

Iterative thresholding methods for longest minimal length partitions

Shilong Hu, Hao Liu, and Dong Wang

Abstract. In this paper, we introduce two iterative methods for the longest minimal length partition problem, which asks whether the disk (ball) is the set maximizing the total perimeter of the shortest partition that divides the total region into sub-regions with given volume proportions, under a volume constraint. The objective functional is approximated by a short-time heat flow using indicator functions of regions and Gaussian convolution. The problem is then represented as a constrained max-min optimization problem. Auction dynamics is used to find the shortest partition in a fixed region, and threshold dynamics is used to update the region. Numerical experiments in two-dimensional and three-dimensional cases are shown with different numbers of partitions, unequal volume proportions, and different initial shapes. The results of both methods are consistent with the conjecture that the disk in two dimensions and the ball in three dimensions are the solution of the longest minimal length partition problem.

1. Introduction

The fencing problem has a long history and is rooted in the division of land using fences. One of the earliest studies in this field can be traced back to [1], which investigated the scenario of a cylinder floating in water. This problem was further explored in subsequent studies such as [9, 15, 29], in which one considers dividing a convex set into two equal-area sets and studies the optimal arc length of the fence.

In the 1950s, Pólya raised a question in [28]: among all convex sets in $\mathbb{R}^2$ with a fixed area, which one has the longest shortest arc that bisects the area? This question remained unresolved throughout the last century. According to [8], Santaló made a conjecture that the disk is the answer. In 2012, the conjecture was proven in [13]. A similar conjecture in high dimensions was proposed in [41] and [26] after Pólya. It considers the case of dividing a region in $\mathbb{R}^d$ space into two sub-regions of arbitrary given volume proportions, and asks whether the least length of arc to divide a ball is greater than the least length of arc to divide any other convex body of the same volume. The new conjecture generalizes the bisection problem to arbitrary volume constraints. In the literature, some analytical

Mathematics Subject Classification 2020: 49M20 (primary); 49Q05, 49M05, 41A30 (secondary).

Keywords: threshold dynamics, auction dynamics, fencing problem, longest minimal length partitions, volume constrained partitions.

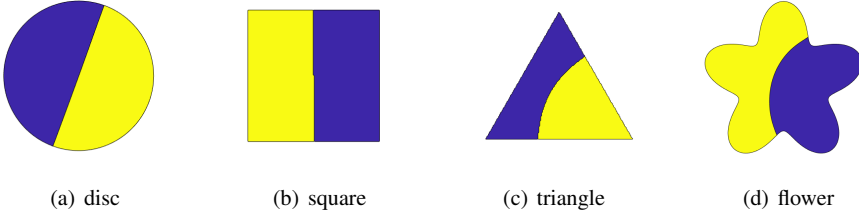


Figure 1. The shortest partitions of different Ω in Pólya's question. The partitions are computed numerically by auction dynamics in Section 3.

results have been presented in [3] and [36], which offer partial proofs for this conjecture. As of now, the conjecture remains unanswered for the case of three-dimensional space and for large fraction areas in the two-dimensional case (except for bisection).

In [4], the authors extend the conjecture even further. They investigate whether the disk in two dimensions and the ball in three dimensions still provide the longest minimal length partitions when the region is divided into more than two parts. The mathematical description of it is given below.

Given $\Omega \subset \mathbb{R}^d$ an open, connected region with Lipschitz boundary and the number of partition $n > 1$, $(\omega_1, \dots, \omega_n)$ is said to be a partition of Ω if $\bigcup_{i=1}^n \omega_i = \Omega$ and $\omega_i \cap \omega_j = \emptyset$ for any $1 \leq i < j \leq n$. Given a vector $\mathbf{c} = (c_1, \dots, c_n) \in \mathbb{R}^n$ with $c_i > 0$ and $\sum_{i=1}^n c_i = 1$, the shortest partition of Ω with volume constraint $\mathbf{c}$ is defined by

$$\text{SP}(\Omega, \mathbf{c}) = \arg \min_{(\omega_i)_{i=1}^n} \left\{ \sum_{i=1}^n \text{Per}_{\Omega}(\omega_i) : (\omega_i)_{i=1}^n \text{ is a partition of } \Omega, |\omega_i| = c_i |\Omega| \right\},$$

where $|\cdot|$ is the Lebesgue measure, and $\text{Per}_{\Omega}(\omega_i)$ is the relative perimeter of ω_i on Ω . Some examples of the shortest partition for different regions are shown in Figure 1. The relative isoperimetric profile of the partition is defined as

$$\text{PI}(\Omega, \mathbf{c}) = \min \left\{ \sum_{i=1}^n \text{Per}_{\Omega}(\omega_i) : (\omega_i)_{i=1}^n \text{ is a partition of } \Omega, |\omega_i| = c_i |\Omega| \right\}.$$

Then the longest minimal length partitions problem is formulated by

$$\max_{|\Omega|=V} \text{PI}(\Omega, \mathbf{c}). \quad (1.1)$$

When $n = 2$, problem (1.1) degenerates into the original fencing problem. If we further set the dimension $d = 2$ and $\mathbf{c} = (0.5, 0.5)$, the problem is exactly the same one that Pólya raised.

The relative perimeter here is defined by using the total variation. Suppose χ_{ω} is the indicator function of ω . Then

$$\text{Per}_{\Omega}(\omega) = \text{TV}(\chi_{\omega}, \Omega),$$

where

$$\mathrm{TV}(u, \Omega) = \sup \left\{ \int_{\Omega} u \operatorname{div} g : g \in C_c(\Omega, \mathbb{R}^d), \|g\|_{\infty} \leq 1 \right\}$$

is the total variation of u on Ω , and $C_c(\Omega, \mathbb{R}^d)$ means the space of C^{∞} vector-valued functions on Ω with compact support. Intuitively speaking, $\operatorname{Per}_{\Omega}(\omega) = |\partial\omega \setminus \partial\Omega|$, and $\sum_{i=1}^n \operatorname{Per}_{\Omega}(\omega_i)$ in $\operatorname{PI}(\Omega, \mathbf{c})$ is $\frac{1}{2}(\sum_{i=1}^n |\partial\omega_i| - |\partial\Omega|)$. More details about the relative perimeter can be found in [6].

In [4], the authors propose a phase-field method based on shape derivatives to numerically simulate the longest minimal length partitions. The method involves using a Modica–Mortola approximation to represent the perimeter, and the total region is parametrized using a radial function discretized with Fourier coefficients. The authors update the total region by iteratively modifying the Fourier coefficients based on shape derivatives. To maintain the volume constraint, they project the Fourier coefficients using a homothety at each iteration step. The authors claim that the results of their method provide numerical evidence for the conjecture that for any fixed $\mathbf{c}$, the disk is always the maximizer in two dimensions, and the ball is always the maximizer in three dimensions.

In [25] the authors present a short-time heat flow approximation for the boundary integral energy of functions with bounded variation. This approximation utilizes indicator functions and Gaussian convolution and does not require taking derivatives of the functions. In [23, 24], Merriman, Bence, and Osher (MBO) developed a threshold dynamics method for interface motion. The method has been proved to converge to mean curvature flow in [2, 14]. Esedoglu and Otto further extended this method to the multiphase case in [11]. The method can be applied to general multiphase problems with arbitrary surface tensions. The MBO method and its ideas have gained recognition for simplicity and stability, and have been widely applied in various areas such as motion by mean curvature for filaments [31], volume-preserving interface motion [17, 32], image processing [12, 33, 34, 39, 40], graph data processing [5, 21, 22], foam bubbles [38], grain boundary motion [10], and wetting phenomena on solid surfaces [42]. Recently, connections among the MBO method, operator-splitting methods, and deep neural networks are pointed out in [19, 20, 35].

Numerous algorithms and rigorous error analyses have been developed to refine and expand upon the original MBO method [16, 24, 30]. For instance, in [17], the authors extend the MBO method to multiphase volume-constrained curvature motion using an auction dynamics scheme. This extension allows for solving problems related to minimal surface area with multiphase volume constraints.

In this paper, two new iterative methods are introduced for longest minimal length partitions. The objective functional is approximated by a short-time heat flow. Then, the method of threshold dynamics is used to maximize $\operatorname{PI}(\Omega, \mathbf{c})$ under volume constraints, and auction dynamics is used to find $\operatorname{SP}(\Omega, \mathbf{c})$ for a fixed Ω . Effectiveness of the proposed methods are demonstrated by comprehensive numerical experiments in two-dimensional and three-dimensional cases with different numbers of partitions, unequal volume proportions, and different initial shapes. The results of both methods are consistent with the

conjecture that the disk in two dimensions and the ball in three dimensions are the solution of the longest minimal length partition problem.

The rest of this paper is organized as follows: Section 2 introduces approximations for the objective functional. These approximations serve as the foundation for the subsequent methods. Section 3 presents the derivation of the two-step methods based on the introduced approximations. In Section 4, we present numerical experiments in two and three dimensions to demonstrate the performance of the proposed methods. We propose in Section 5 an objective functional-monotone method. We then draw conclusions and discussions in Section 6.

2. Approximations of the objective functional

Let $\Omega \subset \mathbb{R}^d$ be an open and connected set to be partitioned. We define the objective functional (the length of the partition of Ω) by

$$E(\Omega, (\omega_i)_{i=1}^n) = \sum_{i=1}^n \text{Per}_{\Omega}(\omega_i). \quad (2.1)$$

Given a volume constraint $V > 0$ and partition proportions $\mathbf{c} = (c_1, \dots, c_n)$, the relative isoperimetric profile is written as

$$\text{PI}(\Omega, \mathbf{c}) = \min_{|\omega_i|=c_i V} E(\Omega, (\omega_i)_{i=1}^n).$$

The longest minimal length problem can be formulated as the following max-min optimization problem:

$$\max_{|\Omega|=V} \text{PI}(\Omega, \mathbf{c}) = \max_{|\Omega|=V} \min_{|\omega_i|=c_i V} E(\Omega, (\omega_i)_{i=1}^n). \quad (2.2)$$

To approximate the objective functional, we use indicator functions to implicitly represent the regions Ω and $(\omega_i)_{i=1}^n$. Define u_{Ω} and $(u_i)_{i=1}^n$ by

$$u_{\Omega}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega, \\ 0 & \text{if } \mathbf{x} \notin \Omega, \end{cases} \quad u_i(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \omega_i, \\ 0 & \text{if } \mathbf{x} \notin \omega_i, \end{cases} \quad \text{for } i = 1, \dots, n.$$

The volume constraints in (2.2) can be expressed as $\int_{\mathbb{R}^d} u_{\Omega} \mathbf{d}\mathbf{x} = V$ and $\int_{\mathbb{R}^d} u_i \mathbf{d}\mathbf{x} = c_i V$. Since $(\omega_i)_{i=1}^n$ is a partition of Ω , we also have $\sum_{i=1}^n u_i = u_{\Omega}$.

