steepest-slope orientation, *i.e.* the angle between the $\mathbf{X}$ axis and the projection of $\mathbf{n}_{(u_i,v_i)}$ onto the horizontal plane $(\mathbf{X}, \mathbf{Y})$.

For the metric, we choose a weighted Euclidean distance, as proposed also in [6]. More precisely, the distance between two given points S_i and S_j is given by:

$$\sqrt{w_1(u_i - u_j)^2 + w_2(v_i - v_j)^2 + w_3(s_i - s_j)^2 + w_4(\theta_i - \theta_j)}$$

where the values of the weights, $0 \leqslant w_1, w_2, w_3, w_4 \leqslant 1$, can be considered as further optimization variables. In the sequel we shall denote by W the vector whose components are w_1, w_2, w_3 and w_4.

This first step produces a partition of the surface into K zones. Let I_k be the index set of the sample points that belong to zone k, $k = 1, 2, \ldots, K$. The clustering step is interpreted as the following blackbox constraint:

$$I = c(K, W), \tag{1}$$

where c is a function that represents this clustering step procedure that outputs a partition, I, of the index set of the data points; I is a vector whose k^{th} component is I_k, $k = 1, 2, \ldots, K$.

The machining step addresses each zone separately according to the parallel-plane strategy. The distance between two adjacent planes (called step-over distance) is defined as the maximum distance such as the scallop height constraint is respected for the whole path. At a given plane, the position of the adjacent plane is easy to determine: it suffices to consider the worst point, in terms of scallop height, of the path defined by the given plane. This is enough to ensure that the quality constraint will be satisfied for the entire trajectory. From the optimization point of view, this strategy is very useful because the constraints are thereby satisfied by construction. The scallop height may be calculated from the surface swept by the cutter during milling [24, 25], but using a fast scallop height calculation method, such as in [26], the above-described toolpath planning process (including milling time calculation [27]) is rapid enough to be used as the objective function of an optimization procedure. In the case of 3-axis machining, for each cluster k, the toolpath planning depends only on the machining direction. The angle, noted γ_k, between this direction and the $\mathbf{X}$ axis is an optimization variable.

When 3+2-axis machining is considered, two more orientation angles must be determined for each zone k. Firstly, we define the angle, φ_k, between the tool axis and the $\mathbf{Z}$ axis. Secondly, we define the angle, ψ_k, between the $\mathbf{X}$ axis and the projection of the tool axis on the $\mathbf{XY}$ plane. As for γ_k, these orientation angles, illustrated on Figure 5, are optimization variables. In the sequel we shall use the following vector notation: $\Gamma_K = (\gamma_1, \gamma_2, \ldots, \gamma_K)$, $\Phi_K = (\varphi_1, \varphi_2, \ldots, \varphi_K)$ and $\Psi_K = (\psi_1, \psi_2, \ldots, \psi_K)$ (each component corresponds to one zone).

All points of the zone should be accessible by the tool. More precisely, consider a zone k and a given point of that zone, of parametric coordinates (u, v). Let $\beta_{(u,v)}(\varphi_k, \psi_k)$ be the angle between the $\mathbf{Z}$ axis and the vector, $\mathbf{n}_{(u,v)}(\varphi_k, \psi_k)$, normal to the zone k at (u, v). This given point is considered accessible by the tool if $|\beta_{(u,v)}(\varphi_k, \psi_k)| \leqslant \frac{\pi}{2}$. In practice, this constraint is evaluated only over the sample points $(u_i, v_i) \in I_k$ (belonging to zone k). This will be ensured within the objective-function blackbox evaluation, f.

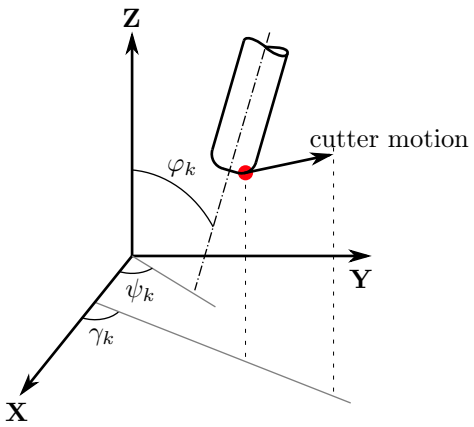


Figure 5: The three angle variables used for 3+2-axis optimization for a zone k.

To sum up, the resulting optimization problem involves a blackbox objective function (the machining time), and the blackbox clustering constraint (1). A chart summarizing the working flow is shown on Figure 6, and the constrained mixed variable optimization problem considered is:

$$\min_{K, W, \Gamma_K, \Phi_K, \Psi_K} F(K, W, \Gamma_K, \Phi_K, \Psi_K)$$

subject to

$$\begin{array}{ll} I = c(K, W) & \\ 0 \leqslant w_i \leqslant 1, & i = 1, 2, 3, 4 \\ 0 \leqslant \gamma_k \leqslant \pi, & k = 1, 2, \ldots, K \\ -\frac{\pi}{2} \leqslant \varphi_k \leqslant \frac{\pi}{2}, & k = 1, 2, \ldots, K \\ 0 \leqslant \psi_k \leqslant \pi, & k = 1, 2, \ldots, K \\ K \in \{1, 2, \ldots, K_{\max}\}, & \end{array} \tag{2}$$

where

$$F(K, W, \Gamma_K, \Phi_K, \Psi_K) = \sum_{k=1}^{K} f(I_k, \gamma_k, \varphi_k, \psi_k).$$

This problem is partially separable per zone, the only linking constraint being constraint $I = c(K, W)$ through the components, I_k, $k = 1, 2, \ldots, K$, of the sample-point index-set partition I.

Remark that the number of variables and constraints of our problem depends directly on the number of zones, K, which is itself a variable. Although it is integer valued, the optimization variable K plays a role of *categorical* variable: its continuous relaxation is meaningless.

