



**HAL**  
open science

## Milling plan optimization with an emergent problem solving approach

Sonia Djebali, Alexandre Perles, Sylvain Lemouzy, Stéphane Segonds, Walter Rubio, Jean-Max Redonnet

► **To cite this version:**

Sonia Djebali, Alexandre Perles, Sylvain Lemouzy, Stéphane Segonds, Walter Rubio, et al.. Milling plan optimization with an emergent problem solving approach. *Computers & Industrial Engineering*, 2015, 87, pp.506-517. 10.1016/j.cie.2015.05.025 . hal-03832286

**HAL Id: hal-03832286**

**<https://hal.science/hal-03832286>**

Submitted on 27 Oct 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Milling plan optimization with an emergent problem solving approach

Sonia Djebali<sup>a,\*</sup>, Alexandre Perles<sup>b</sup>, Sylvain Lemouzy<sup>b</sup>, Stephane Segonds<sup>a</sup>, Walter Rubio<sup>a</sup>, Jean-Max Redonnet<sup>a</sup>

<sup>a</sup> Institut Clément Ader de Toulouse, France

<sup>b</sup> Institut de Recherche en Informatique de Toulouse, France

---

## ARTICLE INFO

### Article history:

Received 22 March 2014

Received in revised form 25 March 2015

Accepted 22 May 2015

Available online 12 June 2015

### Keywords:

Free-form surface

Machining zones

Path length tool

Emergent problem solving

Mutli-agent algorithm

---

## ABSTRACT

With elaboration of products having the more complex design and good quality, minimize machining time becomes very important. The machining time is assumed, by hypothesis, to be proportional to the paths length crossed by the tool on the surface. The path length depends on the feed direction and the surface topology. To get an optimal feed direction at all points of surface, this study concerns machining with zones of the free-form surfaces on a 3-axis machine tool. In each zone, the variation of the steepest slope direction is lower, total path length is minimized and the feed direction is near the optimal feed direction. To resolve this problem, the Adaptive Multi-Agent System approach is used. Furthermore, a penalty reflecting the displacement of the tool from a zone to another one is taken into account. After several simulations and comparisons with the machining in one zone (what is being done at present), the results obtained present a significant saving about 22%.

---

## 1. Introduction

In various fields of activity, such as aeronautic, automotive industry or capital goods, the competition leads to the elaboration of products having a design more and more complex and increasing quality. These complex surfaces are named free-form surfaces and require a high quality level and reduced shape defects. The machining of the free-form surfaces, manufacturing moulds or matrix used to make the models is time consuming, not optimized and costly. The main factor influencing the global cost of the free-form surfaces production is the machining time. The free-form surfaces are modeled using computer-assisted design software. The associated mathematical models are parametric surfaces with poles, as NURBS, B-Spline, Bézier curves... (Faux & Pratt, 1985).

Their machining is done by material removal using Numerically Controlled Machine Tools, with hemispherical or toric end mills (Marciniak, 1992). It has been demonstrated (Senatore, Segonds, Rubio, & Dessein, 2012) that using the toric end mill cutter allows to decrease the machining time if cutting feed direction is well chosen. It is the type of tool that will be used in the present study.

There are several machining strategies: parallel planes milling (Huang & Oliver, 1994), guide surfaces (Kim & Choi, 2000), and iso-parametric milling (Loney & Ozsoy, 1987). These strategies are obtained directly from 3-axis machining researches. The one that's used in this study is the parallel planes milling, it's the most used and mastered in the industry. It consists in determining the tool paths using the intersection between the work piece and the parallel planes oriented along one machining direction. Advantages of parallel planes milling strategy are:

- Do not generate an overlapping tool path, allowing a considerable time saving.
- To avoid the appearance of non-machined areas.

The parallel planes milling strategy is not optimal. Indeed, during the machining of the surfaces with a large variation of the normal direction, the successive paths become nearer to respect scallop height criteria, thus increasing manufacturing time. When a tool moves on the surface, it sweeps a volume by leaving an imprint on this surface, it's a swept surface (Steiner, Peternell, Pottman, & Zhao, 2005). Between two adjacent paths, the intersection of the swept surface produces a scallop. Usually, a maximum given scallop height criteria must be respected.

The main object of this study is to minimize the machining time of free-form surfaces while respecting the quality imposed by the engineering consulting firm, by 3-axis machining with toric end mill cutter. A considerable number of works have been devoted

---

\* Corresponding author.

E-mail addresses: [Sonia\\_djebali@yahoo.fr](mailto:Sonia_djebali@yahoo.fr) (S. Djebali), [perles@irit.fr](mailto:perles@irit.fr) (A. Perles), [Sylvain.Lemouzy@irit.fr](mailto:Sylvain.Lemouzy@irit.fr) (S. Lemouzy), [stephane.segonds@univ-tlse3.fr](mailto:stephane.segonds@univ-tlse3.fr) (S. Segonds), [walter.rubio@univ-tlse3.fr](mailto:walter.rubio@univ-tlse3.fr) (W. Rubio), [redonnet@lgmt.ups-tlse.fr](mailto:redonnet@lgmt.ups-tlse.fr) (J.-M. Redonnet).

to reduce machining time (Chen, Vickers, & Dong, 2003; Maeng, Ly, & Vickers, 1996). Indeed, the first path is calculated according to the optimal feed rate direction, then secondary paths are calculated, respecting a fixed step-over distance that guaranties a maximal scallop height, inferior to the imposed scallop height criteria. The problem is that secondary path deviate rapidly and may not be in optimal direction engendering time loss.

Due to the complexity of the machining of free form surfaces respecting a scallop height criterion in a minimal tool path length, and since the choice of a unique machining direction for the whole surface could not be efficient, a new approach is detailed in the following paper. The main innovation consists in using the best local advantages of both toric end mill use and parallel planes strategy by extending it to a whole surface by zoning it in several zones. The zones will be machined using a specific optimal feed rate direction. Furthermore, a penalty reflecting the displacement of the tool from a zone to another one is taken into account. The problem appearing during the machining of the zones is the determination of the number and geometry of zones. Indeed:

- The number of zones should not be too high to avoid spending too much time going from a zone to another one.
- The number of zones should not be too small to keep the feed rate direction not so far from optimal value in every point of the zone.

To minimize the global machining time of a given surface, this study concerns the optimization of the number and geometry of the zones. This paper is organized as follows: Section 2 presents the notions related to the machining strategy that will be used in the optimization problem definition presented in Section 3. After having reviewed the main optimization methods that could be used, Section 4 presents their limits in the context of this paper and will argue the choice of the Adaptive Multi-Agent System (AMAS) approach. Section 5 presents the multi-agent system implemented as well as behavior of the agent. Finally, after having presented in the part Section 6 the experimental results and their analysis, Section 7 concludes and mentions the perspectives of improvement that are envisaged.

## 2. Reminder

### 2.1. Machining direction

The surfaces studied in this paper are the free-form surfaces, they present a set of curved zones. The geometry is obtained by successive passes of a tool. For this study the toric end mill with cutter radius  $R$  and torus radius  $r$  is used (Fig. 1), and the adopted machining strategy is the parallel planes of type one-way. During the machining, tool moves tangentially to the workpiece at the contact point  $C_c$ . At this point the tool can move in any direction  $\alpha$ . This parameter  $\alpha$  is called feed motion direction, it is directly influenced by the topology of the surface. The optimal direction

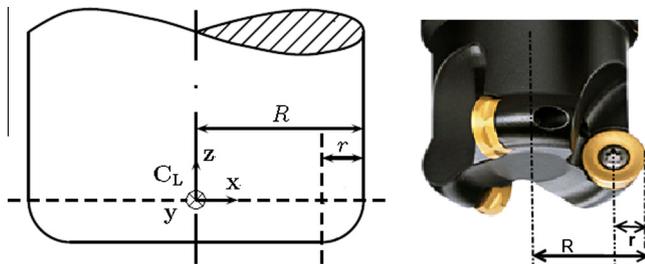


Fig. 1. The toric tool.

at a given point of the surface is the direction of steepest slope. For example, let be an inclined plane ( $S = 30^\circ$ ) and a toric end mill cutter ( $R = 5 \text{ mm}, r = 2 \text{ mm}$ ) presented partially by a quarter of torus. Fig. 2 illustrates two cases of movement of this tool. The blue-lined curve presents the trace left by the tool on the workpiece at  $C_c$ .

1. During the movement of the tool following a direction perpendicular to the steepest slope direction ( $\alpha = 90^\circ$ ), the trace left by the tool at contact point  $C_c$  is a curve with a radius of curvature greater than  $R$ .
2. During the movement of the tool along the horizontal direction ( $\alpha = 0^\circ$ ), the trace radius of curvature left by the tool is equal to  $r$ .