To approximate the objective functional, instead of using a Modica–Mortola approximation as in [4], we use indicator functions and Gaussian convolution. Note that $\text{Per}_{\Omega}(\omega_i) = \sum_{j=1}^n \text{Per}_{\omega_i \cup \omega_j}(\omega_i)$, and $\text{Per}_{\omega_i \cup \omega_i}(\omega_i) = \text{Per}_{\omega_i}(\omega_i) = 0$. For any $i \neq j$, we have $\text{Per}_{\omega_i \cup \omega_j}(\omega_j) = \text{Per}_{\omega_i \cup \omega_j}(\omega_i)$, which can be approximated by

$$\text{Per}_{\omega_i \cup \omega_j}(\omega_j) \approx \sqrt{\frac{\pi}{\tau}} \int_{\mathbb{R}^d} u_i(G_{\tau} * u_j) \mathbf{d}\mathbf{x}, \quad (2.3)$$

where

$$G_\tau(\mathbf{x}) = \frac{1}{(4\pi\tau)^{d/2}} \exp\left(-\frac{|\mathbf{x}|^2}{4\tau}\right)$$

is the Gaussian kernel and $*$ denotes convolution in $\mathbb{R}^d$. The convergence of this approximation for closed curves has been shown in [25] as τ goes to 0^+ and the multiphase formula was proposed in [11]. By utilizing (2.3), we approximate $E(\Omega, (\omega_i)_{i=1}^n)$ in (2.1) by

$$\begin{aligned} E(\Omega, (\omega_i)_{i=1}^n) &= \sum_{i=1}^n \sum_{j \neq i} \text{Per}_{\omega_j \cup \omega_i}(\omega_i) \\ &\approx \sqrt{\frac{\pi}{\tau}} \sum_{i=1}^n \sum_{j \neq i} \int_{\mathbb{R}^d} u_i (G_\tau * u_j) d\mathbf{x} \end{aligned} \quad (2.4)$$

$$\begin{aligned} &= \sqrt{\frac{\pi}{\tau}} \sum_{i=1}^n \int_{\mathbb{R}^d} u_i \left(G_\tau * \left(\sum_{j \neq i} u_j \right) \right) d\mathbf{x} \\ &= \sqrt{\frac{\pi}{\tau}} \sum_{i=1}^n \int_{\mathbb{R}^d} u_i (G_\tau * (u_\Omega - u_i)) d\mathbf{x} \\ &= \sqrt{\frac{\pi}{\tau}} \sum_{i=1}^n \int_{\mathbb{R}^d} (G_{\tau/2} * u_i) (G_{\tau/2} * (u_\Omega - u_i)) d\mathbf{x}. \end{aligned} \quad (2.5)$$

Since $\sqrt{\frac{\pi}{\tau}} (G_{\tau/2} * u_i) (G_{\tau/2} * (u_\Omega - u_i))$ concentrates on Ω as τ goes to 0^+ , that is,

$$\lim_{\tau \rightarrow 0^+} \sqrt{\frac{\pi}{\tau}} \int_{\mathbb{R}^d \setminus \Omega} (G_{\tau/2} * u_i) (G_{\tau/2} * (u_\Omega - u_i)) d\mathbf{x} = 0,$$

when τ is sufficiently small, the objective functional will not change too much if we replace the integrating domain in (2.5) from $\mathbb{R}^d$ to Ω . To achieve this, recall that u_Ω is the indicator function of Ω , and we multiply the integrand in (2.5) by u_Ω . Thus, the objective functional in (2.5) is further approximated by

$$\begin{aligned} (2.5) &\approx \sqrt{\frac{\pi}{\tau}} \sum_{i=1}^n \int_{\mathbb{R}^d} u_\Omega (G_{\tau/2} * u_i) (G_{\tau/2} * (u_\Omega - u_i)) d\mathbf{x} \\ &= \sqrt{\frac{\pi}{\tau}} \int_{\mathbb{R}^d} u_\Omega \left(S_{\tau/2} - \sum_{i=1}^n (G_{\tau/2} * u_i) \right) d\mathbf{x} \\ &\quad - \sqrt{\frac{\pi}{\tau}} \int_{\mathbb{R}^d} u_\Omega S_{\tau/2} (G_{\tau/2} * (1 - u_\Omega)) d\mathbf{x}, \end{aligned}$$

where $S_{\tau/2}$ denotes $S_{\tau/2}(\mathbf{x}) := G_{\tau/2} * (\sum_{i=1}^n u_i)(\mathbf{x})$. By [37], in the sense of τ going to 0^+ , the last quadratic term can be approximated by

$$\sqrt{\frac{\pi}{\tau}} \int_{\mathbb{R}^d} u_\Omega S_{\tau/2} (G_{\tau/2} * (1 - u_\Omega)) d\mathbf{x} \approx \sqrt{\frac{\pi}{\tau}} \int_{\mathbb{R}^d} u_\Omega S_{\tau/2}^{\frac{1}{2}} (G_{\tau/2} * (S_{\tau/2}^{\frac{1}{2}} (1 - u_\Omega))) d\mathbf{x}.$$

Denote the approximate objective functional by

$$\begin{aligned}\tilde{E}_\tau(u_\Omega, (u_i)_{i=1}^n) &:= \sqrt{\frac{\pi}{\tau}} \int_{\mathbb{R}^d} u_\Omega \left(S_{\tau/2} - \sum_{i=1}^n (G_{\tau/2} * u_i)^2 \right) dx \\ &\quad - \sqrt{\frac{\pi}{\tau}} \int_{\mathbb{R}^d} u_\Omega S_{\tau/2}^{\frac{1}{2}} (G_{\tau/2} * (S_{\tau/2}^{\frac{1}{2}} (1 - u_\Omega))) dx.\end{aligned}\quad (2.6)$$

The original problem given by (2.2) is approximated by

$$\max_{\substack{u_\Omega \in \mathcal{B} \\ \int_{\mathbb{R}^d} u_\Omega dx = V}} \min_{\substack{u_i \in \mathcal{B} \\ \int_{\mathbb{R}^d} u_i dx = c_i V \\ \sum u_i = u_\Omega}} \tilde{E}_\tau(u_\Omega, (u_i)_{i=1}^n), \quad (2.7)$$

where $\mathcal{B} := \{u \in \text{BV}(\Omega, \mathbb{R}) : u = \{0, 1\}\}$, and $\text{BV}(\Omega, \mathbb{R})$ denotes the bounded-variation functional space.

Note that (2.4) can also be used for approximation, leading to another approximation of the objective functional

$$\hat{E}_\tau(u_\Omega, (u_i)_{i=1}^n) := \sqrt{\frac{\pi}{\tau}} \sum_{i=1}^n \sum_{j \neq i} \int_{\mathbb{R}^d} u_i (G_\tau * u_j) dx. \quad (2.8)$$

The corresponding modified problem is

$$\max_{\substack{u_\Omega \in \mathcal{B} \\ \int_{\mathbb{R}^d} u_\Omega dx = V}} \min_{\substack{u_i \in \mathcal{B} \\ \int_{\mathbb{R}^d} u_i dx = c_i V \\ \sum u_i = u_\Omega}} \hat{E}_\tau(u_\Omega, (u_i)_{i=1}^n). \quad (2.9)$$

Here, $\tilde{E}_\tau$ and $\hat{E}_\tau$ are equivalent to each other in the sense of τ converging to 0^+ , and hence are (2.7) and (2.9).

We will propose two algorithms to solve the longest minimal length partitions. Both algorithms contain two steps: the first step updates u_Ω and the second step finds the shortest partition of the updated u_Ω . Approximations (2.6) and (2.8) are designed so that both steps can be computed effectively. Approximation (2.6) is a functional of u_Ω , making it well suited to be used in the first step. In the second step, we use approximation (2.8) so that the auction dynamics algorithm can be directly used to update u_i 's. Details on how the two approximations are used are presented in the next section.

3. Derivation of the methods

Based on the two objective functional approximations in (2.6) and (2.8), in this section, two iterative methods are derived for longest minimal length partitions. Denote the indicator functions of the total region and the shortest partition in k -th iteration by u_Ω^k and $(u_i^k)_{i=1}^n$ respectively. Each method contains two steps:

Step 1: Fix $(u_i^k)_{i=1}^n$ and find u_Ω^{k+1} by maximizing $\widetilde{E}_\tau$.

Step 2: Fix u_Ω^{k+1} and find $(u_i^{k+1})_{i=1}^n$ by minimizing $\widehat{E}_\tau$.

In Step 2, due to the good form of $\widehat{E}_\tau$, auction dynamics [17] can be directly employed in both methods. The $(u_i^k)_{i=1}^n$ given by the two methods preserve $\sum_{i=1}^n u_i^k = u_\Omega^k$ to ensure that the collection of subsets is a partition of the total region. In Step 1, the update is based on the objective functional $\widetilde{E}_\tau$ and threshold dynamics is used. The constraint $\sum_{i=1}^n u_i = u_\Omega$ is relaxed in this part to update the total region. Otherwise, if $u_\Omega^{k+1} = \sum_{i=1}^n u_i^k = u_\Omega^k$, the total region would remain unchanged during iterations. To maintain the stability of the method, in the first method, the update of u_Ω^{k+1} is fully led by the behavior of $\widetilde{E}_\tau$, while $(u_i^{k+1})_{i=1}^n$ are updated by running auction dynamics several times to avoid local minimizers of $\widehat{E}_\tau$. But this treatment makes the method slow. In the second method, only a one-time auction dynamics is used to updated $(u_i^{k+1})_{i=1}^n$. It makes the method more efficient but less stable. In order to balance the efficiency and stability, we introduce a proximal term to control the evolution of u_Ω , which prevents u_Ω from evolving too fast. In general, the first method is more stable and the second method is more efficient.

3.1. The first method

3.1.1. Update of u_Ω^{k+1} . In this step, we relax the constraint $\sum_{i=1}^n u_i = u_\Omega$. We fix $(u_i^k)_{i=1}^n$ and find u_Ω^{k+1} by maximizing $\widetilde{E}_\tau$ in (2.6). Fixing $(u_i^k)_{i=1}^n$, we update u_Ω^{k+1} by

$$u_\Omega^{k+1} = \arg \max_{\substack{u_\Omega \in \mathcal{B} \\ \int_{\mathbb{R}^d} u_\Omega dx = V}} \widetilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n). \quad (3.1)$$

Directly solving (3.1) is difficult due to the constraint $u_\Omega \in \mathcal{B}$, which is a non-convex set. To solve (3.1), we relax the constraint $u_\Omega \in \{0, 1\}$ to $u_\Omega \in [0, 1]$. Specifically, we define $\mathcal{K} := \{u \in \text{BV}(\Omega, \mathbb{R}) : u \in [0, 1]\}$ and consider the following problem instead:

$$u_\Omega^{k+1} = \arg \max_{\substack{u_\Omega \in \mathcal{K} \\ \int_{\mathbb{R}^d} u_\Omega dx = V}} \widetilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n). \quad (3.2)$$

The following lemma shows that (3.1) and (3.2) have the same maximizer:

Lemma 3.1. *Problems (3.1) and (3.2) have the same maximizer, that is,*

$$\arg \max_{\substack{u_\Omega \in \mathcal{B} \\ \int_{\mathbb{R}^d} u_\Omega dx = V}} \widetilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n) = \arg \max_{\substack{u_\Omega \in \mathcal{K} \\ \int_{\mathbb{R}^d} u_\Omega dx = V}} \widetilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n).$$

Proof. Note that

$$\max_{\substack{u_\Omega \in \mathcal{B} \\ \int_{\mathbb{R}^d} u_\Omega dx = V}} \widetilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n) \leq \max_{\substack{u_\Omega \in \mathcal{K} \\ \int_{\mathbb{R}^d} u_\Omega dx = V}} \widetilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n),$$

since $\mathcal{B} \subset \mathcal{K}$. To prove the equivalence, it suffices to show

$$\arg \max_{\substack{u_\Omega \in \mathcal{K} \\ \int_{\mathbb{R}^d} u_\Omega \mathrm{d}\mathbf{x} = V}} \tilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n) \in \left\{ u_\Omega \in \mathcal{B} : \int_{\mathbb{R}^d} u_\Omega \mathrm{d}\mathbf{x} = V \right\}.$$

