

International Journal of Computational Geometry & Applications
© World Scientific Publishing Company

MINIMUM AREA CONVEX PACKING OF TWO CONVEX POLYGONS

KAI TANG*

*Department of Mechanical Engineering
Hong Kong University of Science and Technology, Clear Water Bay
Kowloon, Hong Kong, P.R. China
mektang@ust.hk*

CHARLIE C. L. WANG

*Department of Automation and Computer-Aided Engineering
Chinese University of Hong Kong, Shatin, N.T., Hong Kong, P.R. China
cwang@acaе.cuhk.edu.hk*

DANNY Z. CHEN

*Department of Computer Science and Engineering
University of Notre Dame, Notre Dame, IN 46556, USA
chen@cse.nd.edu*

Received 11 August 2004

Revised 22 November 2005

Communicated by JSB Mitchell

ABSTRACT

Given two convex polygons P and Q in the plane that are free to translate and rotate, a convex packing of them is the convex hull of a placement of P and a placement of Q whose interiors do not intersect. A minimum area convex packing of P and Q is one whose area is minimized. The problem of designing a deterministic algorithm for finding a minimum area convex packing of two convex polygons has remained open. We address this problem by first studying the contact configurations between P and Q and their algebraic structures. Crucial geometric and algebraic properties on the area function are then derived and analyzed which enable us to successfully discretize the search space. This discretization, together with a delicate algorithmic design and careful complexity analysis, allows us to develop an efficient $O((n+m)nm)$ time deterministic algorithm for finding a true minimum area convex packing of P and Q , where n and m are the numbers of vertices of P and Q , respectively.

Keywords: Convex packing; minimum area; convex polygon; contact configuration; configuration space.

*Corresponding author.

2 *K. Tang, C.C.L. Wang, and D.Z. Chen*

1. Introduction

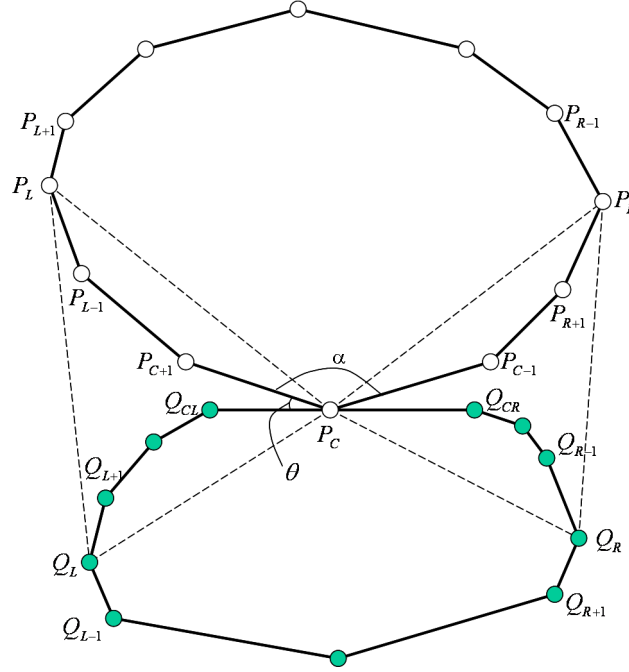
This paper studies the problem of packing two convex polygons by a minimum area convex polygon. Given two arbitrary convex polygons $P = \langle P_0, P_1, \dots, P_{n-1} \rangle$ and $Q = \langle Q_0, Q_1, \dots, Q_{m-1} \rangle$ in the plane, such that P and Q are free to translate and rotate, a *convex packing* (or *packing* for brevity) of them is the convex hull of a placement of P and a placement of Q whose interiors do not intersect. A *minimum area convex packing* (or *minimum packing*) of P and Q then is one whose area is minimized. When cast with different objectives and constraints, this minimum packing problem finds applications in a variety of problems, such as the Pallet Loading problem,^{6,17,8,19} the Cutting-Stock problem,^{1,9} the Bin-Packing problem,^{3,13,4} and the Assortment problem.^{5,11,12,10} From a theoretical point of view, it will be meritorious if an exact and deterministic algorithm can be found for computing a minimum packing efficiently. Earlier work on this problem all has a sub-optimal nature, such as either fixing the orientation between P and Q ⁷ or enclosing the objects by rectangular boxes.^{16,14} Whether there exists a deterministic algorithm for finding a true minimum packing of two arbitrary convex polygons P and Q has remained open.

We present an $O((n+m)nm)$ time deterministic algorithm for finding a true minimum packing of two convex polygons P and Q , thus providing an affirmative answer to this open problem. Our algorithm is based on a careful geometric analysis of the configurations of the convex packing which leads to a characterization of some critical geometric and algebraic properties on the area function of the packing. These properties discretize the search space and thus allow us to search among only a finite number of the packing configurations. An efficient algorithm is then developed to perform this search to find a true minimum area packing, which, based on a careful complexity analysis, takes $O((n+m)nm)$ time and linear space.

The rest of the paper is organized as follows. After giving necessary definitions and preliminaries, Section 2 derives the analytical formation of a special contact configuration space called *mosaic map*, and determines the area function $A(\theta, t)$ of the packing in a mosaic map. Detailed mathematical description of the area function is given in Section 3, which helps lead to an important discovery on the area function — the area function can obtain a minimum only at a finite number of special points in the mosaic map. Section 4 gives a delicate algorithm that efficiently searches through these finite special points and finds one with the minimum area. A careful analysis on the computation and complexity of the algorithm is presented in Sections 5 and 6 that ensures the $O((n+m)nm)$ upper-bound on the running time of the algorithm. Finally, we show some implementation results (Section 7) and conclude the paper (Section 8).

2. Preliminary

Let a placement of P and a placement of Q be said to be in contact of (or touch) each other if they touch each other on their boundaries only (i.e., their interiors do

Fig. 1. Two convex polygons in contact at P_C and $Q_{CL}Q_{CR}$.

not intersect). We can assume that Q is fixed, while P can move freely but keeps in contact with Q . The *contact configuration space* of P and Q is the set of all possible placements of P that are in contact with Q . Without loss of generality, it can be further assumed that a minimum packing of P and Q is realized by a placement of P that has a vertex in contact with an edge of Q . Consider the configuration that P and Q are required to be in contact at a vertex P_C and an edge $Q_{CL}Q_{CR}$, as shown in Fig. 1. There are two parameters, t and θ , that completely determine such a contact. The parameter t is used to specify the position of P_C on $Q_{CL}Q_{CR}$, with $t = \frac{\|P_C Q_{CL}\|}{\|Q_{CR} Q_{CL}\|} \in [0, 1]$. The parameter θ is used to determine the orientation between P and Q . When θ changes, P rotates around the point P_C . Since the interiors of P and Q are not allowed to intersect, all vertices of P can be assumed to lie above the contact edge of Q (when P_C is at Q_{CL} or Q_{CR}), and θ must satisfy $\theta \in [0, \pi - \alpha]$, where $\alpha = \angle P_{C-1}P_C P_{C+1}$. The four special vertices, P_L , P_R , Q_L , and Q_R , are called *hull vertices*, and they form a *hull-quadruple*. The two special edges on the packing — $P_L Q_L$ and $P_R Q_R$ — are called *link edges*. The following definition is in order.

Definition 1. A hull configuration is a subspace in the contact configuration space of P and Q such that the contact vertex of P , the contact edge of Q , and the hull-quadruple all remain the same.

4 *K. Tang, C.C.L. Wang, and D.Z. Chen*

Hereafter it will be assumed that the contact vertex P_C and the contact edge $Q_{CL}Q_{CR}$ are fixed; consequently, a hull configuration is completely determined by the hull-quadruple. The set of points in the $\theta - t$ domain $[0, \pi - \alpha] \times [0, 1]$ that realize a same hull configuration will be referred to as a *cell*. Conceivably, a cell is a contiguous region. Since every (θ, t) point in $[0, \pi - \alpha] \times [0, 1]$ corresponds to a hull configuration, the $\theta - t$ domain $[0, \pi - \alpha] \times [0, 1]$ is thus a partitioning made of cells, which will be called a *mosaic map*. As an example, for the contact vertex P_6 and the contact edge Q_0Q_1 shown in Fig. 2(a), their mosaic map is depicted in Fig. 2(b), where the hull-quadruple of each cell is shown embedded in the cell. Note that a mosaic map, in fact, is conceptually a planar structure called *arrangement* in computational geometry²; but as we will show later, by exploiting the geometric properties of the problem, our algorithm is able to avoid computing the entire structure of the arrangement. As clearly revealed in the above example, excluding those points on the boundary of the domain $[0, \pi - \alpha] \times [0, 1]$, the boundary of a cell consists of some continuous curves, which will be referred to as *extreme curves*. Explicitly, all points on a same extreme curve correspond to an *extreme hull configuration* where a current hull vertex is about to be replaced by a new vertex. Geometrically, this designates a contact situation when an edge of P or Q becomes collinear with a link edge, as entailed below.

Definition 2. A point $(\theta_0, t_0) \in [0, \pi - \alpha] \times [0, 1]$ is called a P_L -*extreme point* in the $\theta - t$ plane if the hull vertex Q_L is collinear with either the edge $P_L P_{L-1}$ or edge $P_L P_{L+1}$. Similarly, (θ_0, t_0) is a Q_L -*extreme point* if the hull vertex P_L is collinear with one of the two edges $Q_L Q_{L-1}$ and $Q_L Q_{L+1}$. Analogously, we define the other two types of extreme hull configuration points, which are all listed in Table 1.

Table 1. Extreme hull configuration points.

Type of collinear configuration points	Geometric condition
P_L - <i>extreme point</i>	Q_L is collinear with $P_L P_{L-1}$ or $P_L P_{L+1}$
P_R - <i>extreme point</i>	Q_R is collinear with $P_R P_{R-1}$ or $P_R P_{R+1}$
Q_L - <i>extreme point</i>	P_L is collinear with $Q_L Q_{L-1}$ or $Q_L Q_{L+1}$
Q_R - <i>extreme point</i>	P_R is collinear with $Q_R Q_{R-1}$ or $Q_R Q_{R+1}$

For instance, in the example given in Fig. 2, the cell of hull-quadruple $P_0 - P_4 - Q_7 - Q_3$ and the cell of $P_0 - P_5 - Q_7 - Q_3$ share an extreme curve of the P_R type; on this curve, both the vertices P_4 and P_5 are the P_R vertex. Also exemplified by the example, which will become clear later, is the fact that every end point of the extreme curves, except those on the boundary of $[0, \pi - \alpha] \times [0, 1]$, is shared by exactly four cells. We call the end points of the extreme curves *critical points*. Obviously, at a critical point in the open domain $(0, \pi - \alpha) \times (0, 1)$, two edges of P and/or Q are involved in the collinearity condition, e.g., the critical point in Fig. 2(b) shared by the cells of the four hull-quadruples — $P_0 - P_4 - Q_7 - Q_3$,

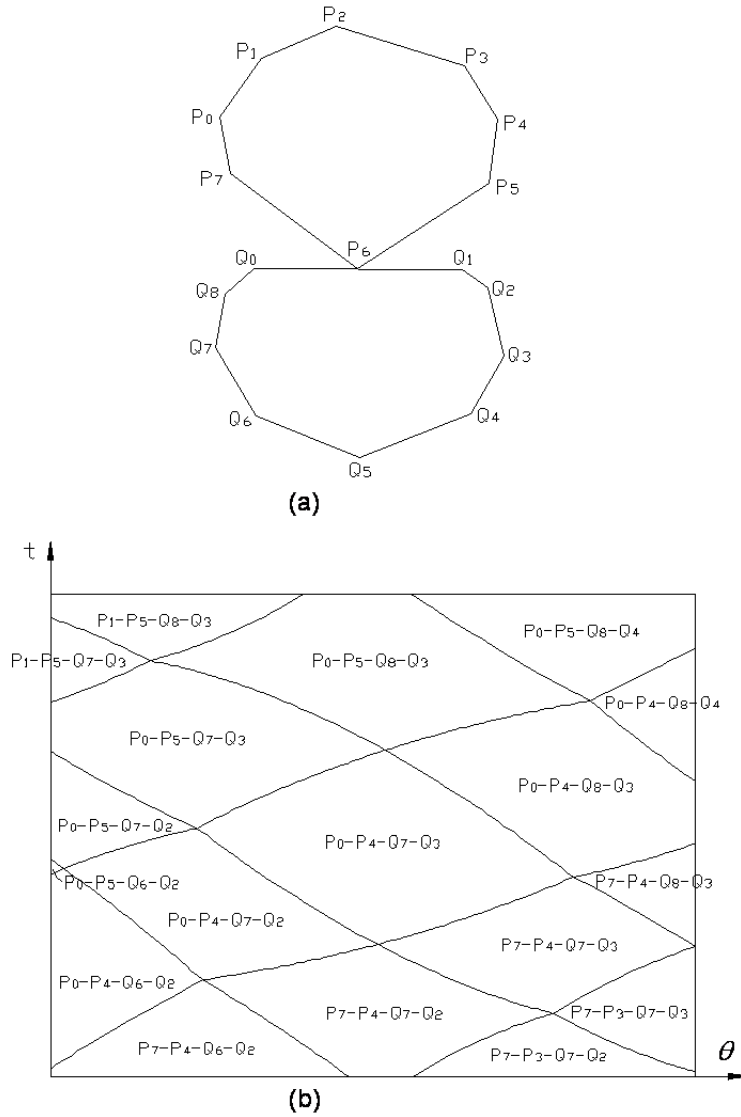


Fig. 2. Example of the mosaic map of a pair of contact vertex and edge: (a) Two convex polygons, and (b) Mosaic map of pair $P_6 - Q_0Q(1)$.