For the same path length, the quantity of material removed in the case (1) is much more important than in the case (2), then the number of paths required to machine the whole surface in the case (1) is inferior than in the case (2). So, the time required to machine the surface in case (1) is less than the machining time for the same surface in case (2).

The previous example illustrates that the choice of the feed direction has an important role on the radius of the trace left by the tool on the surface. The more the radius of this trace is important and the more the machining time decreases. The radius of curvature of the trace left by the tool at  $C_c$  is called effective radius  $R_{eff}$ . The toric end mill enables to keep a large effective radius while avoiding unsightly marks.

### 2.2. Effective radius

At the contact point  $C_c$ , the trace left by the tool on the surface is the swept curve. Effective radius corresponds to the radius of curvature of the swept curve projected in a plane normal to the feed direction along a direction parallel to the feed direction. Redonnet, Djebali, Segonds, Senatore, and Rubio (2013) demonstrates that, at contact point  $C_c$ , the effective radius is equal to the ellipse radius - resulting from the projection along the feed direction in a plane normal to the feed direction of the center-torus circle-increased by an exterior offset with a value equal to the torus radius of tool  $r$ . This gives us the analytical equation of the effective radius:

$$R_{eff} = \frac{(R - r) \cos(\alpha - \varphi_{cc})^2}{\sin(S)(1 - \sin(\alpha - \varphi_{cc})^2 \sin(S)^2)} + r \quad (1)$$

With:

- $\begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix}$ : the normal of the surface at the contact point  $C_c$ .
- $\varphi_{cc} = \arctan\left(\frac{n_y}{n_x}\right)$ : steepest slope direction.
- $S = \arccos(n_z)$ : the slope angle of the workpiece at a given point.
- $Z_s$ : tool axis.
- $\alpha$ : the feed direction angle relatively to  $x$  axis.

This analytical expression of  $R_{eff}$  is easy to handle. It depends on the feed direction angle  $\alpha$  and on the slope angle (Fig. 3).

### 2.3. Step-over distance and toric end mill conditions

Step-over distance  $P_t$  corresponds to the distance between two successive and parallel tool paths while not overpassing the imposed scallop height criteria  $h_c$ . Generally  $h_c$  is in the order of 0.01 mm, then the following hypothesis are made:

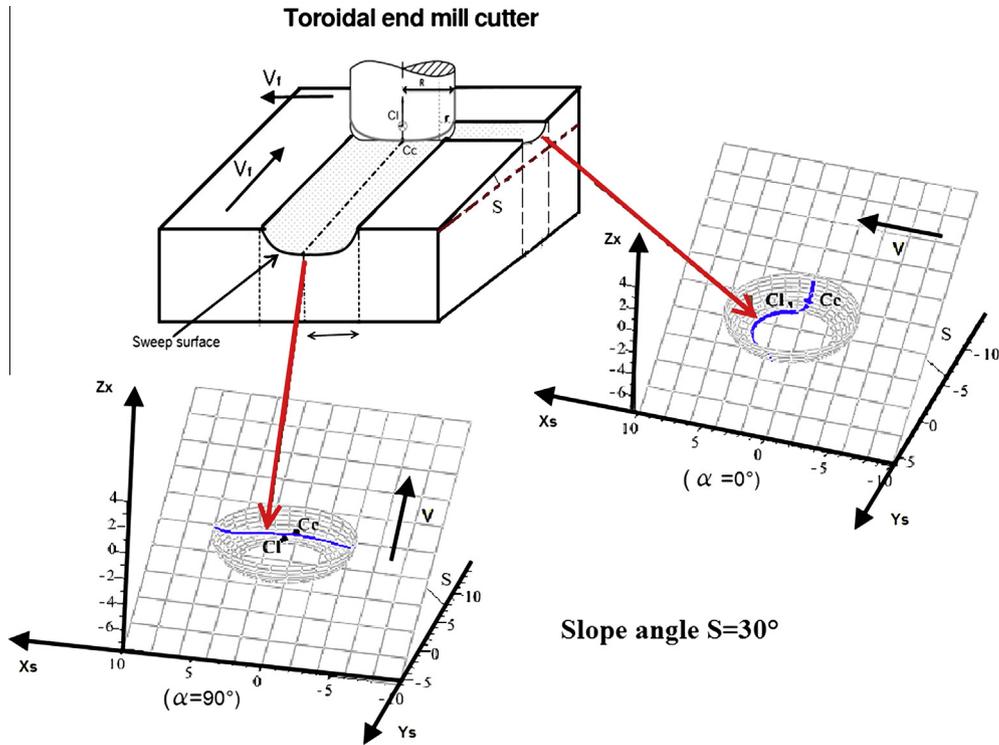


Fig. 2. Trace left by the tool on the workpiece along the feed rate direction.

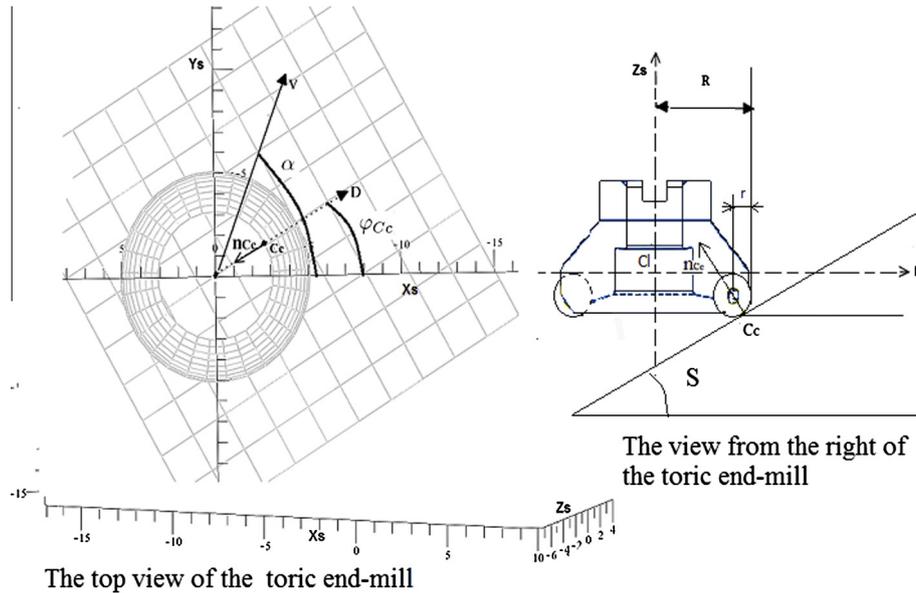


Fig. 3. Different angles of the effective radius.

- The tool makes a circular trace with radius equal to  $R_{eff}$  in the neighborhood of  $C_c$ .
- The radius of curvature  $\rho$  of the surface in a plane perpendicular to the feed rate direction is assumed to be locally constant.
- $h_c$  is smaller than  $R_{eff}$  and  $\rho$ .

The calculation of the step-over  $P_t$  distance (Fig. 4) is made in the perpendicular plan to the feed direction, it is evaluated by:

$$P_t = \sqrt{\frac{8 * h_c * R_{eff} * (R_{eff} + \rho)}{\rho}} \quad (2)$$

The effective radius has a direct influence on machining time. Indeed, when the effective radius increases the step-over distance increases too. The number of paths required to machine the surface decreases significantly and reduces the machining time.

Using a toric tool is relevant only when allowing machining time reduction compared to spherical tool use. Also, this condition is satisfied when the effective radius of the toric tool is greater than  $r$ . This is true when the difference between the steepest slope direction and the machining direction belongs to an interval  $I$  depending on the slope angle of the surface to be machined. When the slope angle is low, the toric end mill cutter is

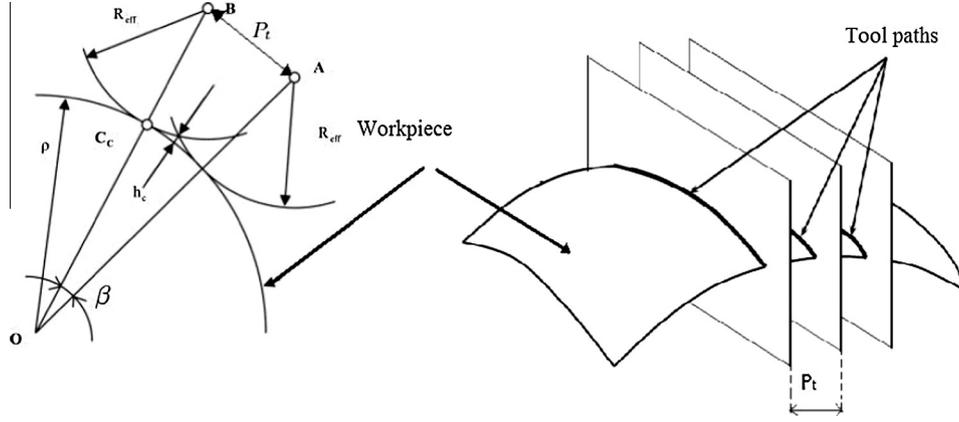


Fig. 4. Step-over distance.

recommended because the effective radius is important in a large feed direction interval.

