A Discrete Global Minimization Algorithm for Continuous Variational Problems

Danil Kirsanov
and
Steven J. Gortler

TR-14-04

Computer Science Group
Harvard University
Cambridge, Massachusetts

A Discrete Global Minimization Algorithm for Continuous Variational Problems

Danil Kirsanov
Harvard University
kirsanov@fas.harvard.edu

Steven J. Gortler
Harvard University
sjg@eecs.harvard.edu

Abstract

In this paper, we apply the ideas from combinatorial optimization to find *globally* optimal solutions to continuous variational problems. At the heart of our method is an algorithm to solve for globally optimal discrete minimal surfaces. This discrete surface problem is a natural generalization of the planar-graph shortest path problem.

1 Introduction

We are given a variational problem to solve for the function $f(x) : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}$ that minimizes some functional

$$\int_{\Omega} \cdots \int G(f(x), \nabla f(x), x)\, dx. \tag{1.1}$$

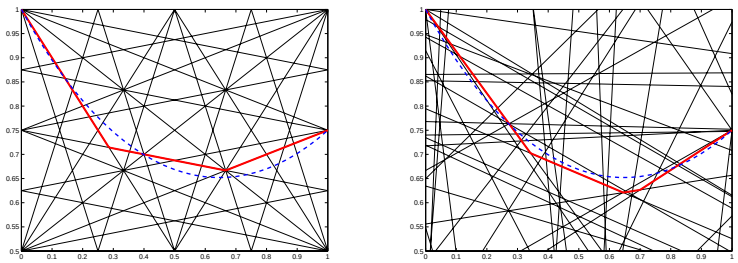
with boundary conditions $f(\partial\Omega) = \Gamma$.

The standard numerical approach is to discretize the domain Ω and then use floating point numbers to represent functions $\Omega \subset \mathbb{R}^n \mapsto \mathbb{R}$. Then, descent methods are used to find a local optimum. In this paper we study how combinatorial algorithms may be applied to these problems to find global minima. First we discretize the space $\Omega \times \mathbb{R}$. With this discretization, the optimization becomes completely combinatorial in nature. We then solve the resulting combinatorial optimization problem with a polynomial time discrete algorithm.

For example, suppose $\Omega = \mathbb{R}$, then a discretization of $\Omega \times \mathbb{R}$ might be describable as a planar graph (described with a set of vertices and edges) over $\mathbb{R}^2$. In this case, a two point boundary value problem reduces to the problem of finding the optimal path in a graph, and it is well known that one can solve for the global discrete minimal path using Dijkstra's algorithm [5] (see Figure 1).

Contribution In particular, in this paper we present the following contributions

- In order to solve the given continuous variational problem, we must be assured that, with enough discretization, the solution to the combinatorial problem will be close to the continuous one. Here we show conditions sufficient to ensure this, and we demonstrate that these conditions can be met with a sequence of deterministic and random grids we construct.
- We prove that, in arbitrary dimension, the resulting combinatorial optimization problem can be reduced to an instance of min-cut, over an appropriate dual graph. In 3D, for example, this gives us a simple algorithm for finding globally minimum discrete surfaces.
- Finally, we describe some important details about the implementation of our algorithm and we show some experimental results demonstrating our method's superiority to traditional numerical approaches.



(a) (b)

Figure 1: Solution for the curve that generates the "minimal surface of rotation" on the regular (a) and random (b) grids. Exact solution is shown as a dashed line.

1.1 Previous work

The literature on finding solutions to variational problems is extensive. For a detailed review, see [6, 8]. Most of the described numerical methods use some form of local descent and can only guarantee convergence to some local minimum. In contrast to all of these methods, our approach has global guarantees even in the presence of local minima.

In the literature, there are a few algorithms that are able to find global minima for certain classes of problems. Here we review and compare them along the following set of axes.