$P_0 - P_5 - Q_7 - Q_3$, $P_0 - P_4 - Q_8 - Q_3$, and $P_0 - P_5 - Q_8 - Q_3$ — defines a contact configuration as shown in Fig. 3(a), where the edges P_4P_5 of P and Q_7Q_8 of Q are collinear with the Q_R vertex Q_3 and the P_L vertex P_0 , respectively. A hull-quadruple could degenerate into a triangle, which corresponds to the case when two critical points are coincident (e.g., the example in Fig. 3(b)).

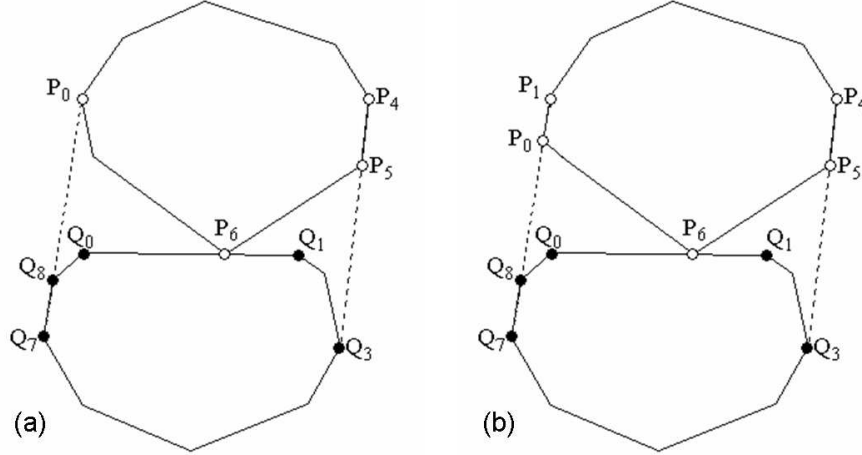
6 *K. Tang, C.C.L. Wang, and D.Z. Chen*

Fig. 3. Geometric interpretation of a critical point: (a) a general case, and (b) an extreme case.

The last item to be defined is the area function of the packing. For any point $(\theta, t) \in [0, \pi - \alpha] \times [0, 1]$, let $A_{hull}(\theta, t)$ denote the area of the packing of P and Q corresponding to the parameter (θ, t) . Referring to Fig. 1,

$$A_{hull}(\theta, t) = A(P_L P_C Q_L) + A(P_R P_C Q_R) + A(P_C Q_L Q_R) \\ + A(P_C P_L \dots P_R) + A(Q_L Q_{L-1} \dots Q_{R+1} Q_R)$$

where we use $A(v_1 v_2 \dots v_n)$ to represent the area of a closed polygon with vertices v_1, v_2, \dots, v_n . When (θ, t) is restricted to be within a cell, the hull-quadruple remains the same, and hence the last two items in the above expression are constants. Therefore, as a function of (θ, t) , we only need to consider the area of the three triangles.

$$A_{tri}(\theta, t) = A(P_L P_C Q_L) + A(P_R P_C Q_R) + A(P_C Q_L Q_R) \quad (1)$$

It is trivial to verify that the function $A_{tri}(\theta, t)$ is continuous in the entire domain $[0, \pi - \alpha] \times [0, 1]$. In addition, within each single cell, $A_{tri}(\theta, t)$ is C^1 continuous in both θ and t . In the next section, a crucial property of $A_{tri}(\theta, t)$ will be shown: the function $A_{tri}(\theta, t)$ can achieve a minimum only at some critical points in $[0, \pi - \alpha] \times [0, 1]$.

3. Minimum Packing Area and Search Space Discretization

Let us define two types of iso-parametric curves on the surface $A_{tri}(\theta, t)$ given in Eq. (1): the t -curve that is the function $A_{tri}(\theta, t)$ when parameter θ is fixed at some θ_0 , and the θ -curve, i.e., $A_{tri}(\theta, t_0)$, for some fixed t_0 . The following property is introduced.

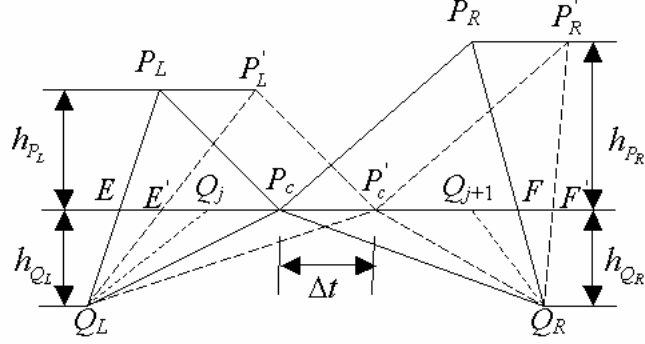


Fig. 4. Proof of Property 1.

Property 1. The surface $A_{tri}(\theta, t)$ is a ruled surface and its t -curves are the rulings.

Proof. As shown in Fig. 4, first we fix θ , and then move the contact vertex P_C along the contact edge $Q_j Q_{j+1}$ to a new position P'_C by a distance Δt , with the new triangle area

$$\begin{aligned} A'_{tri}(\theta, t) &= A(P'_L P'_C Q_L) + A(P'_R P'_C Q_R) + A(P'_C Q_L Q_R) \\ &= A(P_L P_C Q_L) + A(P'_L P_L P_C P'_C) + A(P'_C P_C Q_L) - A(P'_L P_L Q_L) + \\ &\quad A(P_R P_C Q_R) + A(P'_R P_R Q_R) - A(P'_R P_R P_C P'_C) - A(P'_C P_C Q_R) + \\ &\quad A(P_C Q_L Q_R) + A(P'_C P_C Q_R) - A(P'_C P_C Q_L) \end{aligned}$$

Since $\|P_L P'_L\| = \|P_R P'_R\| = \|P_C P'_C\| = \Delta t$, we have

$$\begin{aligned} \Delta A &= A'_{tri} - A_{tri} \\ &= A(P'_L P_L P_C P'_C) + A(P'_C P_C Q_L) - A(P'_L P_L Q_L) + \\ &\quad A(P'_R P_R Q_R) - A(P'_R P_R P_C P'_C) - A(P'_C P_C Q_R) + \\ &\quad A(P'_C P_C Q_R) - A(P'_C P_C Q_L) \end{aligned}$$

i.e.,

$$\Delta A = \frac{h_{P_L} + h_{Q_R} - h_{P_R} - h_{Q_L}}{2} \Delta t. \quad (2)$$

The above equation indicates that any t -curve is linear in t , and consequently the surface $A_{tri}(\theta, t)$ is ruled. \square

As a direct consequence of Property 1, $A_{tri}(\theta, t)$ can achieve its minimum only on the boundary of its parameter domain, as stipulated below.

8 *K. Tang, C.C.L. Wang, and D.Z. Chen*

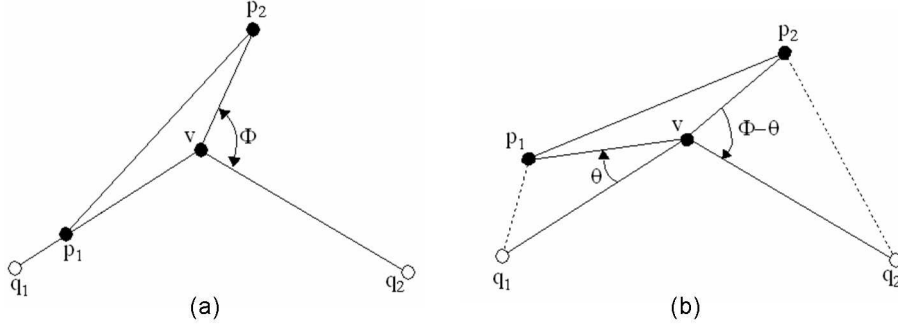


Fig. 5. Rotating a triangle about a shared vertex.

Corollary 1. *The minimum of function $A_{tri}(\theta, t)$ occurs only on the boundary of the cell in which the function is defined.*

Next, we provide a useful lemma on the area of the packing of two triangles sharing a common vertex. Referring to Fig. 5, suppose initially the edge p_1v of triangle Δp_1p_2v overlaps with edge q_1v of triangle Δq_1q_2v , and the initial angle $\angle p_2vq_2$ is Φ (Fig. 5(a)). The triangle Δp_1p_2v is then rotated clockwise about the shared vertex v (Fig. 5(b)). Considering the sum $A(\theta)$ of the areas of the two triangles Δp_1vq_1 and Δp_2vq_2 as a function of the rotation angle θ , we have the following lemma.

Lemma 1. *The minimum of function $A(\theta): \theta \in [0, \Phi]$ occurs only at $A(0)$ or $A(\Phi)$.*

Proof. We have

$$A(\theta) = \frac{1}{2}l_{p_1v}l_{q_1v} \sin(\theta) + \frac{1}{2}l_{p_2v}l_{q_2v} \sin(\Phi - \theta) \quad (3)$$

whose 1st derivative is

$$A'(\theta) = \frac{1}{2}l_{p_1v}l_{q_1v} \cos(\theta) - \frac{1}{2}l_{p_2v}l_{q_2v} \cos(\Phi - \theta) \quad (4)$$

and we can also determine its 2nd derivative as

$$A''(\theta) = -\frac{1}{2}l_{p_1v}l_{q_1v} \sin(\theta) - \frac{1}{2}l_{p_2v}l_{q_2v} \sin(\Phi - \theta) \quad (5)$$

Since both θ and $\Phi - \theta$ are internal angles of two valid triangles Δp_1q_1v and Δp_2q_2v , we have $0 \leq \theta \leq \pi$ and $0 \leq \Phi - \theta \leq \pi$. The second derivative of $A(\theta)$ thus remains negative in $\theta \in [0, \Phi]$, which means it cannot have a minimum in the open domain $\theta \in (0, \Phi)$. There are two extreme cases: 1) $\theta > \pi$ and $0 < \Phi - \theta < \pi$, or 2) $0 < \theta < \pi$ and $\Phi - \theta > \pi$. However, when such a case occurs, for case 1), the area function in Eq. (1) degenerates to

$$A_{tri}(\theta, t) = A(P_R P_C Q_R) + A(P_C Q_L Q_R)$$

where $A(\theta) = \frac{1}{2}l_{p_2v}l_{q_2v}\sin(\Phi - \theta)$ leads to $A''(\theta) = -\frac{1}{2}l_{p_2v}l_{q_2v}\sin(\Phi - \theta) < 0$; for case 2),

$$A_{tri}(\theta, t) = A(P_L P_C Q_L) + A(P_C Q_L Q_R)$$

and similarly, $A(\theta) = \frac{1}{2}l_{p_1v}l_{q_1v}\sin(\theta)$ leads to $A''(\theta) = -\frac{1}{2}l_{p_1v}l_{q_1v}\sin(\theta) < 0$. \square

With Lemma 1, the following property on the other iso-parametric curve, the θ -curve, is established.

Property 2. The θ -curve of surface $A_{tri}(\theta, t)$, i.e., $A_{tri}(\theta, t_0)$ with $\theta \in [0, \pi - \alpha]$, is either strictly monotone, or first strictly increases and then strictly decreases.

Proof. Referring to Fig. 1, when t is fixed, only the areas of the two triangles $A(P_L P_C Q_L)$ and $A(P_R P_C Q_R)$ change when θ varies. The geometric configuration of these two triangles is however identical to that given in Fig. 5. Due to Lemma 1, thus, $A_{tri}(\theta, t)$ cannot have a minimum in the open domain $\theta \in (0, \pi - \alpha)$ with $\alpha = \angle P_{C+1} P_C P_{C-1} \geq \angle P_L P_C P_R$. Furthermore, since $\alpha > 0$, $A_{tri}(\theta, t)$ can have at most one maximum in $\theta \in (0, \pi - \alpha)$. \square

Before proceeding further, it is imperative to ensure that the set of all mosaic maps covers the *entire* contact configuration space. To illustrate the importance of this, Fig. 6(b) displays a particular contact configuration in which P and Q share a common vertex v . By inferring from Fig. 6(a) and 6(c), it is obvious that this configuration belongs to neither the mosaic map of $v - e_1$ nor the mosaic map of $v - e_2$. However, if we switch the roles between P and Q and let Q move around the stationary P , then we can show that the contact of Fig. 6(b) must lie in one of the mosaic maps of $v - E_1$ or $v - E_2$. To see this, refer to Fig. 6(d); in order for a contact configuration (in which the two polygons share the vertex v) not to be covered by any one of the mosaic maps of $v - e_1$ and $v - e_2$, the inequality $\alpha < \beta$ must hold so that there exists a configuration with P covering both sides of the edges e_1 and e_2 . On the other hand, in order for this contact configuration to be covered by the mosaic maps of neither $v - E_1$ nor $v - E_2$, we must have $\alpha > \beta$. The only possibility is $\alpha = \beta$, in which case the corresponding contact configuration is a corner point in a mosaic map, as manifested in Fig. 6(e).