The more the slope angle increases, the more the maximal effective radius decreases and rapidly reaches the torus radius  $r$  and more the interval  $I$  is reduced, in this case the use of the spherical tool is recommended.

To define this angular interval the equation  $R_{eff} = R$  is solved:

$$\alpha = \arcsin \left( \frac{1}{\sqrt{\sin(S) + \sin^2(S) + 1}} \right) \quad (3)$$

The interval  $I$  is defined then as follows:

$$I = \left[ -\arcsin \left( \frac{1}{\sqrt{\sin(S) + \sin^2(S) + 1}} \right), \arcsin \left( \frac{1}{\sqrt{\sin(S) + \sin^2(S) + 1}} \right) \right]$$

$$I = [-35^\circ, 35^\circ], \forall S \in [-90^\circ, 90^\circ]$$

Observation: this interval is only a function of the slope angle and it's not dependent on tool dimensions  $R$  and  $r$ .

### 3. The optimization problem

#### 3.1. Problem formulation

Let  $\mathbf{S}(u, v)$  be a parametric free-form surface,  $u$  and  $v$  being the parameters of this parametric surface. At any point of this surface the optimal feed direction corresponds to the steepest slope direction. Machining the surface with a single feed direction cannot ensure the maximum effective radius at any point, because of the variation of the normal direction on the surface. Indeed, an optimal feed direction at one point of a path can be very far from the optimal feed direction at another point of the same path. This induces decreasing of the average value of the effective radius, contracting the distance between the parallel planes and increasing machining distance.

In order to allow a machining with a near optimal feed rate direction at any point, the surface  $\mathbf{S}(u, v)$  is mapped with a parametric meshing following  $(u, v)$ . Each mesh  $m_i \in \mathcal{M}$  presents an elementary surface, with a low variation of the normal direction and then an optimal feed direction. Each mesh has a fixed size, determined during the meshing, an average slope angle  $S_{m_i}$  and an optimal machining direction  $\alpha_{m_i}^{opt}$  associated to the maximum effective radius on the mesh  $R_{eff_{m_i}}^{opt}$ .

If the crossing time of the tool from one mesh to another one is neglected, the optimal solution to machine the mapped surface is

to machine each mesh independently following its optimal direction. But, in practice this approximation is not acceptable. For this, the meshes will be combined into zones, reducing then the unproductive time of the tool. A zone is a composition of a set of associated meshes. Two meshes can be associated if they are contiguous, meaning that they have at least one border in common.

The optimization problem studied in this paper relates to the minimization of the total distance crossed by the tool, respecting the constraint of an imposed maximum scallop height. The optimization parameters are the geometry of the zones and the feed direction used in each one.

More formally, the problem of minimization of the machining time is formulated as follows:

$$\text{Min} \sum_{i=0}^{|Z|} Lg_{z_i}^{\alpha_{z_i}^{opt}} + (|Z| - 1) \times \Delta L \quad (4)$$

With:

- $|Z|$  number of zones covering the whole surface.
- $Lg_{z_i}^{\alpha_{z_i}^{opt}}$  the path length crossed by the tool to machine the zone  $z_i$  following the optimal direction  $\alpha_{z_i}^{opt}$ .
- $\Delta L$  penalty due to the withdrawal of the tool outside the surface (material), while moving from a zone to the following one.

The calculation of the path length crossed by the tool to machine the zone  $z_i$  following the optimal direction is very costly. Indeed, it requires:

1. The use of the simplex algorithm to calculate  $\alpha_{z_i}^{opt}$ .
2. The calculation of the total distance crossed by the tool on the zone.

Because of the combinatorial explosion of this optimization problem  $|Z|$  evolves more than exponentially according to the fineness of the discretization the surface. So, it's not envisaged to resolve the problem on the basis of the minimization of length path.

#### 3.2. Local approach

In order to overcome this combinatorial problem, we have chosen an indirect method, that relies on several local and lightweight optimizations rather than a global and extremely time consuming optimization. To keep consistency with our goal, these local optimizations must rely on criteria that are coherent and correlated with the global optimization.

As already said in the previous sections, maximizing the effective machining radius  $R_{eff_{m_i}}$  over all the meshes  $m_i$  of a given zone will increase the step-over distance and consequently may reduce the machining path. However, the normals of all the meshes belonging to a zone are not necessarily equal. This implies that more the  $\alpha_{m_i}^{opt}$  and  $S_{m_i}$  of each meshes are close over a zone, more the  $R_{eff_{m_i}}$  will be the same for any machining direction. More precisely, as shown in Fig. 5, the more  $S_{m_i}$  approach to  $0^\circ$ , the more  $m_i$  could be efficiently machined with a wide range of direction –  $R_{eff_{m_i}}^x$  being greater than  $R$  for a wide range of  $\alpha$  values. Consequently a zone  $z_i$  will be efficiently machined if it is composed of meshes  $m_i$  that have a very close  $\alpha_{m_i}^{opt}$  or have a  $S_{m_i}$  value sufficiently close to  $0^\circ$  to keep a large effective radius whatever the machining direction  $\alpha_{z_i}^{opt}$  is used in the zone. The first local criteria will then be to *define zones  $z_i$  composed of meshes that have similar  $\alpha_{m_j}^{opt}$  or  $S_{m_j}$  that let them keep compatibility with  $\alpha_{z_i}^{opt}$ .*

On the other hand, and depending on the milling machine characteristics, the time spent during the transition from machining a zone to another is generally significant, so the less the number of zones is, the less the machine will spend time out of matter (in other words, the shortest the machining path out of matter will be). This criterion can be formulated in term of size of a zone as follows: the larger are the zones, the smaller the number of zone is. Thus, the second local criterion will be to *define zones that are as big as possible.* An optimal solution for a given milling machine should be obtained thanks to a balance between those two criteria that depends on the machine characteristics.

In order to design a solver for this problem formulation, we chose to use the Adaptive Multi-Agent System (AMAS) approach that is especially suited to reach such kind of equilibrium between the satisfaction of several, eventually opposite, local criteria.

#### 4. The AMAS approach

The Adaptive Multi-Agent System (AMAS) approach aims at building artificial systems which reach their functional adequacy in their environment while agents composing them only seek to pursue their individual objectives and do not have a global view of the system. This approach is an organizational one, where the designer of the application has only to focus on the agent level. He has only to define when and how an agent has to locally decide to change its interaction links with other agents, in order to reach an organization giving rise to the intended global function.

To reach this functional adequacy, it had been proven that each autonomous agent composing the system has to keep relations as cooperative as possible with its social (other agents) and physical environment. In order, to lead a coherent collective behavior, whereas agents only seek to achieve individual goals, techniques of cooperation avoiding failure such as conflict, competition were developed. These failures are called “Non Cooperative Situations” (NCS) and can be assimilated to “exceptions” in traditional programming. The definition of cooperation we use is more not a conventional one (simple sharing of resources, common work, etc.).

Our definition is based on three meta-rules the designer has to instantiate according to the problem to be solved: An agent is in a (NCS) when:

- Meta-rule 1: ( $\neg C_{per}$ ) a perceived signal is not understood or is ambiguous;
- Meta-rule 2: ( $\neg C_{dec}$ ) perceived information does not produce any new decision;
- Meta-rule 3: ( $\neg C_{act}$ ) the consequences of its actions are not useful to others.

The adopted approach is proscriptive because agents have, first of all, to anticipate, to avoid or to repair the NCSs. Such NCS occurs when at least one of the three previous meta-rules is not locally proved correct by an agent. Different generic NCSs can then be highlighted: *incomprehension* or *ambiguity* if ( $\neg C_{per}$ ) is not checked, *incompetence* or *unproductiveness* if ( $\neg C_{dec}$ ) is not obeyed and finally *uselessness*, *competition* or *conflict* when ( $\neg C_{act}$ ) is not checked. Such detection involves a local reorganization between involved agents.

