Geometric Optimization Algorithms in Manufacturing

Kai Tang

Hong Kong University of Science and Technology, mektang@ust.hk

ABSTRACT

In this paper we review the recent progress in the design of efficient algorithms for several geometric optimization problems in manufacturing processes. We first describe the strong geometric characteristics of optimization problems in manufacturing. Three illustrative problems are then presented in details: the shortest contour touring problem in laminated object manufacturing, the minimum tool retractions problem in zigzag machining, and the minimum workpiece setups in 4-axis surface machining. We survey representative algorithms and techniques used to solve these three problems, with emphasis on both their practical significance and theoretical contributions.

Keywords: Introductory and survey; analysis of geometric algorithms; computational geometry; manufacturing automation; optimization.

1. INTRODUCTION

With the advances of manufacturing equipments such as CNC machines and laser machining systems, concerning workpiece machining, various problems in geometric optimization subject to physical constraints have been raised. Finding solutions to these geometric optimization problems requires knowledge linking several scientific disciplines — optimization algorithms, computational geometry, and mechanical engineering — and has became an active area of research in manufacturing industry. During the last decade, many elegant and sophisticated geometric optimization algorithms have been developed and successfully applied in manufacturing industry. By showing that the optimization problems in manufacturing have strong geometric characteristics, in this paper we present a survey on some key techniques used to handle these characteristics and efficient algorithms for several important optimization problems in manufacturing.

Due to the material removal nature of machining operations, most optimization problems in manufacturing have a fundamental geometric character. Albeit non-exclusiveness, in terms of computational solutions, these problems can be in general categorized into three groups below.

Combinatorial solutions

For an optimization problem in this group, the input is usually of very simple geometric types and often discrete, such as line segments. On the other hand, the amount of input is huge. For example, in layered machining, where the input is given in STL format, tens of thousands of line segments are often encountered. To solve this kind of problems, traditional techniques from computational geometry, which emphasize on computational speed but are only suitable for discrete data, are most appropriate. Unfortunately, in most cases these problems are NP-*hard* and heuristic and sub-optimal solutions have to be sought.

Gradient-based solutions

In this case, the optimization can be expressed as a traditional constrained function optimization of a continuous function f($\boldsymbol{x}$): $\boldsymbol{x}\in\boldsymbol{R}^n$ for some n. Traditional gradient-based search methods, such as the Conjugate Gradient Method and Lagrange Multipliers' Method, can then be utilized to find an optimum of f($\boldsymbol{x}$). However, although gradient-based methods are popular in many engineering disciplines, especially in engineering design where the optimization objective is often a single scalar and the design variables can be conveniently represented by a vector in the $\boldsymbol{R}^n$ space, it is usually difficult or even impossible to formulate the optimization as a function f($\boldsymbol{x}$) in real space $\boldsymbol{R}^n$, let alone the existence of continuous gradient of f($\boldsymbol{x}$). Approximation must be made in order for a problem to be able to be placed in this class, thus achieving only sub-optimal (and some times drastically inferior) results. As a result, most optimization problems in manufacturing are not suitable for gradient-based methods.

Domain-partition solutions

In many scenarios of manufacturing optimization, the output required is often of very simple geometric type, e.g., a line or a direction in 3D space, which realizes a scalar optimum. Since a simple geometric entity can be described by a few real-number parameters (e.g., a plane ax + by +cz +1 = 0 is uniquely decided by the coefficients (a,b,c)), this type of optimization problems can be characterized as the optimization of a scalar function f($\boldsymbol{x}$): $\boldsymbol{x}\in\Omega\subset\boldsymbol{R}^k$ over some sub-space Ω with a small integer *k*. Different than the ordinary functional optimization problems, however, the function f($\boldsymbol{x}$) has no gradients! Specifically, the domain Ω is characterized by a quotient-space partitioning $\Omega=\bigcup_{i=1}^{m}\Omega_i$ ($I(\Omega_i)\cap I(\Omega_j)=Null$ for and any $i\neq j$ and i,j = 1,2,…,m), for some finite m, such that function f($\boldsymbol{x}$) remains a constant on each Ω_i. Their solutions require the establishment of this partitioning and identification of an optimum $\boldsymbol{x}$ in Ω. The key to the success of this domain-partitioning strategy is that the partitioning is usually never explicitly constructed – the explicit construction of the partitioning is in most cases either too computationally formidable or practically infeasible. Rather, the partitioning is constructed in an implicit form – each Ω_i is characterized by one or a few representative points and the search of optimum is carried out over the function values at these points. How to identify these representative points often requires algorithmic ingenuity and geometric insight of the problem at hand.

In the rest of the paper, we present three optimization problems in manufacturing to illustrate the above classification. For each of the three problems, some key techniques and algorithms are described.

2. LAMINATED OBJECT MANUFACTURING

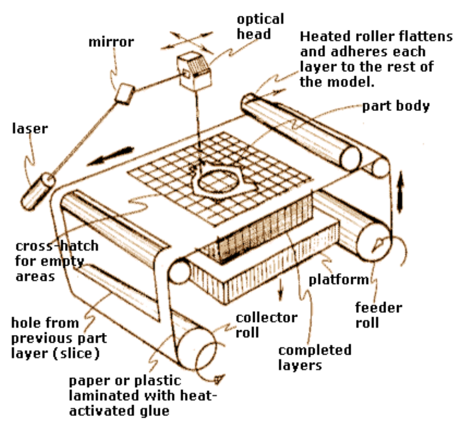


Fig. 1. Laminated object manufacturing (picture from http://www.efunda.com/processes/rapid_prototyping/lom.cfm)

Laminated object manufacturing (LOM) is a rapid prototyping technique to create low cost 3D models from CAD data. Refer to Fig. 1. The system uses a laser beam to trace the outline of the layer on a sheet of paper. The sheet is then adhered to a substrate with a heated roller. After cutting the geometric features of a layer, the worktable holding the part is moved down and then back up one layer below its previous position. At this step new sheet is also rolled into the cutting position. The method is self-supporting for overhangs and undercuts. Areas of cross sections which are to be removed in the final object are heavily crosshatched with the laser to facilitate removal. The final prototype part has a wood-like texture composed of paper layers which are sealed with paint or lacquer to avoid moisture being absorbed by the paper.