Having assured the equivalence between the union of the mosaic maps and the contact configuration space, we next prove that no corner point, i.e., $(0, 0)$, $(0, 1)$, $(\pi - \alpha, 0)$, and $(\pi - \alpha, 1)$ (the cases shown in Fig. 7), in any mosaic map can achieve the minimum area unless it is on some extreme curve. This relieves us from worrying about those points at the corners of mosaic maps, which in turn will help reduce the time complexity of the final algorithm tremendously. Hereafter, we use A_{\min} to denote the area of a minimum packing of P and Q .

Lemma 2. *If the minimum area A_{\min} is achieved by a packing C corresponding to a corner of a mosaic map, then this corner point must be on an extreme curve.*

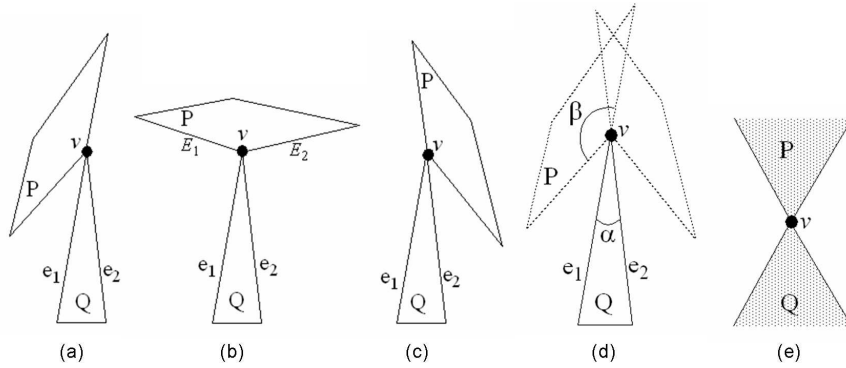


Fig. 6. Contact configuration not covered by mosaic maps.

Proof. It is easily discernible that any of the four corners of a mosaic map can be geometrically categorized as one of the two patterns, namely, “dangling” and “overlapping”, shown as packing C in Fig. 7(a) and 7(b), respectively. In the case of a “dangling” C , according to Lemma 1, we can rotate P about the shared vertex v , either clockwise or counter-clockwise depending on in which direction the area function decreases, and have the comfort of knowing that after rotating a small angle $\Delta\theta$ the resulting packing C' will have less area than that of C . For the latter case when C is of the “overlapping” pattern, by Property 1, we translate P , either to the left or right (again depending on in which direction the area function decreases), by a small distance Δt to get a new packing C' whose area is assured to be smaller than that of C .

The above analysis is valid as long as $\Delta\theta$ or Δt exists; that is, during the movement of $\Delta\theta$ or Δt , the hull-quadruple remains the same. Since the hull-quadruple changes only on extreme curves (for example, Fig. 7(c) and 7(d)), we have proved our lemma. \square

We note on the proof of Lemma 2 that the example given in Fig. 7(d) for the “overlapping” case is indeed of an extreme nature, albeit a degenerate one, as in this case the hull vertex P_L of P coincides with the hull vertex Q_L of Q . (Notice the important difference between (b) and (d) in Fig. 7: In (b), the hull vertex P_L does not coincide with the hull vertex Q_L , whereas in (d) they do.) Also as the properties 1 and 2 are held, the min-area packing cannot be achieved at a point on one of the 4 edges of a mosaic map that is not on an extreme curve.

We are now ready to present a key result that completes the task of the search space discretization, by eliminating the interior points on any extreme curves from the search space. We have the following lemma.

Lemma 3. *The minimum area packing cannot be obtained in the interior of any extreme curve.*

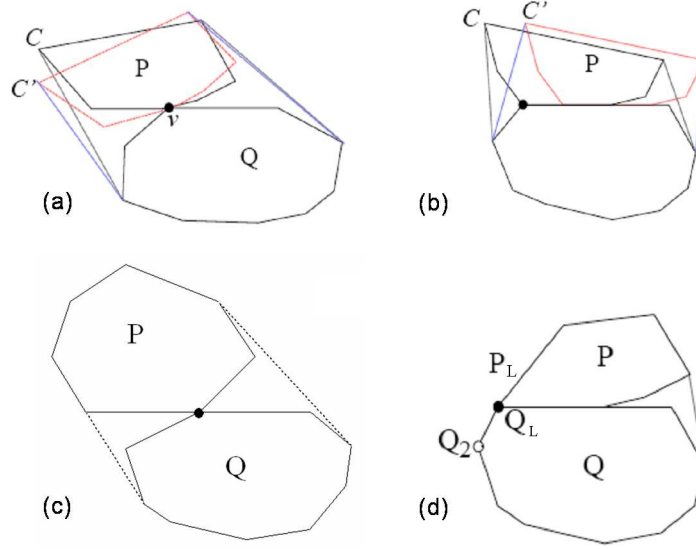


Fig. 7. Proof of Lemma 2: (a) “dangling”, (b) “overlapping”, (c) extreme “dangling”, and (d) extreme “overlapping”.

Proof. We only prove the lemma for the Q_R -extreme case, and the other three extreme cases can be analyzed analogously. Let $\sigma(s)$ be a Q_R -extreme curve between two cells χ_1 and χ_2 , and A_1 and A_2 be the area function A_{hull} defined on the two cells respectively, where s is the arc length of curve σ . Obviously, on σ , A_1 and A_2 agree with each other, i.e., $A_1(s) = A_2(s)$ for all points on σ . Suppose that the univariate function $A_1(s)$ ($A_2(s)$) obtains a local minimum at an interior point $(t_0, \theta_0) = \sigma(s_0)$ of σ . Without loss of generality, let Q_1Q_2 be the edge that contributes to the Q_R -extremity. To make a local minimum, referring to Fig. 8(a) and (b), at point (t_0, θ_0) , according to Eq. (2), we should have

$$\frac{\partial A_1}{\partial t} \Big|_{(\theta_0, t_0)} = \frac{d_1}{2} + K < 0$$

and

$$\frac{\partial A_2}{\partial t} \Big|_{(\theta_0, t_0)} = \frac{d_2}{2} + K > 0$$

where d_1 and d_2 are the h_{Q_R} length for cells χ_1 and χ_2 respectively, and K is $\frac{1}{2}(h_{P_L} - h_{P_R}) + h_{Q_L}$. Since d_2 is always larger than d_1 , the above two inequalities indicate that there is a unique point Q_2^* on edge Q_1Q_2 such that the h_{Q_R} length d^* of it leads to $\frac{d^*}{2} + K = 0$, as shown in Fig. 8(c). Now, suppose we replace the original polygon Q with the new one Q^* obtained from Q by replacing its vertex Q_2 with Q_2^* , which also alters functions A_1 and A_2 to A_1^* and A_2^* respectively. It is not difficult to see that, as can be validated from the formulas in Section 5.1 — Eq. (13), for the new polygon Q^* the original Q_R -extreme curve σ remains unchanged locally. Moreover,

12 *K. Tang, C.C.L. Wang, and D.Z. Chen*

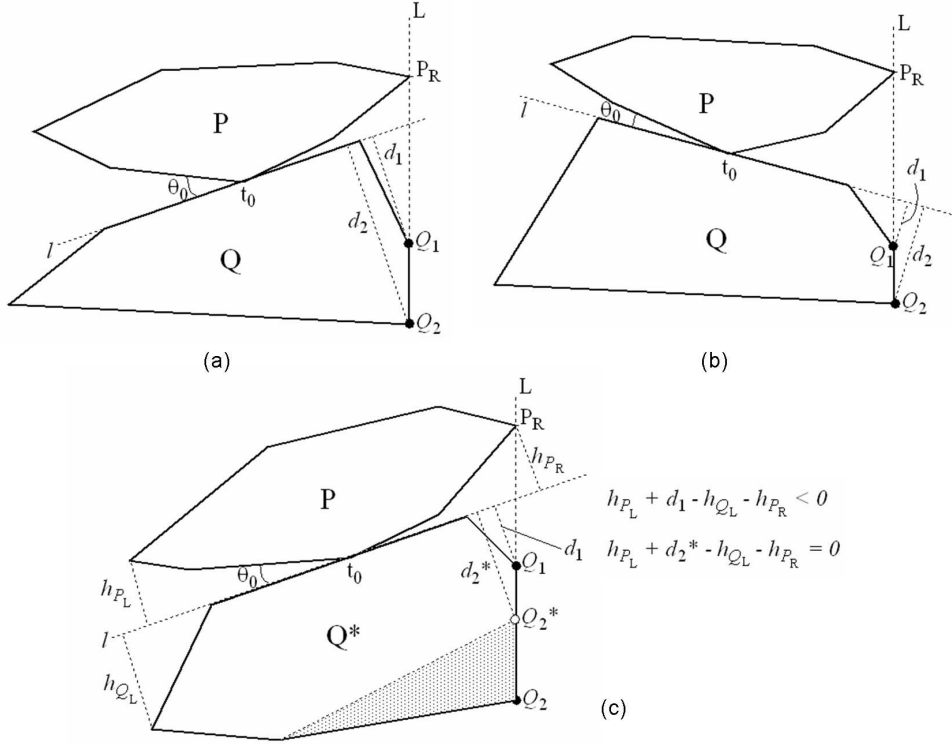


Fig. 8. Proof of Lemma 3.

function A_1^* only differs from A_1 by a constant (the area of the shaded triangle in Fig. 8(c)). Therefore, $(t_0, \theta_0) = \sigma(s_0)$ is still expected to be a local minimum point of $A_1^*(s)$. Since $A_2^*(s)$ and $A_1^*(s)$ are the same, $(t_0, \theta_0) = \sigma(s_0)$ should also be a local minimum point of $A_2^*(s)$. By classical optimization theory of multivariate functions, the tangent to curve σ at $(t_0, \theta_0) = \sigma(s_0)$ thus should be perpendicular to both the gradients of A_1^* and A_2^* , which means $\left. \frac{\partial A_1^*}{\partial t} \right|_{(t_0, \theta_0)} = k \left. \frac{\partial A_2^*}{\partial t} \right|_{(t_0, \theta_0)}$, where k is just some negative number. However, this equality cannot hold since $\left. \frac{\partial A_2^*}{\partial t} \right|_{(t_0, \theta_0)} = \frac{d_2^*}{2} + K = 0$ but $\left. \frac{\partial A_1^*}{\partial t} \right|_{(t_0, \theta_0)} = \frac{d_1}{2} + K < 0 \neq 0$. Therefore, point $(t_0, \theta_0) = \sigma(s_0)$ cannot be a local minimum on $A_1^*(s)$ and $A_2^*(s)$, a contradiction. \square

With Lemmas 2 and 3, and Properties 1 and 2, we conclude the section by the following theorem, in which the *critical points* are the intersections of extreme curves on the mosaic map.

Theorem 1. *A minimum area packing C of P and Q can be obtained only at certain critical points in some mosaic maps.*

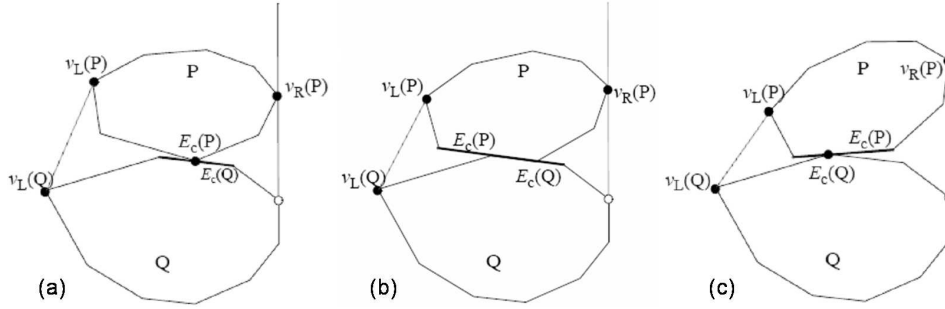


Fig. 9. Types of states: (a) and (c) General states; (b) singular state.

4. Details of the Algorithm

The successful discretization of the search space as stipulated by Theorem 1 enables us to develop a deterministic method for finding a minimum packing — there are only a finite number of critical points in all mosaic maps. Our next task is then to design an efficient algorithm to perform this deterministic search.