This local and cooperative approach has great methodological implications: designing an AMAS consists in defining and assigning cooperation rules to agents. In particular, the designer, according to the problem to solve, has (i) to define the nominal behavior of an agent, then (ii) to deduce the NCSs the agent can be confronted with, and (iii) finally to define the actions, the agents have to perform to come back to a cooperative state.

The ADELFE methodology (Bernon, Camps, Gleizes, & Picard, 2005) allows engineers to design AMAS software in which self-organization makes the solution emerge from the interactions of their parts. It also gives some hints to the designer to tell him if using the AMAS theory is relevant to build his application. ADELFE mainly focuses on the identification of agents, on the detection of all the NCS that may appear during the system functioning and then on the definition of the actions the agents have to perform to come back to a cooperative state.

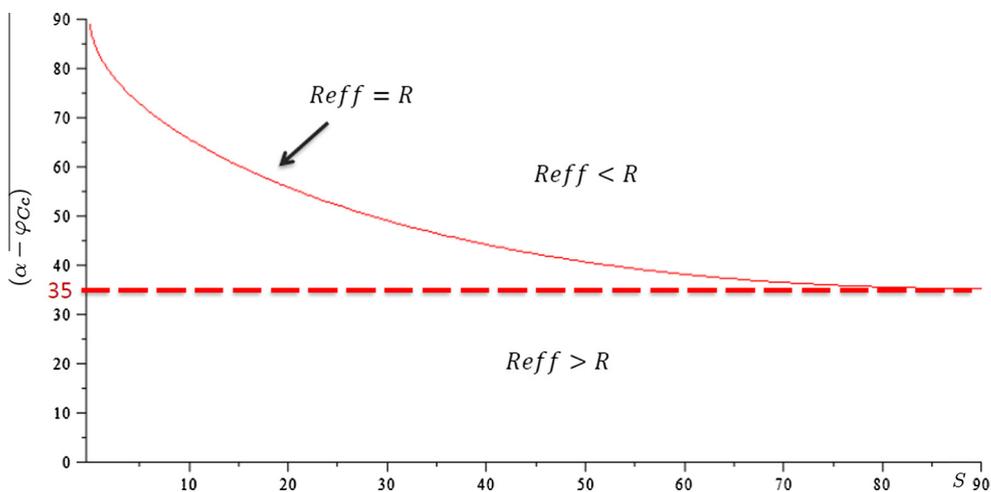


Fig. 5. Slope angle according to  $(\alpha - \varphi_c)$ .

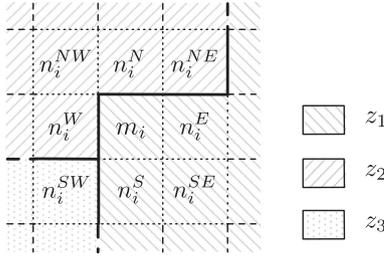


Fig. 6. Mesh agent neighborhood and zones.

In the following section, we will detail the main ADELFE task results in order to introduce the proposed multi-agent algorithm.

## 5. Application to milling plan optimization

### 5.1. Agent identification

The ADELFE method requires the system designed to identify the entities of the system and to define the entities that will act as agents. We have identified two kind of entities:

1. *zone*: an entity that represents a partition of the surface to machine. It is composed of a collection of connected meshes (at least one mesh). Let  $\mathcal{Z}$  be the set of all possible zones. A zone  $z_i \in \mathcal{Z}$ , is characterized by the number of meshes it contains  $z_i \cdot \text{size}$  and an optimal machining angle  $\alpha_{z_i}^{opt}$  (that is intended to lead to the shortest machining path over the zone). Because this angle  $\alpha_{z_i}^{opt}$  needs an optimization procedure to be figured out, it is not always known by zones. In our algorithm, we will not use this information but rely on correlated informations that does not need such a loud computation procedure.
2. *mesh*: an atomic element of the surface to machine, it is a quadrilateral sufficiently small to be supposed being flat. It is characterized by a slope angle  $S_{m_i}$ , an optimal machining direction  $\alpha_{m_i}^{opt}$  and belongs to a zone  $z_i$ .

Because the local criteria we want to satisfy can be both manipulated at the mesh level, there is only one type of agent that is the *mesh agent*. Zone will be only considered as passive entities, which can be created and manipulated by *mesh agents*. A *mesh agent*  $m_i$  inherits the characteristics of *mesh* entities they represents. Moreover, it has up to eight neighbours that it can perceive their respective characteristics. Let note  $n_i^j$  with  $j \in D = \{N, NE, E, SE, S, SW, W, NW\}$  the neighbour of the mesh  $m_i$  which is situated in the direction  $j$  from  $m_i$  (cf. Fig. 6). Finally, a *mesh agent* can act by changing the zone it belongs to. More precisely, it can execute the following actions:

1. Quit its current zone and join one of it's neighbour's zone. Because a zone is only composed of strictly connected meshes (i.e. that share one edge), a *mesh agent* can only join the zone of it's  $N, E, S, W$  situated neighbour (i.e. that belongs to  $\{n_i^d | d \in \{N, E, S, W\}\}$ ). Thus, up to four different actions of this class is possible depending on its neighborhood:  $join^N, join^E, join^S, join^W$ . Naturally, if the mesh is situated at the south east corner of the surface, it will just be able to execute  $join^S$  and  $join^W$ .
2. Quit its current zone and join a new zone only composed of itself: *create*.

We note  $\mathcal{A} = \{join^N, join^E, join^S, join^W, createZone\}$  the set of possible actions.

Thus, through the actions of each *mesh agent* a given topology of zones will be reached from the agent's collective activity.

### 5.2. Agent's local goals

The local goal of a *mesh agent* being directly related to the local criteria we want to optimize (cf. Section 3.2), we can easily define that a *mesh agent* will try to act in order to reach a local optimal equilibrium between the two following local criteria:

#### Closest Machining Directions (CMD):

All perceived zones contains *mesh agents*  $m_i$  with as close as possible machining direction (i.e.  $\alpha_{m_i}^{opt}$ ).

#### Largest Zones (LZ):

All perceived zones contains as much as possible meshes.

Fig. 7, illustrates 3 different zone organizations through a coarse grained discretized surface (9 meshes). The first one, shows a surface with a zone for each mesh (double directed arrows stand for the optimal machining direction of meshes  $\alpha_{m_i}^{opt}$ ). The second one shows 3 zones composed of 3 meshes (meshes 1, 4 and 7 for the zone 1; meshes 2, 5 and 8 for zone 2 and meshes 3, 6 and 9 for zone 3). And the last one illustrates a single zone composed of all the 9 meshes. These figures will be used in the following paragraphs to clarify how the agents will evaluate the two local criteria.

#### 5.2.1. Closest milling directions criterion

A zone is able to compute the minimal angle (sector size) that contains all optimal machining directions  $\alpha_{m_i}^{opt}$  of the meshes it is composed of.

Fig. 8 illustrates some possible sectors that contain all the directions of the first zone of Fig. 7b. In that case, the minimal angle is obviously  $\alpha_1$ . The minimal sector angle of a zone  $z_i$  is noted  $z_i \cdot \gamma \in [0 \dots \pi]$ . Also  $z_i \cdot e_1$  and  $z_i \cdot e_2$  are the two extreme directions that define this sector. We have then:

$$\gamma_i = e_i^2 - e_i^1$$

A *mesh agent* can perceive  $z_i \cdot \gamma, z_i \cdot e_1$  and  $z_i \cdot e_2$  from all its neighbours zones. Thus, the smaller  $z_i \cdot \gamma$  is, the more the *Closest Milling Direction* criterion is satisfied. Because this value is defined over  $[0 \dots \pi]$ , *mesh agents* compute the satisfaction of this criteria as follows:

$$CMD(z_i) = 1 - \frac{z_i \cdot \gamma}{\pi} \text{ with } CMD(z_i) \in [0 \dots 1]$$

When  $CMD(z_i) = 1$  the criterion is completely satisfied. On the contrary, when  $CMD(z_i) = 0$  the criterion is as far as possible from its objective.

In order to illustrate the evaluation of this criteria:

- In Fig. 7a all the zones have  $CMD(z_i) = 1$ .
- In Fig. 7b the middle zone is only composed of meshes having the same machining angle  $CMD(z_2) = 1$ ; whereas the two side zones have approximately:  $CMD(z_1) = CMD(z_3) = \frac{4}{5}$  (as  $z_1 \cdot \gamma = z_3 \cdot \gamma = \frac{\pi}{5}$ ).
- In Fig. 7c the single zone has approximately  $CMD(z_1) = \frac{1}{3}$  (as  $z_1 \cdot \gamma = 2 \cdot \frac{\pi}{3}$ ).