On each sheet (layer), the part is represented by a cross-section obtained by intersecting a plane with the CAD geometry of the part. A laser beam traces on the layer along the contours inside this cross-section, resulting in their extrications from the rest of the sheet. Since the movement of the laser head is continuous, the contours in the cross section need to be connected by a polygonal chain (or tour). To reduce the total manufacturing time, the length of the tour (the jumping distance) should be minimized. Simulation experiments [26] show that in many cases an LOM operation with optimized tours vs. non-optimized ones can lead up to 30% reduction in total manufacturing time.

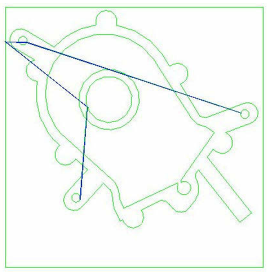
The most general geometric optimization problem related to this tour minimization can be stated as follows.

Problem 1. Given a sequence $\{C_1, C_2, \cdots, C_k\}$ of k contours in the plane, each contour C_i being a simple closed polygon, find a set of points $\{p_1, p_2, \cdots, p_k : p_i \in C_i\}$ and a permutation $\pi : \{1, \cdots, k\} \rightarrow \{1, \cdots, k\}$, such that $\sum_{i=1}^{k-1} d(p_{\pi(i)}, p_{\pi(i+1)})$ is minimized, where $d(,)$ is a metric.

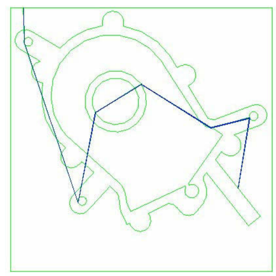
This is a typical computational geometry problem and, as usually, can be easily seen to be NP-*hard*. In the following we describe several heuristic or special-case solutions.

Maximum-intersection-line based greedy approach

A maximum-intersection-line (MIL) is a line that intersects the maximum number of given contours. For contours represented by lines or simple curves, efficient algorithms are known for finding a MIL [28, 29]. The greedy heuristic works as follows [26]. First we find a MIL of the contours. Then remove the intersected contours from the current set of contours. This process is recursively performed on the current set of contours until it is empty. As a comparison, a GA based approach is also implemented in [26]. Experiments showed that the MIL-based greedy algorithm has a promising improvement over the GA based approach, both in terms of jumping distance reduction and, especially, the computing time requirement. Fig. 2 shows an example of Problem 1 for a layer of a mechanical part.



(a) MIL-based greedy result



(b) GA result

Fig. 2. Finding a minimum length tour of contours

Fixed-ordering based approach

In the context of LOM, the number of contours in a cross section is usually small, say less than 10. On the other hand, the number of line segments on a contour can easily exceed several thousands. Therefore, a plausible approach to the minimization is to try all the orders of the contours and for each ordering find a shortest tour. Some known heuristics on finding a "good" ordering (e.g., variable *r*-opt procedure [15]) can also be utilized for "intelligent" rather than brute-force search of the ordering. Unfortunately, even when the ordering of the contours to tour is fixed, the minimum length tour problem is still NP-*hard* [7]. However, if all the contours are convex and mutually disjoint from each other, and if one further requires that the first and the last contours degenerate to single points*, the corresponding minimum length touring problem becomes computationally attainable and an $O(kn\log(n/k))$ deterministic and exact minimization algorithm is recently reported by Dror et al. [7], where n is the total number of vertices on the polygonal contours. Dror et al. [7] also present an $O(nk^2\log n)$ algorithm for the case that the convex polygons are allowed to intersect each other but the subpath between any two consecutive polygons is constrained to lie within a simply connected region.

Curve-interpolation based approach

In the fixed-ordering case, if the first and last contours are not degenerate or some contours are concave, the algorithm in [7] will not work. An alternative solution is to first approximate every contour by an interpolating smooth parametric curve such as a B-spline. The tour length then can be expressed as a function $L(u_1, u_2, \ldots u_k)$, where u_i is the parameter of the point of the tour on the interpolated contour C_i. As L now has continuous partial derivatives, well-known gradient-based search algorithms, e.g., the Conjugate Gradient Method can be applied to minimize L. In Fig. 3 we give an example from our implementation to illustrate this approach, in which the average number of vertices on an original contour is 200 and each contour is approximated by a cubic B-spline with an average number of 25 control points. (Note that the first and last contours are degenerated points A and B respectively.)

Local-minimum based approach

* In LOM, the laser head sometimes is required to be placed at a fixed position called *homing position* when not in machining operation.

Still pertaining to the fixed-ordering case, one practical heuristic is the local-minimum based iterative search. Basically, with the current tour L(A=p_0, p_1, p_2, …, p_k, p_{k+1}=B): $p_i \in C_i$, for i = 1 to k in turn we fix p_{i-1} and p_{i+1} and find a best point p* on contour C_i that minimizes the sum $d(p_{i-1}, p^*) + d(p^*, p_{i+1})$ and replace the current p_i on the tour with p*. Owing to its local search nature, this algorithm usually runs very fast. In terms of final minimization, one can show [27] that it always outputs the true minimum in the special case when all the contours are convex and disjoint. Even for the general case, our tests demonstrate that it often finds near-optimum results. For instance, the local-minimum based iterative method finds the same true optimum tour in Fig. 3 as did the curve-interpolation based method, within the specified tolerance. Understandably, due to its inherent local nature, the initial tour is extremely critical. As an improvement, heuristics on initial conditions (e.g., multiple re-starts) can be used.