Refer to Fig. 9. Again, suppose Q is fixed while P can be rotated and translated. Also assume that the rightmost edge of Q is vertical (on line L). Consider all those packings C whose hull vertex P_R lies on line L . Then a general configuration of C can be specified by a 5-tuple:

$$\langle E_c(P), E_c(Q), v_L(P), v_R(P), v_L(Q) \rangle$$

where $E_c(P)$ and $E_c(Q)$ stand for the boundary entities (vertices or edges) on P and Q that are in contact with each other, $v_L(P)$ and $v_R(P)$ are respectively the two hull vertices P_L and P_R on P , and $v_L(Q)$ is the corresponding hull vertex Q_L on Q . We call such a 5-tuple a **state** of C . A state is said to be **singular** if one or more of the following types occur:

- (1) both $E_c(P)$ and $E_c(Q)$ are vertices;
- (2) both $E_c(P)$ and $E_c(Q)$ are edges;
- (3) some vertex P_i , in addition to $v_R(P)$, lies on line L ;
- (4) some vertex P_i is collinear with $v_L(P)$ and $v_L(Q)$;
- (5) some vertex Q_i is collinear with $v_L(P)$ and $v_L(Q)$.

A 5-tuple that is not singular is called a **general** state, e.g., the states shown in Fig. 9(a) and 9(c), whereas the state given in Fig. 9(b) belongs to the singular type (2). Geometrically, singular states of type (1) correspond to those critical points on the boundary $t = 0$ or $t = 1$ of a mosaic map, and type (2) singular states pertain to critical points on the boundary $\theta = 0$ or $\theta = \pi - \alpha$. All the last three types are associated with the interior critical points in the mosaic map (i.e., the critical points not on the 4 edges of a mosaic map), where an interior critical point is the

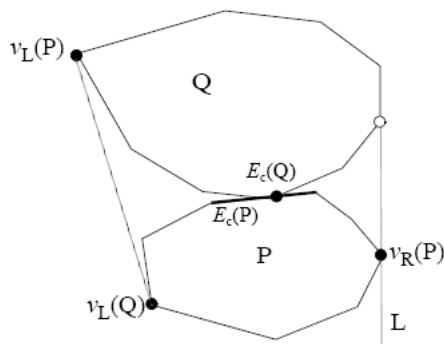


Fig. 10. A state of bottom-support type.

intersection between the Q_R -extreme curve and a P_R -extreme curve (type (3)), or a P_L -extreme curve (type (4)), or a Q_L -extreme curve (type (5)).

A state as exemplified by Fig. 9 actually belongs to the category of **top-support** type, as the hull vertex P_R is above the hull vertex Q_R on line L . Accordingly, a state can also be in the **bottom-support** category, as shown in Fig. 10, where P_R is below Q_R . With both top- and bottom-support types defined, and by switching the roles between P and Q , based on Theorem 1, it immediately follows that any minimum packing must correspond to some singular state of the top-support or bottom-support type.

We distinguish a state from its *instances*. An instance of a state is a specific placement of P that meets the 5-tuple condition of that state. Obviously, a general state has an infinite number of instances, whereas any singular state has exactly one (isolated) instance. An instance of state S_i thus can be represented as (S_i, θ) , where θ is the θ -value of the placement of P in the mosaic map associated with S_i . The following lemma is then given.

Lemma 4. *The set of all instances of any state S_i forms exactly a contiguous portion on an extreme curve in some mosaic map.*

Consider using vertical lines (i.e., θ is a constant) to cut the extreme curves in a mosaic map; then the part between two intersections belongs to the same general state, and the intersection points are on singular states. Similarly, when horizontal lines are used (i.e., t is a constant on the line), we can draw the same conclusion — the part between two intersections represents the same general state. Together with the definitions of extreme curves, the above lemma can be easily proved.

Next, let $\{S_0, S_1, \dots, S_{K-1}\}$ be the set of all distinct top-support general (non-singular) states of C (restricted with respect to the vertical line L), sorted according to P 's orientations of their instances, in which the vertices of P touch the line L in the counterclockwise order. State S_i is called the **successor state** of state S_{i-1} ,

and S_{i-1} is called the **predecessor state** of state S_i ($i-1$ and $i \bmod K$). For example, the state shown in Fig. 9(c) succeeds the state given in Fig. 9(a). We will use $next(S_i)$ to denote the successor state of S_i , i.e., $S_{i+1} = next(S_i)$. It is easy to see that between two consecutive states S_i and S_{i+1} , there is one and only one singular state, e.g., the state in Fig. 9(b) is the sole singular state between the two states shown in Fig. 9(a) and 9(c). As to be shown in the next section, given S_i , its ensuing singular state is uniquely determined. Let us use $singular(S_i)$ to denote the singular state between S_i and $next(S_i)$.

We are now ready to present the outline of our algorithm for finding a minimum area convex packing of P and Q . The algorithm given below (in Table 2) only considers the case of the top-support singular states; the treatment for the bottom-supported case can be derived analogously. In the algorithm, we use **Area_singular** (S) to denote the function that returns the area of the packing defined by the singular state S . Two other procedures will be used, both taking as input an instance (S_i, θ_0) : **Get_singular** (S_i, θ_0) which returns the sole instance of the singular state $singular(S_i)$, and **Get_next** (S_i, θ_0) which returns the state $next(S_i)$.

Table 2. Algorithm **Min_area_convex_hull_top** (P, Q).

```

Algorithm Min_area_convex_hull_top ( $P, Q$ )
/* Find a top-supported minimum packing of  $P$  and  $Q$  */
Begin
Step 1. For every edge  $e$  of  $Q$  do {
Step 1.1. Rotate  $Q$  so that  $e$  becomes vertical and also its right-most edge;
Step 1.2.  $(S_0, \theta_0) \leftarrow$  an instance of an arbitrary non-singular state of  $P$ ;
         $S \leftarrow S_0$ ;
         $S_{next} \leftarrow$  Get_next ( $S, \theta_0$ );
Step 1.3. While ( $S_{next} \neq S_0$ ) do {
         $(S_s, \theta_s) \leftarrow$  Get_singular ( $S, \theta_0$ );
         $A_m \leftarrow$  Area_singular ( $S_s$ ); /* the area of singular state  $S_s$  */
        If ( $A_m < A_{min}$ ) then { /*  $A_{min}$  is initialized to  $+\infty$  */
             $\{S_{min}, A_{min}\} \leftarrow \{S_s, A_m\}$ 
        }
         $S \leftarrow S_{next}$ ;
         $\theta \leftarrow \theta_s$ ;
         $S_{next} \leftarrow$  Get_next ( $S, \theta_0$ );
    } /* end of While */
} /* end of For statement */
End.

```

In the next section, details will be given on how to mathematically as well as algorithmically perform the three functions **Get_next**(), **Get_singular**(), and **Area_singular**(). As we will show, based on incremental updating, these three operations all take *constant* time. In Section 6, we will prove that, with respect to a vertical supporting line, P can have at most $O(nm)$ distinct non-singular states (top-supported). Consequently, the entire While loop at Step 1.3 takes $O(nm)$ time. All the other steps in the For loop at Step 1 obviously require no more than linear

16 *K. Tang, C.C.L. Wang, and D.Z. Chen*

$O(n + m)$ time. Therefore, the whole For loop at Step 1 takes $O(nm^2)$ time. Step 2 is easily seen to take $O(n + m)$ time. Thus, algorithm **Min_area_convex_hull_top** has a running time of $O(nm^2)$. In the same spirit, we can design an $O(nm^2)$ time algorithm for those bottom-supported singular states. Adding the consideration that we also need to switch the roles between P and Q so as to capture all the possible contact configurations, we arrive at the following result.

Theorem 2. *A minimum area convex packing of two arbitrary convex polygons P and Q , when both are allowed to translate and rotate, can be found in $O(nm^2 + n^2m) = O((n + m)nm)$ time and linear space.*

5. Algebraic Formulation and Computation of Singular and Successor States

The core of the algorithm **Min_area_convex_hull_top** consists of three functions **Get_next()**, **Get_singular()**, and **Area_singular()**. They all depend on the mathematical formulation of extreme curves. In this section, we first derive the exact algebraic formulations for the extreme curves; we then elaborate on how the two functions **Get_next()** and **Get_singular()** should be implemented; finally, we describe the algorithmic mechanism that ensures a constant time for computing **Get_singular()**.

5.1. Mathematical formulation of extreme curves

There are exactly four kinds of extreme curves, each corresponding to one of the four hull vertices in a hull configuration. As already alluded, an extreme curve is a simple curve in a mosaic map. This curve can always be put in an explicit form $t = f(\theta)$, which we derive one by one next. In the following derivations, we use \bar{L} to denote the length of the contact edge $Q_{CL}Q_{CR}$, h for the length $t \cdot \bar{L}$, β for $\angle P_L P_C P_R$, β_L for $\angle P_L P_C P_{C+1}$, and β_R for $\angle P_R P_C P_{C-1}$.

5.1.1. P_L -configuration extreme curve

The P_L -configuration extreme curve consists of those points in the $\theta - t$ plane for which the three hull vertices P_L , P_{L-1} , and Q_L are collinear, as shown in Fig. 11(a). To maintain this condition, the values of t and θ must ensure the equation

$$\frac{\|P'_{L-1}Q_L\|}{\|P_{L-1}P'_L\|} = \frac{\|P'_L Q_L\|}{\|P_L P'_L\|}, \text{ which leads to} \quad (6)$$

$$\frac{h - l_{Q_{CL}Q_L} \cos \varphi_1 - l_{P_{L-1}P_C} \cos(\theta + \beta_L - \alpha_2)}{l_{P_{L-1}P_C} \sin(\theta + \beta_L - \alpha_2) + l_{Q_{CL}Q_L} \sin \varphi_1} = \frac{h - l_{Q_{CL}Q_L} \cos \varphi_1 - l_{P_L P_C} \cos(\theta + \beta_L)}{l_{P_L P_C} \sin(\theta + \beta_L) + l_{Q_{CL}Q_L} \sin \varphi_1}.$$

Simplifying the above equation, the P_L -configuration extreme curve $t = f_{PL}(\theta)$ is determined by

$$f_{PL}(\theta) = \frac{1}{L(l_{P_{L-1}P_C} \sin(\alpha_2 - \theta - \beta_L) + l_{P_L P_C} \sin(\theta + \beta_L))} [l_{P_{L-1}P_C} l_{P_L P_C} \sin \alpha_2 + l_{P_L P_C} l_{Q_{CL}Q_L} \sin(\theta + \beta_L - \varphi_1) + l_{P_{L-1}P_C} l_{Q_{CL}Q_L} \sin(\alpha_2 - \theta - \beta_L + \varphi_1)]. \quad (7)$$

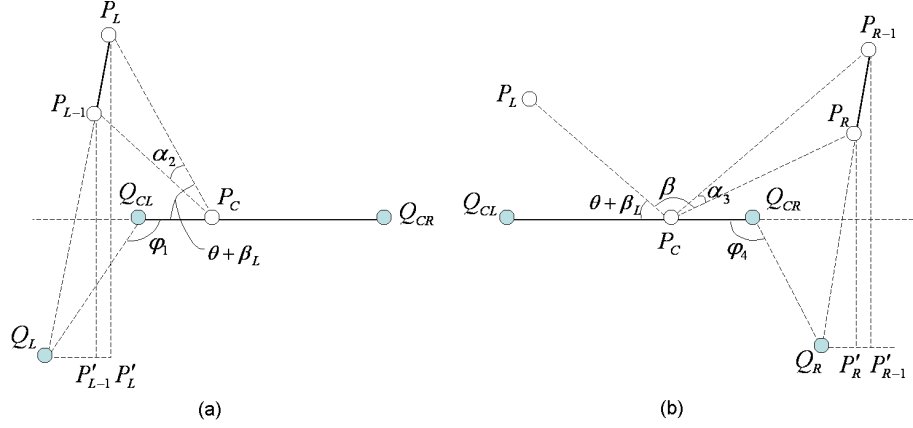


Fig. 11. Conditions of P_L - and P_R -extreme curves: (a) for P_L -extreme curve, and (b) for P_R -extreme curve.

For the collinearity of P_L , P_{L+1} , and Q_L , the formula is similar — by just replacing P_{L-1} and P_L with P_L and P_{L+1} . The following three extreme curves have the same property.