- Functional: Some of the methods can only solve for minimal *area* surfaces, where others can minimize energy measured by any first-order functional.
- Boundary condition: Some of the methods search for surfaces that have some proscribed boundary, while others look for closed surfaces that surround certain points.
- Topology: Some methods solve for surfaces in R^3, while others solve for scalar functions over the plane (height fields). Surface-based methods typically find surfaces of arbitrary topology. It should be noted that to correctly define the continuous minimal surface problem one needs to employ the mathematical machinery of "integral currents" [14].
- Discrete solution: All of these methods solve the continuous problem by discretizing space and solving a combinatorial optimization problem. They show that in the limit, the discrete problem produces a solution to the continuous problem. For some of the methods, the discrete algorithm outputs a specific piecewise linear surface, other methods only output a volumetric "slab" in which the solution must lie.
- Algorithm Some of the methods reduce to MIN-CUT which has the complexity of $O(N^2 \log N)$, while some reduce to circulation problems with higher complexity.
- Experimentation: Some of the methods are described in publications which include experimental results and implementation details.

Hu et al. describe a method that uses MIN-CUT to compute minimal *area* functions [11, 12]. In particular they find the minimal area a two-dimensional function $f : \Omega \in \mathbb{R}^2 \mapsto [a, b]$ that satisfies boundary conditions $f(\partial\Omega) = \Gamma$. The boundary curve must be "extremal", which means that it is on the exterior of volume $\Omega \times [a, b]$ in which the surface must lie. In their solution, they discretize space into N volumetric nodes. They then proceed to find the "slab" of nodes of some thickness $d \ll 1$ that satisfies the boundary conditions, separates 3D volume $\Omega \times [a, b]$ into two pieces and has minimal volume. The discretized version of the latter problem can be solved using a MIN-CUT algorithm over an appropriate graph with N nodes and $O(N^2 d^3)$ edges.

Algorithm	HKR [12]	BK [2]	Sullivan [17]	Our Method
Continuous functional	Euclidean	Riemannian	general	general
Boundary	external	closed surface	general	external
Boundary condition	exact or free	exact or free	exact	exact or free
Topology	function only	surface only	surface only	function and surface
Discrete solution	volume	volume	piece-wise linear surface	piece-wise linear surface
Method	min-cut	min-cut	min-circulation	min-cut
Experimental justification	no	yes	no	yes
Grids	regular	regular	regular	regular and random

Table 1: Comparative properties of the graph-cut functional minimization algorithms.

Boykov and Kolmogorov have introduced a similar but more sophisticated approach for solving for minimal surfaces using MIN-CUT [2] Similar to Hu, this algorithm has two major degrees of freedom: spatial discretization and connectivity. By cleverly selecting the weights for the edges in their graph they can solve for a wider class of functionals than simply "area". In particular, they allow one to choose an arbitrary Riemannian metric over space, and then can solve for the minimal area surface, where area is defined using this new metric. Unfortunately, if one is given a problem in the form of equation (1.1) where the function G depends non-quadratically on the surface normal direction, then their algorithm is not applicable.

In their paper, they solve for closed surfaces; to avoid surface collapse, they also impose an energy term which measures which points are inside and outside the closed surface. It seems likely that their approach could be generalized to handle extremal boundary curves like Hu et al.

In both the above algorithms, MIN-CUT is used to divide the volumetric nodes of R^3 into slabs. They show that in the limit, the actual minimal solution lies somewhere in a shrinking volumetric "slab" determined by the cut. These algorithms do not compute a specific discrete surface that minimizes the functional over some discrete space.

In contrast to the above methods, our method is applicable to any first order functional. Our discrete algorithm outputs a specific minimal discrete surface, and can thus be seen as a generalization of the "shortest path problem over a planar graph" to a "smallest surface problem in a spatial CW complex".

The closest work to ours is the algorithm described in the Ph.D. thesis of John Sullivan [17]. More generally than the above methods, Sullivan solves for the surface that minimizes any arbitrary first order functional. His discrete algorithm outputs a specific discrete surface. In addition he allows his boundary to be any set of loops, including non-extremal loops and knots. Because of his general treatment of boundary conditions his algorithm reduces not to MIN-CUT but to a more complicated network circulation problem. He then solves this problem using an algorithm with time complexity $O(N^2 A \log N)$, where N is the number of nodes in his discrete graph, and A is the complexity of the solution surface. It is unclear if his approach was ever implemented or compared with numerical methods. In addition, due to the amount of difficult machinery employed, this work has remained somewhat inaccessible and unfortunately unappreciated outside of the minimal surface community.