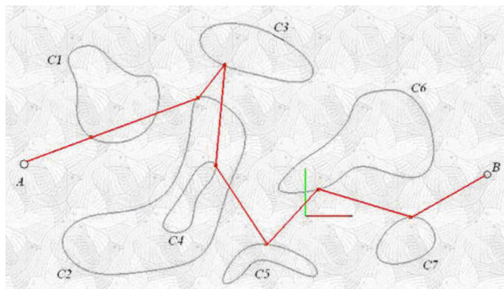


Fig. 3. A minimum length tour obtained via curve interpolation and Conjugate Gradient method.

3. MINIMIZING TOOL RETRACTIONS

Up to 80% of mechanical parts can be machined using 2.5D milling [11]. The milling machining takes place in a displaced plane (say, normal to z-axis). The task of 2.5D milling is known as *pocket machining*. The ultimate goal of pocket machining is the generation of optimal tool paths in order to remove material in areas with specified boundaries and z-values. There are two basic types of tool path generation in practice: contour milling and zigzag milling. Contour milling uses successive offsets of the pocket contour as the cutting paths of the tool. To facilitate elimination of self-intersections of the offset curves, Voronoi diagram needs to be computed that offers a planar subdivision of the pocket where each face in the subdivision corresponds to exactly one contour element (cf. [5, 11, 13]). We remark that computing offset is a rather difficult and time-consuming task if the pocket contour consists of elements other than lines and circular arcs [16]. In contrast, the paths for zigzag milling are conceptually much simpler: milling takes place over line segments (called zigzag segments) parallel to a specified reference line in alternating directions between adjacent passes, and the cutter is required to move between neighboring passes only on the un-cut boundary of the region. A tool retraction is one when the cutter is raised from the surface being cut, moves to a new position, re-contacts the surface, and resumes the cutting again. The main task in zigzag tool path generation is how to connect the zigzag segments so that the number of tool retractions can be minimized. Fig. 4 depicts an example of zigzag milling tool path where the tool retractions are represented by arrows.

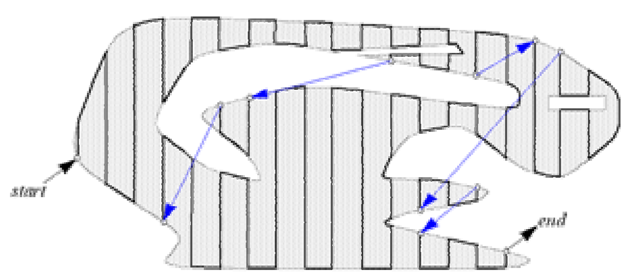


Fig. 4. A zigzag tool path with five tool retractions.

We follow the terminologies in [1] to formulate this minimization problem in a more formal and convenient manner. Refer to Fig. 5. Let a pocket, denoted by P, be a compact, simply or multiply, connected planar shape (shaded area in Fig. 5) bounded by a contour C. Let $\delta > 0$ be the step-over distance which is determined by the cutter used for machining. Given a reference line l and a starting point Δ, define the set $ZL(P)$ of zigzag lines with respect to P to be the set of lines parallel to l and one of which passes Δ. The set $ZS(P)$ of zigzag segments induced by P is $ZS(P) = \cup_{l \in ZL(P)} \{P \cap l\}$.

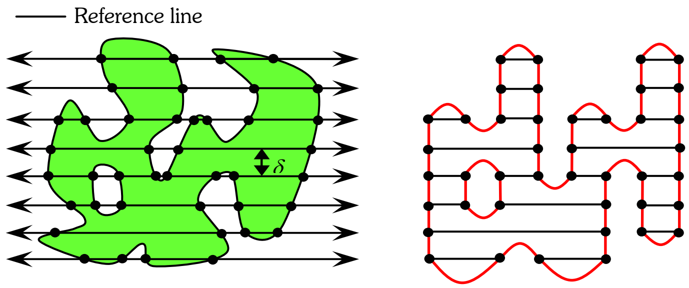


Fig. 5. Zigzag machining graph. Left: Zigzag lines intersect pocket contour; Right: corresponding machining graph

Definition 1. The machining graph $MG(P)$ induced by P is an undirected graph (N, E), where the set N of nodes is the union of all intersection points of zigzag lines in $ZL(P)$ with the contour C and the set E of edges consists of two parts: one is the subset of *noncompulsory edges* (nc-edges) which corresponds to the contour C of P (the red edges in Fig. 5) and the other is the subset of *compulsory edges* (c-edge) which corresponds to the zigzag segments (the black edges in Fig. 5).

Define a *path* from v_1 to v_{k+1} in the graph (N, E) to be a sequence $\{v_1, e_1, v_2, \cdots, v_k, e_k, v_{k+1}\}$ such that $e_i = (v_i, v_{i+1}) \in E$, $1 \le i \le k$, and $e_i \ne e_j$ if $i \ne j$. Now the minimum tool retraction problem becomes the following,

Problem 2. Given a machining graph $MG(P) = (N, E)$, find a finite set of paths $\{\pi_1, \pi_2, \cdots\}$ with minimum cardinality called *retraction number*, such that (1) every c-edge is traversed exactly once and (2) every nc-edge is traversed at most once.