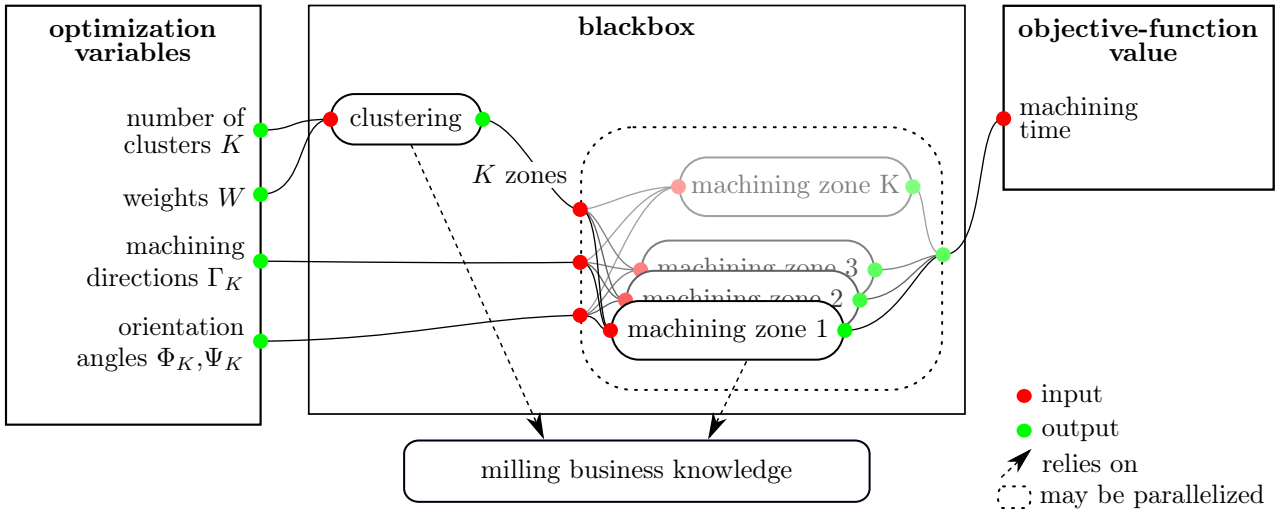


Figure 6: Working flow chart of the two-step optimization model.

4 Surrogates models and their numerical evaluation

This section proposes and evaluates surrogate functions to improve the blackbox optimization process.

In this section, we focus on *static* surrogates. The aim is to construct the cheaper surrogates to reduce the computational time, without losing too much precision. An efficient surrogate does not need to be a good approximation of the exact values of the blackbox, it may solely approximate its variations and the location of its optimal points.

In our context, the machining time is the result of a blackbox simulation based on numerical calculation of intersections of the part surface and parallel planes. The intersection curves (the machining toolpath) rely on an isoparametric meshing of the part surface. The precision of the curves and the time for their calculation increase with the mesh step size. Therefore, the mesh step size is a parameter that allows simplification of the physical model (cheapest evaluations but less precision) and can be used to create surrogates. This will be used to create our Surrogates 1 and 2. Note that the mesh step size cannot be directly defined on the actual surface, it has to be defined in the parametric variable space. Consequently, a reverse approximation procedure is implemented to set the step size, targeting a given value in the 3D space.

A second parameter that could be used to shorten the simulation time is the scallop height tolerance. In fact, increasing this value leads to larger step-over distances which reduces the number of parallel planes, and subsequently the number of intersections to be computed with the part surface. Of course, this will impact the objective-function values (reduced machining times). Nevertheless, it is reasonable if the variations and optimal points were preserved. This will be investigated further in this section and it motivate our Surrogate 4.

So far, simplifications of the model have concerned the machining step rather than the clustering step. One way of simplifying the clustering step is to increase the clustering mesh step size. This leads to the reduction of the computational time required for clustering, since the K-means algorithm (and any other clustering algorithm) needs less time to run for fewer data points. However, the precision of the part surface zones (the clusters) is then deteriorated; the borders of the zones are roughly approximated, which will affect the machining times values. The simulation time reduction is more or less efficient depending on the algorithm used. In our case, empirical tests show that the K-means algorithm is fast enough, and that this does not yield a sufficient benefit.

Yet another alternative to simplify the blackbox would be to approximate the part surface zones to be machined by rectangular planes, so that machining times could be calculated analytically, which is expected to reduce drastically the computational time of the blackbox evaluation. The approximation by planes should be accurate, even though some zones may not be adapted for such an approximation. Principal Component Analysis (PCA) is used to calculate the planes that best fit the surface zones. This procedure is the same one as that introduced in [28], but we shall apply it here to *zones* instead of applying it to the whole surface. The machining time is thereafter calculated, via Surrogate 4, using the analytical approach detailed in A.

To summarize, we propose the four surrogate functions to approximate the blackbox function f:

- **Surrogate 1** approximates coarsely the blackbox function, targeting a value of 5 mm for the mesh step size (instead of the original value 0.5 mm).
- **Surrogate 2** is a finer approximation of the black-

box function, targeting a value of 2 mm for the mesh step size.

- **Surrogate 3** approximates the blackbox function using an analytical calculation based on the above-mentionned PCA-based approach.
- **Surrogate 4** approximates the blackbox function using a scallop height tolerance increased to 1 mm (instead of the original value 0.1 mm).

In order to investigate the efficiency of the four surrogates, we are content with the case of 3-axis machining (*i.e.* the components of the orientation variables Φ_K and Ψ_K are set to the constant value zero).

In this paper, we use two test surfaces: a 4×4 bicubic Bézier surface (named Surface 1) and a 3×3 quadratic Bézier surface (named Surface 2). The corresponding control points are given in B.

These surfaces are partitioned using the K-means algorithm (clustering step) and the Euclidean metric (*i.e.* $w_1 = w_2 = w_3 = w_4 = 1$), leading to a total of seven zones to be studied (Figure 7).

Figures 8 and 9 display the variations of the four surrogates and those of the blackbox machining-time function, f, for each zone k of Surface 1 and of Surface 2, respectively. To compare the different surrogates and their correlations with the blackbox, we calculate a correlation coefficient for each zone as follows. Consider a particular zone k. Let $(\gamma^i)_{i=1,2,\ldots,m}$ be a uniform sampling of the domain $[0, \pi]$. Let h denote one of the four surrogate functions. The correlation coefficient ρ between f (the blackbox) and the surrogate function h is computed from the sample values $f(I_k, \gamma^i, 0, 0)_{i=1,2,\ldots,m}$ and $h(I_k, \gamma^i, 0, 0)_{i=1,2,\ldots,m}$.

Table 1 gives the percentages of the computation times of each surrogate compared to the blackbox function f for the different zones, while Table 2 provides the correlation coefficient of each surrogate with the blackbox.

All surrogates show a relatively good correlation with the blackbox to approximate. We shall therefore eliminate the two candidates (Surrogates 1 and 2) that are the slowest in terms of computational time. Surrogate 3 is particularly interesting with respect to its very low computational time. This is due to its PCA-based analytical computation, which renders Surrogate 3 especially relevent when dealing with zones whose shapes are almost rectangular (such as the zones of Surface 2). For zones whose shape is far from being rectangular, one should rely on Surrogate 4. But, somewhat unexpectedly, at least for surfaces tested in Section 5, Surrogate 3 also provides very good results.

5 Tests and results

In this section, we present computational results of our optimization methodology using Surrogates 3 and 4 on Surfaces 1 and 2.