Assume this is not true and the maximizer is u^* . Then there exists a set $A \subset \Omega$ with $|A| > 0$ and $0 < c < \frac{1}{2}$ such that

$$u^*(\mathbf{x}) \in (c, 1 - c), \quad \text{for any } \mathbf{x} \in \Omega.$$

Divide A into two subsets A_1 and A_2 such that $A_1 \cup A_2 = A$, $A_1 \cap A_2 = \emptyset$ and $|A_1| = |A_2| = \frac{1}{2}|A|$. Define $u^t = u^* + t\chi_{A_1} - t\chi_{A_2}$ where χ_{A_1} and χ_{A_2} are indicator functions of A_1 and A_2 respectively. Then $u^t \in \{u_\Omega \in \mathcal{K} : \int_{\mathbb{R}^d} u_\Omega \mathrm{d}\mathbf{x} = V\}$ for any $|t| < \frac{c}{2}$. Observe that

$$\begin{aligned} \frac{\mathrm{d}^2}{\mathrm{d}t^2} \tilde{E}_\tau(u^t, (u_i^k)_{i=1}^n) &= 2\sqrt{\frac{\pi}{\tau}} \int_{\mathbb{R}^d} (\chi_{A_1} - \chi_{A_2})(S_{\tau/2}^k)^{\frac{1}{2}} (G_{\tau/2} * ((\chi_{A_1} - \chi_{A_2})(S_{\tau/2}^k)^{\frac{1}{2}})) \mathrm{d}\mathbf{x} \\ &= 2\sqrt{\frac{\pi}{\tau}} \int_{\mathbb{R}^d} (G_{\tau/4} * ((\chi_{A_1} - \chi_{A_2})(S_{\tau/2}^k)^{\frac{1}{2}}))^2 \mathrm{d}\mathbf{x} > 0, \end{aligned}$$

where $S_{\tau/2}^k$ denotes $S_{\tau/2}^k(\mathbf{x}) = G_{\tau/2} * (\sum_{i=1}^n u_i^k)(\mathbf{x})$. In particular, when taking $t = 0$, we get $u^0 = u^*$, $\frac{\mathrm{d}^2}{\mathrm{d}t^2} \tilde{E}_\tau(u^*, (u_i^k)_{i=1}^n) > 0$, which contradicts with that u^* is the maximizer. Therefore, the maximizer is in $\mathcal{B}$ and this ends the proof. $\blacksquare$

By Lemma 3.1, solving (3.1) is converted to solving (3.2), for which we follow [11] to linearize $\tilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n)$ as

$$L_\tau(u_\Omega, u_\Omega^k, (u_i^k)_{i=1}^n) = \int_{\mathbb{R}^d} u_\Omega \phi^k \mathrm{d}\mathbf{x},$$

where the dominant function is

$$\phi^k(\mathbf{x}) = \sqrt{\frac{\pi}{\tau}} \left(S_{\tau/2}^k - \sum_{i=1}^n (G_{\tau/2} * u_i^k)^2 + (S_{\tau/2}^k)^{\frac{1}{2}} G_{\tau/2} * ((S_{\tau/2}^k)^{\frac{1}{2}} (2u_\Omega^k - 1)) \right). \quad (3.3)$$

By threshold dynamics, u_Ω^{k+1} is updated by maximizing the linearized objective functional under the volume constraint

$$u_\Omega^{k+1} = \arg \max_{\int_{\mathbb{R}^d} u_\Omega \mathrm{d}\mathbf{x} = V} L_\tau(u_\Omega, u_\Omega^k, (u_i^k)_{i=1}^n).$$

Let $\sigma > 0$ be such that

$$\int_{\mathbb{R}^d} \chi_\sigma(\mathbf{x}) \mathrm{d}\mathbf{x} = V, \quad (3.4)$$

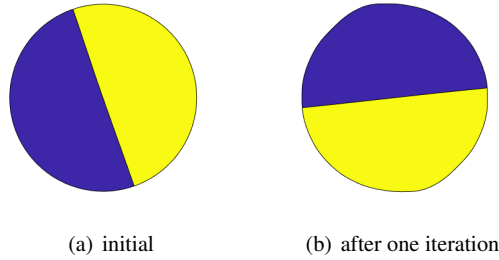


Figure 2. The initial condition is a disk. After one iteration by (3.5), the total region changes instead of remaining round.

where $\chi_\sigma(\mathbf{x})$ is the indicator function of the set $\{\mathbf{x} \in \mathbb{R}^d : \phi^k(\mathbf{x}) \geq \sigma\}$. It is easy to see that the maximizer u_Ω^{k+1} is determined pointwise by

$$u_\Omega^{k+1}(\mathbf{x}) = \begin{cases} 1 & \text{if } \phi^k(\mathbf{x}) \geq \sigma, \\ 0 & \text{if } \phi^k(\mathbf{x}) < \sigma, \end{cases} \quad (3.5)$$

which is exactly $u_\Omega^{k+1}(\mathbf{x}) = \chi_\sigma(\mathbf{x})$.

Remark 3.2. In [4], during the update of Fourier coefficients of the radial function that represents the total region, the volume constraint is broken. Then the authors project the updated coefficients onto the volume constraint using a homothety. For the methods in this paper, volume preserving is kept in the way of (3.4) and (3.5). For regular discretization, such a σ is easy to find. Besides, since σ is a threshold of the dominant function, it makes sure that the update maximizes the approximate objective functional in this time step under the volume constraint.

In fact, if we just update the total region by (3.5) in each iteration, u_Ω^{k+1} will not converge to a stationary point, and the iterations will not stop. This is because the update of u_Ω^{k+1} depends on $(u_i^k)_{i=1}^n$, which represents a shortest partition of u_Ω^k but may not be unique. For example, each diameter of a disk gives a shortest partition in the two-dimensional bisection case, and there are infinity many diameters. When we update the total region, u_Ω^{k+1} differs from u_Ω^k in the direction of increasing the objective functional of $(u_i^k)_{i=1}^n$. However, this may decrease the objective functional of other shortest partitions and eventually decrease the objective function we want to maximize. As Figure 2 shows, if we take a disk as the initial total region, the update by (3.5) will give a region that is not a disk after one iteration. This contradicts the analytical result in [13] that the disk is the optimal solution of bisection in the two-dimensional case.

To address this issue, one approach is to identify all the shortest partitions of u_{Ω}^k , and then update the total region in the direction of increasing the objective functional of all of them. However, this is impractical to implement as the number of shortest partitions can be infinite. An alternative approach is to reduce the update rate gradually when the change of objective functional is small. Then when u_{Ω}^k is close to the optimal solution, it will not change too much and go far away.

Instead of using all the points in the level set $\{\mathbf{x} \in \mathbb{R}^d : \phi^k(\mathbf{x}) \geq \sigma\}$, we only use a part of it during the update of the total region. Denote $0 \leq \beta \leq 1$ the update step-length, and $u_{\Omega}^{\text{process}}(\mathbf{x}) = \chi_{\sigma}(\mathbf{x})$ the indicator function obtained by (3.5). Denote Ω_{process} the region corresponding to $u_{\Omega}^{\text{process}}$, $A = \Omega_{\text{process}} - \Omega^k$, and $B = \Omega^k - \Omega_{\text{process}}$. Intuitively speaking, to increase the approximate objective functional in the direction of $(u_i^k)_{i=1}^n$, the original algorithm tends to delete the points in B and add the points in A into the total region. Since $|\Omega_{\text{process}}| = |\Omega^k|$ by volume conservation, $|A| = |B|$ and $A \cap B = \emptyset$. In order to reduce the update rate, we find σ_1, σ_2 such that

$$\int_A \chi_{\sigma_1}(\mathbf{x}) d\mathbf{x} = \beta|A|, \quad \int_B \chi_{\sigma_2}(\mathbf{x}) d\mathbf{x} = (1 - \beta)|B|,$$

and define $\tilde{A} = \{x \in A : \phi^k(\mathbf{x}) \geq \sigma_1\}$ and $\tilde{B} = \{x \in B : \phi^k(\mathbf{x}) \leq \sigma_2\}$. Then $\tilde{A}$ is a subset of A with $|\tilde{A}| = \beta|A|$, and for any $\mathbf{x}_1 \in \tilde{A}$, $\mathbf{x}_2 \in (A - \tilde{A})$, we have

$$\phi^k(\mathbf{x}_1) \geq \sigma_1 > \phi^k(\mathbf{x}_2),$$

which means $\tilde{A}$ is the part of A with higher dominant function values. Similarly, $|\tilde{B}| \subset |B|$, $|\tilde{B}| = \beta|B|$, and $\tilde{B}$ is the part of B with lower dominant function values. We update the total region by

$$u_{\Omega}^{k+1}(\mathbf{x}) = u_{\Omega}^k(\mathbf{x}) + \chi_{\tilde{A}} - \chi_{\tilde{B}}, \quad (3.6)$$

where $\chi_{\tilde{A}}$ and $\chi_{\tilde{B}}$ are the indicator functions of $\tilde{A}$ and $\tilde{B}$. In other words, we delete the points in $\tilde{B}$ and add the points in $\tilde{A}$ into the total region, instead of the whole of B and A . Note that $|\tilde{A}| = |\tilde{B}|$ and $\tilde{A} \cap \tilde{B} = \emptyset$. The update of the total region follows volume conservation. Here β controls the degree of differences between u_{Ω}^k and u_{Ω}^{k+1} . When $\beta = 0$, the total region will not update during iterations; when $\beta = 1$, the update is the same as (3.5). Now the update step-length can be controlled by setting a different value for β .

When β is large, u_{Ω}^k may be difficult to converge due to the non-uniqueness of the solution and the two-step updating strategy. In fact, it oscillates around a disk. To obtain a stable and convergent scheme, we need to control the updating rate of u_{Ω}^k , which is β here. For a fixed β , after sufficiently many iterations, even though u_{Ω}^k oscillates, its objective functional stays at a certain scale. To evaluate the change of objective functional during iterations, we calculate the change of averaged objective functional over a few (M in Algorithm 3) consecutive iterations. When the change is less than a tolerance r_{tol} , it is regarded to be small, and we reduce β by multiplying it by a factor γ ($0 < \gamma < 1$). Here the averaged objective functional of a few steps is used instead of the objective functional in one iteration. This can mitigate the impact of local shortest partitions on objective functional calculation.

3.1.2. Update of $(u_i^{k+1})_{i=1}^n$. In the update of shortest partition, we fix u_Ω^{k+1} and use the approximate objective functional $\hat{E}_\tau$ defined in (2.8). Functions $(u_i^{k+1})_{i=1}^n$ are computed by solving

$$(u_i^{k+1})_{i=1}^n = \arg \min_{\substack{u_i \in \mathcal{B} \\ \int_{\mathbb{R}^d} u_i dx = c_i V \\ \sum u_i = u_\Omega^{k+1}}} \hat{E}_\tau(u_\Omega^{k+1}, (u_i)_{i=1}^n).$$

Observe that $\hat{E}_\tau$ is proportional to the heat content energy in [17]

$$HC_\tau = \frac{1}{\sqrt{\tau}} \sum_{i \neq j} \int u_i(\mathbf{x})(G_\tau * u_j)(\mathbf{x})d\mathbf{x}.$$

Therefore, given u_Ω^{k+1} with parameters of stopping criterion μ , auction dynamics with volume constraints from [17] directly finds a shortest partition $(u_i^{k+1})_{i=1}^n$. Here $\mu = (m, \varepsilon_{\min}, \alpha, \varepsilon_0)$, m is the number of maximum steps, $\varepsilon_{\min}$ is the auction error tolerance, α is the epsilon scaling factor, and ε_0 is the initial epsilon value. Denote the auction dynamics method by ADM. We compute $(u_i^{k+1})_{i=1}^n$ as

$$(u_i^{k+1})_{i=1}^n = \text{ADM}(u_\Omega^{k+1}, V, \mathbf{c}, \tau; \mu),$$

where details of ADM are summarized in Algorithm 1, with the sub-routine membership auction summarized in Algorithm 2.