The above problem is again, unfortunately, NP-*hard* as shown in [1], even for a simple P. Actually, it is not hard to convert it to the HAMILTONIAN CIRCUIT problem which is well known to be NP-*hard* [9,14]. So once again heuristic algorithms have to be sought.

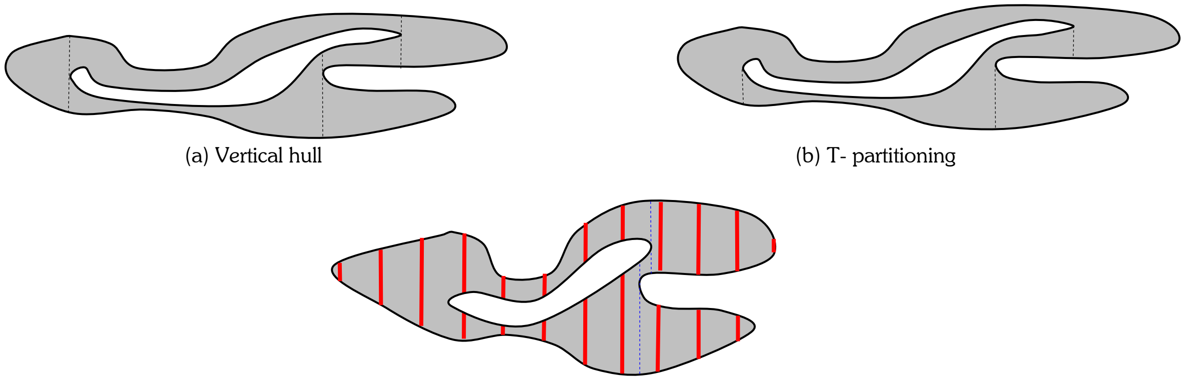
Arkin's algorithm

Arkin et al. developed a linear algorithm [1] which guarantees to find a tool path of an $MG(P)$ with a retraction number no greater than 5K* +6h, where K* is the unknown true minimum retraction number of that $MG(P)$ and h the number of holes in P. Their algorithm assumes no restrictions on $MG(P)$. They also show the upperbound "5K*" is tight; that is, there does exist an $MG(P)$ for some simple P to which their algorithm will output a tool path whose retraction number is 5 times the true minimum.

The T-partitioning method

In [21] a heuristic algorithm based on the concept of T-partitioning is proposed. Refer to Fig. 6. Assuming the reference line l to be vertical, the vertical hull of a P is the partitioning of P made by the boundary of P and the internal vertical supporting lines of P, e.g., the vertical hull of P in Fig. 6(a) has 6 sub-regions contributed by 3 internal vertical supporting lines (dotted). Each internal vertical supporting line segment is made of an upper part and a lower part which are delimited by the involved reflex point on the boundary of P. The T-partitioning of P then is the vertical hull of P but relaxed by removing the upper parts of all the internal vertical supporting lines, e.g., Fig. 6(b). It is not hard to verify that within any constituting sub-region of the T-partitioning all the zigzag segments can be connected without any retractions. Since there are only N_r–h+1 constituting sub-regions in P (where N_r is the number of internal vertical supporting lines) [21], we effectively obtain a tool path with at most N_r–h retractions.

The N_r–h result is nonetheless by no means the optimum, as correctly pointed out by Held et al in [12]. Later a sufficient condition is offered in [18] for the optimality of the T-partitioning: if every constituting sub-region of the vertical hull of P strictly contains 3 or more zigzag segments, then the N_r–h result from the T-partitioning is also the true minimum. However, the proof given in [12] overlooked the fact that the T-partitioning scheme is no longer valid if P has collinear internal vertical supporting lines or some two internal vertical supporting lines are "sandwiched" between two neighboring zigzag segments (see Fig. 6(c)). Also, the requirement of every sub-region strictly containing 3 zigzag segments seems to be too stringent. Catering to these concerns, a more rigorous algorithms was recently proposed in [22] which is based on a novel computational model called block transition graph, which we describe next.



(a) Vertical hull

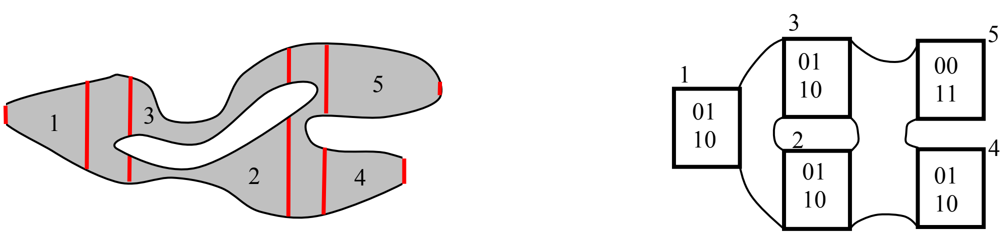
(b) T- partitioning

(c) Two internal vertical supporting lines (blue) sandwiched between two zigzag segments

Fig. 6. T-partitioning algorithm and its deficiency

Block-transition-graph based method

We first make some definitions. A strip in an *MG*(*P*) is the region bounded by two neighboring zigzag segments and the two nc-edges between them. Two strips are neighbors if they share a common zigzag segment. A block is a set of strips such that every strip in the block is and only is a neighbor of another strip in the block. A zigzag segment in a block is said to be internal if it is shared by two neighboring strips, otherwise it is a bounding zigzag segment. The bounding zigzag segments of all the blocks together with the nc-edges between them form a partitioning of *MG*(*P*) called block partitioning, e.g., Fig. 7(a) depicts the block partitioning of *MG*(*P*) in Fig. 6(c), where each block is assigned with an integer ID. Taking each block as a special node with a geometric shape of a rectangle and nc-edges as edges between the nodes, a block partitioning is uniquely converted to a special graph (N, E) called Block Transition Graph (BTG). For example, the corresponding BTG of the block partitioning in Fig. 7(a) is displayed in Fig. 7(b). As shown in Fig. 7(b), each node in a BTG is associated with a pair of mutually complemented 2-bit strings called its state strings – it is {“00”, “11”} if the number of zigzag segments in the corresponding block is even (e.g., node 5), and {“01”, “10”} otherwise. A state string “b_1b_2” identifies the four corners of the rectangle node: b_1= “0” and “1” identify the “upper left” or “lower left” corners respectively, whereas b_2= “0” and “1” identify the “upper right” or “lower right” corners respectively.