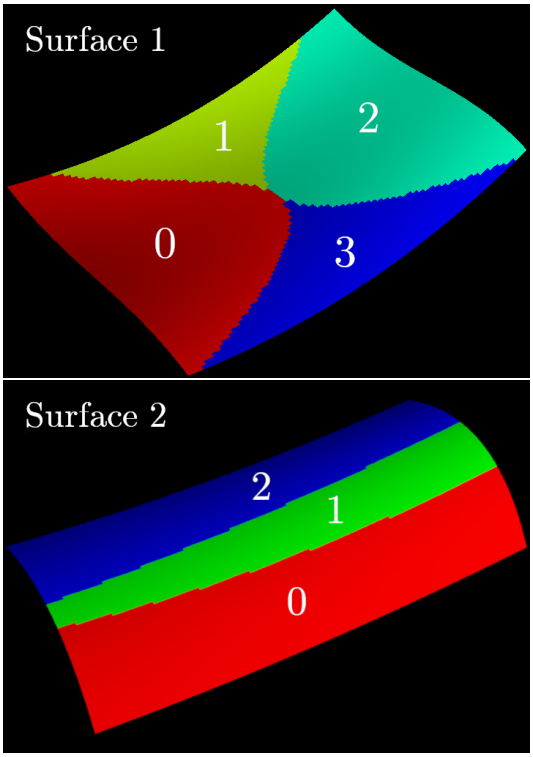


Figure 7: Clustering of the two surfaces (zones numbers are displayed in white).

5.1 Test protocol

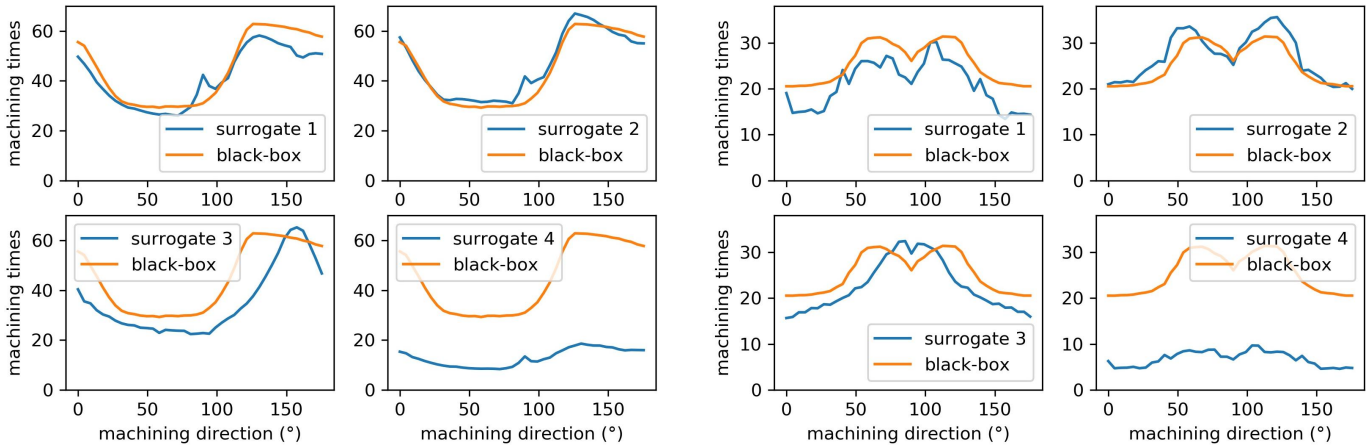
Machining is performed with a toroidal cutter whose radii are $R = 5\,\text{mm}$ (outer radius) and $r = 2\,\text{mm}$ (torus radius). The parallel-plane strategy is used. The scallop height tolerance is set to $sh = 0.01\,\text{mm}$, a value commonly used in industry. The kinematic parameters of the CNC machine tool are:

- jerk (constant): $J = 40\,\text{m/s}^3$;
- maximum acceleration: $A_{max} = 6\,\text{m/s}^2$;
- nominal feedrate: $V_f = 0.083\,\text{m/s}$ ($5\,\text{m/min}$).

For the sake of simplicity, in this study we are content with using an enumerative strategy to determine the value of the decision variable K (the number of zones). More precisely, the optimization problem is solved successively for $K = 2, 3, \ldots, K_{\max}$. This proves, in our tests, to be more efficient than introducing K as an optimization variable. Remark that the resulting $K_{\max}$ optimization problems could then even be solved in parallel. The upper limit on the number of clusters is set to $K_{\max} = 10$.

Let us now consider one fixed value of K. The mesh used for clustering is a uniform 80×80 isoparametric grid.

The maximum number evaluations of the blackbox (Figure 6) is set empirically to 1,000. The mesh direc-



(a) zone 0 of Surface 1

(b) zone 1 of Surface 1

(c) zone 2 of Surface 1

(d) zone 3 of Surface 1

Figure 8: Evolution of machining times of the four surrogates compared to the real blackbox function for the zones of Surface 1 .

tion type parameter of the MADS algorithm is set to `ORTHO 2N` (*i.e.*, poll directions are generated using the maximum positive spanning set produced from an orthogonal basis of the search space and its opposite).

The initial point given to NOMAD is defined with $\omega_i = 1$, $i = 1, 2, 3, 4$, and $\varphi_k = \psi_k = 0$, , $k = 1, 2, \ldots, K$ and γ_k, $k = 1, 2, \ldots, K$, is set to the average steepest slope direction of zone k. This starting solution is considered by practitioners to be a good machining configuration.

For each optimization run, the gain η is defined in relation to this initial point:

$$\eta = \frac{F_0 - F^*}{F_0}, \tag{3}$$

where F_0 is the value of the objective function at the initial point, and F^*, the best value of the objective function.

As it is common practice in blackbox optimization, the evaluation of surrogates is counted in the computation time in proportion to their relative cost in time with respect to that of the blackbox. For the sake of comparison, the optimization process is stopped after a computational time equivalent to 1,000 blackbox evaluations.

5.2 Optimization results for 3-axis machining

To simulate 3-axis machining, angles ϕ_k and ψ_k are fixed constant, to zero, throughout the optimization process. For each test surface, three optimizations are run: one without surrogate, the other with Surrogate 3 and the last with Surrogate 4.

Table 3 summarizes the optimization results for each of these three cases for Surface 1. This example illustrates how using an efficient surrogate may allow the optimization to explore a larger part of the search space. Indeed, the best objective-function value is found using Surrogate 3 (with $K = 8$). The optimization history graphs, presented in Figure 10, show that the surrogate functions, especially Surrogate 3, accelerate convergence, requiring

Table 1: Average CPU time ratios between each surrogate and the blackbox f, for each zone k.

Surface	zone	Surrogate 1	Surrogate 2	Surrogate 3	Surrogate 4
1	0	26.2%	62.5%	5.4%	14.9%
	1	25.2%	66.7%	3.1%	17.4%
	2	23.9%	82.0%	3.0%	18.8%
	3	18.1%	59.4%	1.6%	14.0%
2	0	18.8%	84.0%	3.0%	8.0%
	1	19.8%	106%	1.3%	11.3%
	2	13.6%	103%	1.2%	9.4%

Table 2: Correlation coefficient between each surrogate and the blackbox f, for each zone k.

Surface	zone	Surrogate 1	Surrogate 2	Surrogate 3	Surrogate 4
1	0	0.962	0.979	0.854	0.966
	1	0.925	0.915	0.824	0.925
	2	0.973	0.982	0.850	0.980
	3	0.920	0.932	0.650	0.915
2	0	0.813	0.992	0.931	0.820
	1	0.990	0.999	0.936	0.991
	2	0.810	0.981	0.931	0.815

fewer blackbox evaluations, this is especially true when the number of variables is high (*i.e.*, for large values of K).