5.1.2. P_R -configuration extreme curve

A point (θ, t) is on the P_R -configuration extreme curve when P_{R-1} , P_R , and Q_R are collinear, which corresponds to the equation $\frac{\|Q_R P'_R\|}{\|P_R P'_R\|} = \frac{\|Q_R P'_{R-1}\|}{\|P_{R-1} P'_{R-1}\|}$ (see Fig. 11(b)). Accordingly, we have

$$\frac{l_{P_R P_C} \cos(\pi - \beta - \theta - \beta_L) - [(\bar{L} - h) - l_{Q_{CR} Q_R} \cos(\varphi_4)]}{l_{P_R P_C} \sin(\pi - \beta - \theta - \beta_L) + l_{Q_{CR} Q_R} \sin \varphi_4} = \frac{l_{P_{R-1} P_C} \cos(\pi - \beta - \theta - \beta_L + \alpha_3) - [(\bar{L} - h) - l_{Q_{CR} Q_R} \cos \varphi_4]}{l_{P_{R-1} P_C} \sin(\pi - \beta - \theta - \beta_L + \alpha_3) + l_{Q_{CR} Q_R} \sin \varphi_4}. \quad (8)$$

After simplification, the P_R -configuration extreme curve $t = f_{P_R}(\theta)$ becomes

$$\begin{aligned} f_{P_R}(\theta) = & \frac{1}{\bar{L}(l_{P_{R-1} P_C} \sin(\alpha_3 - \beta - \theta - \beta_L) + l_{P_R P_C} \sin(\beta + \theta + \beta_L))} [l_{P_{R-1} P_C} l_{P_R P_C} \sin \alpha_3 \\ & + l_{P_R P_C} \bar{L} \sin(\beta + \theta + \beta_L) + l_{P_{R-1} P_C} \bar{L} \sin(\alpha_3 - \beta - \theta - \beta_L) \\ & - l_{P_R P_C} l_{Q_{CR} Q_R} \sin(\beta + \varphi_4 + \theta + \beta_L) \\ & - l_{P_{R-1} P_C} l_{Q_{CR} Q_R} \sin(\alpha_3 - \beta - \varphi_4 - \theta - \beta_L)]. \end{aligned} \quad (9)$$

5.1.3. Q_L -configuration extreme curve

The Q_L -configuration extreme curve identifies the collinearity among the three hull vertices P_L , Q_L , and Q_{L-1} . This collinearity requires the equation $\frac{\|Q_{L-1} Q'_L\|}{\|Q_L Q'_L\|} =$

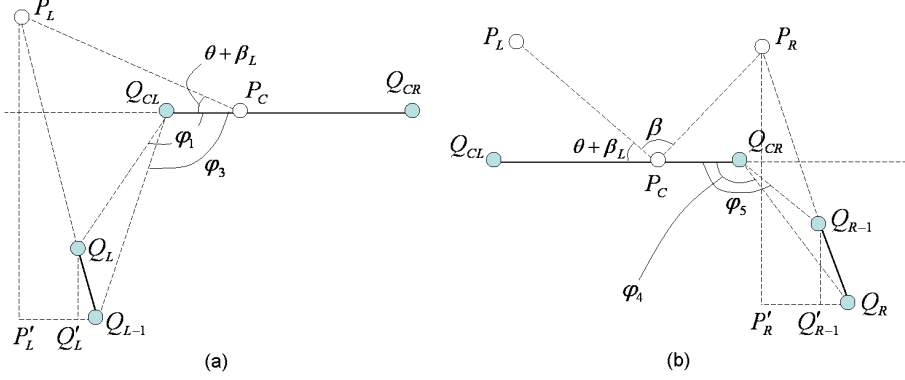
18 *K. Tang, C.C.L. Wang, and D.Z. Chen*


Fig. 12. Conditions of Q_L - and Q_R -extreme curves: (a) for Q_L -extreme curve, and (b) for Q_R -extreme curve.

$\frac{\|Q_{L-1}P'_L\|}{\|P_L P'_L\|}$ (see Fig. 12(a)), which leads to

$$\frac{(h-l_{Q_L Q_{CL}} \cos \varphi_1) - (h-l_{Q_{L-1} Q_{CL}} \cos \varphi_3)}{l_{Q_{L-1} Q_{CL}} \sin \varphi_3 - l_{Q_L Q_{CL}} \sin \varphi_1} = \frac{l_{P_L P_C} \cos(\theta + \beta_L) - (h-l_{Q_{L-1} Q_{CL}} \cos \varphi_3)}{l_{P_L P_C} \sin(\theta + \beta_L) + l_{Q_{L-1} Q_{CL}} \sin \varphi_3}. \quad (10)$$

Thus, the Q_L -configuration extreme curve $t = f_{Q_L}(\theta)$ has the following mathematical formulation

$$f_{Q_L}(\theta) = \frac{1}{\bar{L}(-l_{Q_L Q_{CL}} \sin \varphi_1 + l_{Q_{L-1} Q_{CL}} \sin \varphi_3)} [-l_{Q_L Q_{CL}} l_{Q_{L-1} Q_{CL}} \sin(\varphi_1 - \varphi_3) - l_{P_L P_C} l_{Q_L Q_{CL}} \sin(\varphi_1 - \theta - \beta_L) + l_{P_L P_C} l_{Q_{L-1} Q_{CL}} \sin(\varphi_3 - \theta - \beta_L)]. \quad (11)$$

5.1.4. Q_R -configuration extreme curve

Finally, for a point (θ, t) to be on the Q_R -configuration extreme curve when P_R , Q_{R-1} , and Q_R are collinear, it must maintain the equation $\frac{\|Q_R Q'_{R-1}\|}{\|Q_{R-1} Q'_{R-1}\|} = \frac{\|Q_R P'_R\|}{\|P_R P'_R\|}$ (see Fig. 12(b)). It leads to

$$\frac{[(\bar{L}-h) - l_{Q_R Q_{CR}} \cos \varphi_4] - [(\bar{L}-h) - l_{Q_{R-1} Q_{CR}} \cos \varphi_5]}{l_{Q_R Q_{CR}} \sin \varphi_4 - l_{Q_{R-1} Q_{CR}} \sin \varphi_5} = \frac{(\bar{L}-h) - l_{Q_R Q_{CR}} \cos \varphi_4 - l_{P_R P_C} \cos(\pi - \beta - \theta - \beta_L)}{l_{Q_R Q_{CR}} \sin \varphi_4 + l_{P_R P_C} \sin(\pi - \beta - \theta - \beta_L)}. \quad (12)$$

Thus, the Q_R -configuration extreme curve $t = f_{Q_R}(\theta)$ is given by

$$f_{Q_R}(\theta) = \frac{1}{\bar{L}(l_{Q_R Q_{CR}} \sin \varphi_4 - l_{Q_{R-1} Q_{CR}} \sin \varphi_5)} [l_{Q_R Q_{CR}} \bar{L} \sin \varphi_4 - l_{Q_{R-1} Q_{CR}} l_{Q_R Q_{CR}} \sin(\varphi_4 - \varphi_5) + l_{P_R P_C} l_{Q_R Q_{CR}} \sin(\beta + \varphi_4 + \theta + \beta_L) - l_{Q_{R-1} Q_{CR}} \bar{L} \sin \varphi_5 - l_{P_R P_C} l_{Q_{R-1} Q_{CR}} \sin(\beta + \varphi_5 + \theta + \beta_L)]. \quad (13)$$

5.2. Determining $singular(S_i)$ and $next(S_i)$

Given an instance (S_i, θ_0) , we want to identify the successor state $next(S_i)$ and the in-between singular state $singular(S_i)$, which are found by the two procedures

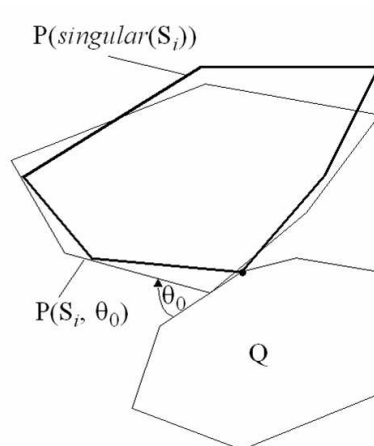


Fig. 13. Geometric derivation of type I singular state.

$\mathbf{Get_next}(S_i, \theta_0)$ and $\mathbf{Get_singular}(S_i, \theta_0)$, respectively. As to be seen soon, state $\mathit{next}(S_i)$ is readily derivable once $\mathit{singular}(S_i)$ is available. Therefore, we focus on $\mathbf{Get_singular}(S_i, \theta_0)$ first.

By definition, state S_i falls on a Q_R -configuration extreme curve $t = f_{QR}(\theta)$ in some mosaic map; that is, the three hull vertices P_R , Q_{R-1} , and Q_R are collinear (on the vertical line L) for all instances of S_i . Given an instance (S_i, θ_0) , the singular state $\mathit{singular}(S_i)$ can be of any of the five types; we analyze them one by one.

5.2.1. Type I singular state

If $\mathit{singular}(S_i)$ is of type I, then P and Q are in vertex-vertex contact at $\mathit{singular}(S_i)$. Assume that $E_c(P)$ is a vertex and $E_c(Q)$ is an edge in S_i . Conceivably, at $\mathit{singular}(S_i)$, vertex $E_c(P)$ coincides with an end point of edge $E_c(Q)$. Therefore, the t value of the single corresponding point of $\mathit{singular}(S_i)$ in the mosaic map must be 0 or 1, and its θ value is determined by solving the following two equations:

$$\begin{cases} 0 = f_{QR}(\theta) \\ 1 = f_{QR}(\theta) \end{cases} \quad (14)$$

The solution to the above two equations (which are simple sinusoidal equations) will be at most two θ values θ_1 and θ_2 . They must, however, be validated to be within the θ -domain $[0, \pi - \alpha]$ of the mosaic map. Assuming the most general case that both are valid, then the one which is smaller than θ_0 of the given instance (S_i, θ_0) , along with the corresponding t value (0 or 1), identifies the singular state $\mathit{singular}(S_i)$. Refer to Fig. 13 for an example.

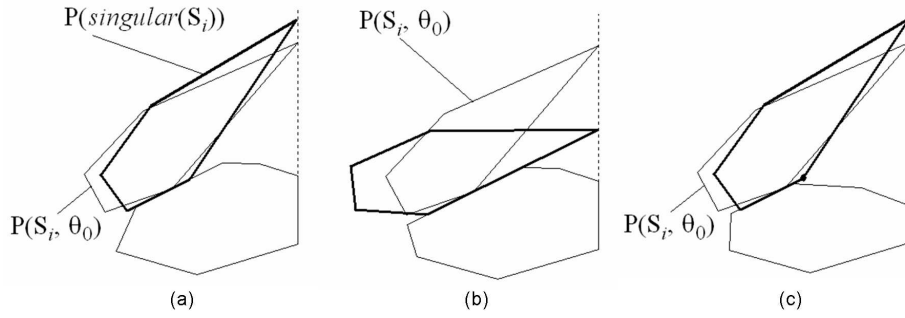
20 *K. Tang, C.C.L. Wang, and D.Z. Chen*

Fig. 14. Type II singular state: (a) valid $singular(S_i)$, (b) $\theta = \pi - \alpha$ case, and (c) t out of $[0, 1]$ case.

5.2.2. Type II singular state

A type II singular state requires that P and Q must be in edge-edge contact at $singular(S_i)$. The only possible $\theta-t$ solution in the mosaic map is point $(0, f_{QR}(0))$, whose t -value $f_{QR}(0)$ should be validated against the interval $[0, 1]$. Note that the other possible singular state $(\pi - \alpha, f_{QR}(\pi - \alpha))$ cannot be $singular(S_i)$, as $\pi - \alpha > \theta_0$. To illustrate our analysis, Fig. 14 depicts three singular states of type II; only the one in Fig. 14(a) is $singular(S_i)$.

5.2.3. Type III singular state

By definition, at a type III singular state, an adjacent vertex of P_R , i.e. P_{R-1} or P_{R+1} , also falls on the vertical line L through P_R , Q_R , and Q_{R-1} . Vertex P_{R-1} must be ruled out, since it is assumed that the states are ordered according to the orientations of the instances they represent. Let $t = f_{PR}(\theta)$ be the P_R -configuration extreme curve for P_R and P_{R+1} as defined in Eq. (9). The (θ, t) point representing the state $singular(S_i)$ then is the solution to the following two equations:

$$\begin{cases} f_{PR}(\theta) = f_{QR}(\theta) \\ t = f_{QR}(\theta) \end{cases} \quad (15)$$

After the validity checking (i.e. $\theta \in [0, \pi - \alpha]$ and $t \in [0, 1]$), the sole solution to the above two equations, if it exists, is the singular state $singular(S_i)$. Fig. 15(a) shows an example for a type III $singular(S_i)$.

5.2.4. Type IV singular state

The analysis for a type IV singular state is similar to that of type III, except that this time the (θ, t) point of $singular(S_i)$ is the intersection of the two curves:

$$\begin{cases} f_{PL}(\theta) = f_{QR}(\theta) \\ t = f_{QR}(\theta) \end{cases} \quad (16)$$

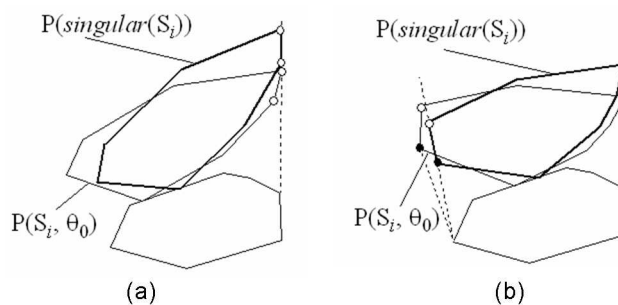


Fig. 15. Type III (a) and type IV (b) singular states.