Remark 3.3. In Algorithm 1, ψ^l is a tensor whose i -th element is $\psi_i^l(\mathbf{x})$. After discretizing the space, it becomes a matrix. The criterion for convergence is $u_{i,l+1}(\mathbf{x}) = u_{i,l}(\mathbf{x})$ for all $i = 1, \dots, n$, that is, when the partitions are no longer updated. In Algorithm 2, $i_{\text{cs}}(\mathbf{x}, \mathbf{p})$ and $i_{\text{next}}(\mathbf{x}, \mathbf{p}, i^*)$ are defined by

$$i_{\text{cs}}(\mathbf{x}, \mathbf{p}) = \arg \max_{1 \leq i \leq n} [a_i(\mathbf{x}) - p_i], \quad i_{\text{next}}(\mathbf{x}, \mathbf{p}, i^*) \in \arg \max_{j \neq i^*} [a_j(\mathbf{x}) - p_j].$$

Here $i_{\text{cs}}(\mathbf{x}, \mathbf{p})$ is a set containing all indices satisfying the right-hand side, $i_{\text{next}}(\mathbf{x}, \mathbf{p}, i^*)$ is just an arbitrary element of the set satisfying the right-hand side.

The results of ADM critically depend on its initialization, which is usually set as random. In many cases, ADM does not always give global shortest partitions, but a local one. To make the partitions more likely to be a global shortest one, in the first method, ADM is repeated several times in each iteration. Every time a random initialization is given and a corresponding partition is created, we compute the objective functional of all partitions. The partitions with minimal objective functional are regarded as the shortest partitions. This increases the stability of the method.

The first two-step iterative method for the longest minimal length partitions problem is summarized in Algorithm 3. The stopping criterion is set to be $\beta < \beta_{\min}$ for some constant $\beta_{\min}$, or the total regions between two iterations are exactly the same.

Remark 3.4. In Algorithm 3 and Algorithm 4, convergence means $u_\Omega^{k+1} = u_\Omega^k$.

Algorithm 1: Auction dynamics

Input: Discrete total region u_Ω , volume constraint V , partition proportions $\mathbf{c}$, time step τ , parameters of stop conditions $\boldsymbol{\mu} = (m, \varepsilon_{\min}, \alpha, \varepsilon_0)$.

Output: Shortest partitions $(u_i)_{i=1}^n$.

- 1 Set $l = 0$; initialize $(u_{i,0})_{i=1}^n$ randomly with $\int u_{i,0} d\mathbf{x} = c_i V$; set $\bar{\varepsilon} = \varepsilon/n$; compute G_τ ;
- 2 **while** $l < m$ and not converged **do**
- 3 Compute the convolutions: $\psi_i^l(\mathbf{x}) = \sum_{i \neq j} (G_\tau * u_{j,l})(\mathbf{x})$;
- 4 Compute the assignment problem coefficients: $\mathbf{a}^l = 1 - \boldsymbol{\psi}^l$;
- 5 Initialize the prices $\mathbf{p} = \mathbf{0}$, and $\varepsilon = \varepsilon_0$;
- 6 **while** $\varepsilon \geq \bar{\varepsilon}$ **do**
- 7 Run Algorithm 2: $((u_i^{out})_{i=1}^n, \mathbf{p}^{out}) = \text{Membership auction}(\varepsilon, V\mathbf{c}, \mathbf{a}^l, \mathbf{p}, u_\Omega)$;
- 8 Set $\mathbf{p} = \mathbf{p}^{out}$;
- 9 Divide ε by α ;
- 10 **if** $\varepsilon < \bar{\varepsilon}$ **then**
- 11 Set $(u_{i,l+1})_{i=1}^n = (u_i^{out})_{i=1}^n$;
- 12 **end**
- 13 **end**
- 14 Set $l = l + 1$;
- 15 **end**
- 16 Return $(u_i)_{i=1}^n = (u_{i,l})_{i=1}^n$

Remark 3.5. In Algorithm 3, although ADM is repeated with the same parameters, the random initialization makes the output $(u_{i,q}^{k+1})_{i=1}^n$ different each time.

3.2. The second method

In the first method, to increase the stability, ADM is repeated several times in one iteration to find the shortest partitions. However, this increases the time complexity. To address this issue, the second method uses a more efficient way for stability. In the updating of $(u_i^k)_{i=1}^n$, the second method follows a similar process as the first method but only employs ADM once. We omit the details here. In the updating of u_Ω^{k+1} , a regularization term is introduced.

3.2.1. Update of u_Ω^{k+1} by $(u_i^k)_{i=1}^n$. In the update of the total region, the approximate objective functional $\tilde{E}_\tau$ in (2.6) is used. Given u_Ω^k and $(u_i^k)_{i=1}^n$, u_Ω^{k+1} is attained by

$$u_\Omega^{k+1} = \arg \max_{\substack{u_\Omega \in \mathcal{B} \\ \int u_\Omega = V}} \tilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n) + \lambda \int_{\mathbb{R}^d} u_\Omega G_{\tau'} * \left(\sum_{i=1}^n u_i^k \right) d\mathbf{x}, \quad (3.7)$$

Algorithm 2: Membership auction

Input: $\varepsilon > 0$, volumes $\mathbf{V}$, coefficients $\mathbf{a}$, initial prices $\mathbf{p}^0$ and indicator function $u_\Omega(\mathbf{x})$.

Output: Final partitions and prices $((u_i)_{i=1}^n, \mathbf{p})$.

```

1 For every  $i \in \{1, \dots, n\}$  set  $u_i(\mathbf{x}) = 0$  for all  $\mathbf{x}$  such that  $u_\Omega(\mathbf{x}) = 1$ ; set  $\mathbf{p} = \mathbf{p}^0$ 
2 while some  $\mathbf{x}$  is unassigned, i.e.,  $u_\Omega(\mathbf{x}) = 1$  but  $\sum_i u_i(\mathbf{x}) = 0$ , do
3   for each unassigned  $\mathbf{x}$  do
4     Compute  $i_{cs}(\mathbf{x}, \mathbf{p})$  and choose one element  $i^* \in i_{cs}(\mathbf{x}, \mathbf{p})$ ;
5     Set  $b(\mathbf{x}) = p_{i^*} + \varepsilon + (a_{i^*}(\mathbf{x}) - p_{i^*}) - (a_{i_{\text{next}}(\mathbf{x}, \mathbf{p}, i^*)}(\mathbf{x}) - p_{i_{\text{next}}(\mathbf{x}, \mathbf{p}, i^*)})$ ;
6     if  $\int u_{i^*} d\mathbf{x} = V_{i^*}$  then
7       Find  $\mathbf{y} = \arg \min_{u_{i^*}(\mathbf{z})=1} b(\mathbf{z})$ ;
8       Set  $u_{i^*}(\mathbf{y}) = 0$  and  $u_{i^*}(\mathbf{x}) = 1$ ;
9       Set  $p_{i^*} = \min_{u_{i^*}(\mathbf{z})=1} b(\mathbf{z})$ ;
10    else
11      Set  $u_{i^*}(\mathbf{x}) = 1$ ;
12      if  $\int u_{i^*} d\mathbf{x} = V_{i^*}$  then
13        Set  $p_{i^*} = \min_{u_{i^*}(\mathbf{z})=1} b(\mathbf{z})$ ;
14      end
15    end
16  end
17 end
18 Return  $((u_i)_{i=1}^n, \mathbf{p})$ 
    
```

where $\lambda > 0, \tau' > 0$. In (3.7), the second term is a regularization term to relax the constraint of $\sum_{i=1}^n u_i = u_\Omega$ and ensure that u_Ω^{k+1} does not deviate significantly from u_Ω^k . In the following, we denote the regularized objective functional as

$$\tilde{E}_{\tau, \tau'}^\lambda(u_\Omega, (u_i^k)_{i=1}^n) = \tilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n) + \lambda \int_{\mathbb{R}^d} u_\Omega G_{\tau'} * \left(\sum_{i=1}^n u_i^k \right) d\mathbf{x}.$$

In fact, (3.7) is a discretized scheme for an evolution equation of u_Ω . Based on equation (2.6), u_Ω^{k+1} is expected to maximize $\tilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n)$. Consider a gradient ascent flow

$$\frac{\partial u_\Omega}{\partial t} - \partial \tilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n) = 0. \quad (3.8)$$

We update u_Ω^{k+1} by solving (3.8) by a short-time (one time step), and discretize (3.8) with the time step $1/\lambda$ by a backward Euler scheme:

$$\frac{u_\Omega^{k+1} - u_\Omega^k}{1/\lambda} - \partial \tilde{E}_\tau(u_\Omega^{k+1}, (u_i^k)_{i=1}^n) = 0.$$

Algorithm 3: The first method for longest minimal length partitions

Input: Volume constraint V , partition proportions $\mathbf{c} = (c_1, \dots, c_n)$ satisfying $c_i > 0$ and $\sum_{i=1}^n c_i = 1$, time step $\tau > 0$, auction dynamics parameters μ , repeat times of auction dynamics p , update step-length parameters $0 \leq \beta_0 \leq 1, 0 < \gamma < 1$, the number of steps in average objective functional $M \in \mathbb{N}, r_{\text{tol}} > 0, 0 < \beta_{\min} < 1$.

Output: u_Ω maximizing the objective functional, shortest partitions $(u_i)_{i=1}^n$.

- 1 Set $k = 0$; initialize u_Ω^0 randomly with $\int u_\Omega^0 d\mathbf{x} = V$; $\beta = \beta_0$;
- 2 **for** $1 \leq q \leq p$ **do**
- 3 Compute $(u_{i,q}^0)_{i=1}^n = \text{ADM}(u_\Omega^0, V, \mathbf{c}, \tau; \mu)$;
- 4 **end**
- 5 Set $(u_i^0)_{i=1}^n = \arg \min_{(u_{i,q}^0)_{i=1}^n} \tilde{E}_\tau(u_\Omega^0, (u_{i,q}^0)_{i=1}^n)$;
- 6 **while** $\beta > \beta_{\min}$ and not converge **do**
- 7 Compute dominant function ϕ^k by (3.3);
- 8 Update u_Ω^{k+1} by (3.6) using β ;
- 9 **for** $1 \leq q \leq p$ **do**
- 10 Compute $(u_{i,q}^{k+1})_{i=1}^n = \text{ADM}(u_\Omega^{k+1}, V, \mathbf{c}, \tau; \mu)$;
- 11 **end**
- 12 Set $(u_i^{k+1})_{i=1}^n = \arg \min_{(u_{i,q}^{k+1})_{i=1}^n} \tilde{E}_\tau(u_\Omega^{k+1}, (u_{i,q}^{k+1})_{i=1}^n)$;
- 13 **if** $k > M$ **then**
- 14 Compute $E_{\text{ave}}^k = \frac{1}{M} \sum_{j=k-M+1}^k \tilde{E}_\tau(u_\Omega^j, (u_i^j)_{i=1}^n)$;
- 15 Compute $E_{\text{ave}}^{k+1} = \frac{1}{M} \sum_{j=k-M+2}^{k+1} \tilde{E}_\tau(u_\Omega^j, (u_i^j)_{i=1}^n)$;
- 16 **if** $|E_{\text{ave}}^{k+1} - E_{\text{ave}}^k| / |E_{\text{ave}}^{k+1}| < r_{\text{tol}}$ **then**
- 17 $\beta = \gamma\beta$;
- 18 **end**
- 19 **end**
- 20 $k = k + 1$;
- 21 **end**
- 22 Return $u_\Omega = u_\Omega^k, (u_i)_{i=1}^n = (u_i^k)_{i=1}^n$