Table 4 summarizes the optimization results for Surface 2. In this configuration, again, the best point is found using Surrogate 3 (for $K = 4$). In this case, however, using surrogates does not yield a better objective-function value. The optimization history graphs presented in Figure 11 show, again, that the surrogates functions, especially Surrogate 3, accelerate convergence, requiring fewer blackbox evaluations. This means that even if using surrogates does not improve the result, it still allows one to reach the best objective-function value faster.

5.3 Optimization results for 3+2-axis machining

Due to the high number of optimization variables involved in 3+2-axis machining, especially for high values of K, NOMAD's search step is skipped (constructing search models may be very time consuming for 3+2-axis machining). Moreover, here, we present only results using Surrogate 3, as it systematically yields results that are better than those of Surrogate 4.

Tables 5 and 6 summarize the optimization results for Surfaces 1 and 2, respectively. As for the results obtained in 3-axis-machining, the optimization history graphs (Figures 12 and 13) here again reveal that Surrogate 3 accelerates convergence, requiring fewer blackbox evaluations.

For each value of K, both optimization runs took nearly the same CPU time, approximately 1 hour.

To sum up, for both 3-axis and 3+2-axis machinings, the optimization process allows improvements of machining times, both on Surfaces 1 and 2, of nearly 15% compared to the reference values. Recall that these reference values are obtained using solely a clustering step with a particular (non-optimized) choice of the weights w_i's that is fairly representative of recent, the-state-of-art results [6].

As expected the machining times are generally better in 3+2-axis machining than in 3-axis machining (more degrees of freedom for optimizing). Some exceptions are due to the fact that the optimization algorithm cannot guarantee global optimality and the variable space is likely to be explored differently from one case to another. The main point to keep in mind here is that for the best values obtained, 3+2 axis machining does better than the 3-axis counterpart.

Surrogate 3 is not only the fastest, because of its analytical formulation, but it is also the one that yields the best results. In addition, the optimization history plots show a faster decrease of machining times when Surrogate 3 is used. They also reveal that our methodology generally reaches best solutions within a number-of-evaluation budget that is much lower that the 1,000 iterations we used in our tests.

K	initial value F_0	F^*, no surrogate	F^*, Surrogate 3	F^*, Surrogate 4	η, no surrogate	η, Surrogate 3	η, Surrogate 4
2	96.19	92.01	92.01	92.01	4.34%	4.34%	4.34%
3	104.77	93.92	92.92	92.52	10.35%	11.31%	11.69%
4	147,68	98.5	95.92	95.06	33.3%	35.04%	35.63%
5	101.90	96.41	96.23	100.32	5.38%	5.56%	1.55%
6	100.1	95.96	98.41	97.3	4.13%	1.68%	2.79%
7	103.78	102.02	100.06	102.13	1.69%	3.58%	1.58%
8	102.4	95.48	**91.93**	92.87	6.75%	10.22%	9.3%
9	153,5	102.7	95.29	95.88	33.09%	37.92%	37.53%
10	143,9	97.25	97.91	97.21	32.41%	31.96%	32.44%

Table 3: Optimization results for 3-axis machining of Surface 1 (machining times in seconds).

K	initial value F_0	F^*, no surrogate	F^*, Surrogate 3	F^*, Surrogate 4	η, no surrogate	η, Surrogate 3	η, Surrogate 4
2	91.54	90.91	90.82	91.01	0.68%	0.78%	0.57%
3	91.62	86.83	90.77	90.63	5.22%	0.92%	1.08%
4	103.53	**78.86**	**78.86**	87.55	23.82%	23.82%	15.43%
5	96.75	90.06	87.26	89.49	6.91%	9.8%	7.5%
6	102.8	88.49	90.03	87.28	13.92%	12.42%	14.52%
7	106.1	89.05	86.89	87.87	16.06%	18.11%	17.18%
8	108.55	93.61	86.52	90.32	13.76%	20.3%	16.79%
9	108.23	88.39	86.75	87.94	18.33%	19.85%	18.75%
10	116.11	88.11	90.97	87.38	24.11%	21.65%	24.74%

Table 4: Optimization results for 3-axis machining of Surface 2 (machining times in seconds).

6 Comparison with other works

This section compares numerically our approach with other works from the literature.

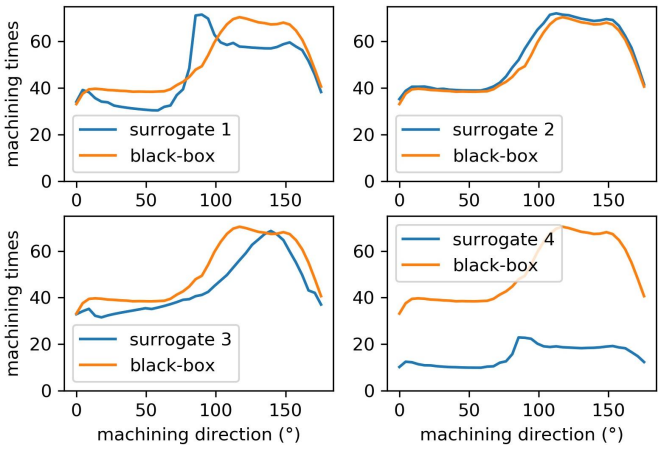
We test on the same test surfaces (Surface 1 and Surface 2) and on third surface from [29]. We use for comparison previous works from the literature that provide sufficient information (machining time obtained, and when it is not given: the path length obtained) and we use the same machining parameters. Remark that, to our knowledge, these studies are the only ones providing sufficient data to make comparisons possible.

Surface 2 is studied in [7] and [8]. In [7], Clarke and Wright algorithm is used, among others, for partitioning, while in [8], an adaptive multi-agent system approach is adopted. Both papers use the parallel-plane strategy for 3-axis machining. Only the best results provided by these two papers are included into the present comparison.

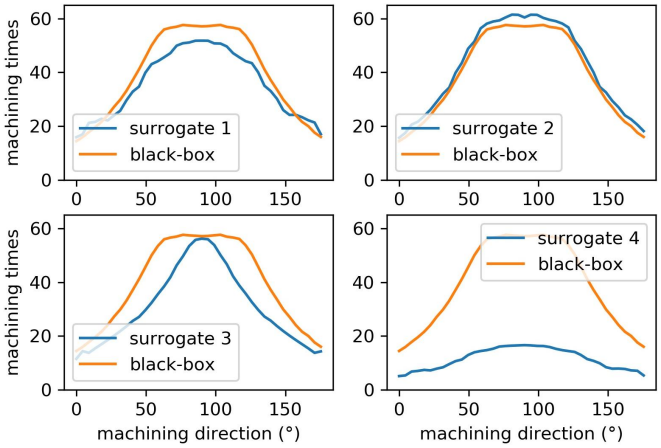
In [7], a toroidal cutter with radii $R = 5\,\text{mm}, r = 2\,\text{mm}$ is used, and the maximum scallop height is set to $sh = 0.01\,\text{mm}$. The shortest toolpath length, provided by this study is 4,988 mm. Our methodology provide a better result: 4,018 mm using 200 blackbox evaluations per value of k ($k = 2, 3, \ldots, 10$).