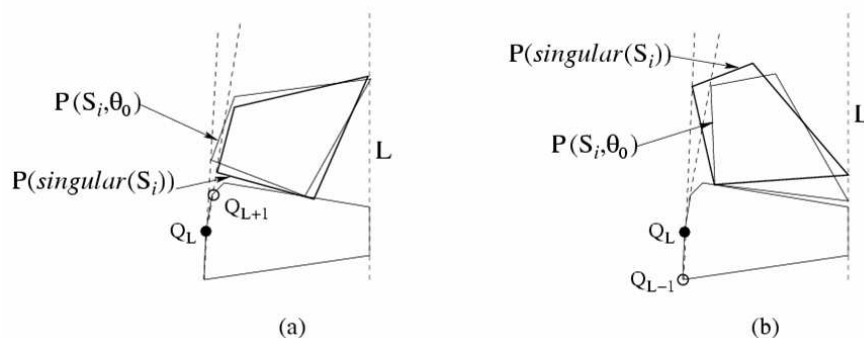


Fig. 16. Type V singular states.

where $t = f_{PL}(\theta)$ is the P_L -configuration extreme curve for P_L and P_{L+1} as defined in Eq. (7). Refer to Fig. 15(b) for an example.

5.2.5. Type V singular state

Finally, for our last type of singularity, type V, the (θ, t) point of $singular(S_i)$ can be shown to be the intersection of the two curves:

$$\begin{cases} f_{QL}(\theta) = f_{QR}(\theta) \\ t = f_{QR}(\theta) \end{cases} \quad (17)$$

Care however must be taken when selecting the Q_L -configuration extreme curve $t = f_{QL}(\theta)$. Unlike types III and IV, in the case of type V, both the adjacent vertices of the current hull vertex Q_L can contribute to $singular(S_i)$. Fig. 16 shows two examples to illustrate this point: in Fig. 16(a), the singular state $singular(S_i)$ is contributed by Q_L and Q_{L+1} , whereas in Fig. 16(b) it is Q_L and Q_{L-1} . Regardless of which one, however, the actual geometry of P and Q dictates that only one of

22 *K. Tang, C.C.L. Wang, and D.Z. Chen*

Q_{L+1} and Q_{L-1} will contribute to $singular(S_i)$. Algebraically, this implies that we should try both $t = f_{Q_L}(\theta)$ curves in Eq. (17) and take the sole solution, if it exists, that meets the validity checking of $\theta \in [0, \pi - \alpha]$ and $t \in [0, 1]$.

Having established geometrically the analysis for a singular state on all the five types, the procedure **Get_singular** (S_i, θ_0) is now easy to define. Basically, given an instance (S_i, θ_0) , we check its ensuing $singular(S_i)$ for all the five types, and the one with the largest θ - value $< \theta_0$ is the correct $singular(S_i)$. Since all extreme curves are in the forms of some explicit and simple sinusoidal functions of θ (see Eqs. (7), (9), (11), and (13)), it is easily seen that solving Eqs. (14) – (17) all takes constant time. As a result, procedure **Get_singular** (S_i, θ_0) takes constant time.

Once $singular(S_i)$ is available, state $next(S_i)$ can then be derived. For the last three types of singularity, namely, types III – V, the derivation is straightforward – only one hull vertex in the 5-tuple of state S_i needs to be replaced. For instance, referring to Fig. 15(b), if $singular(S_i)$ is of type IV, then $next(S_i)$ agrees with S_i except the P_L vertex; that is, if vertex P_i is the P_L vertex of S_i , then the P_L vertex of $next(S_i)$ should be P_{i+1} . The treatments for the first two types though require some geometric insight. These two types only alter the contact elements, and hence the mosaic map, in the state – the four hull vertices will remain the same on $next(S_i)$. First, consider type I and suppose the contact in state S_i is $P_i - Q_{j-1}Q_j$, and at $singular(S_i)$ vertex P_i becomes coincidental with Q_j . Fig. 17 shows two possible scenarios: (a) the next contact is $P_i - Q_jQ_{j+1}$, and (b) the next contact is $Q_j - P_iP_{i+1}$. We stipulate that the decision on choosing the correct one can be made by comparing the orientations of edge P_iP_{i+1} and edge Q_jQ_{j+1} . Explicitly, assuming that the positive z-axis points to the reader's eyes from this paper, at state $singular(S_i)$, if the vector $P_iP_{i+1} \times Q_jQ_{j+1}$ is in the +z-axis direction, then the next contact is $P_i - Q_jQ_{j+1}$ (e.g., Fig. 17(a)); otherwise, the next contact should be $Q_j - P_iP_{i+1}$, as shown in Fig. 17(b). To see the rational behind this criterion, consider the new placement of P_iP_{i+1} after it undergoes an infinitesimal $\Delta\theta$ motion (counterclockwise) from the singular state $singular(S_i)$. Suppose one walks from P_{i+1} first to P_i and then to Q_{j+1} . If this walk makes a left turn at P_i , then edge P_iP_{i+1} cannot contain vertex Q_j since this would cause P_iP_{i+1} to cut into Q . On the other hand, if the walk makes a right turn at P_i , then edge Q_jQ_{j+1} must be clear of P_i in order to satisfy the non-interference criterion. The analysis for type II is similar.

In summary, we have the following lemma for computing the states $singular(S_i)$ and $next(S_i)$.

Lemma 5. *Given an instance (S_i, θ_0) , both $singular(S_i)$ and $next(S_i)$ can be computed in constant time.*

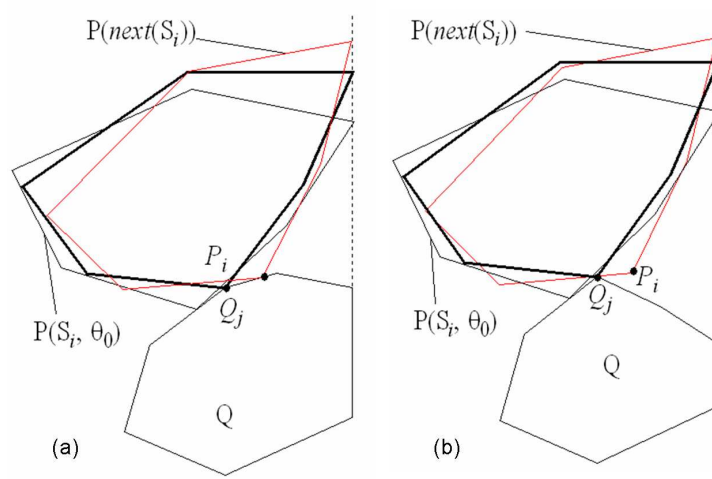


Fig. 17. Two scenarios of the next contact for type I singularity.

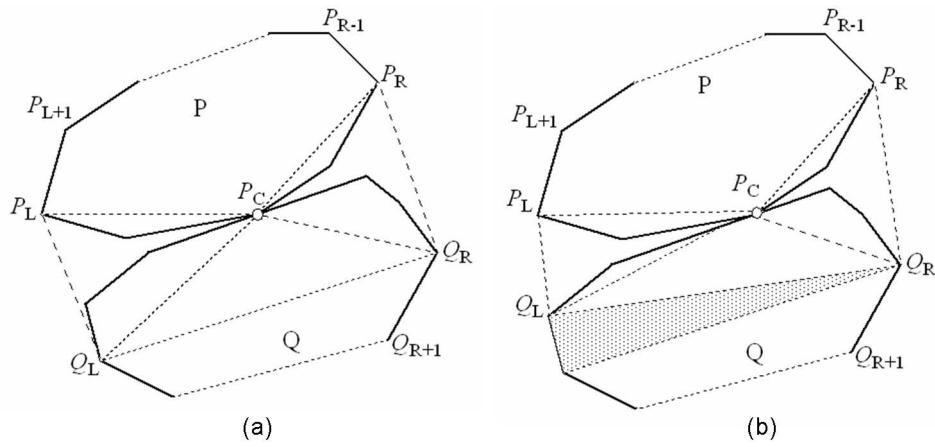


Fig. 18. Computing the area of C : (a) cell χ , and (b) cell χ' .

5.3. Area calculation

Earlier, it was stated that the area calculation function `Area_singular()` for a singular state called in algorithm `Min_area_convex_hull_top()` takes only constant time. Instead of showing our statement for this extreme case, we explain next why the area calculation for an arbitrary general state takes only constant time, if based on a simple incremental updating.

Referring to Fig. 18, the area of the packing C corresponding to a general point

24 *K. Tang, C.C.L. Wang, and D.Z. Chen*

in a cell χ is defined as:

$$A_C(\theta, t) = A(P_L P_C Q_L) + A(P_R P_C Q_R) + A(Q_L P_C Q_R) \\ + A(P_C, P_L, P_{L+1}, \dots, P_R) + A(Q_L, Q_R, Q_{R+1}, \dots, Q_{L-1}).$$

In the above equation, only the first three items, i.e., the areas of the three triangles, are dependent on (θ, t) , while the last two items are constants. Let's call the last two items in the above formula — the *residual areas* of P and Q , respectively. Obviously, given (θ, t) , the areas of the three triangles can be calculated in constant time. Hence, the calculation time for $A_C(\theta, t)$ is dominated by the computation of the residual areas. Now, consider the residual areas in a neighboring cell χ' of χ . Since only one of the four hull vertices differs in any two neighboring cells, only one residual area will change between χ' and χ and their difference is a triangle (shaded in Fig. 18(b)). Obviously, it takes constant time to compute the area of this triangle. As a result, once the residual areas of χ' are known, the area $A_C(\theta, t)$ can be calculated in constant time.

Now, consider any two successive singular states S_s and S'_s in algorithm **Min_area_convex_hull_top**. They can be viewed as belonging to two neighboring cells, albeit at extreme condition — they lie on the boundaries of the cells. Therefore, based on the above reasoning, if the residual areas of the first singular state encountered in algorithm **Min_area_convex_hull_top** are available, then all the subsequent calls **Area_singular** (S_s) take constant time each. This result is summarized in the lemma below.

Lemma 6. *Given the residual areas of the cell corresponding to a state S_i , the area of the packing for any instance of state $next(S_i)$ can be computed in constant time.*

6. Complexity Analysis

In this section, we prove the claim made at the end of Section 4 that the total number of all distinct singular states of P with respect to the vertical line L containing a given rightmost edge of Q is $O(nm)$. This result is needed for completing the proof of Theorem 2. More precisely, we prove the following theorem.

Theorem 3. *For two arbitrary convex polygons P and Q of n and m vertices, respectively, there are $O((n+m)nm)$ distinct singular states in the worst case.*

Proof. Without loss of generality, we only consider the top-supported singular states. First, assume that we fix Q with its rightmost edge lying on the vertical line L and let P rotate and translate while keeping its rightmost vertex tangent to the vertical line L and maintaining contact with the upper boundary of Q as well. Further, assume that the motion of P is counterclockwise in terms of the order of its vertices touching line L .

We begin with proving the following lower bound: For a given rightmost edge of Q lying on the vertical line L , the number of all distinct singular states of P

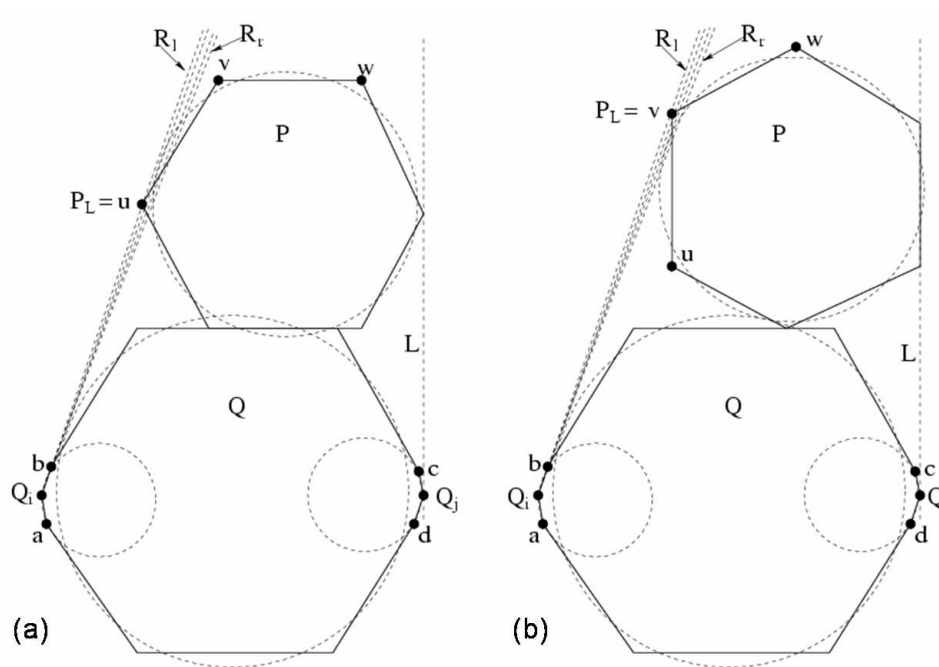


Fig. 19. Illustrating the proof of Theorem 3.

is at least $\Omega(nm)$ in the worst case. We prove this by constructing a pair of P and Q that achieves this lower bound. Refer to Fig. 19. In our construction, P is roughly a regular N -gon, and Q is roughly a regular M -gon, where $N = f(n)$ and $M = g(m)$ (the functions $f(\cdot)$ and $g(\cdot)$ will be determined later). By “roughly”, we mean that (say) P is similar to a regular N -gon but its edge lengths may differ from each other by some small values. (Note that such length differences are necessary for rotating and translating P to search for a minimum packing C — otherwise, it would be sufficient to simply consider one edge of a perfect regular N -gon in order to find a minimum packing C .) Such a roughly regular N -gon can be generated as follows: First generate a perfect regular N -gon from a circle, and then randomly move each vertex of the N -gon inside a small circular region containing that vertex (see Fig. 19).