Then u_Ω^{k+1} solves

$$\begin{aligned}
 u_\Omega^{k+1} &= \arg \min_{u_\Omega} \left[\frac{\lambda}{2} \int_{\mathbb{R}} |u_\Omega - u_\Omega^k|^2 d\mathbf{x} - \tilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n) \right] \\
 &= \arg \max_{u_\Omega} \left[\tilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n) - \frac{\lambda}{2} \int_{\mathbb{R}} |u_\Omega - u_\Omega^k|^2 d\mathbf{x} \right] \\
 &= \arg \max_{u_\Omega} \left[\tilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n) - \frac{\lambda}{2} \int_{\mathbb{R}} ((u_\Omega)^2 + (u_\Omega^k)^2 - 2u_\Omega u_\Omega^k) d\mathbf{x} \right].
 \end{aligned}$$

Using the constraint $u_\Omega \in \mathcal{B}$, $\int_{\mathbb{R}} u_\Omega d\mathbf{x} = V$ and the fact $u_\Omega^k \in \mathcal{B}$, $\int_{\mathbb{R}} u_\Omega^k d\mathbf{x} = V$, we further deduce

$$\begin{aligned} u_\Omega^{k+1} &= \arg \max_{u_\Omega} \left[\widetilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n) - \frac{\lambda}{2} \int_{\mathbb{R}} (u_\Omega + u_\Omega^k - 2u_\Omega u_\Omega^k) d\mathbf{x} \right] \\ &= \arg \max_{u_\Omega} \left[\widetilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n) - \frac{\lambda}{2} \int_{\mathbb{R}} (-2u_\Omega u_\Omega^k) d\mathbf{x} \right] - \lambda V \\ &= \arg \max_{u_\Omega} \left[\widetilde{E}_\tau(u_\Omega, (u_i^k)_{i=1}^n) + \lambda \int_{\mathbb{R}} u_\Omega u_\Omega^k d\mathbf{x} \right]. \end{aligned} \quad (3.9)$$

Notice that $\sum_{i=1}^n u_i^k = u_\Omega^k$. Thus, (3.7) is an approximation of (3.9), and such an approximation becomes exact as $\tau' \rightarrow 0^+$.

To solve (3.7), similar to the first method, we relax the optimization to an equivalent problem: finding u_Ω^{k+1} such that

$$u_\Omega^{k+1} = \arg \max_{\substack{u_\Omega \in \mathcal{K} \\ \int u_\Omega = V}} \widetilde{E}_{\tau, \tau'}^\lambda(u_\Omega, (u_i^k)_{i=1}^n),$$

where $\mathcal{K} := \{u \in \text{BV}(\Omega, \mathbb{R}) : u \in [0, 1]\}$. The proof of equivalence is similar to that of Lemma 3.1, and is omitted here.

The linearity of $\widetilde{E}_{\tau, \tau'}^\lambda(u_\Omega, (u_i^k)_{i=1}^n)$ is computed as

$$L_{\tau, \tau'}^\lambda(u_\Omega, u_\Omega^k, (u_i^k)_{i=1}^n) = \int_{\mathbb{R}^d} u_\Omega \phi^k d\mathbf{x},$$

where the dominant function is

$$\begin{aligned} \phi^k(\mathbf{x}) &= \sqrt{\frac{\pi}{\tau}} \left(S_{\tau/2}^k - \sum_{i=1}^n (G_{\tau/2} * u_i^k)^2 + (S_{\tau/2}^k)^{\frac{1}{2}} G_{\tau/2} * ((S_{\tau/2}^k)^{\frac{1}{2}} (2u_\Omega^k - 1)) \right) \\ &\quad + \lambda G_{\tau'} * \left(\sum_{i=1}^n u_i^k \right), \end{aligned} \quad (3.10)$$

and $S_{\tau/2}^k$ denotes $S_{\tau/2}^k(\mathbf{x}) = G_{\tau/2} * (\sum_{i=1}^n u_i^k)(\mathbf{x})$. Then u_Ω^{k+1} can be computed by (3.6), the same as in the first method.

For the stopping criterion, the second method also uses β to control the update step-length, and reduces β when the rate of change of the averaged objective functional in a few iterations is less than the tolerance r_{tol} . The iteration stops when $\beta < \beta_{\min}$, or when the total region between two consecutive steps are the same. The algorithm of the second method is summarized in Algorithm 4.

In the first method, the auction dynamics method is run several times in each iteration to find a partition, making the algorithm more stable and the partition more likely to be the global shortest one instead of a local one. However, the computational complexity of this method is high. In the second method, a regularization term is added in the approximate objective functional for stability. It does not need to run the auction dynamics several

Algorithm 4: The second method for longest minimal length partitions

Input: Volume constraint V , partition proportions $\mathbf{c} = (c_1, \dots, c_n)$ satisfying $c_i > 0$ and $\sum_{i=1}^n c_i = 1$, time step $\tau > 0$, regularization parameter $\lambda > 0$, time step in regularization term $\tau' > 0$ auction dynamics parameters μ , update step-length parameters $0 \leq \beta_0 \leq 1, 0 < \gamma < 1, M \in \mathbb{N}, r_{\text{tol}} > 0, 0 < \beta_{\min} < 1$

Output: u_Ω maximizing the objective functional, relative shortest partitions $(u_i)_{i=1}^n$

```

1 Set  $k = 0$ ; initialize  $u_\Omega^0$  randomly with  $\int u_\Omega^0 dx = V$ ;
2 Find  $(u_i^0)_{i=1}^n = \text{ADM}(u_\Omega^0, V, \mathbf{c}, \tau; \mu)$ ;
3 while  $\beta > \beta_{\min}$  and not converge do
4   Compute dominant function  $\phi^k$  by (3.10) using  $\tau$  and  $\tau'$ ;
5   Update  $u_\Omega^{k+1}$  by (3.6);
6   Compute  $(u_i^{k+1})_{i=1}^n = \text{ADM}(u_\Omega^{k+1}, V, \mathbf{c}, \tau; \mu)$ ;
7   if  $k > M$  then
8     Compute  $E_{\text{ave}}^k = \frac{1}{M} \sum_{j=k-M+1}^k \tilde{E}_\tau(u_\Omega^j, (u_i^j)_{i=1}^n)$ ;
9     Compute  $E_{\text{ave}}^{k+1} = \frac{1}{M} \sum_{j=k-M+2}^{k+1} \tilde{E}_\tau(u_\Omega^j, (u_i^j)_{i=1}^n)$ ;
10    if  $|E_{\text{ave}}^{k+1} - E_{\text{ave}}^k| / |E_{\text{ave}}^{k+1}| < r_{\text{tol}}$  then
11       $\beta = \gamma\beta$ ;
12    end
13  end
14   $k = k + 1$ ;
15 end
16 Return  $u_\Omega = u_\Omega^k, (u_i)_{i=1}^n = (u_i^k)_{i=1}^n$ 

```

times at every iteration in order to maintain the stability of the algorithm. Thus, it is faster than the first method. Since the partition in each step may not be the global minimizer but a local one, the second method may need more iterations to converge.

4. Numerical experiments

We demonstrate the effectiveness of the proposed methods by experiments in two and three dimensions. Our methods are implemented by MATLAB. Convolutions are implemented using the Fast Fourier Transform, and for both methods, we stop iterating when $u_\Omega^{k+1} = u_\Omega^k$.

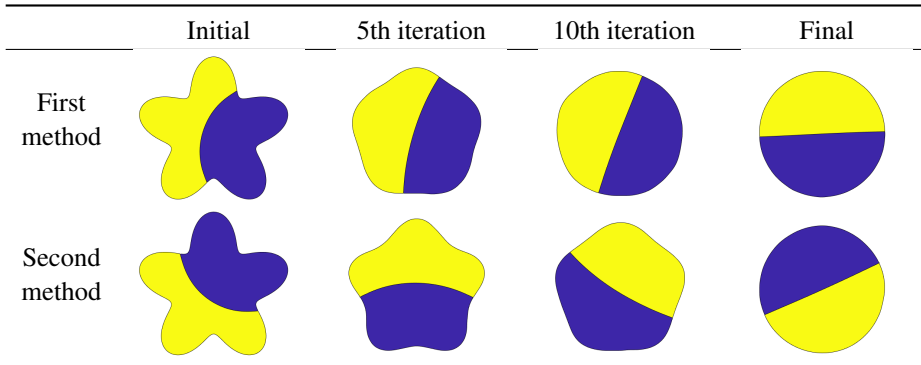


Figure 3. Two uniform partitions in two dimensions. The initial condition is set to be a five-petal flower. The total regions and their shortest partitions of both methods in the initial, 5th, 10th, and final iterations are plotted.

4.1. Numerical experiments in two dimensions

To simulate the longest minimal length partitions in two dimensions, a 256×256 regular grid is formed in $[-\pi, \pi]^2$ with a space step $\Delta x = 2\pi/256$. The time step is set to be $\tau = 2\Delta x \approx 0.049$, and $\tau' = 0.5\Delta x \approx 0.013$. The stopping parameters in the auction dynamics method are set to be $\mu = (m, \varepsilon_{\min}, \alpha, \varepsilon_0) = (1000, 10^{-7}, 4, 0.1)$. Other parameters are set as $\beta_0 = 1$ (the initial update rate for u_Ω), $\gamma = \frac{1}{2}$ (the decay factor of β), $\beta_{\min} = 0.05$ (the lower bound of β), $M = 5$ (the number of steps in averaged objective functional), and $r_{\text{tol}} = 10^{-4}$. For the first method, auction dynamics repeat-time is $p = 5$. For the second method, regularization parameter is $\lambda = 10$.

4.1.1. Two partitions. We first conduct experiments for the two-dimensional example with two partitions of equal proportions. The initial condition is set to be a five-petal flower with the indicator function $\rho^2 < \pi^2(0.4 + 0.2 \sin 5\theta)$, where ρ and θ are the radius and angle in the polar coordinates respectively. The partition proportion is $\mathbf{c} = (\frac{1}{2}, \frac{1}{2})$. Both methods derived in Section 3 are implemented. The first method stops after 34 iterations, and the second method stops after 63 iterations. The total regions and their shortest partitions of both methods in the initial, 5th, 10th and final iterations are plotted in Figure 3. For visualization of the results, the indicator functions of the region and partitions are smoothed using a Gaussian kernel convolution before being plotted in Figure 3 and the figures in the following.