(a)

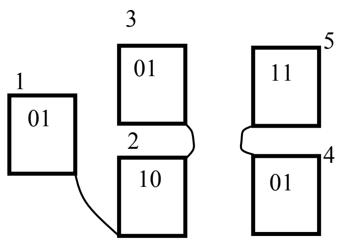
(b)

Fig. 7. Block partitioning and its corresponding Block Transition Graph.

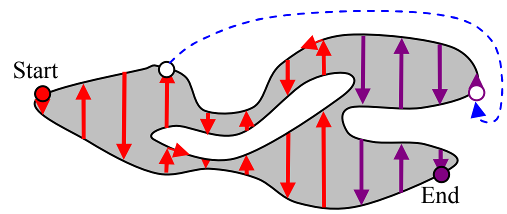
For a node *v* in a BTG we define a compatibility relation between a state string “b_1b_2” of *v* and an incident edge *e* of *v*: *e* is compatible to “b_1b_2” if and only if its incident corner is identified by “b_1b_2”. An instance of a BTG (N, E) then is an assignment of an state string to every node in N together with the subgraph (N, E’⊂E) where E’ contains only those edges in E that are compatible to the assigned state strings. For example, Fig. 8(a) shows an instance of the BTG of Fig. 7(b) which has two connected components. It is straightforward to see that all the zigzag segments in the blocks corresponding to a component can be connected together without using any retraction, e.g., for the *MG*(*P*) of Fig. 6(c) and the instance of its BTG shown in Fig. 8(a), the corresponding tool path is depicted in Fig. 8(b) which requires one retraction, as there are only two components in the instance.

The problem of minimizing the retractions is therefore plausibly transformed to a graph traversal problem: for a given BTG we want to find an instance with the minimum number of components (the minimum instance). In [22] a linear time algorithm based on the idea of ring-removal is presented which guarantees to find a minimum instance of the BTG of *MG*(*P*), if *P* has no holes. For an arbitrary non-simple *P* with h holes, the linear time algorithm given in [22] can find an instance with at most K*+h components, where K* is the true minimum. Nevertheless, one more thing needs to be settled – we still haven't proven the equivalence between the minimization of retractions of *MG*(*P*) and an minimum instance of the BTG of *MG*(*P*). Actually, these two must not be equivalent in general for otherwise Problem 2 would not be NP-*hard*. The culprits of this inequivalence are those "empty" blocks in the block partitioning, that is, the blocks which do not strictly contain a zigzag segment. Calling these "empty" blocks reducible blocks, we thus have achieved the following result [22]:

Given the *MG*(*P*) of an arbitrary *P* with h holes, one can in linear time find a tool path with at most K*+m+h retractions, where m is the number of reducible blocks in the BTG of *MG*(*P*).



(a) Instance



(b) Tool path

Fig. 8. Instance of a BTG and its tool path

4. MINIMUM WORKPIECE SETUPS IN 4-AXIS SURFACE MACHINING

The machining of free-form surfaces usually requires highly-sophisticated numerically controlled (NC) machines. Depending on the number of DOFs enjoyed by the tool, an NC machine can be classified as a 3-, 4- or 5-axis machine. On a 3-axis machine, the tool can only move in translation in *x*-, *y*-, and *z*- directions with respect to the workpiece [20]. If the machine is of 4- or 5-axis type, then in addition to the translation motion, the tool can also rotate (with respect to the workpiece) about one or two axes. This added rotation motion not only significantly improves the quality of the machined surface, but also tremendously increases the accessibility of the surface to the tool, therefore greatly reducing the number of setups. A setup refers to a placement of the workpiece on the worktable with a fixed orientation, which requires dismount and remount of the partially machined part and recalibration of machine coordinate system, plus other necessary adjustments. The minimization of set-ups then refers to the careful selection of set-ups so that the entire part surface can be correctly machined without any gouging and with as few set-ups as possible.

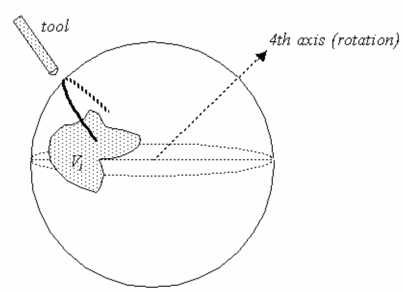


Fig. 9. Visibility map and its relation with the representative arc of the 4th axis.