In contrast to the work of Sullivan, by only allowing extremal boundary conditions, we are able to apply a very simple reduction to obtain an instance of MIN-CUT with time complexity $O(N^2 \log N)$. We also describe specific constraints so that we can solve for minimal functions, which is the focus of this paper, though removal of these constraints would let us solve for minimal surfaces of arbitrary topology as well. In our work, we use random grids for finding approximate solutions which allows us to satisfy complicated boundary conditions and to employ practical local grid refinement techniques. Finally, we describe the details of an implementation and demonstrate how it compares in practice to standard numerical approaches.

To summarize, we compare the relative properties of these approaches in table (1).

Our research was specifically inspired by recent success of the combinatorial methods in the discrete energy minimization problems emerged from computer vision [4], [3]. Though the problems that arise in this area are completely discrete from the very beginning, their method of construction of the directed dual graph can be adopted for our purpose.

Duality between the shortest path and min-cut problem for both directed and undirected planar graphs is well known and described in the literature for different applications (see [10] pg.156, [13]). As we will see, this duality can be generalized to higher dimension, and applied to solve variational problems.

2 Problem definition

Our problem is to find the function $f(x) : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}$ that minimizes some functional

$$F(f) = \int_{\Omega} \cdots \int G(f(x), \nabla f(x), x)\, dx. \tag{2.1}$$

It is common to apply the strict boundary conditions

$$f(\partial\Omega) = \Gamma(\partial\Omega), \tag{2.2}$$

but our method will also work with the less restrictive conditions

$$\Gamma_0(\partial\Omega) \leq f(\partial\Omega) \leq \Gamma_1(\partial\Omega). \tag{2.3}$$

We require that the domain Ω is a simply-connected polytope, with boundary $\partial\Omega$ homeomorphic to S^{n-1}. (For $n = 1$ this says that the $\partial\Omega$ is two points. And for $n = 2$, $\partial\Omega$ is the boundary of a simple polygon with no holes).

In addition, we require our boundary condition(s) Γ to be described as a piece-wise linear function with a finite number of pieces.

We require that the solution of the variational problem $f_s(x)$ belongs to a function space $\mathcal{P}$ of continuous bounded piece-wise twice differentiable functions with bounded first and second derivatives [1]. By "piece-wise" we mean that if $g \in \mathcal{P}$ then there exists a subdivision of the domain Ω into a finite set of non-intersecting polytopes $\Omega = \bigcup \varpi_i^g$, such that on the internal points of every polytope ϖ_i^g the function g is a continuous twice differentiable function with bounded first and second derivatives. Without the loss of generality, we can scale the problem to make sure that

$$0 \leq g(x) \leq 1, \quad \left|\frac{\partial g}{\partial x_i}(x)\right| \leq C_1^g, \quad \left|\frac{\partial^2 g}{\partial x_i \partial x_j}(x)\right| \leq C_2^g. \tag{2.4}$$

We also require the kernel of the functional (2.1) to be nonnegative [2]. In other words, for all x, y such that $x \in \Omega$, $0 \leq y \leq 1$, and n-dimensional vectors $p \in \mathbb{R}^n$ we have

$$G(y, p, x) \geq 0. \tag{2.5}$$

This ensures that the weights of all edges of graphs that we construct will be nonnegative [3].

[1] Though we will strongly rely on it in the proofs of the correctness of our method, we believe that this assumptions can be weakened in many cases.

[2] We can weaken the condition (2.5), requiring the kernel to be limited from below. Indeed, we know that if the function $f_s(x)$ is a solution of the variational problem with the kernel $G(f, \nabla f, x)$ then it is also a solution of a problem with the kernel $G(f, \nabla f, x) + C$ for any constant C. Therefore, if there exists a constant C such that for all x, y, and p in (2.5) we have $G(y, p, x) \geq -C$, then the problem with the kernel $G(f, \nabla f, x) + C$ will satisfy (2.5) and have the same solution $f_s(x)$.

[3] This is not a necessary condition for the shortest path algorithms in 1D, but will be required in higher dimensions.