In [8], a toroidal cutter with radii $R = 3.175\,\text{mm}, r = 1\,\text{mm}$ is used, and $sh = 0.254\,\text{mm}$. The shortest toolpath length obtained is 1183 mm, but this is not taking into account the connections between adjacent paths. Our approach yields again a better toolpath length: 995 mm.

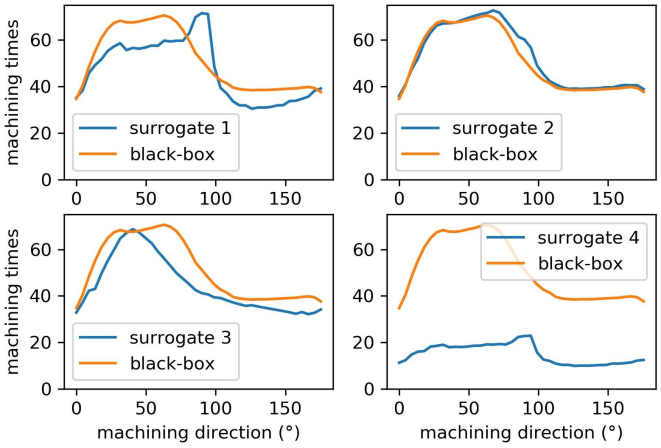
Surface 1 is studied in a more recent work [9] where a partitioning approach based on the concept of efficient machining intervals is proposed. The surface is machined in 3 axes using a toroidal tool with the same radii as in Section 5. Two feed rates are considered: a low feed rate of 1 m/min (for hard materials and/or large tools), and a



(a) zone 0 of Surface 2

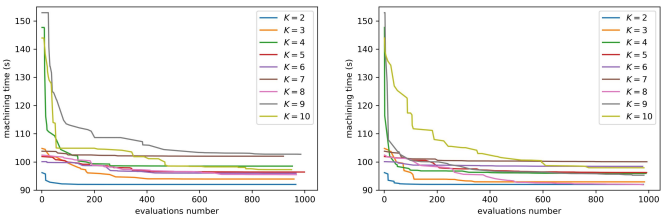


(b) zone 1 of Surface 2

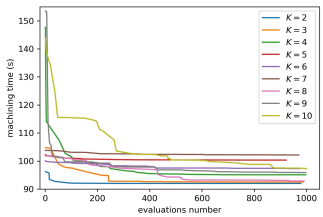


(c) zone 2 of Surface 2

Figure 9: Evolution of machining times of the four surrogates compared to the real blackbox function for the zones of Surface 2.

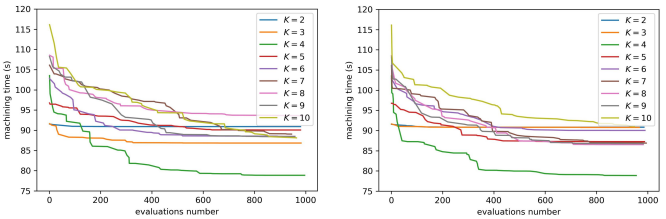


(a) without surrogate (b) Surrogate 3

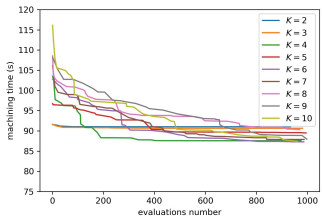


(c) Surrogate 4

Figure 10: Optimization history for 3-axis machining of Surface 1.

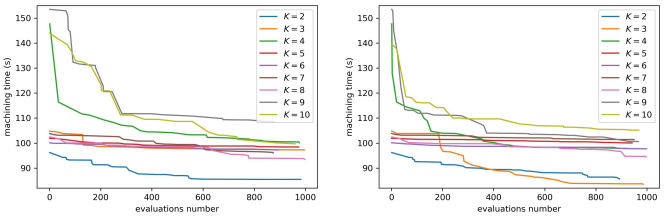


(a) without surrogate (b) Surrogate 3



(c) Surrogate 4

Figure 11: Optimization history for 3-axis machining of Surface 2.



(a) without surrogate (b) Surrogate 3

Figure 12: Optimization history for 3+2-axis machining of Surface 1 .

K	initial value F_0	F^*, no surrogate	F^*, Surrogate 3	η, no surrogate	η, Surrogate 3
2	96.19	85.46	85.68	11.15%	10.92%
3	104.77	97.24	**83.59**	7.18%	20.21%
4	147,68	100.11	100.14	32.21%	32.19%
5	101.90	98.42	96.2	3.41%	5.59%
6	100.1	97.19	97.41	2.9%	2.69%
7	103.78	96.00	97.8	7.49%	5.76%
8	102.4	93.48	94.36	8.71%	7.85%
9	153,5	108.54	91.93	29.28%	40.11%
10	143,9	99.70	96.09	30.71%	33.22%

Table 5: Optimization results for 3+2-axis machining of Surface 1 (machining times in seconds).

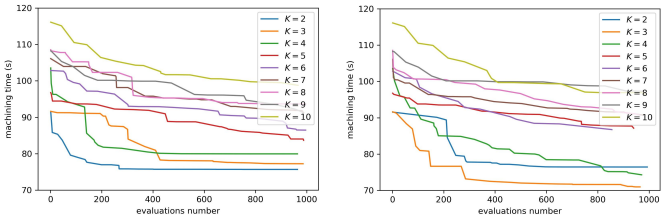
K	initial value F_0	F^*, no surrogate	F^*, Surrogate 3	η, no surrogate	η, Surrogate 3
2	91.54	75.67	74.96	17.33%	18.11%
3	91.62	77.23	73.84	15.70%	19.4%
4	103.53	79.94	**71.95**	22.78%	30.5%
5	96.75	83.67	82.01	13.51%	15.23%
6	102.8	86.46	85.43	15.89%	16.89%
7	106.1	91.29	82.23	13.95%	22.5%
8	108.55	93.65	84.03	13.72%	22.58%
9	108.23	91.28	93.91	15.66%	13.23%
10	116.11	98.76	92.04	14.94%	20.73%

Table 6: Optimization results for 3+2-axis machining of Surface 2 (machining times in seconds).

high feed rate equal to 10 m/min (for light alloys and/or small tools). The low feed rate leads to a machining time equal to 311 seconds, and the high feed rate leads to 81 seconds. Our results are again better: 291 seconds for the low feed rate, and 48 seconds for the high speed rate.