To put this minimization problem in a viable computational format, the part surface is usually first divided into a number of faces f_1, f_2, …, f_n with each of them associated with a set of directions, called accessible orientations, along which the tool can access any point on the face without interfering with part or fixtures. The set of accessible orientations of f_i is best represented as a spherical polygon V_i on the Gaussian sphere S (i.e., the unit sphere) called the visibility map [2,3,8]. In the case of a 4-axis machine, for a particular set-up, the orientations of the tool as provided by the 4^{th} axis (rotation) can be represented by a great arc of some length L on S, called L-arc, where L is the allowable range of rotation about the 4^{th} axis by the tool. Obviously, as illustrated by Fig. 9, f_i is accessible in a set-up if and only if the representative arc of that set-up intersects V_i. To minimize the number of set-ups is then equivalent to finding a minimum set of L-arcs such that every f_i is intersected by at least one arc. Heuristic solutions have been sought for sub-optimal solutions for this obviously NP-*hard* problem. Among them is a popular greedy approach: each time we find an L-arc that intersects a maximum number of V_i; we then find another L-arc which intersects a maximum number of V_i that have not be intersected before; this iteration continues until all the V_i are intersected. Accordingly, the following geometric optimization is called for.

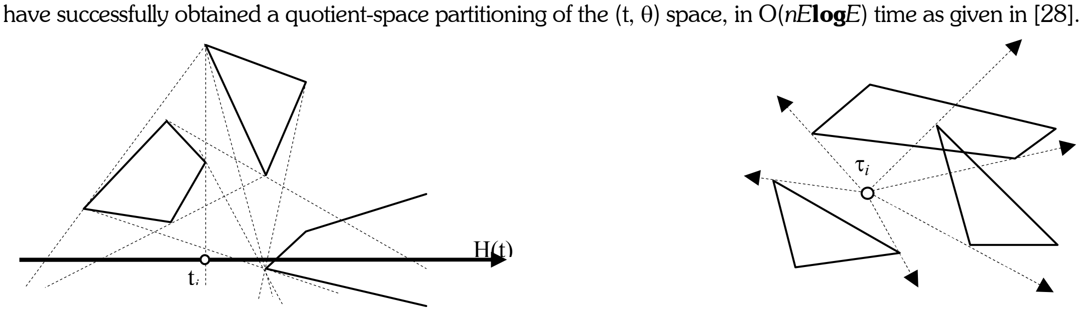
Problem 3. Given a set Ψ of n spherical polygons and a real number $L \le 2\pi$, find a minimum set A of great arcs of length L such that every spherical polygon in Ψ intersects at least one of the arcs in A.

We next survey several key algorithms for solving Problem 3, in the order of degree of restriction on L, i.e., when the L-arc is a whole great circle, a semi-great circle, and an arbitrary arc.

Case of a great circle

When the L-arc is a great circle, i.e., $L = 2\pi$, an immediate benefit is that all the spherical polygons in Ψ can be assumed to be convex, since a great circle intersects a spherical polygon if and only if it intersects the convex hull of the polygon. In [28] a simple algorithm was proposed for solving this special case of Problem 3. The n spherical polygons are first projected to the plane $z = 1$, resulting in $n' \ge n$ convex polygons in the plane, where a spherical polygon across the equator generates two unbounded polygons in the plane. A great circle is mapped to a unique line in the plane under the central projection. Through proper registration, the maximum arc intersection problem on the sphere is thus transformed to the equivalent maximum line intersection problem in the plane: given n convex polygons in the plane, find a line intersecting as many of them as possible. This is a typical domain-partition problem. Since a line can be represented by $y = ax + b$, the maximization function $f(a,b)$ is defined in $(a,b) \in \Omega \subset \boldsymbol{R}^2$. To solve this kind of problems, a crucial step is to find a transformation of domain Ω so that the quotient-space partitioning of the new Ω is more attainable. A novel transformation is given in [28]. Let H(t) be an arbitrary horizontal line with a natural parameter t. A non-horizontal line l thus is uniquely represented by (t, θ) where H(t) is the intersection point between H and l and θ is the slope angle of l. We need to consider only non-horizontal lines, as the maximization restricted to lines of a fixed orientation can be easily solved in $O(E\mathbf{log}E)$ time using the well-known plane-sweep technique, where E is the total number of edges on the polygons.

To correctly partition the (t, θ) space, let $t_1, t_2, \ldots t_m$ be the intersection points between H and the edges on the polygons and the pair-wise common supporting lines of the polygons, e.g., Fig. 10(a), sorted from left to right, with $t_0 = -\infty$ and with $t_{m+1} = \infty$. Within each interval (t_i, t_{i+1}), for any two arbitrary points $\tau, \tau' \in (t_i, t_{i+1})$, the 1D restricted functions $f(\tau, \theta)$ and $f(\tau', \theta)$ $(\theta \in (0, \pi))$ always have the same maximum. This effectively tells that (t_i, t_{i+1}) can be represented by a single point in it, say τ_i. Restricted to a fixed $t=\tau_i$, the partitioning of θ is achieved by finding the supporting rays from point $H(\tau_i)$ to all the polygons not containing $H(\tau_i)$ (see Fig. 10(b)); these rays partition the slope angle θ of the lines through $H(\tau_i)$ into congruent intervals – any lines l and l' within the angular interval of any two neighboring rays intersect the same set of polygons. We thus have successfully obtained a quotient-space partitioning of the (t, θ) space, in $O(nE\mathbf{log}E)$ time as given in [28].



(a) (b)

Fig. 10. Quotient-space partitioning of the (t, θ) domain.

There can be other types of transformation for the domain Ω and the corresponding partitioning schemes. In [10] a transformation based on the point-line duality was proposed, and the corresponding partitioning algorithm was presented which requires $O(E^2)$ running time.