3 Grids

Our basic approach is to approximate the continuous variational problem with a discretized combinatorial one which we can globally solve in polynomial time. For this scheme to work, it is essential that the discrete solution gives us meaningful information about the continuous minimum. Here we present sufficient conditions for such an approach to succeed. Then we describe a grid structure that satisfies these conditions.

Sufficient conditions. Let us assume that our variational problem is to be solved over a space of possible functions $\mathcal{P}$ with a norm $\|\cdot\|$ in it. Our discretized version of the problem will only allow for the representation of some other set $\mathcal{L} \subset \mathcal{P}$. The following lemma states the sufficient conditions to guarantee that our discretized solution will be "close enough" to the actual solution.

Lemma 3.1. *Let $\mathcal{P}$ be a function space, and $\|\cdot\|$ a norm over $\mathcal{P}$. Let $F(f)$ be a functional that is continuous in $\mathcal{P}$ with respect to $\|\cdot\|$. Suppose that $\mathcal{L} \subset \mathcal{P}$ is dense in $\mathcal{P}$ with respect to the norm. Then for any $\bar{f} \in \mathcal{P}$, any $\varepsilon > 0$, any $\delta > 0$ there exists $g \in \mathcal{L}$ such that $\left\|g - \bar{f}\right\| < \delta$ and $\left|F(g) - F(\bar{f})\right| < \varepsilon$.*

Proof. Due to the density we can pick a sequence $\{u_n\}_{n=0,1,\ldots,\infty} \subset \mathcal{L}$ such that $\lim_{n\to\infty} \left\|u_n - \bar{f}\right\| = 0$. In particular, there exists an integer N_1 such that for all $n > N_1$ we have $\left\|u_n - \bar{f}\right\| < \delta$.

Because the functional is continuous, there also exists an integer N_2 such that $\left|F(u_n) - F(\bar{f})\right| < \varepsilon$ for all $n > N_2$. Therefore, for any $n > \max(N_1, N_2)$ function $g = u_n$ satisfies the requirements of the lemma. □

Corollary 3.2. *Let us denote $F_{min} = \inf_{f\in\mathcal{P}} F(f)$. If $\mathcal{L}$ is dense in $\mathcal{P}$ and $\mathcal{P}$ is dense in $\mathcal{L}$, then $F(g) \geq F_{min}$ for any $g \in L$, and for any $\varepsilon > 0$ there exists $g \in \mathcal{L}$ such that $|F(g) - F_{min}| < \varepsilon$.*

So, if the functional is continuous and we chose a set of the functions $\mathcal{L}$ that satisfies the conditions of the corollary, we can solve the problem over this set $\mathcal{L}$ and our the energy of our result will be as close to the minimal energy of the original problem as we want. In most cases we can also guarantee that the solution of the discrete problem is close to the solution of the continuous problem under the norm $\|\cdot\|$.

Definition 3.3. Let f_{min} be the global minimum, $F(f_{min}) = F_{min}$. It is called an *isolated global minimum* if for any $\varepsilon > 0$ there exists $\delta > 0$ such that if $\|f - f_{min}\| > \varepsilon$, then $F(f) - F_{min} > \delta$.

Corollary 3.4. *If there exists an isolated global minimum f_{min}, then for any $\varepsilon, \delta > 0$, there exists $g \in \mathcal{L}$ such that $\|g - f_{min}\| < \delta$ and $|F(g) - F_{min}| < \varepsilon$.*

In other words, under very general assumptions we can find "good enough" solution in our numerical subset $\mathcal{L}$, which can be much simpler then $\mathcal{P}$. In the next sections we are going to construct such a subset.

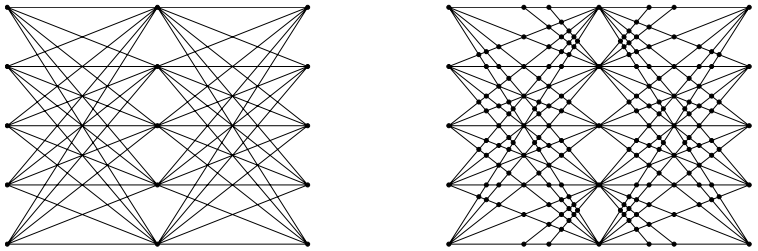
3.1 Univariate variational problem

Here we describe grids for the univariate minimization problems that satisfy conditions (2.1 - 2.5). Though many constructions are possible, we have chosen this one because it is easily generalizable to arbitrary dimension.