We can also notice that, even though the notion of mesh slope could be helpful in order to evaluate if a given set of meshes could be machined in an efficient way, we have not yet used this information in the agent behaviors for the sake of simplicity. Taking

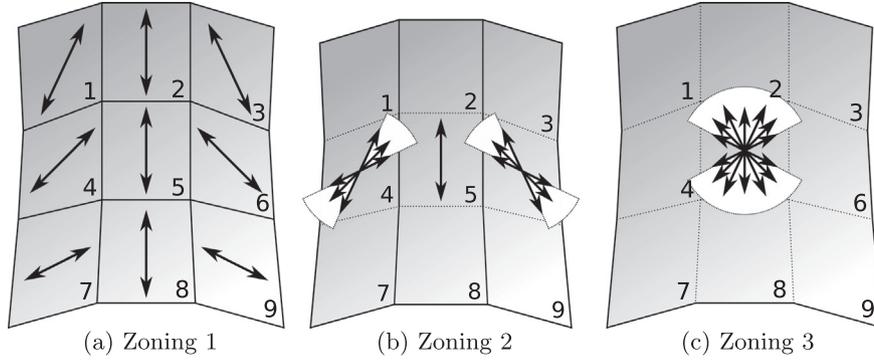


Fig. 7. Zone organization.

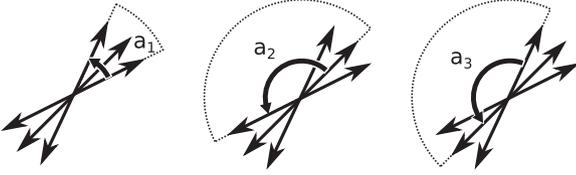


Fig. 8. Example of sectors that contains all machining direction.

into account the slope will be discussed in the perspective part of this paper.

### 5.2.2. Largest Zone criterion

In order to compute the *Largest Zone* criterion for a zone  $z_i$   $LZ(z_i)$ , the *mesh agents* are able to perceive  $z_i.size$ , the number of *mesh agents* a zone is composed of. The greater this value is, the more the criterion is satisfied. Thus, the satisfaction of *Largest Zone* criterion for the zone  $z_i$  is defined as follows:

$$LZ(z_i) = \frac{z_i.size}{|\mathcal{M}|}$$

with  $|\mathcal{M}|$  being the total number of meshes on the surface to machine. As a zone cannot be empty, thus  $z_i.size \in \{1; \dots; |\mathcal{M}|\}$ . Consequently:

$$LZ(z_i) \in \left\{ \frac{1}{|\mathcal{M}|}; \frac{2}{|\mathcal{M}|}; \dots; 1 \right\}$$

As  $CMD(z_i)$ , when  $LZ(z_i) = 1$  the criterion is completely satisfied. When  $LZ(z_i) = \frac{1}{|\mathcal{M}|}$  the criterion is as far as possible from its objective.

In order to illustrate the evaluation of this criteria:

- In Fig. 7a all the zones have  $LZ(z_i) = \frac{1}{9}$ .
- In Fig. 7b all the zones have  $LZ(z_i) = \frac{5}{9} = \frac{1}{3}$ .
- In Fig. 7c the single zone has approximately  $LZ(z_1) = \frac{9}{9} = 1$ .

### 5.3. Agent's cooperative behavior

In order to define the behavior of the *mesh agent* the AMAS approach requires to define which kind of local situation agents will try to hold on. These searched situations are called *cooperative* ones. Thus defining what is a *cooperative* situation will help what is a *non cooperative* one and how agents can recover a *cooperative* situation.

A *mesh agent* is in a *cooperative situation*, if there is not any action it can execute that will lead to a better situation (in term of satisfaction of the *CMD* and *LZ* criteria).

Thus, a *mesh agent* is in a *non cooperative situation*, if there is at least one action it can execute that will lead to a better situation.

We can notice that it is the only kind of NCS that could appear in this system. This kind of NCS belongs to the  $-c_{act}$  class of NCS introduced in Section 4.

#### 5.3.1. Definition of a situation

From these two criteria, a *mesh agent* is able to evaluate the satisfaction level of a given *situation*. From the point of view of a *mesh agent*, a *situation* is the representation of its environment at a given instant. Let be  $\mathcal{S}$  the set of all the situations a mesh could encounter. A situation  $s \in \mathcal{S}$  is composed of the representation of the zones that could be perceived by  $m_i$ . Thus a situation is defined as a subset of the all possible zone representations:  $s \subset \mathcal{Z}$ . Actually, Fig. 6 describes a situation for the *mesh agent*  $m_i$ .

#### 5.3.2. Evaluation of a situation's satisfaction

Although it is possible to find sometime an action that will increase the satisfaction of both criteria, this is not always the case. Indeed, some situations imply that increasing the size of a zone  $z_i$  – that higher  $LZ(z_i)$  – will generally lead to add a new mesh that will increase  $z_i \cdot \gamma$  and mechanically lower the satisfaction of  $CMD(z_i)$ . Moreover, when a *mesh agent* leave a given zone to join another one, it will increase the size of the joined zone but it will also mechanically decrease the size of the left zone. For all these reasons, *mesh agents* must be able to compare the satisfaction of the both criteria for each zone it perceives.

If the target machine loses a lot of time out of matter when it changes its milling direction, we had better take into account the “Largest Zone” criterion before the “Closest Machining Direction” one. On the contrary, if the machine is very quick to change its direction, we will choose the inverse order of preference. Thus, depending on the target machine characteristics we normalize each criteria with an appropriate coefficient as follows:

$$CMD_n(z_i) = CMD(z_i) \cdot k_{CMD}$$

$$LZ_n(z_i) = LZ(z_i) \cdot k_{LZ}$$

with,

- $k_{CMD} \in [0 \dots 1]$  being the normalization coefficient for the “Closest Machining Direction”,
- $k_{LZ} \in [0 \dots 1]$  being the normalization coefficient for the “Largest Zone” criterion.

Given values of these coefficients, will lead to more or less optimal solutions depending on the target machine characteristics. These coefficients have to be set by the user and could be modified at runtime during the resolution. In this study we have not

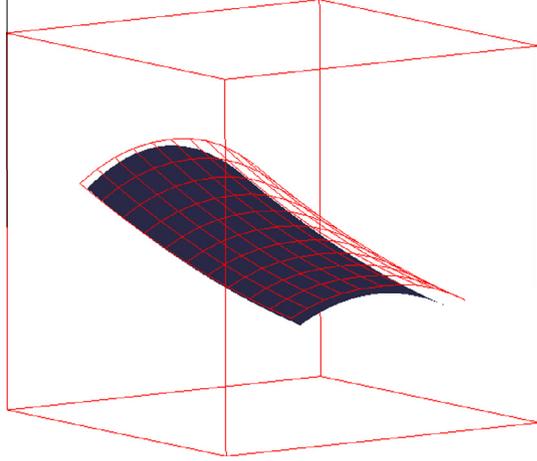


Fig. 9. The tile surface.

addressed the problem of the definition of such optimal coefficient values.

In order to maximize the satisfaction of the both normalized criteria, the situation satisfaction – noted  $Sat(s)$  – will be defined from the worst satisfied criterion:

$$Sat(s) = \min_{z \in S} (\min(CMD_n(z), LZ_n(z)))$$

### 5.3.3. The mesh agent cooperative behavior

In order to reach a *cooperative situation* as defined above, a *mesh agent* will try to act to reach a situation that maximize its satisfaction. Thus, *mesh agents* need a special skill that allows to compute the probable situation resulting of a given action execution. More precisely, it computes the resulting zones, their probable size and their probable  $z_i \cdot e1$  and  $z_i \cdot e2$  value. This skill is noted as the following function:

$$FutureSituation : S \times \mathcal{A} \rightarrow S$$

For the sake of simplicity, the way the probable future situation is figured out is not introduced in this paper. Yet, it is important to notice that, this skill only relies on the local and limited agent's perceptions and representations. Consequently the predicted resulting situation could not exactly match with the effective resulting situation. As we have noticed in our experimentations (cf. Section 6), those potential errors don't seem to have an impact on the collective behavior (as they seem to occur quite rarely).

Thanks to this skill, the agent will choose the action that will probably lead to the situation that maximize the satisfaction. The function that will make this choice is defined as follows:

$$NextAct : S \rightarrow \mathcal{A}$$

$$s \mapsto \arg \max_{a \in \mathcal{A}} (Sat(FutureSituation(s, a)))$$

In the case where there is more than one action  $a$  that maximize the satisfaction, the agent will choose randomly one of these actions. This choice implies that the obtained system has a non-deterministic behavior.