Case of a semi-great circle

Restricting the L-arc to be a great circle is impractical, as the allowable rotation angle range for the 4th axis is always less than π (note that the worktable is usually flat). We made an improvement in [19] by considering the case of a semi-great circle, i.e., $L = \pi$. Several difficulties are introduced due to this relaxation. First, if still using central projection, the mapping between semi-great circles and lines are no longer bijective. Second, polygons can no longer be assumed to be convex. To resolve the first, we now assign colors to the central projection: a point in the plane is "white" if its source point is on the upper hemisphere, and "black" otherwise. Under this color scheme, a spherical polygon is mapped to either a single polygon of single color if it does not cross the equator, or two or more polygons with different colors. A semi-great circle is mapped to a unique pair of rays R and $\overline{R}$, called a conjugate pair, which originate at a same origin point and point at opposite directions. Unlike a line, we need three parameters (x, y, θ) to decide a conjugate pair $\{R, \overline{R}\}$: (x, y) is the common origin of the two rays and θ is the angle between the $+x$-axis and the ray R. The objective function can now be expressed as $f(x, y, \theta) = \omega(R)+\beta(\overline{R})-\phi(R, \overline{R})$, with $\omega(R)$ being the number of white polygons intersected by ray R, $\beta(\overline{R})$ the black polygons intersected by ray $\overline{R}$, and $\phi(R, \overline{R})$ a number to resolve the double counting if two intersected polygons in $\omega(R)$ and $\beta(\overline{R})$ have a same ID (i.e., they have the same source spherical polygon). Refer to Fig. 11 for an illustration.

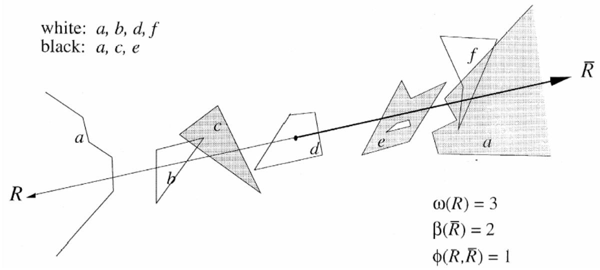
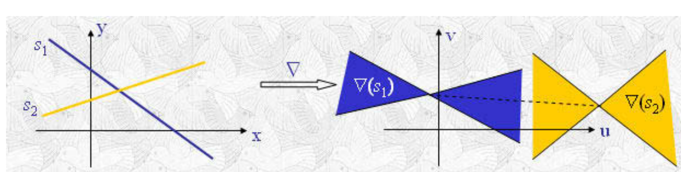


Fig. 11. Intersecting black and white polygons by a conjugate pair $\{R, \overline{R}\}$.

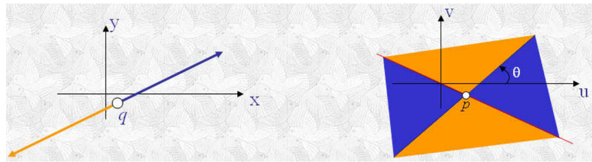
A direct partitioning of (x, y, θ) appears to be formidable. We now introduce a transformation based on the bijective duality ∇: the dual of a point (a,b) in the x-y plane is a line $au + bv + 1 = 0$ in the u-v plane, with the same color. Under this mapping, a line segment is mapped to a double wedge and two line segments intersect each other iff their double wedges in u-v cover each other's apexes (see Fig. 12(a)); a pair $\{R, \overline{R}\}$ becomes a pair of special conjugate double wedges that cover the entire u-v plane (Fig. 12(b)) – they share a common bounding line which passes through the origin of the u-v plane. This pair of conjugate double wedges can be parametrically represented by $(p(u,v), \theta)\in\boldsymbol{R}^2\times[0, 2\pi)$, e.g., Fig. 12(b). Therefore, $f(x,y,\theta)$ becomes $f(u,v,\theta)$. To partition the (u,v,θ) space, let's consider the arrangement of all the bounding lines of the double wedges, as illustrated in Fig. 12(c). It is proven in [19] that, within each cell of this arrangement, $f(p, \theta):\theta\in[0,2\pi)$ has the same minimum for any p. Hence, each cell can be represented by an arbitrary point in it. For a fixed point p, to maximize $f(p, \theta):\theta\in[0,2\pi)$, we go back to the primal x-y plane and the problem becomes: given a line $\nabla(p)$ which the pair $\{R, \overline{R})$ are required to lie on, find the origin of the pair that will maximize the function $f(R, \overline{R})$ (see Fig. 12(d)). This problem is easily seen to be solvable in $O(E\mathbf{log}E)$ time. To summarize, the entire Problem 3 for a semi-great circle $L = \pi$ can be solved in $O((E+I_{wb})^2 n)$ time, where I_{wb} is the number of intersections between the edges of the polygons of different colors, [19].

The general case

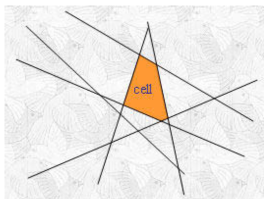
The treatment for the general case of an arbitrary length $L < 2\pi$ follows the same strategy of transformation + partitioning, although the algorithm is more complicated. Due to the limit of space, it is not presented here. For details the reader is referred to Ref. [23,24].

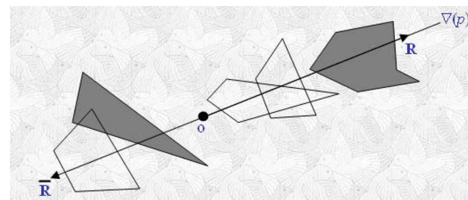


(a) Duals of two line segments and their intersection relationship in the u-v plane



(b) Dual of a conjugate pair





(c) Arrangement of the double wedges

(d) Find the best origin o for the pair $\{R, \overline{R}\}$ on their line

Fig. 12. Duality and its application

5. CONCLUSIONS

In this survey we have reviewed several techniques and their applications in geometric optimization problems raised from three important manufacturing processes, i.e., laminated object manufacturing, zigzag pocket machining, multi-axis surface machining. Besides the review purpose, it is hoped that, rather than scattered around, the presented techniques and algorithms can be systematized to help form a core set of methodologies applying to a large set of optimization problems in manufacturing.