The evolution of the approximate objective functionals $\tilde{E}_\tau(u_\Omega^k, (u_i^k)_{i=1}^n)$ of both methods are plotted in Figure 4. As we can see, the final approximate objective functionals of two methods are both about 9.32. In the 10th iteration of the first method, the objective functional is near the final maximum objective functional, and the shape of total region shown in Figure 3 is close to a disk. Compared with the approximate objective functional of the second method, the approximate objective functional of first method has less fluc-

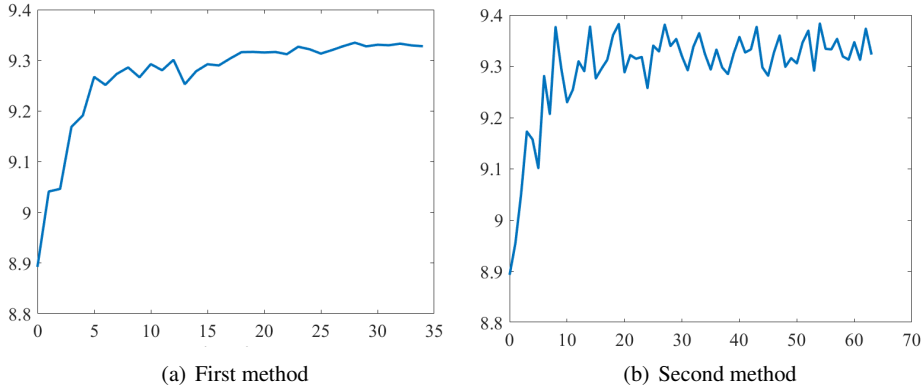


Figure 4. Two uniform partitions in two dimensions. The approximate objective functionals $\tilde{E}_\tau(u_\Omega^k, (u_i^k)_{i=1}^n)$ of two methods versus iteration times. The horizontal axis represents the number of iterations. The vertical axis represents the functional value.

tuation. This is because auction dynamics is repeated several times in each step of the first method, which increases the likelihood of obtaining a global shortest partition instead of a local minimizer. Thus, the approximate objective functional in the first method should have a smaller gap with the exact objective functional compared to the second method.

To further confirm that we get a disk as the optimal total region, the isoperimetric inequality is checked for the final iteration results in Figure 3. For any simply connected region Ω in 2D, we have

$$4\pi|\Omega| \leq |\partial\Omega|^2,$$

and the equality holds only when Ω is a disk. Using the indicator function of the initial condition, the area is computed to be $0.4\pi^3$. For the perimeter, we use the approximation

$$|\partial\Omega| \approx \sqrt{\frac{\pi}{\tau}} \int_{\mathbb{R}^d} u_\Omega G_\tau * (1 - u_\Omega) dx.$$

Then the values of $4\pi|\Omega|/|\partial\Omega|^2$ for the results given by both methods are computed, which are 1.0056 and 1.0055. Here the final values are bigger than 1 due to the numerical error. Since they are close to 1, the isoperimetric inequality corroborates that the optimal total region by each method is a disk.

For the longest minimal partition problem, it is difficult to define accuracy, as there are infinitely many solutions. Suppose the conjecture (the solution is a disk in two dimension and a ball in three dimension) is true. If one can find one partition solving the problem, rotate the partition by any angle is also a solution to the problem. One surrogate for measuring accuracy is to check how close u_Ω is to a disk. Besides checking it visually, the quantity $4\pi|\Omega|/|\partial\Omega|^2$ provides another perspective to quantify the accuracy. However, when we compute the quantity, the approximation of $|\partial\Omega|$ introduces additional error,

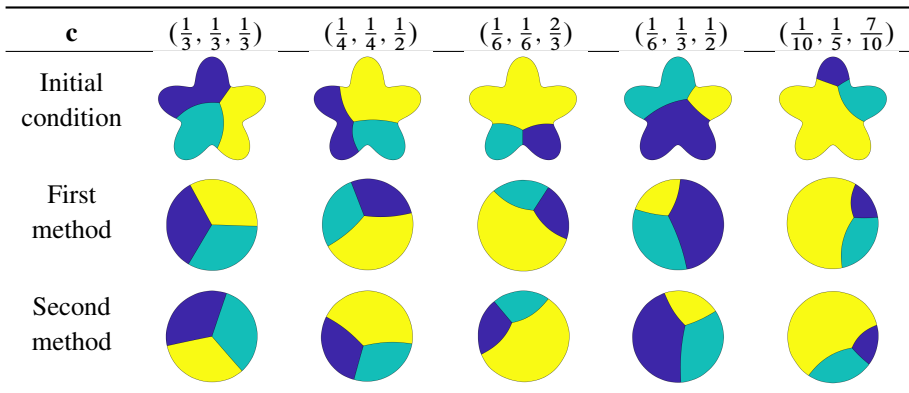


Figure 5. Three partitions in two dimensions. The initial condition is set to be a five-petal flower. The total regions and their shortest partitions of both methods in the final iterations are plotted for different values of $\mathbf{c}$ of the three-partition case.

making a convergence test unreliable. We leave a comprehensive study of the solution property and design of reliable accuracy measurement as our future work.

As the experiments show, for the uniform two-partition case, numerical simulations imply that the disk is the maximizer for longest minimal length partitions, which is compatible with the analytical results proved in [13].

4.1.2. Three partitions. We then test the proposed methods to find the optimal solution for three partitions with equal and unequal proportions. The volume proportion parameter $\mathbf{c}$ is set to be $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, $(\frac{1}{4}, \frac{1}{4}, \frac{1}{2})$, $(\frac{1}{6}, \frac{1}{6}, \frac{2}{3})$, $(\frac{1}{6}, \frac{1}{3}, \frac{1}{2})$ and $(\frac{1}{10}, \frac{1}{5}, \frac{7}{10})$. The results of two methods are shown in Figure 5. As we can see, although the value of $\mathbf{c}$ varies, the optimal total regions given by both methods continue to be discs, and the partition results of both methods look similar.

4.1.3. Sensitivity to initial shape. We test the sensitivity of the proposed methods to the initial shape. In this experiment, both methods are used to find the optimal solution for three partitions with equal proportions. The shape of initial condition is set to be a triangle, a rectangle, and randomly generated pentagons. The results are shown in Figure 6. Both methods give the optimal total region to be a disk no matter how the shape of initial condition changes.

4.1.4. Six and nine partitions. In this set of experiments, the initial condition is set to be the five-petal flower, whose indicator function is given in Section 4.1.1. The two methods are tested to solve the six-partition and nine-partition problems. For each problem, the volume proportions of every part in the partitions are set to be equal. The total regions and their shortest partitions of both methods in the initial, 5th, 10th, 20th, and final iterations are plotted in Figure 7.

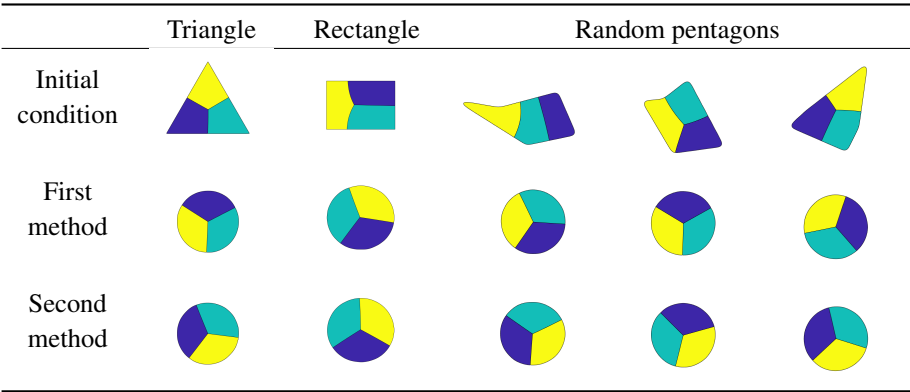


Figure 6. Three partitions in two dimensions, with sensitivity to initial shape. The proportion is set to be $\mathbf{c} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. The total regions and their shortest partitions of both methods in the final iterations are plotted for different shape of initial conditions.

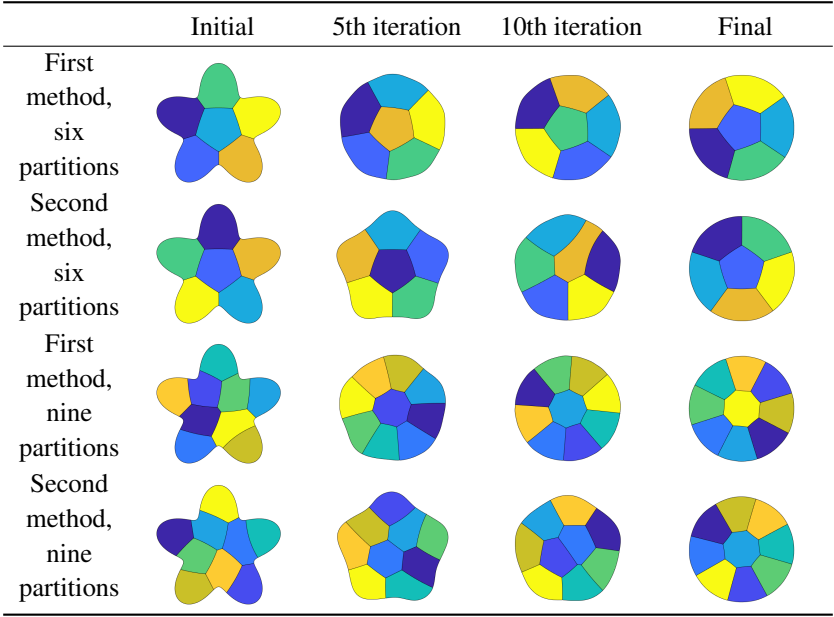


Figure 7. Six and nine uniform partitions in two dimensions. The initial condition is set to be a five-petal flower. The total regions and their shortest partitions of both methods in the initial, 5th, 10th, and final iterations are plotted.

The evolution of the approximate objective functional $\tilde{E}_\tau(u_\Omega^k, (u_i^k)_{i=1}^n)$ for both problems is plotted in Figure 8. We observe some irregular protrusions in the objective functional evolution. This is because the auction dynamics may give a local shortest partition

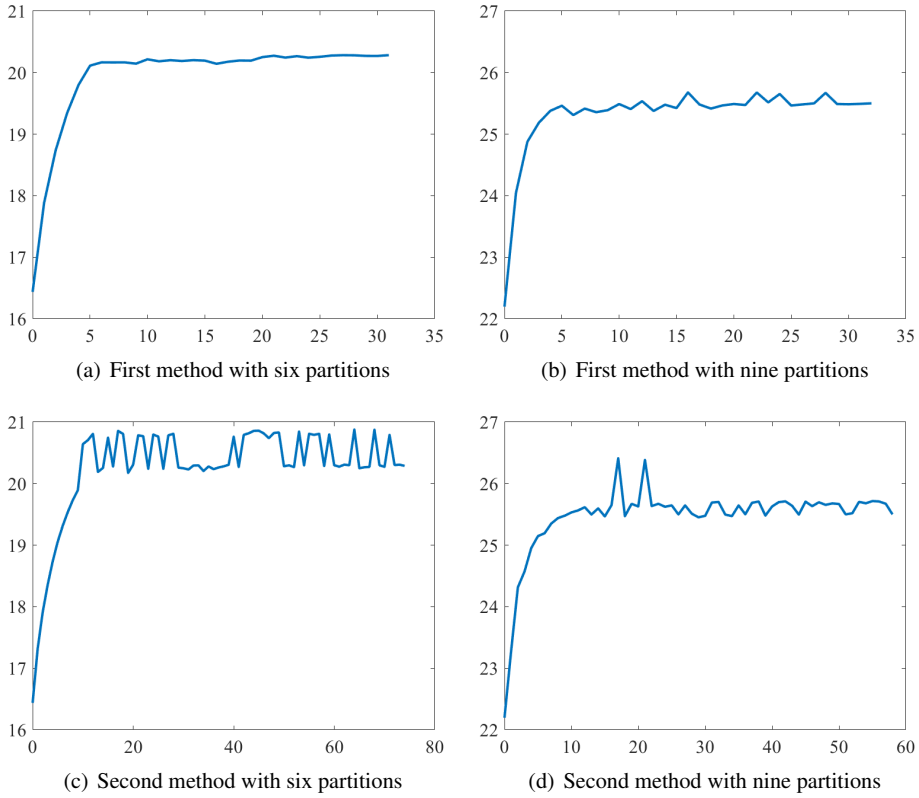


Figure 8. Six and nine uniform partitions in two dimensions. The approximate objective functionals $\tilde{E}_\tau(u_\Omega^k, (u_i^k)_{i=1}^n)$ of two methods versus iteration times. The horizontal axis represents the number of iterations. The vertical axis represents the functional value.

instead of a global one in some iterations. Then the approximate objective functional is computed to be larger than the real objective functional, which leads to a protrusion. Since the first method repeats the auction dynamics several times, it is more likely to get a global shortest partition than the second method. Thus, the first method produces fewer protrusions in the evolution of the objective functional. This implies that the first method is more stable than the second one. However, since it repeats auction dynamics many times, the first method requires more time than the second one in general. We show the CPU times of both methods with different numbers of partitions in Table 1. As mentioned above, even though the first method requires fewer iterations, since it repeats auction dynamics several times, its overall CPU time is higher than that of the second method.