We can then sum up the life cycle of a *mesh agent* as follows:

1. *perception*: the agent updates its representation of the current situation.
2. *decision*: the agent decides the next action to execute thanks to the function *NextAct*.
3. *action*: the agent executes the decided action.

Thanks to the execution of this life cycle for each *mesh agent* of the surface, the obtained collective behavior results to a gradually evolving zone topology which finally reaches a steady state when each agent cannot increase locally its situation satisfaction. In other words and from a more global point of view, this steady state is reached when  $CMD_n$  and  $LZ_n$  has reached an optimal equilibrium among all zones (with respect to the user defined normalization coefficients  $k_{CMD}$  and  $k_{LZ}$ ).

We have implemented this multi-agent system so that allows the user:

- to control its running process of the system (run a step, run or pause);
- to view in real time the topology of the zones over the surfaces (cf. Fig. 11b);
- to adjust in real time the values of  $k_{CMD}$  and  $k_{LZ}$ .

This system has been also used to achieve the experimentations introduced in the following section.

## 6. Experimentations and analysis

Our experimentations have been made on two different surfaces. First of all, a tile (Fig. 9) as it is easy to understand what kind of zoning seems to be consistent. Next, a water blade (Fig. 10) which have a more complex structure than the tile. The tile is a  $2 \times 2$  degree Bézier surface included in a  $40 \times 80 \times 30$  mm ( $x \times y \times z$ ) volume, while the water blade is a  $7 \times 7$  degree Bézier surface included in a  $160 \times 150 \times 120$  mm ( $x \times y \times z$ ) volume. For both of them, the maximum scallop height criterion  $h_c = 0.01$  mm.

And then, the presented approach has been applied to a reference study case presented in (Choi, 2004). The machining of this surface (see here after for reminder of the equation) is optimized studied using a single zone with a ball end mill of radius  $R = 3.175$  mm ( $1/8^\circ$  of inch), the maximum permitted scallop height criterion is  $h_c = 0.254$  mm ( $0.01$  in.).

In this analysis, we talk about “Solution Quality”. This one is evaluated for a given zoning and for a given surface. This evaluation is made by simulating the tool path will make on the surface if we do mill it. The solution quality obtained is the length of this path. For a  $n$  zones cutting up, we evaluate for each one the distance to cross then we add a constant multiplied by  $n - 1$  to take into account the penalty  $\Delta L$  caused by changes of direction (cf. Eq. (4)). First, we have studied the efficiency of such a method compared to a milling made by maintaining the same direction during the whole milling process. Then, we have tried to evaluate the complexity of the algorithm by observing the evolution of the necessary time to find a solution relatively to the meshing thinness. And finally, we managed to evaluate the stability of the obtained solution by varying the meshing thinness to observe the evolution of the solution quality.

These experimentations have been made with a toric end mill cutter where  $R = 5$  mm and  $r = 2$  mm for the tile and where  $R = 10$  mm and  $r = 4$  mm for the water blade.

### 6.1. Comparison with a straightforward method

In this experimentation, we have attempted to compare the quality of the solutions obtained by the system and a straightforward method. The compared solutions are the following:

1. the “straightforward” method solution: obtained by defining a unique zone composed of all the meshes with the milling direction set to a specific optimal value for the whole surface;

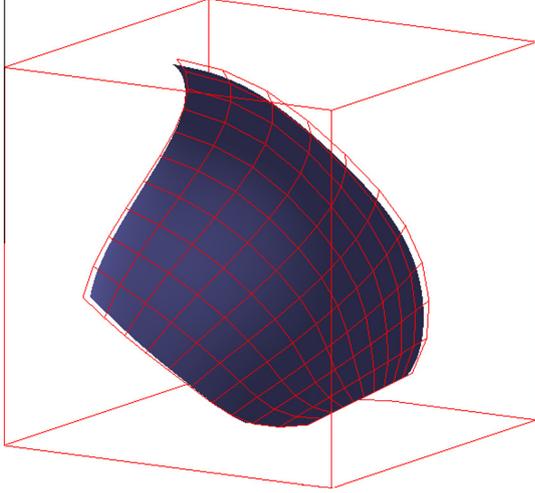


Fig. 10. The water blade surface.

2. the fast direction changing milling machine optimized plan, called the machine 1 solution: obtained from a  $20 \times 20$  zoning with our MAS algorithm where  $k_{CMD} = 1$  and  $k_{LZ} = 0.15$ .
3. the slow direction changing milling machine optimized plan, called the machine 2: obtained from a  $20 \times 20$  zoning with our MAS algorithm where  $k_{CMD} = 1$  and  $k_{LZ} = 0.35$ .

Figs. 11a and b and 12a and b represent the zoning solutions respectively obtained for the machine 1 and the machine 2. As expected, the fast direction changing milling machine optimized plans contains more zones (6 for the tile and 8 for the water blade) than the slow direction changing milling machine optimized plans (3 for both surfaces).

$\Delta L$  is a way to represent the time wasted by the machine to change the machining direction. The quality of compared solutions is evaluated thanks to two measures, each one corresponding to one machine characteristic:

1. Machine 1 path length: path length measure obtained with  $\Delta L = 40$ .
2. Machine 2 path length: path length measure obtained with  $\Delta L = 300$ .

Obtained results are introduced by Tables 1 and 2.

By observing these results, we observe a noticeable benefit of performance, about 22%, whatever it is for a fast direction changing machine or not. This demonstrates the relevance of our solution and the gain that we can benefit with the proposed method.

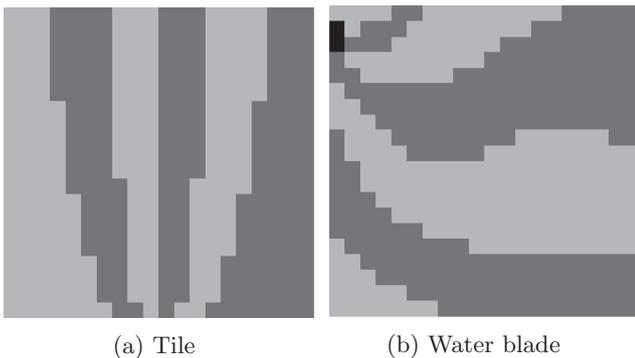


Fig. 11. Machine 1 solutions ( $k_{CMD} = 1$  and  $k_{LZ} = 0.15$ ).

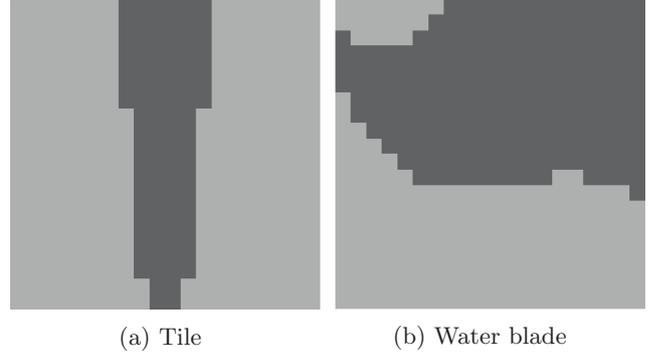


Fig. 12. Machine 2 solutions ( $k_{CMD} = 1$  and  $k_{LZ} = 0.35$ ).

Table 1  
Tile's obtained paths length.

	Machine 1: path length $k_{CMD} = 1, k_{LZ} = 0.15$ $\Delta L = 40$	Machine 2: path length $k_{CMD} = 1, k_{LZ} = 0.35$ $\Delta L = 300$
Straightforward solution	5202	5202
Machine 1 sol. (Fig. 11a)	4029	5329
Machine 2 sol. (Fig. 12a)	4159	4679

## 6.2. Evaluation of the algorithm complexity

For this experimentation, we have tried to observe the evolution of the number of cycles required by the system to find a steady state according to the meshing thinness. A cycle corresponds with the execution of each agent life cycle.

First, we have set the parameters  $k_{CMD}$  and  $k_{LZ}$  respectively to 1 and 0.2. Then, starting with a  $1 \times 1$  zoning, we have been to  $20 \times 20$ . Finally the path length is evaluated with  $\Delta L = 40$  (the machine 1 path length).

Because of the non-deterministic behavior of the agent, the obtained solutions may slightly differed. We have then launched the system times for each case in order to obtain representative results (average, minimum and maximum measured values). We have systematically taken as a start state the one in which each mesh is in a different zone.