We assume w.l.o.g. that $\Omega = [0, 1]$ and therefore the solution of the univariate variational problem lies in the unit square of the plane $\Phi = \{(x, f) \mid x \in [0, 1], f \in [0, 1]\}$. Define the space $\mathcal{P}$ of the allowable solutions as a set of continuous finite piece-wise twice differentiable functions $f : [0, 1] \mapsto [0, 1]$ with bounded first and second derivatives. We define the integral norm to be

$$\|f\| = \int_0^1 |f(x)| + \left|f'(x)\right| \, dx. \tag{3.1}$$

This norm includes a term for both the value *and* the derivative of the f. As a result, our functional F, which depends on both value and derivative, is continuous with respect to this norm. Essentially, this



(a) (b)

Figure 2: (a) The graph C_2 can be used to obtain a combinatorial problem that is arbitrarily close to the continuous one. (b) The planar graph C_2^{pl} is constructed by intersection of the set of lines.

means that two "close" functions must have similar energy measurements. This continuity is needed in the preconditions of Lemma 3.1.

Our strategy is to piecewise linearly embed a planar graph in this square, and then represent possible functions $g(x)$ as piecewise linear paths in this embedded graph. Every path in the graph with monotonically increasing x, from 0 to, 1 corresponds to a piecewise linear function $g : [0, 1] \mapsto [0, 1]$. The set of all possible piecewise linear functions generated by the grid R we denote as $\mathcal{M}(R)$. For a graph with a finite number of vertices the set $\mathcal{M}(R)$ is also finite. Obviously, $\mathcal{M}(R) \subset \mathcal{P}$.

Regular grids. For univariate minimization problems, we will use a grid such as one shown in Figure (2a). This grid, C_n, has $n+1$ columns and n^2+1 rows of vertices. The distance between the columns is therefore equal to $h = n^{-1}$ and the distance between the rows is $d = h^2 = n^{-2}$. Edges are included between each pair of vertices in adjacent columns. This graph has about n^3 vertices and n^5 edges. The number of the possible functions generated by this grid $|\mathcal{M}(C_n)| = n^{(2n+2)}$. Fortunately, we will be able to find the global minimum of these possible functions with efficient combinatorial minimization techniques. This grid has been chosen so that it can represent functions with a wide variety of values *and* derivatives. In particular we can state the following:

Theorem 3.5. *The set* $\mathcal{L} = \bigcup_{n=1}^{\infty} \mathcal{M}(C_n)$ *is dense in* $\mathcal{P}$ *with respect to our norm.*

The proof of the theorem is given in the appendix A.

Combined with the corollary (3.4), this theorems states that if the continuous problem has an isolated minimal solution, then it is possible to solve the variational problems on the grids C_n, and the discrete solution will converges to the actual solution of the continuous problem when $n \to \infty$. Note, that our definition of the grid does not depend on the variational problem. In fact, any continuous variational problem that has a solution in the set $\mathcal{P}$ can be solved on this grid.

Unfortunately, the grid C_n is not planar. As described below, for higher dimensional problems, we will need to have an embeddable grid. Fortunately, we can easily modify the graph to obtain the planar graph C_n^{pl} as shown in Figure(b).

Random grids. The described regular grid structure can be created as the intersection of sets of lines. As n increases, the lines become more "dense", meaning that in the vicinity of every point of the plane there pass more and more lines with different positions and orientations. Random grids have the same property, and thus can be used for solving our continuous variational problems.

The random grid R_n in the unit square of the plane Φ is created by intersection of n random lines. One vertex is created at each line intersection, and the resulting graph is planar. Each line is constructed by taking the point $p \in \Phi$ and orientation $-\pi \le c \le \pi$ from some probability distributions [4]. The grid R_{n+1}

[4]In this paper we will consider uniform distributions unless the different distribution is mentioned.

can be constructed by adding one random line to R_n. Such families of the grids has a nice property that $\mathcal{M}(R_{n+1}) \supset \mathcal{M}(R_n)$. The set of all functions generated by the family we will denote as $\mathcal{L}_r = \bigcup_{n=1}^{\infty} \mathcal{M}(R_n)$.