We generate both P and Q in this manner. Next, we modify Q as follows. Choose two arbitrary “symmetric” vertices of Q , say Q_i and Q_j . Position Q such that Q_i and Q_j are the leftmost and rightmost vertices of Q respectively. Modify Q by adding two vertices a and b to Q such that a and b are adjacent and very close to Q_i ; similarly add two other vertices c and d near to vertex Q_j . Furthermore, let the three vertices a , Q_i , and b (resp., c , Q_j , and d) be on a circular arc (see Fig. 19(a)). Note that we can make the difference between the x -coordinates of vertex b and Q_i

(resp., c and Q_j) as small as we want; this implies that the two circular arcs are two nearly “flat” segments, which allow us to control the angle of the wedge bounded by the two rays R_l and R_r in Fig. 19(a) (where R_l passes through Q_i and P_L , and R_r passes through Q_i and b). Now, let $M = g(m) = m/2$ (i.e., Q was first generated as a roughly regular $(m/2)$ -gon). Randomly put $m/4$ points on the arc containing a , Q_i , and b (resp., c , Q_j , and d), and connect these points by edges along the order of the arc. These are the vertices of Q from vertex a (resp., c) clockwise to vertex b (resp., d). Q thus constructed has m vertices. Without loss of generality, we assume that the vertical line L in Fig. 19 passes through the right most edge of Q .

Suppose we rotate P counterclockwise, say from the placement of Fig. 19(a) to the placement of Fig. 19(b). Assume that all the dashed rays between the two rays R_l and R_r are collinear with a subset of the $m/4$ edges of Q from vertex a clockwise to vertex b . Then, there are $O(m)$ such dashed rays between R_l and R_r . During the rotation of P from the placement in Fig. 19(a) to the one in Fig. 19(b), the vertex u of P is first the P_L vertex for a while, and then the vertex v becomes the P_L vertex. In this rotation, $P_L = u$ in Fig. 19(a) cuts through each of the corresponding $O(m)$ dashed rays from Q , meaning that u forms a distinct type V singular state with each of the corresponding edges of Q (i.e., u is collinear with each such edge of Q while being the P_L vertex). Similarly, $P_L = v$ in Fig. 19(b) also cuts through each of the $O(m)$ rays from Q . In fact, a complete rotation of P in Fig. 19 will make every vertex z of P (when $P_L = z$) cut through each of these $O(m)$ rays of Q at most twice, once as z moves from right to left, and once as it moves from left to right in this example.

Note that given any roughly regular N -gon P , we can always make a wedge between the two rays R_l and R_r sufficiently small so that the above argument remains valid.

For the time being, let $N = f(n) = n$. Then the above argument implies that for a given rightmost vertical edge of Q from vertex c clockwise to vertex d , each of the n vertices of P can participate in $O(m)$ distinct type V singular states. This implies a total of $O(nm)$ distinct type V singular states (over the n vertices of P) for the given rightmost vertical edge of Q .

Next, observe that the above situation can be repeated $O(m)$ times, one for each of the $m/4$ very short edges close to the current rightmost vertical edge of Q , since every such edge of Q can in turn be used as its rightmost edge lying on the vertical line L . Thus, the same process can be repeated for each of these $m/4$ edges Q close to its current leftmost edge. This means that there can be altogether $O(nm^2)$ distinct type V singular states.

On the other hand, we can also establish the following upper bound: For a given rightmost edge of Q lying on the vertical line L , the number of all distinct singular states of P is at most $O(nm)$. For example, it is easy to see that every vertex of P , while it is being the hull vertex P_L , can be collinear with at most $O(m)$ different edges of Q that are incident to the current hull vertex Q_L , implying

that there can be at most $O(nm)$ distinct type V singular states. In general, note that there are $O(n)$ boundary entities of P and $O(m)$ boundary entities of Q , and each pair between them may define a singular state of a certain type (out of the five singularity types). There can be at most $O(nm)$ such boundary entity pairs between P and Q in the worst case. Each of them may “realize” a distinct singular state, one pair after another, as P is in motion. Therefore, there can be no more than $O(nm)$ distinct singular states for a given rightmost edge of Q .

Letting each of the m edges of Q in turn be the right most vertical edge, the above argument leads to that there are no more than $O(nm^2)$ distinct singular states for the given roles of P and Q (i.e., fixing the rightmost edge of Q and letting P rotate and translate).

We can switch the roles of P and Q (i.e., fix the rightmost edge of P and let Q rotate and translate). Before doing that, we let $N = f(n) = n/2$ (i.e., P was first a roughly regular $(n/2)$ -gon) and apply the described modification on Q to P , that is, we add $n/4$ vertices respectively close to the leftmost and rightmost vertices of P . P thus constructed has n vertices. Note that the structure of Q is similar to that of P (i.e., Q is still like a roughly regular $(m/2)$ -gon but has m vertices, with $m/4$ vertices close to each of the two chosen “symmetric” vertices). Then the above analysis can be applied to this setting of P and Q , proving that the lower and upper bounds of all distinct singular states for this case are both $O(mn^2)$.

In summary, there are $O(nm^2) + O(mn^2) = O((n+m)nm)$ distinct singular states between two arbitrary convex polygons P and Q in the worst case. \square

7. Implementation and Examples

The algorithm presented above has been implemented on a PC platform with a modest configuration and tested on a number of examples. When “designing” the test examples, we consciously looked for cases that can verify all the five singularity types. Explicitly, we wanted to answer the question “can a minimum packing be of any of the five singularity types?”. Our answer is “yes”, and Fig. 20 through Fig. 24 give one example for each type.

The last example (given in Fig. 25) demonstrates the robustness of the algorithm (and our implementation). In this example, a link edge shrinks to a single point; geometrically, this indicates a singular state of multiple types (e.g., types I, II, and III in Fig. 25). The correct minimum packing is successfully captured in our test.

8. Conclusion

In this paper, we have presented an efficient deterministic algorithm for computing a minimum area convex packing of two arbitrary convex polygons P and Q , thus settling this open problem. Our algorithm is based on a novel characterization of the geometric and algebraic structures of the problem, which enables us to successfully discretize the search space to only a finite number of special points called critical

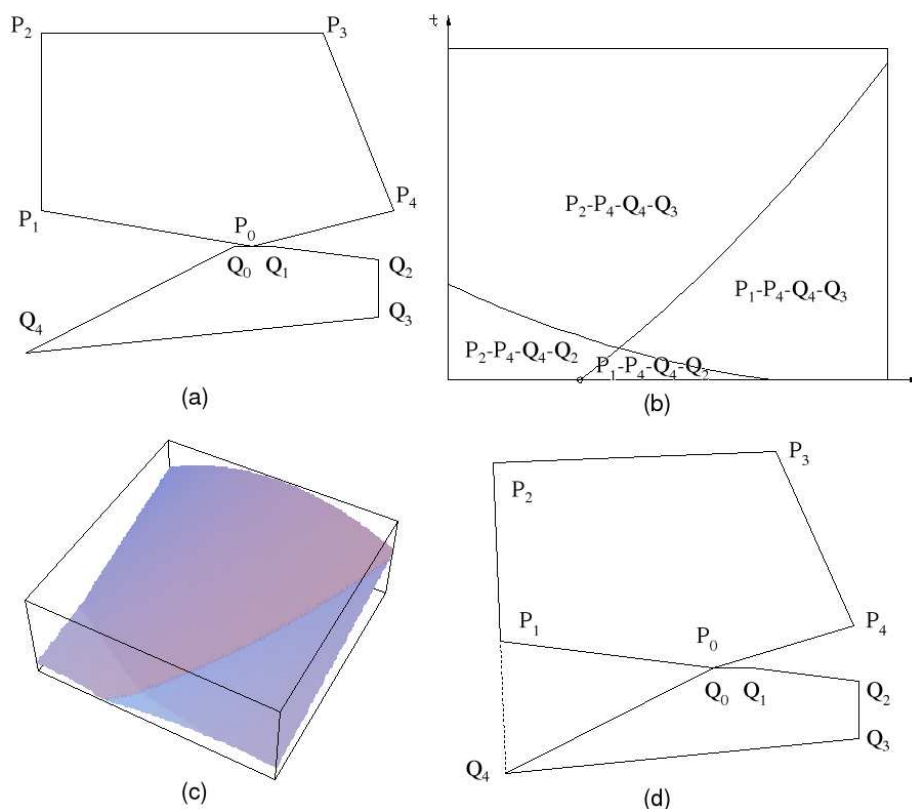
28 *K. Tang, C.C.L. Wang, and D.Z. Chen*

Fig. 20. A minimum packing of type I: (a) the given two convex polygons, (b) the mosaic map containing the result, (c) 3D display of the area function of the mosaic map of the result, and (d) the minimum packing convex hull.

points. A careful complexity analysis shows that our algorithm runs in $O((n + m)nm)$ time, where n and m are the numbers of vertices of P and Q , respectively.

Along the direction of this work, some interesting open problems still remain. The first one is either to prove that our $O((n + m)nm)$ time algorithm is optimal, or to design a faster algorithm for solving this problem. The second problem is to extend our method to solving the problem of computing a minimum area convex packing of more than two arbitrary convex polygons. The third problem is to consider packing polygons that are not necessary convex. Packing geometric objects in the 3-dimensional space is also a very interesting and useful topic.

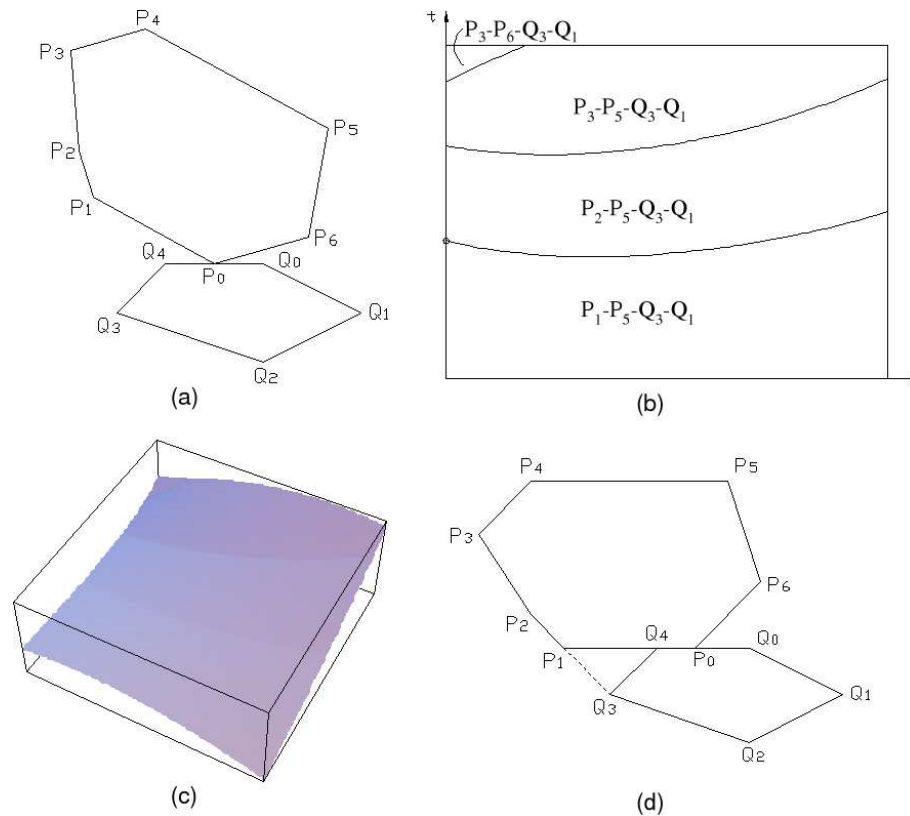


Fig. 21. A minimum packing of type II: (a) the given two convex polygons, (b) the mosaic map containing the result, (c) 3D display of the area function of the mosaic map of the result, and (d) the minimum packing convex hull.

Acknowledgements

The first author, Kai Tang, was partially supported by Hong Kong RGC Grant 05/06.EG.620105. The research of the third author, Danny Z. Chen, was supported in part by the National Science Foundation under Grants CCR-9988468 and CCF-0515203; this work was partially done while D.Z. Chen was visiting the Department of Computer Science, Hong Kong University of Science and Technology.