In order to count the necessary cycles to stabilize the system, we have considered that the system is stable when, from one cycle to another, there is no actions made. Indeed, the system only reacts to agent's actions. So, if no agent acts, the system is stable. Fig. 13a represents the evolution of the necessary cycle's count over the meshing thinness for the tile. We can see in this figure that the maximum, average and minimum number of cycles evolves linearly at least until a  $12 \times 12$  meshing thinness. Then, there is some peaks on the maximum count line. Especially for high meshing thinness value. However, even if there is some peak, the average value stays close to the minimal value. We think that these peaks are caused by the lack of precision of the *FutureSituation* function used by agents. Indeed, more there are agents, more the interaction between local errors can raise collective erroneous errors that takes several cycles to be "absorbed".

Fig. 13b represents the evolution of the necessary cycle's count over the meshing thinness of the blade. We can see here the same kind of behavior than in the figure for the tile except that the perturbation appears after a  $6 \times 6$  meshing thinness. This observation confirms the analysis made for the tile. Indeed, the blade has a more complicated structure where the variation of the meshes optimal milling angle is greater than the tile. This implies that an

**Table 2**

Water blade's obtained paths length.

	Machine 1: path length $k_{CMD} = 1, k_{LZ} = 0.15$ $\Delta L = 40$	Machine 2: path length $k_{CMD} = 1, k_{LZ} = 0.35$ $\Delta L = 300$
Straightforward solution	38,171	38,171
Machine 1 sol. (Fig. 11b)	25,127	27,010
Machine 2 sol. (Fig. 12b)	25,734	26,640

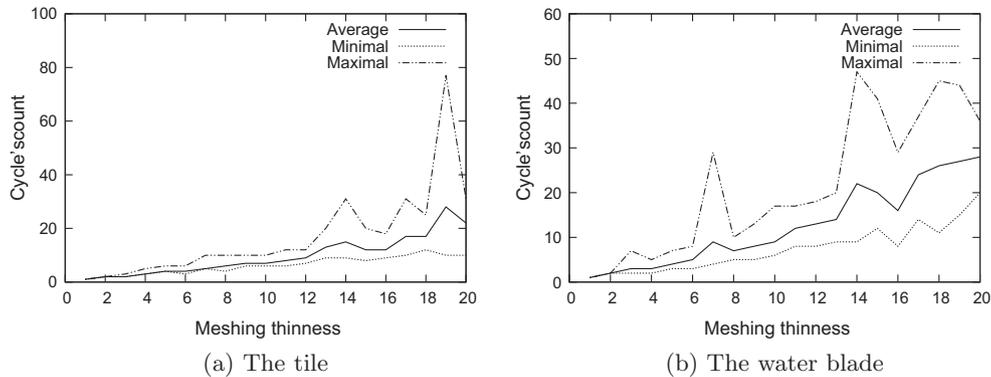
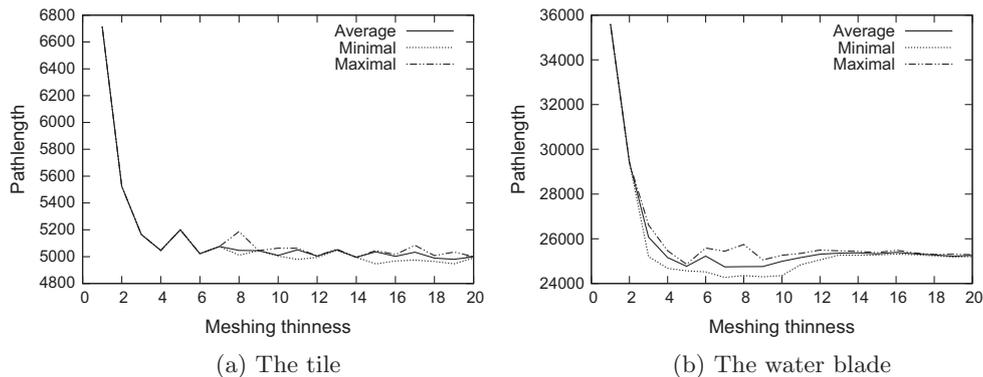
error made by an agent will probably have more impact for the agents belonging to its zone than on the tile topology.

The observed peaks on the maximum values observed through these two experiments let us conclude that the agent behavior still needs to be improved. Nevertheless, the average values seem to increase linearly as the meshing thinness is increasing. This shows that the proposed method is really efficient and will scale up to very thin meshing.

### 6.3. Evaluation of the stability of the solution

The aim of this experiment is to ensure that we obtain the same kind of solution regardless of the meshing thinness. The meshing thinness should only affect the frontier's precision. Through this experimentation, we have been looking forward to observing the evolution of the meshing path length benefit over the meshing thinness.

We have set  $k_{CMD} = 1$  and  $k_{LZ} = 0.2$  and realized the experimentation with the tile and the water blade for each zoning between  $2 \times 2$  and  $20 \times 20$ . As already said for the previous experimentations, and because of the non-deterministic behavior of the agent, we have launched the system several times for each case in order to obtain representative results (average, minimum and maximum

**Fig. 13.** Cycle's count over meshing thinness.**Fig. 14.** Path length over meshing thinness.

measured values). Even if the system resolves the problem quickly, the evaluation of the quality of the solution can take a lot of time. It's for that reason that for this experimentation, each cases have only been tested out 10 times. Fig. 14a and b point out the results.

In these figures, we can see that there is a large gain between a  $1 \times 1$  zoning and a  $4 \times 4$ . We can also notice that for the tile and for the water blade, we come to a stable state with as few as a  $6 \times 6$  zoning. Also, we can see that the use of this method generates an important benefit regarding the solution with a  $1 \times 1$  zoning. We can conclude that, although there is a kind of indeterminism with high meshing thinness, we can obtain a large gain with a  $4 \times 4$  zoning.

Fig. 15 presents the comparison between the solutions obtained with different meshing thinness on the water blade. The similarity of the obtained zone shape from  $6 \times 6$  to  $14 \times 14$  is consistent with Fig. 14a and b. This fact confirms that the topology of the solution proposed by the system remains the same as the meshing thinness increases.

This property could be helpful in the sense that users could first launch a "fast interactive" resolution of the system with a relative thick meshing in order to try and adjust the values of  $k_{CMD}$  and  $k_{LZ}$ . Once the user obtains a satisfying solution, he will launch a finer grained resolution in order to obtain precise zone frontiers.

### 6.4. Application to a reference test case

The following test case is presented by Choi in (Choi, 2004). It's based on a  $3 \times 3$  degree Bezier surface defined by

$$\mathbf{S}(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 B_i^3(u) B_j^3(v) \mathbf{P}_{ij}$$

with  $(u, v) \in [0, 1]^2$ ,  $B_i^3(u)$  and  $B_j^3(v)$  are Bernstein polynomial.

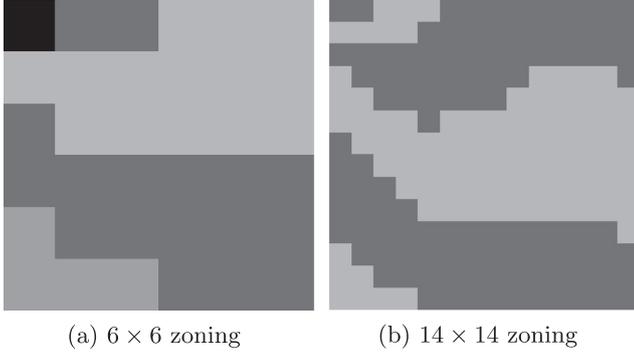


Fig. 15. Water blade solutions ( $k_{CMD} = 1$  and  $k_{LZ} = 0.2$ ).

The surface poles  $\mathbf{P}_{ij}$  are presented in Table 3.

The surface is included in a volume sizing along  $x, y$  and  $z$ :  $[0-50.8]$ ,  $[0-76.2]$ ,  $[24-38]$ , and is symmetrical in relation to the plan  $Y = 38.1$  mm ( $\nu = 0.5$ ). It is represented on Fig. 16.

In (Choi, 2004), using the conditions previously described, the total length path is 1717 mm long using an iso-scallop strategy. Using a  $70 \times 70$  meshes zoning, with our MAS algorithm with  $k_{CMD} = 1$  and  $k_{LZ} = 0.15$ , (equivalent to a 40 mm $\Delta L$ ), the following final zoning presented on Fig. 17 is obtained.