No. of partitions	No. of Iter.		CPU time(s)	
	First method	Second method	First method	Second method
2	34	63	281.49	112.72
6	31	74	995.67	427.05
9	32	59	1611.79	650.43

Table 1. Uniform partitions in two dimensions. Number of iterations and CPU times of both methods with different numbers of partition.

Partition number	First method			Second method		
	3	6	9	3	6	9
Final iteration						
An expanded view						
A translucent expanded view						

Figure 9. Uniform partitions in three dimensions. The initial condition is set to be a cube. The total regions and their shortest partitions of both methods in the final iteration (together with an expanded view and a translucent expanded view) are plotted for three-partition case, six-partition case, and nine-partition case.

4.2. Numerical experiments in three dimensions

To simulate the problem in three-dimensional cases, a $128 \times 128 \times 128$ regular grid is formed in $[-\pi, \pi]^3$. The tolerance of objective functional rate is set to be $r_{\text{tol}} = 0.0005$. For the first method, auction dynamics repeats $p = 3$ times in each iteration. Other parameters are the same as in two-dimensional cases. The initial total region is set to be a cube. We test both methods for three-partition, six-partition, and nine-partition cases. The volume proportions of every part in the partitions are set to be equal in each case. The results of both methods are shown in Figure 9, together with an expanded view and a translucent expanded view.

In Figure 9, the solutions of the total region after convergence by the first method is consistent with the solutions of the second method. They both give a ball as the optimal solution for different numbers of partitions. This numerically supports the conjecture

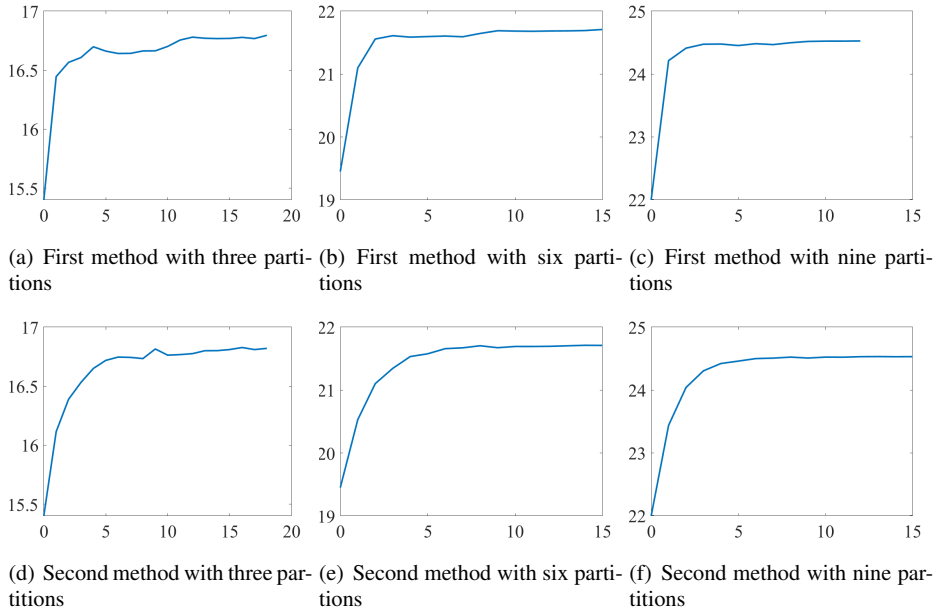


Figure 10. Uniform partitions in three dimensions. The approximate objective functionals $\tilde{E}_\tau(u_\Omega^k, (u_i^k)_{i=1}^n)$ of the two methods versus iteration times. The horizontal axis represents the number of iterations. The vertical axis represents the functional value.

No. of partitions	No. of Iter.		CPU time(s)	
	First method	Second method	First method	Second method
3	18	18	10678.05	2768.09
6	15	15	11053.22	4260.25
9	12	15	15129.92	6282.31

Table 2. Uniform partitions in three dimensions. Number of iterations and CPU times of both methods.

that the ball is the maximizer of the longest minimal length partitions problem in three dimensions.

The evolution of approximate objective functionals for each experiment are shown in Figure 10. As the number of partitions increases, the approximate objective functional of final total region also increases. In each case, the objective functional by the first method is consistent with the objective functional by the second method.

The CPU time of both methods with different numbers of partitions are given in Table 2. Again, the second method uses more iterations but less GPU time compared to the first method.

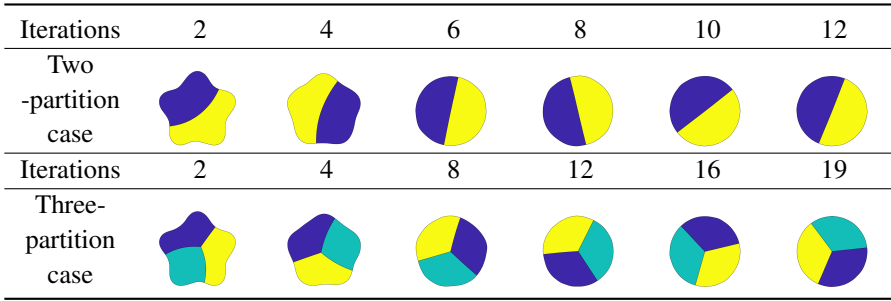


Figure 11. Objective-functional-monotone method. The initial condition is set to be a “five-petal flower”. The total regions and their shortest partitions of the objective functional monotone method in some iterations are plotted for two-partition and three-partition cases with equal proportions.

5. An objective-functional-monotone method

Based on the first method, we further develop a method that promises the approximate objective functional is increasing during iterations. The idea is to check whether $\tilde{E}_\tau(u_\Omega^{k+1}, (u_i^{k+1})_{i=1}^n)$ is greater than or equal to $\tilde{E}_\tau(u_\Omega^k, (u_i^k)_{i=1}^n)$ in each step. If not, then $(u_i^k)_{i=1}^n$ could be a local minimizer instead of global shortest partition, or u_Ω^k is near the optimal solution. In this case, we just reduce β from 1, get new $u_\Omega^{k+1}, (u_i^{k+1})_{i=1}^n$, and check if the objective functional is increasing. If the objective functional is increasing, then the iterations keep going. Otherwise, β is further reduced, and new $u_\Omega^{k+1}, (u_i^{k+1})_{i=1}^n$ are computed to compare with the objective functional. If $u_\Omega^{k+1}, (u_i^{k+1})_{i=1}^n$ are not found to increase the objective functional, and β is reduced to 0, then $(u_i^k)_{i=1}^n$ is not a global shortest partition, or u_Ω^k is likely to be the optimal total region. To check which case it is, auction dynamics is repeated multiple times to find a partition of u_Ω^k with smaller objective functional than $(u_i^k)_{i=1}^n$. If such partition is found, then it is set to be the new $(u_i^k)_{i=1}^n$, and we take a step back to check whether $\tilde{E}_\tau(u_\Omega^k, (u_i^k)_{i=1}^n)$ is greater than or equal to $\tilde{E}_\tau(u_\Omega^{k-1}, (u_i^{k-1})_{i=1}^n)$. If such partition is not found, then we accept u_Ω^k to be the result, and $(u_i^k)_{i=1}^n$ is its shortest partition.

This method is tested for two-dimensional examples with two and three partitions and equal proportions. The total regions and their shortest partitions of both cases in some iterations are shown in Figure 11. The approximate objective functionals are shown in Figure 12. For the two-partition case, this method uses 12 iterations and 3147.61 seconds. For the three-partition case, this method uses 19 iterations and 3896.23 seconds. Although the method promises the objective functional is increasing, it spends ten times more CPU time than the first method for the two-partition case. This is because the new method uses a lot of time (many repetitions of auction dynamics) for checking whether u_Ω^k and $(u_i^k)_{i=1}^n$ are global optimizers.

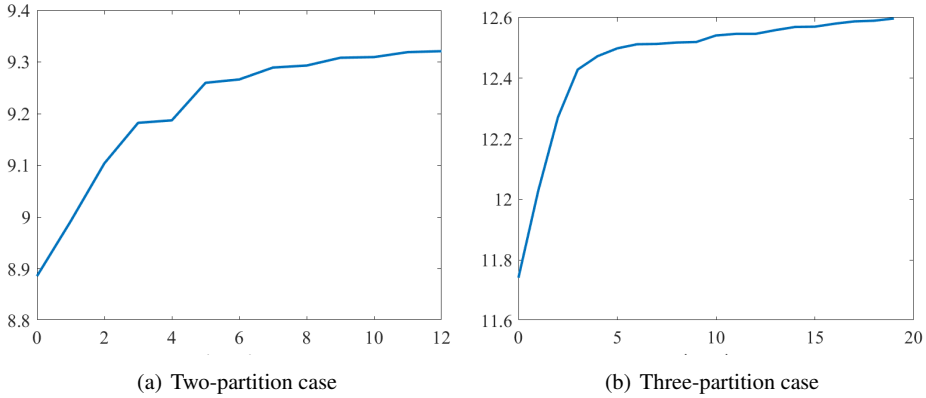


Figure 12. Objective-functional-monotone method for uniform partitions in two dimensions. The approximate objective functionals $\tilde{E}_\tau(u_\Omega^k, (u_i^k)_{i=1}^n)$ of both cases versus iteration times are plotted. The horizontal axis represents the number of iterations. The vertical axis represents the functional value.

6. Conclusions

In this paper, two numerical methods are derived for the simulation of the longest minimal length partitions problem. The methods are based on approximations of the objective functional by a short-time heat flow, threshold dynamics, and splitting methods to solve the relative constrained max-min optimization problem. In both methods, auction dynamics is used to find the shortest partitions, and the update of total region is based on threshold dynamics. The shortest partitions and total region are updated in an alternative manner. To reduce the influence of local minimal partitions, the first method repeats auction dynamics several times in each iteration, and the second method introduces a proximal regularization term, which is shown to be equivalent to a gradient ascent flow. Both methods update the total region by introducing an update step-length, which decreases gradually according to some criterion during iterations. The effectiveness of both methods are demonstrated by comprehensive numerical experiments, which validate the conjecture that the disk is the maximizer for two-dimensional problems and the ball is the maximizer for three-dimensional problems.

This paper focuses on the simulation to an approximate problem to provide the numerical evidence to the conjecture for the longest minimal length partitions problem. It would be interesting to consider the proof of the conjecture in the sense of the approximate objective functional and further consider the limit as τ goes to 0. It would be also interesting to use the level-set-based framework [7, 18, 24, 27] to approximate such problems.