30 *K. Tang, C.C.L. Wang, and D.Z. Chen*

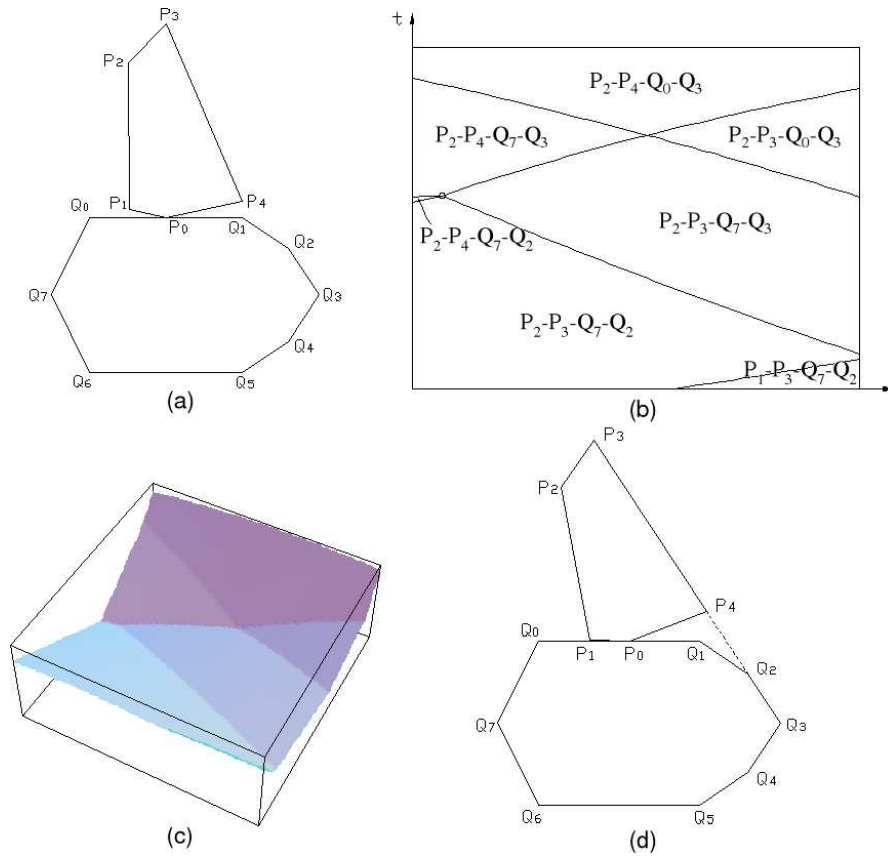


Fig. 22. A minimum packing of type III: (a) the given two convex polygons, (b) the mosaic map containing the result, (c) 3D display of the area function of the mosaic map of the result, and (d) the minimum packing convex hull.

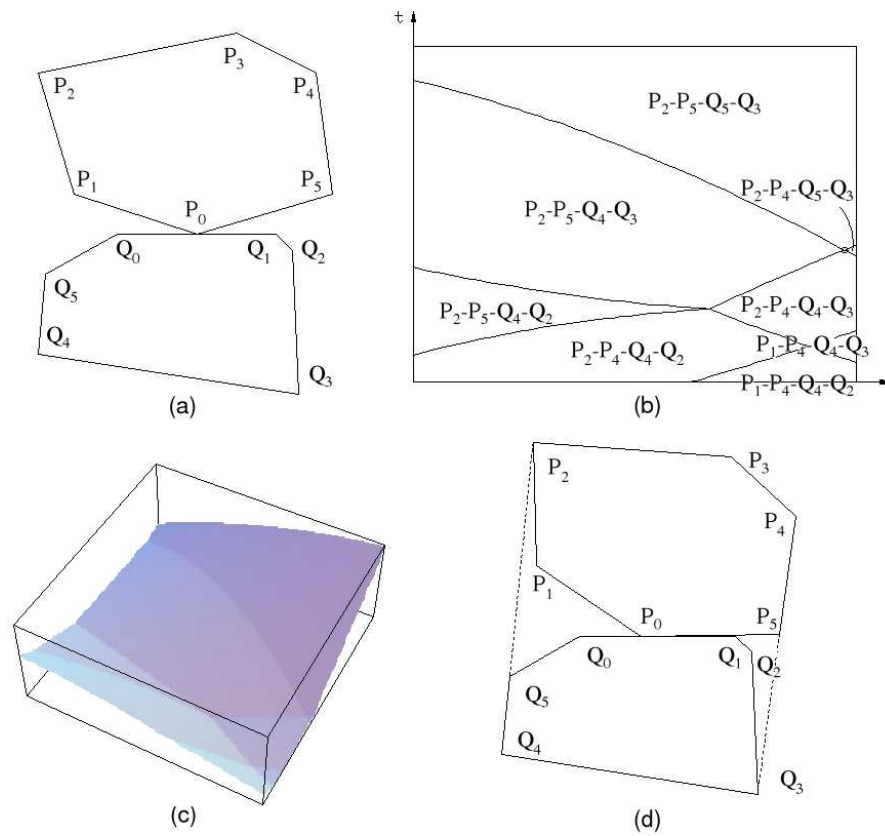


Fig. 23. A minimum packing of type IV: (a) the given two convex polygons, (b) the mosaic map containing the result, (c) 3D display of the area function of the mosaic map of the result, and (d) the minimum packing convex hull.

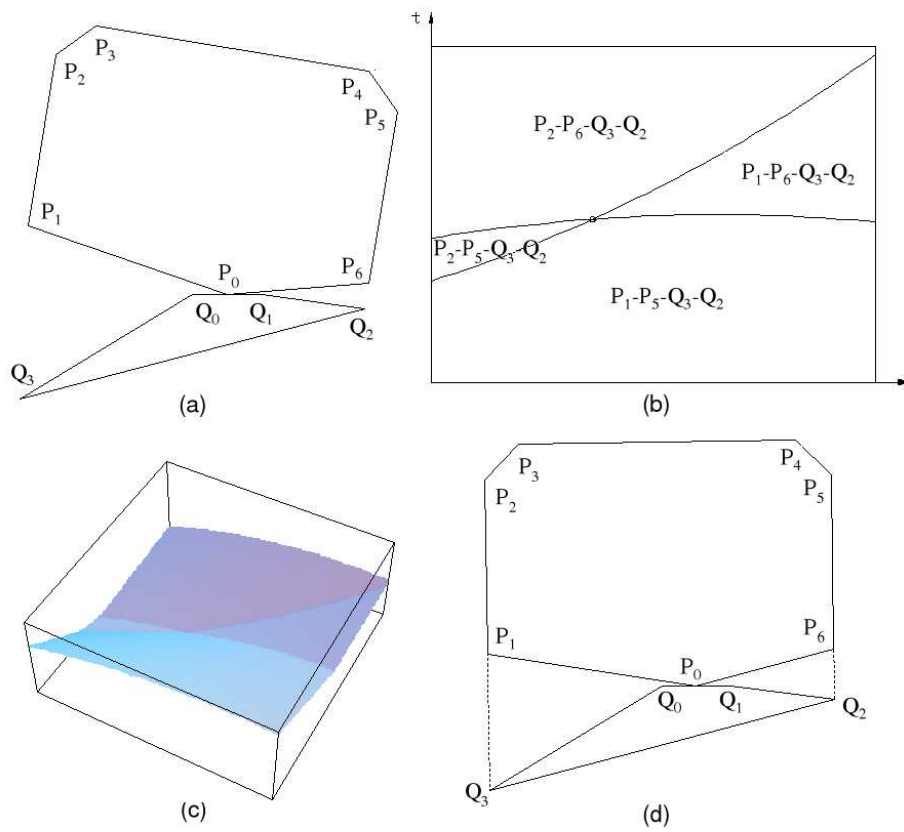
32 *K. Tang, C.C.L. Wang, and D.Z. Chen*

Fig. 24. A minimum packing of type V: (a) the given two convex polygons, (b) the mosaic map containing the result, (c) 3D display of the area function of the mosaic map of the result, and (d) the minimum packing convex hull.

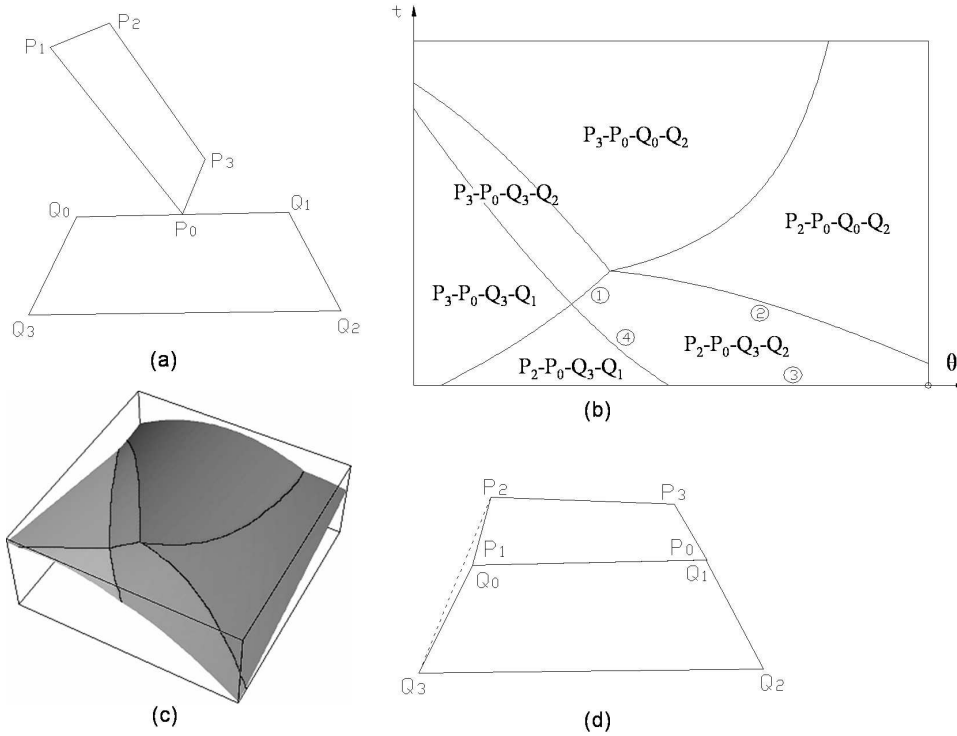


Fig. 25. Degenerate singular state: (a) the given two convex polygons, (b) the mosaic map containing the final result, (c) 3D display of the area function of the mosaic map of the final result, and (d) the minimum packing convex hull.

34 *K. Tang, C.C.L. Wang, and D.Z. Chen*

References

1. A.R.S. Amaral, and M. Wright, Efficient algorithm for the constrained two-dimensional cutting stock problem, *International Transactions in Operational Research*. **8** (2001) 3–13.
2. M. de Berg, M. Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, Springer, Berlin, 1997.
3. M. Dell'Amico, S. Martello, and D. Vigo, A lower bound for the non-oriented two-dimensional bin packing problem, *Discrete Applied Mathematics*. **118** (2002) 13–24.
4. X. Gu, G. Chen, J. Gu, L. Huang, and Y. Jung, Performance analysis and improvement for some linear on-line bin-packing algorithms, *Journal of Combinatorial Optimization*. **6** (2002) 455–471.
5. A.I. Hinxman, The trim-loss and assortment problems: A survey, *European Journal of Operational Research*. **5** (1980) 8–18.
6. T.J. Hodgson, A combined approach to the pallet loading problem, *IIE Transactions*. **14** (1982) 175–182.
7. H.C. Lee, and T.C. Woo, Determining in linear time the minimum area convex hull of two polygons, *IIE Transactions*. **20** (1988) 338–345.
8. A.N. Letchford, and A. Amaral, Analysis of upper bounds for the pallet loading problem, *European Journal of Operational Research*. **132** (2001) 582–593.
9. T.W. Leung, C.H. Yung, and M.D. Troutt, Applications of genetic search and simulated annealing to the two-dimensional non-guillotine cutting stock problem, *Computers & Industrial Engineering*. **40** (2001) 201–214.
10. H.L. Li, C.T. Chang, and J.F. Tsai, Approximately global optimization for assortment problems using piecewise linearization techniques, *European Journal of Operational Research*. **140** (2002) 584–589.
11. H.L. Li, and J.F. Tsai, An approximately global optimization method for assortment problems, *European Journal of Operational Research*. **105** (1998) 604–612.
12. H.L. Li, and J.F. Tsai, A fast algorithm for assortment optimization problems, *Computers & Operations Research*. **28** (2001) 1245–1252.
13. A. Lodi, S. Martello, and D. Vigo, Recent advances on two-dimensional bin packing problems, *Discrete Applied Mathematics*. **123** (2002) 379–396.
14. R.R. Martin, D.K. Wright, and I.N. Jordanov, Putting objects in boxes, *Computer-Aided Design*. **20** (1988) 506–514.
15. Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, 1996.
16. J. O'Rourke, Finding minimal enclosing boxes, *International Journal of Computer & Information Science*. **14** (1985) 183–199.
17. J. Terno, G. Scheithauer, U. Sommerweiss, and J. Riehme, An efficient approach for the multi-pallet loading problem, *European Journal of Operational Research*. **123** (2000) 372–381.
18. H. William, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1992.
19. G. Young-Gun, and Maing-Kyu Kang, A fast algorithm for two-dimensional pallet loading problems of large size, *European Journal of Operational Research*. **134** (2001) 193–202.