Using a  $R = 3.175$  mm,  $r = 1$  mm toroidal end mill having the same boundary dimension than the one used in (Choi, 2004), and using the same  $h_c = 0.254$  mm scallop height criterion with parallel planes strategy conduces to a 1183 mm length of machining. A reduction of 31% of the machining length is thus obtained compared to the 1717 mm initial length with the optimal one zone strategy with ball end mill presented in (Choi, 2004). This result has been obtained on a personal computer using an Intel(R) Core(TM) i7-4800MQ CPU @ 2.70 GHz processor with 16 Go RAM within a 160 s computation time, the illustration of the zoning with different tool paths is presented in Fig. 18 hereafter.

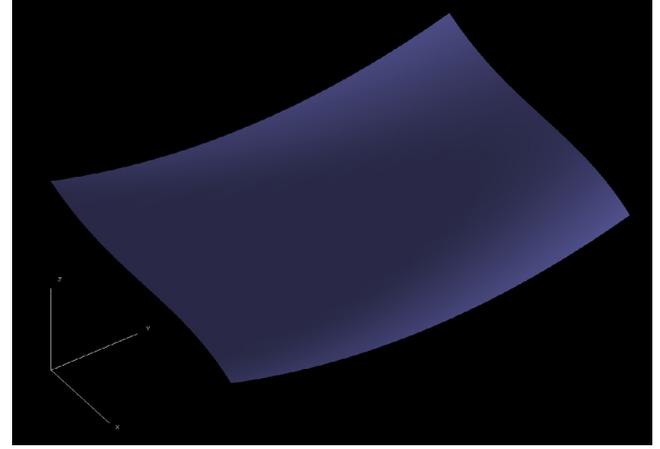


Fig. 16. Reference test case surface.

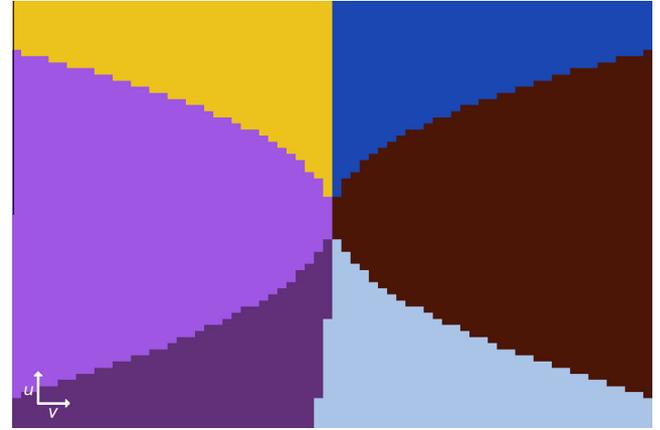


Fig. 17. MAS algorithm final zoning.

## 7. Conclusion and perspectives

In this paper, we have introduced a preliminary study of complex surface milling plan optimization. The proposed approach consists in: to divide the surface into meshes and to find the ones that will be machined together in the same milling direction (*i.e.* the zones). After a formal definition of this optimization problem and a short analysis of its combinatorial complexity we have proposed to cope with this complexity thanks to a local problem solving approach based on Multi-Agent Systems called AMAS.

The obtained results are really encouraging. Indeed, the first experimentations have shown that the proposed system (*i*) is able to converge to solutions that are coherent to the surface topology, (*ii*) can lead to solutions that will be milled up to 26.4% faster than if the surface would be milled in a standard way (one unique zone, milled with the same optimal direction over all the surface) and (*iii*) can be fitted to the target milling machine characteristics through the adjustment of  $k_{CMD}$  and  $k_{LZ}$  coefficients. Moreover, we have shown that, in spite of some non-deterministic agent

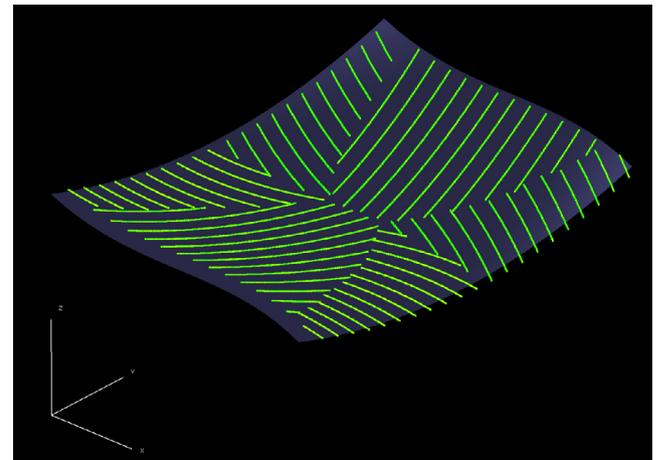


Fig. 18. Final optimized zoning for machining reference test case surface.

Table 3  
Poles for reference surface definition.

X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
0	0	38.1	17.78	0	30.48	35.56	0	38.1	50.8	0	30.48
0	25.4	30.48	17.78	25.4	22.86	35.56	25.4	30.48	50.8	25.4	22.86
0	50.8	30.48	17.78	50.8	22.86	35.56	50.8	30.48	50.8	50.8	22.86
0	76.2	38.1	17.78	76.2	30.48	35.56	76.2	38.1	50.8	76.2	30.48

behavior, the system obtains the same kind of topology from one run to another whatever the meshing thinness is.

Another very interesting property obtained thanks to this multi-agent approach is that the user can see in real time the solving process of the system. The user can also modify in real time the values of the  $k_{CMD}$  and  $k_{LZ}$  coefficient in order to explore different solutions. These features have been found to be really useful, they especially ease user decision making.

However, there are still some aspects to improve. First, although the complexity seem to be linear – *i.e.* the average convergence time seems to be linearly connected to the meshing thinness – we have noticed that the convergence time could be twice as the average in the worst cases. We then plan to study the causes of this phenomenon to be able to improve our agent's behavior.

Another way of improving the obtain milling plan will be to take into account more information in the two local criteria functions. The *Largest Zone* criteria could be more accurate using the real surface size of each meshes (as for the moment we assumes that the  $(u, v)$  dividing fits with regular mesh surfaces on the real surface). On the other hand, the *Closest Milling Direction* criterion could also be improved by taking into account the meshes slope angle (*cf.* Section 5.2.1).

Finally, in order to ease the accuracy of agent reasoning (especially the *FutureSituation* function), we have decided to run agents in a sequential way (the effect of the previous agent action is taken in account before the execution of the next agent). We plan to study more sophisticated agents behavior in order to be able to run agents in an asynchronous way and then obtain a system that can gain performance from multi-core and parallel computation architecture.

## References

- Bernon, C., Camps, V., Gleizes, M. P., & Picard, G. (Juin 2005). Engineering adaptive multi-agent systems: The adelfe methodology, pp. 172–202, ISBN1-59140-581-5.
- Chen, Z. C., Vickers, G. W., & Dong, Z. (2003). Integrated steepest-directed and iso-cusped toolpath generation for three-axis CNC machining of sculptured parts. *Journal of Manufacturing Systems*, 22(3), 190–201.
- Choi, Y. K. (2004). *Tool path generation and 3D tolerance analysis for free-form surfaces*. PhD thesis, Texas A&M University.
- Faux, I. D. & Pratt. M. J. (1985). Computational geometry for design and manufacture.
- Huang, Y., & Oliver, J. H. (1994). Non-constant parameter nc tool path generation on sculptured surfaces. *The International Journal of Advanced Manufacturing Technology*, 9(5), 281–290.
- Kim, B. H., & Choi, B. K. (2000). Guide surface based tool path generation in 3-axis milling: An extension of the guide plane method. *Computer-Aided Design*, 32(3), 191–199.
- Loney, G. C., & Ozsoy, T. M. (1987). Nc machining of free form surfaces. *Computer-Aided Design*, 19(2), 85–90.
- Maeng, Hee Y., Ly, Minh H., & Vickers, Geoffrey W. (1996). Feature-based machining of curved surfaces using the steepest directed tree approach. *Journal of Manufacturing Systems*, 15(6), 379–391.
- Marciniak, K. (1992). *Geometric modelling for numerically controlled machining*. Oxford university press.
- Redonnet, J.-M., Djebali, S., Segonds, S., Senatore, J., & Rubio, W. (2013). Study of the effective cutter radius for end milling of free-form surfaces using a torus milling cutter. *Computer-Aided Design*, 45(6), 951–962.
- Senatore, J., Segonds, S., Rubio, W., & Dessein, G. (2012). Correlation between machining direction, cutter geometry and step-over distance in 3-axis milling: Application to milling by zones. *Computer-Aided Design*, 44(12), 1151–1160.
- Steiner, T., Peterzell, M., Pottman, H., & Zhao, H. (2005). Swept volumes. *Computer Aided Design*, 2(5), 599–608.