A fuzzy C-means algorithm is used in [29] with a 5-parameter feature vector, $(u, v, \mathbf{n})$, to partition, for 3+2-axis machining, a test surface defined in the same paper. Our approach is tested with the same toroidal cutter of radii $R = 6.7\,\text{mm}$,$r = 0.7\,\text{mm}$, using the maximum scallop height: $sh = 0.0254\,\text{mm}$, the nominal feed rate $V = 2\,\text{m/min}$, and $K = 4$ zones. The authors of [29] report a machining time of 634 seconds, while our approach still leads to a better machining time: 601 seconds.

The computational time for all the algorithms in studies presented hereabove (including ours) are of the same



(a) without surrogate (b) Surrogate 3

Figure 13: Optimization history for 3+2-axis machining of Surface 2.

order of magnitude, and their cost is negligeable with regards to the cost of industrial machining processes (hourly rate for exploiting of a machine tool is far above hourly rate for the use of a computer). Furthermore, toolpath planification is done only once while real world parts may be produced a very large number of times.

7 Real-world part application

The tests and results presented in subsections 5.2 et 5.3 focused on surfaces extracted from the literature and that provides enough data for comparison purposes (see Section 6). However, the methodology we introduced obviously also applies to surfaces with more complex geometries. As an example of such an application, below is provided the result of our optimization workflow to an industrial surface: the blade of a propeller (shown on Figure 14).

On such a surface, optimal partitioning and optimal machining directions are far from being obvious.

Figure 15 displays the four zones resulting from clustering, and relative scallop height distribution. The relative scallop height is defined as the ratio:

$$\frac{\text{effective scallop height}}{\text{maximum allowed scallop height}}.$$

Recall that since the scallop height constraint is integrated in our toolpath planning algorithm, it is necessarily satisfied, by construction. Thus, the relative scallop

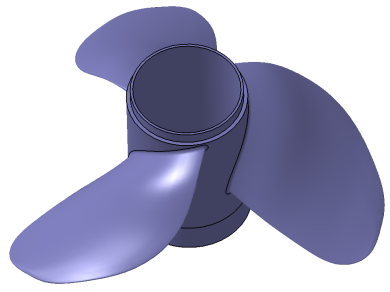


Figure 14: An industrial application: a propeller blade extrado

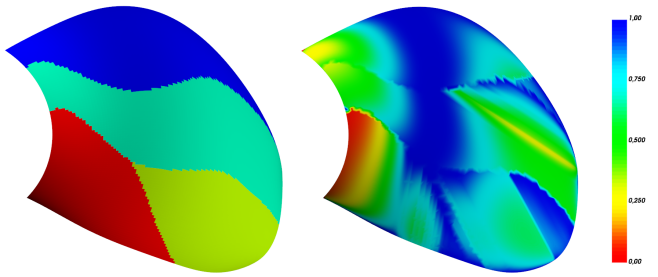


Figure 15: Four zones obtained by clustering (left) and the relative scallop height distribution over the blade surface (right)

height ratio is necessarily included in the $[0, 1]$ interval. The optimization results (machining time) is better when the relative scallop height ratio is close to one, since an effective scallop height near to the maximum allowed scallop height implies faster machining process. Indeed, as it is common when optimizing in presence of a constraint that is strongly conflicting with the objective, the scallop height constraint is active (or almost active) over most of the surface.

8 Conclusion and perspectives

The blackbox optimization approach for machining free-form surfaces introduced in this article involves an objective function that is composed of two blackboxes: a clustering step producing a partition of the surface into zones, followed by a machining-time computation for each zone. The clustering step is driven by the number of zones, and by the weighting coefficients for the metric used for clustering. The machining-time computation step addresses, for each zone, the choice of the machining direction (and also orientation angles in the case of 3+2-axis machining). In order to improve the performance of the optimization process, we proposed and evaluated the efficiency of four surrogate functions, one of which is analytical and thereby cheap to evaluate. Numerical experiments on several benchmark test surfaces from the literature show that our methodology outperforms other approaches. Figures comparisons show that our algorithms always provide a better result than other approaches. These gains can go up to 40% in terms of machining time. However, numerical comparisons should be taken with caution because the performances of algorithms are strongly dependant on the topology of the test surfaces. In any case, the main benefit of our methodology is to allow a broader exploration of the search space, in particular thanks to the use of surrogates that takes advantage of specific business knowledge.

Future tracks of research may consider integrating more categorical decision variables such as: the cutter type to be used, the machining strategy (other than the parallel planes), the clustering algorithm used in the partitioning step or even the metric upon which it is based to define the zones. Finally, the partial-separability of the optimization formulation introduced in this paper could probably be advantageously exploited.

References

[1] S.-H. Suh, J.-J. Lee, Five-axis part machining with three-axis CNC machine and indexing table, Journal of Manufacturing Science and Engineering (1998).

[2] R. E. Barnhill, G. Farin, M. Jordan, B. R. Piper, Surface/surface intersection, Computer Aided Geometric Design 4 (1) (1987) 3–16.

[3] S. Bedi, F. Ismail, M. Mahjoob, Y. Chen, Toroidal versus ball nose and flat bottom end mills, The International Journal of Advanced Manufacturing Technology 13 (1997) 326–332.

[4] A. Lasemi, D. Xue, P. Gu, Recent development in CNC machining of freeform surfaces: A state-of-the-art review, Computer-Aided Design 42 (2010) 641–654.

[5] R. A. Mali, T. V. K. Gupta, J. Ramkumar, A comprehensive review of free-form surface milling– Advances over a decade, Journal of Manufacturing Processes 62 (2021) 132–167.

[6] M. Herraz, J.-M. Redonnet, M. Mongeau, M. Sbihi, Analysis of clustering algorithms for by-zone machining of free-form surfaces, Tech. rep., Université Paul Sabatier, Toulouse (2022).
URL https://hal.archives-ouvertes.fr/hal-03568407

[7] S. Djebali, S. Segonds, J.-M. Redonnet, W. Rubio, Using the global optimisation methods to minimise the machining path length of the free-form surfaces in three-axis milling, International Journal of Production Research 53 (17) (2015) 5296–5309.

[8] S. Djebali, A. Perles, S. Lemouzy, S. Segonds, J.-M. Redonnet, W. Rubio, Milling plan optimization with an emergent problem solving approach, Computers & Industrial Engineering 87 (2015) 506–517.

[9] V. Duc, F. Monies, S. Segonds, W. Rubio, Automatic minimal partitioning method guaranteeing machining efficiency of free-form surfaces using a toroidal tool, The International Journal of Advanced Manufacturing Technology 107 (2020) 4239–4254.

[10] S. Makhanov, Optimization and correction of the tool path of the five-axis milling machine: Part 1. Spatial optimization, Mathematics and Computers in Simulation 75 (5) (2007) 210–230.

[11] D. He, Z. Li, Y. Li, K. Tang, Quasi-Developable and Signed Multi-Strip Approximation of a Freeform Surface Mesh for Efficient Flank Milling, Computer-Aided Design 140 (2021) 103083.