6. REFERENCES

[1] Arkin E. M., Held M., and Smith C. L., Optimization algorithms related to zigzag pocket machining, *Algorithmica*, Vol. 26, No. 2, 2000, pp 197-236.

[2] Chen L. L., Chou S., and Woo T., Separating and intersecting spherical polygons: computing machinability on three-, four-, and five-axis numerically controlled machines, *ACM Transaction on Graphics*, Vol. 12, No. 4, 1993, pp 305-326.

[3] Chen L. L., and Woo T., Computational geometry on the sphere with application to automated machining, *Journal of Mechanical Design*, Vol. 114, No. 2, 1992, pp 288-295.

[4] Choi J., Sellen J., and Yap C. K., Approximate Euclidean shortest paths in 3-space, *International Journal of Computational Geometry & Applications*, Vol. 7, No. 4, 1997, pp 271-295.

[5] Chou S. Y., Woo T., Chen L. L., Tang K., and Shin S. Y., Scallop hull and its offset, *Computer-Aided Design*, Vol. 26, No. 7, 1994, pp 537-542.

[6] Chvatal V., A greedy heuristic for the set cover problem, *Mathematics of Operations Research*, Vol. 4, No. 3, 1979, pp 233-235.

[7] Dror M., Efrat A., Lubiw A., and Mitchell J. S. B., Touring a sequence of polygons, *Proceedings of 13th Annual ACM Symposium on Theory of Computing*, 2003, pp 473-482.

[8] Gan J., Woo T., and Tang K., Spherical maps: their construction, properties, and approximation, *Journal of Mechanical Design*, Vol. 116, No. 2, 1994, pp 357-363.

[9] Gary M. R., and Johnson D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.

[10] Gupta P., Janardan R., Majhi J., and Woo T., Efficient geometric algorithms for workpiece orientation in 4- and 5-axis NC machining, *Computer-Aided Design*, Vol. 28, No. 8, 1996, pp 577-587.

[11] Held M., *On the Computational Geometry of Pocket Machining*, Lecture Notes in Computer Science, Vol. 500, Berlin : Springer, 1991.

[12] Held M. and Arkin E. M., Letter to the editor: an algorithm for reducing tool retractions in zigzag pocket machining, *Computer-Aided Design*, Vol. 32, No. 10, 2000, pp 617-9.

[13] Held M., Lukacs G., and Andor L., Pocket machining based on contour-parallel tool paths generated by means of proximity maps, *Computer-Aided Design*, Vol. 26, No. 3, 1994, pp 189-203.

[14] Hopcroft J. E., Motwani R. and Ullman J. D., *Introduction to Automata Theory, Languages, and Computation*, 2nd edition, Addison-Wesley, 2001.

[15] Lin S. and Kernighan B., An effective heuristic algorithm for the traveling salesman problem, *Operations Research*, Vol. 21, No. 2, 1973, pp 498-516.

[16] Maekawa T., An overview of offset curves and surfaces, *Computer-Aided Design*, Vol. 31, No. 3, 1999, pp 165-173.

[17] Slavik P., A tight analysis f the greedy algorithm for set cover, *Journal of Algorithms*, Vol. 25, No. 2, pp 237-54.

[18] Tang K., On minimum traceable partitioning and minimum tool retractions, *Computer-Aided Design*, Vol. 32, No. 10, 2000, pp 621-4.

[19] Tang K., Chen L. L., and Chou S. Y., Optimal workpiece setups for 4-axis numerical control machining based on machinability, *Computers in Industry*, Vol. 37, No. 1, 1998, pp 27-41.

[20] Tang K., Chen C. C., and Dayan Y., Offsetting surface boundaries and 3-axis gouge-free surface machining, *Computer-Aided Design*, Vol. 27, No. 12, 1995, pp 915-927.

[21] Tang K., Chou S. Y., and Chen L. L., An algorithm for reducing tool retractions in zigzag pocket machining, *Computer-Aided Design*, Vol. 30, No. 2, 1998, pp 123-129.

[22] Tang K. and Joneja A., Traversing the machining graph of a pocket, *Computer-Aided Design*, Vol. 35, No. 11, 2003, pp 1023-1040.

[23] Tang K. and Liu Y. J., Maximum intersection of spherical polygons by an arc with applications to 4-axis machining, *Computer-Aided Design*, Vol. 35, No. 14, 2003, pp 1269-1285.

[24] Tang K. and Liu Y. J., A geometric method for determining intersection relations between a movable convex object and a set of planar polygons, *IEEE Transactions on Robotics*, Vol. 20, No. 4, 2004, pp 636-650.

[25] Tang K. and Liu Y. J., An optimization algorithm for free-form surface partitioning based on weighted Gaussian image, *Graphical Models*, Vol. 67, No. 1, 2005, pp 17-42.

[26] Tang K. and Pang A., Optimal connection of loops in laminated object manufacturing, *Computer-Aided Design*, Vol. 35, No. 11, 2003, pp 1011-1022.

[27] Tang K. and Wang W. P., Geometric methods for computing the shortest link of an ordered list of contours, *Unpublished Manuscript*.

[28] Tang K., Woo T., and Gan J., Maximum intersection of spherical polygons and workpiece orientation for 4- and 5-axis machining, *Journal of Mechanical Design*, Vol. 114, No. 3, 1992, pp 477-485.

[29] Tang K., *Congruent Partitioning and Certain Computational Problems*, PhD thesis, Department of Electrical and Engineering and Computer Science, The University of Michigan, 1990.