Funding. H. Liu was partially supported by National Natural Science Foundation of China 12201530, HKRGC ECS 22302123, HKRGC GRF 12301925 and Guangdong and Hong Kong Universities “1+1+1” Joint Research Collaboration Scheme 2025A0505000007.

D. Wang was partially supported by National Natural Science Foundation of China (Grant No. 12422116), Shenzhen Science and Technology Innovation Program (Grant No. JCYJ20220530143803007, RCYX20221008092843046), and Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone Project (No.HZQSWs-KCCYB-2024016).

References

- [1] H. Auerbach, [Sur un problème de M. Ulam concernant l'équilibre des corps flottants](#). *Studia Math.* **7** (1938), no. 1, 121–142 Zbl [64.0733.04](#)
- [2] G. Barles and C. Georgelin, [A simple proof of convergence for an approximation scheme for computing motions by mean curvature](#). *SIAM J. Numer. Anal.* **32** (1995), no. 2, 484–500 Zbl [0831.65138](#) MR [1324298](#)
- [3] J. Berry, E. Bongiovanni, W. Boyer, B. Brown, M. Dannenberg, P. Gallagher, D. Hu, J. Liang, A. Loving, Z. Martin, M. Miller, B. Perpetua, S. Tammen, and Y. Zeng, The convex body isoperimetric conjecture in $\mathbb{R}^2$. *Rose-Hulman Undergrad. Math. J.* **18** (2017), no. 2, article no. 2, 9–33 Zbl [1398.52003](#) MR [3766822](#)
- [4] B. Bogosel and E. Oudet, [Longest minimal length partitions](#). *Interfaces Free Bound.* **24** (2022), no. 1, 95–135 Zbl [1486.49041](#) MR [4395705](#)
- [5] Z. M. Boyd, E. Bae, X.-C. Tai, and A. L. Bertozzi, [Simplified energy landscape for modularity using total variation](#). *SIAM J. Appl. Math.* **78** (2018), no. 5, 2439–2464 Zbl [1398.65133](#) MR [3853618](#)
- [6] A. Braides, [Approximation of free-discontinuity problems](#). Lecture Notes in Math. 1694, Springer, Berlin, 1998 Zbl [0909.49001](#) MR [1651773](#)
- [7] T. F. Chan and X.-C. Tai, [Level set and total variation regularization for elliptic inverse problems with discontinuous coefficients](#). *J. Comput. Phys.* **193** (2004), no. 1, 40–66 Zbl [1036.65086](#) MR [2022688](#)
- [8] H. T. Croft, K. J. Falconer, and R. K. Guy, [Unsolved problems in geometry](#). Probl. Books in Math., Springer, New York, 1991 Zbl [0748.52001](#) MR [1107516](#)
- [9] H. G. Eggleston, [The maximal length of chords bisecting the area or perimeter length of plane convex sets](#). *J. London Math. Soc.* **36** (1961), 122–128 Zbl [0109.14902](#) MR [125494](#)
- [10] M. Elsey, S. Esedoğlu, and P. Smereka, [Diffusion generated motion for grain growth in two and three dimensions](#). *J. Comput. Phys.* **228** (2009), no. 21, 8015–8033 Zbl [1175.82065](#) MR [2573343](#)
- [11] S. Esedoğlu and F. Otto, [Threshold dynamics for networks with arbitrary surface tensions](#). *Comm. Pure Appl. Math.* **68** (2015), no. 5, 808–864 Zbl [1334.82072](#) MR [3333842](#)
- [12] S. Esedoğlu and Y.-H. R. Tsai, [Threshold dynamics for the piecewise constant Mumford-Shah functional](#). *J. Comput. Phys.* **211** (2006), no. 1, 367–384 Zbl [1086.65522](#) MR [2168882](#)
- [13] L. Esposito, V. Ferone, B. Kawohl, C. Nitsch, and C. Trombetti, [The longest shortest fence and sharp Poincaré-Sobolev inequalities](#). *Arch. Ration. Mech. Anal.* **206** (2012), no. 3, 821–851 Zbl [1262.52001](#) MR [2989444](#)

- [14] L. C. Evans, [Convergence of an algorithm for mean curvature motion](#). *Indiana Univ. Math. J.* **42** (1993), no. 2, 533–557 Zbl 0802.65098 MR 1237058
- [15] P. Goodey, [Area and perimeter bisectors of planar convex sets](#). In *Integral geometry and convexity*, pp. 29–35, World Scientific, Hackensack, NJ, 2006 Zbl 1124.52003 MR 2240971
- [16] K. Ishii, [Optimal rate of convergence of the Bence-Merriman-Osher algorithm for motion by mean curvature](#). *SIAM J. Math. Anal.* **37** (2005), no. 3, 841–866 Zbl 1101.35006 MR 2191779
- [17] M. Jacobs, E. Merkurjev, and S. Esedoğlu, [Auction dynamics: a volume constrained MBO scheme](#). *J. Comput. Phys.* **354** (2018), 288–310 Zbl 1380.52018 MR 3738108
- [18] J. Lie, M. Lysaker, and X.-C. Tai, A piecewise constant level set framework. *Int. J. Numer. Anal. Model.* **2** (2005), no. 4, 422–438 Zbl 1125.49031 MR 2177631
- [19] H. Liu, J. Liu, R. H. Chan, and X.-C. Tai, [Double-well net for image segmentation](#). *Multiscale Model. Simul.* **22** (2024), no. 4, 1449–1477 Zbl 1556.68146 MR 4818409
- [20] H. Liu, X.-C. Tai, and R. Chan, [Connections between operator-splitting methods and deep neural networks with applications in image segmentation](#). *Ann. Appl. Math.* **39** (2023), no. 4, 406–428 Zbl 1549.68029 MR 4668938
- [21] E. Merkurjev, E. Bae, A. L. Bertozzi, and X.-C. Tai, [Global binary optimization on graphs for classification of high-dimensional data](#). *J. Math. Imaging Vision* **52** (2015), no. 3, 414–435 Zbl 1333.90111 MR 3345405
- [22] E. Merkurjev, T. Kostić, and A. L. Bertozzi, [An MBO scheme on graphs for classification and image processing](#). *SIAM J. Imaging Sci.* **6** (2013), no. 4, 1903–1930 Zbl 1279.68335 MR 3115457
- [23] B. Merriman, J. Bence, and S. Osher, Diffusion generated motion by mean curvature. Report no. 92-18, 1992, available in *UCLA CAM Rep.* <https://ww3.math.ucla.edu/camreport/cam92-18.pdf>, visited on 20 June 2024
- [24] B. Merriman, J. K. Bence, and S. J. Osher, [Motion of multiple functions: a level set approach](#). *J. Comput. Phys.* **112** (1994), no. 2, 334–363 Zbl 0805.65090 MR 1277282
- [25] M. Miranda, Jr., D. Pallara, F. Paronetto, and M. Preunkert, Short-time heat flow and functions of bounded variation in $\mathbb{R}^N$. *Ann. Fac. Sci. Toulouse Math. (6)* **16** (2007), no. 1, 125–145 Zbl 1142.35445 MR 2325595
- [26] F. Morgan, Convex body isoperimetric conjecture. Unpublished notes 2010, available at <https://sites.williams.edu/Morgan/2010/07/03/convex-body-isoperimetric-conjecture/>, visited on 1 June 2024.
- [27] S. Osher and J. A. Sethian, [Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations](#). *J. Comput. Phys.* **79** (1988), no. 1, 12–49 Zbl 0659.65132 MR 965860
- [28] G. Pólya, Aufgabe 283. *Elem. Math* **13** (1958), 40–41
- [29] K. Radziszewski, Sur les cordes qui partagent l’aire d’un ovale en 2 parties égales. *Ann. Univ. Mariae Curie-Skłodowska Sect. A* **8** (1954), 89–92 (1956) Zbl 0071.38005 MR 80940
- [30] S. J. Ruuth, [Efficient algorithms for diffusion-generated motion by mean curvature](#). *J. Comput. Phys.* **144** (1998), no. 2, 603–625 Zbl 0946.65093 MR 1638032
- [31] S. J. Ruuth, B. Merriman, J. Xin, and S. Osher, [Diffusion-generated motion by mean curvature for filaments](#). *J. Nonlinear Sci.* **11** (2001), no. 6, 473–493 Zbl 1037.35033 MR 1871280
- [32] S. J. Ruuth and B. T. R. Wetton, [A simple scheme for volume-preserving motion by mean curvature](#). *J. Sci. Comput.* **19** (2003), no. 1–3, 373–384 Zbl 1035.65097 MR 2028850

- [33] X.-C. Tai, O. Christiansen, P. Lin, and I. Skjælaaen, A remark on the MBO scheme and some piecewise constant level set methods. Report no. 05-24, 2005, available in *UCLA CAM Rep.*, <https://ww3.math.ucla.edu/camreport/cam05-24.pdf>, visited on 20 June 2024
- [34] X.-C. Tai, O. Christiansen, P. Lin, and I. Skjælaaen, [Image segmentation using some piecewise constant level set methods with MBO type of projection](#). *Int. J. Comput. Vis.* **73** (2007), 61–76 Zbl [1477.68521](#)
- [35] X.-C. Tai, H. Liu, and R. Chan, [PottsMGNet: a mathematical explanation of encoder-decoder based neural networks](#). *SIAM J. Imaging Sci.* **17** (2024), no. 1, 540–594 Zbl [1539.68322](#) MR [4713796](#)
- [36] B.-H. Wang and Y.-K. Wang, [A note on the convex body isoperimetric conjecture in the plane](#). *Proc. Amer. Math. Soc.* **152** (2024), no. 2, 801–813 Zbl [1533.53006](#) MR [4683859](#)
- [37] D. Wang, [An efficient iterative method for reconstructing surface from point clouds](#). *J. Sci. Comput.* **87** (2021), no. 1, article no. 38 Zbl [1471.49002](#) MR [4232848](#)
- [38] D. Wang, A. Cherkhev, and B. Osting, [Dynamics and stationary configurations of heterogeneous foams](#). *PLOS one* **14** (2019), no. 4, article no. e0215836
- [39] D. Wang, H. Li, X. Wei, and X.-P. Wang, [An efficient iterative thresholding method for image segmentation](#). *J. Comput. Phys.* **350** (2017), 657–667 Zbl [1380.65048](#) MR [3707176](#)
- [40] D. Wang and X.-P. Wang, [The iterative convolution–thresholding method \(ICTM\) for image segmentation](#). *Pattern Recognition* **130** (2022), article no. 108794
- [41] W. Wichiramala, Efficient cut for a subset of prescribed area. *Thai J. Math.* **5** (2007), no. 3, Special Issue, 95–100 Zbl [1221.49075](#) MR [2407479](#)
- [42] X. Xu, D. Wang, and X.-P. Wang, [An efficient threshold dynamics method for wetting on rough surfaces](#). *J. Comput. Phys.* **330** (2017), 510–528 Zbl [1378.76087](#) MR [3581477](#)

Received 2 November 2024; revised 1 August 2025.

Shilong Hu

School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), 2001 Longxiang Road, Longgang District, 518172 Shenzhen, P. R. China; shilonghu@link.cuhk.edu.cn

Hao Liu

Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong; haoliu@hkbu.edu.hk

Dong Wang

School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), 2001 Longxiang Road, Longgang District, 518172 Shenzhen; International Center for Industrial and Applied Mathematics, Shenzhen Research Institute of Big Data, 2001 Longxiang Avenue, Longgang District, 518172 Shenzhen; Shenzhen Loop Area Institute, No. 6 Hongmian Road, Futian Free Trade Zone, Shenzhen Hetao College, 518048 Shenzhen, P. R. China; wangdong@cuhk.edu.cn