[12] M. Bartoň, M. Bizzarri, F. Rist, O. Sliusarenko, H. Pottmann, Geometry and tool motion planning for curvature adapted CNC machining., ACM Transactions on Graphics 40 (4) (2021) 1–16.

[13] R. K. Agrawal, D. K. Pratihar, A. Roy Choudhury, Optimization of CNC isoscallop free form surface machining using a genetic algorithm, International Journal of Machine Tools and Manufacture 46 (7) (2006) 811–819.

[14] K. Castelino, R. D'Souza, P. K. Wright, Toolpath optimization for minimizing airtime during machining, Journal of Manufacturing Systems 22 (3) (2003) 173–180.

[15] I. Lazoglu, C. Manav, Y. Murtezaoglu, Tool path optimization for free form surface machining, CIRP Annals 58 (1) (2009) 101–104.

[16] J. Dong, P. Ferreira, J. Stori, Feed-rate optimization with jerk constraints for generating minimum-time trajectories, International Journal of Machine Tools and Manufacture 47 (12-13) (2007) 1941–1955.

[17] A. R. Conn, K. Scheinberg, L. N. Vicente, Introduction to Derivative-Free Optimization, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2009.

[18] C. Audet, W. Hare, Derivative-Free and Blackbox Optimization, Springer, 2017.

[19] C. Audet, S. Le Digabel, C. Tribes, NOMAD user guide, Tech. Rep. G-2009-37, Les cahiers du GERAD (2009).
URL https://www.gerad.ca/nomad/Downloads/user_guide.pdf

[20] S. Le Digabel, Algorithm 909: NOMAD: Nonlinear Optimization with the MADS algorithm, ACM Transactions on Mathematical Software 37 (4) (2011) 1–15.

[21] N. Ploskas, N. Sahinidis, Review and comparison of algorithms and software for mixed-integer derivative-free optimization., Journal of Global Optimization 82 (2022) 433–462.

[22] C. Audet, J.-E. Dennis Jr., Mesh Adaptive Direct Search Algorithms for Constrained Optimization, SIAM Journal on Optimization 17 (1) (2006) 188–217.

[23] A. J. Booker, J.-E. Dennis Jr., P. D. Frank, D. B. Serafini, V. Torczon, M. W. Trosset, A rigorous framework for optimization of expensive functions by surrogates, Structural Optimization 17 (1) (1999) 1–13.

[24] D. Roth, S. Bedi, F. Ismail, S. Mann, Surface swept by a toroidal cutter during 5-axis machining, Computer-Aided Design 33 (1) (2001) 57–63.

[25] J. Machchhar, D. Plakhotnik, G. Elber, Precise algebraic-based swept volumes for arbitrary free-form shaped tools towards multi-axis CNC machining verification, Computer-Aided Design 90 (2017) 48–58.

[26] J.-M. Redonnet, S. Djebali, S. Segonds, J. Senatore, W. Rubio, Study of the effective cutter radius for end milling of free-form surfaces using a torus milling cutter, Computer-Aided Design 45 (6) (2013) 951–962.

[27] X. Pessoles, Y. Landon, W. Rubio, Kinematic modelling of a 3-axis NC machine tool in linear and circular interpolation, The International Journal of Advanced Manufacturing Technology 47 (2010) 639–655.

[28] M. Herraz, J.-M. Redonnet, M. Mongeau, M. Sbihi, A new method for choosing between ball-end cutter and toroidal cutter when machining free-form surfaces, The international journal of advanced manufacturing technology 111 (5-6) (2020) 1425–1443, publisher: Springer-Verlag.

[29] A. Roman, S. Bedi, F. Ismail, Three-half and half-axis patch-by-patch NC machining of sculptured surfaces, The International Journal of Advanced Manufacturing Technology 29 (5) (2006) 524–531.

A Analytical calculation of machining time for Surrogate 3

For a given zone, the coordinates of the points belonging to the zone are gathered and their covariance matrix is calculated. Let $\mathbf{x_I}$ and $\mathbf{x_{II}}$ be the first and second principal directions resulting from the eigen-decompostion of this covariance matrix . Let λ_I and λ_{II} be the corresponding eigenvalues ($\lambda_I \geqslant \lambda_{II}$).

The rectangular plane that best fits the zone is generated by $\mathbf{x_I}$ and $\mathbf{x_{II}}$: its width is along $\mathbf{x_I}$, and is denoted $w = \sqrt{12\lambda_I}$, while its height is along $\mathbf{x_{II}}$, and is denoted $h = \sqrt{12\lambda_{II}}$. This plane is called $\mathbf{\Pi}$ and its normal is $\mathbf{n} = \mathbf{x_I} \times \mathbf{x_{II}}$. The coordinates of $\mathbf{n}$ are denoted n_x, n_y and n_z.

Let $\mathbf{F}$ be the unit vector defining the machining direction in the $\mathbf{XY}$ plane of the machine tool, and γ the

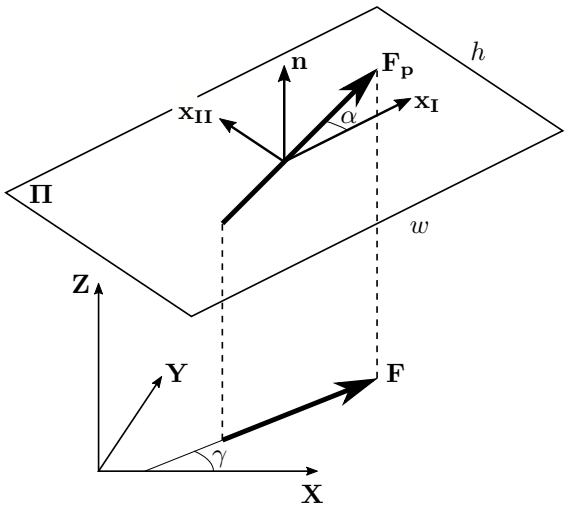


Figure 16: Analytical calculation of machining time for Surrogate 3.

angle between $\mathbf{X}$ and $\mathbf{F}$. Let $\mathbf{F_p}$ be the feedrate vector; it belongs to the plane $\mathbf{\Pi}$, and $\mathbf{F}$ is the projection of $\mathbf{F_p}$ in the $\mathbf{XY}$ plane. The angle between $\mathbf{F_p}$ and $\mathbf{x_I}$ is called α (see Figure 16).

The objective is to compute the machining time of the rectangular plane $\mathbf{\Pi}$ as a function of the machining direction γ. First, we need to express α as function of γ. The coordinates of $\mathbf{F}$ and $\mathbf{F_p}$ are:

$$\mathbf{F} = \begin{pmatrix} \cos(\gamma) \\ \sin(\gamma) \\ 0 \end{pmatrix} \text{ and } \mathbf{F_p} = \begin{pmatrix} \cos(\gamma) \\ \sin(\gamma) \\ z \end{pmatrix},$$

where z is to be determined. We know that $\mathbf{F_p} \cdot \mathbf{n} = 0$, which yields:

$$z = -\frac{n_x \cos(\gamma) + n_y sin(\gamma)}{n_z}.$$

Therefore, the angle α is given by:

$$\cos(\alpha) = \frac{\mathbf{x_I} \cdot \mathbf{F_p}}{\|\mathbf{x_I}\| \, \|\mathbf{F_p}\|}.$$

Now the machining time can be computed as a function of α. Let p be the step-over distance between two adjacent paths; p is constant for planar surfaces, and two cases are considered according to the value of α. Concerning the machine kinematics, we assume that the nominal feedrate, V_f, is achieved on each path, and not achieved for the connections between adjacent paths. The maximum acceleration, A_{max}, is never achieved.

— **1st case:** $\tan(\alpha) \geqslant \frac{h}{w}$**.** In this case, the toolpath is composed of constant-length paths in the center area (in gray on Figure 17), and remaining paths near the sides.

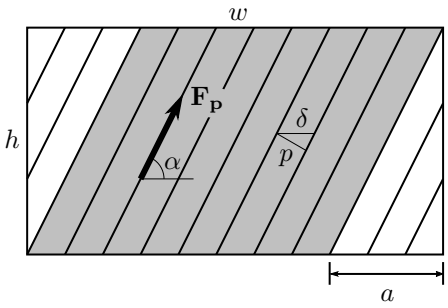


Figure 17: Computation of machining time in the 1st case.

The length, l, of all paths in the central area is given by

$$l = \frac{h}{\sin(\alpha)}$$

The number, n_1, of these paths is the integer part of n which is defined by: $n\,\delta = w - a$, where δ, the projection of p on the $\mathbf{x_I}$ axis, is given by

$$\cos\left(\frac{\pi}{2} - \alpha\right) = \frac{p}{\delta}$$

which implies

$$\delta = \frac{p}{\sin(\alpha)}$$

and a, the size of remaining side areas, is expressed as

$$\tan(\alpha) = \frac{h}{a} \Leftrightarrow a = \frac{h}{\tan(\alpha)}$$

As a result, we obtain

$$n = \frac{\sin(\alpha)}{p}\left(w - \frac{h}{\tan(\alpha)}\right)$$

and the machining time associated to this area is

$$t_1 = n_1 \left(\frac{l}{V_f} + 2\sqrt{\frac{V_f}{J}} + 4\sqrt[3]{\frac{\delta}{2J}}\right),$$

where J is the constant jerk of the machine tool.

For the side areas, paths with different lenghts have to be taken into consideration. The number of these paths, denoted n_2, is the integer part of $\frac{a}{\delta} = \frac{h\cos(\alpha)}{p}$. The length of the i^{th} path is $\frac{i\,p}{\cos(\alpha)\sin(\alpha)}$. Taking into account the connections, the machining time associated to each side area is

$$t_2 = \sum_{i=1}^{n_2} \frac{i\,p}{\cos(\alpha)\sin(\alpha)\,V_f} + 2\sqrt{\frac{V_f}{J}} + 2\sqrt[3]{\frac{\delta}{2J}} + 2\sqrt[3]{\frac{p}{2J\cos(\alpha)}}.$$

Finally, in this case, the total machining time is $t = t_1 + 2\,t_2$.

— **2nd case:** $\tan(\alpha) < \frac{h}{w}$**.** In this case the constant-length paths are ending on the narrower side of the rectangle. Apart from this difference the computation can be carried out analogously.

The length l of the central area is

$$l = \frac{w}{\cos(\alpha)}.$$

The number of these paths, denoted n_1, is the integer part of n defined by

$$n\,\delta = h - b,$$

where δ, the projection of p on the $\mathbf{x_{II}}$ axis, is defined by

$$\delta = \frac{p}{\cos(\alpha)}$$

and b, the size of remaining side areas, is expressed as

$$b = w\tan(\alpha).$$

As a result, we obtain

$$n = \frac{\cos(\alpha)}{p}\left(h - w\tan(\alpha)\right),$$

and the machining time associated to this area is

$$t_1 = n_2\left(\frac{l}{V} + 2\sqrt{\frac{V}{J}} + 4\sqrt[3]{\frac{\delta}{2J}}\right)$$

For the side areas, the number of paths, denoted n_2, is the integer part of $\frac{b}{\delta} = \frac{w\sin(\alpha)}{p}$. The length of the i^{th} path is $\frac{i\,p}{\cos(\alpha)\sin(\alpha)}$. Taking into account the connections, the machining time associated to each side area is

$$t_2 = \sum_{i=1}^{n_2} \frac{i\,p}{\cos(\alpha)\sin(\alpha)\,V_f} + 2\sqrt{\frac{V_f}{J}} + 2\sqrt[3]{\frac{\delta}{2J}} + 2\sqrt[3]{\frac{p}{2J\sin(\alpha)}}.$$

Finally, in this case, the total machining time is $t = t_1 + 2\,t_2$

B Control points of Surfaces 1 and 2

Surface 1 is a 4×4 bicubic Bézier surface whose $\mathbf{P_{i,j}}$ control points are given in Table 7. Surface 2 is a 3×3 quadratic Bézier surface whose $\mathbf{P_{i,j}}$ control points are given in Table 8.

$\mathbf{P_{0,0}}$	$\mathbf{P_{0,1}}$	$\mathbf{P_{0,2}}$	$\mathbf{P_{0,3}}$
0.0	0.0	0.0	0.0
0.0	25.4	50.8	76.2
38.1	30.48	30.48	38.1
$\mathbf{P_{1,0}}$	$\mathbf{P_{1,1}}$	$\mathbf{P_{1,2}}$	$\mathbf{P_{1,3}}$
17.78	17.78	17.78	17.78
0.0	25.4	50.8	76.2
30.48	22.86	22.86	30.48
$\mathbf{P_{2,0}}$	$\mathbf{P_{2,1}}$	$\mathbf{P_{2,2}}$	$\mathbf{P_{2,3}}$
35.56	35.56	35.56	35.56
0.0	25.4	50.8	76.2
38.1	30.48	30.48	38.1
$\mathbf{P_{3,0}}$	$\mathbf{P_{3,1}}$	$\mathbf{P_{3,2}}$	$\mathbf{P_{3,3}}$
50.8	50.8	50.8	50.8
0.0	25.4	50.8	76.2
30.48	22.86	22.86	30.48

Table 7: Cartesian coordinates the control points for Surface 1.

$\mathbf{P_{0,0}}$	$\mathbf{P_{0,1}}$	$\mathbf{P_{0,2}}$
0	0	0
0	20	40
0	10	0
$\mathbf{P_{1,0}}$	$\mathbf{P_{1,1}}$	$\mathbf{P_{1,2}}$
40	40	40
0	20	40
5	15	5
$\mathbf{P_{2,0}}$	$\mathbf{P_{2,1}}$	$\mathbf{P_{2,2}}$
80	80	80
0	20	40
20	35	20

Table 8: Cartesian coordinates the control points for Surface 2.