Similar to the regular case, we can state the density theorem

Theorem 3.6. *The set $\mathcal{L}_r$ is dense in $\mathcal{P}$.*

The proof of the theorem is given in the appendix B.

Therefore, we can solve the variational problems in the domain of the random grids.

3.1.1 Regular versus random

We have shown that univariate variational problems can be numerically solved in the domains of the regular and random grids. What domain should be chosen depends on the type of the problem. Here we emphasize the major tradeoffs between these approaches.

It can be clearly seen that regular grids C_n are based on the simple blocks that can be pre-computed for all reasonable n and stored in data files. The graph for the given problem can be constructed by joining the blocks together. Such a structure gives us another advantage when the kernel of the functional $F(f, f', x)$ does not depend on x and the computation of the edge weight is expensive. In this case we have to compute the weights for the small fraction of the edges and then simply copy these values for the whole structure.

Random grids are more flexible. We can experiment with the density distributions of the lines, making them more "dense" in the vicinity of the possible solution. In order to produce an "average" answer and "deviation" we can run the algorithm several times on different random grids. These advantages are often dominant because for big meshes the time for the grid construction and weight computing is usually small comparing to the other parts of the algorithm. In our implementation we pursued the use of random grids.

The solution for the function generating a minimum surface of rotation using both types of grids can be seen on the Figure (1).

3.2 Grids for higher dimensions

The described grid structure can be created as the intersection of the set of lines. The generalization in higher dimensions is straightforward; the grid is created as the intersection of the set of hyperplanes in the 3D space, $\Omega \times [0, 1]$. With this grid structure, some function space of piecewise linear functions over Ω can be represented using appropriate subsets of facets. The proof of the correctness of these constructions for the higher dimensions is omitted for the sake of space.

Though regular grids have a clear structure and might be easier to build and investigate, random grids are more flexible. We can experiment with the density distributions for the hyperplanes, making them more "dense" in the vicinity of the possible solution. It is also much easier to construct random grids that fit the given boundary conditions. In our experiments we strongly rely on random grids.

3.3 From grids to discrete problem.

The weight of the face can be computed by evaluating the integral (2.1) numerically or analytically. For instance, for bivariate variational problem every face of the grid C_n is a convex polygon located on some plane $z(x, y) = Ax + By + C$ with the constant gradient $\nabla z = (A, B)^T$. The cost of the face can be computed as

$$w_i = \iint_{face} G\left(Ax + By + C,\ \ (A, B)^T,\ \ x, y\right)\, dx\, dy. \tag{3.2}$$

4 Algorithms

We now have approximated our continuous variational problem by a series of discrete optimization problems. In the univariate case, the discrete problem is a shortest path problem, and in the bivariate case is a minimal discrete surface problem. We now wish to find the global optima for these problems. For the shortest path problem, we could simply use a standard shortest path algorithm, such as Dijkstra's algorithm on the grid. Unfortunately it is not clear how to use these algorithms to solve for minimal surfaces.

In this section we show how both the discrete path and surface problems may be globally solved in polynomial time. The basic approach will be to generate an appropriate dual graph from the grid structure. This dual graph will be constructed so that "cuts" of the graph correspond to paths (surfaces) in the original grid structure. We can then apply well known algorithms to solve for the min-cut of the dual graph. For simplicity, we first demonstrate how these ideas work for the univariate problem. The topological machinery we use is possibly a tad bit heavy for the univariate case, but will be sufficient to generalize to bivariate case. In fact, this machinery can be applied to variational problems of any dimension.

4.1 Univariate problem

For the discrete univariate problem, we are given a planar graph (V, E) that is embedded in $\Re^2$, with positive weights on each of its edges. We will label the axes of $\Re^2$ as x and y. Because the graph is embedded, we have not only vertices and edges, but also 2D faces. This collection of vertices edges and faces defines a complex (formally it is a CW complex [9]) that is homeomorphic to the 2D ball (disk) B^2.

We wish to compute the minimum weight path that connects two specified vertices that are on the exterior of this complex. In addition we will want to ensure that the our path corresponds to a function in the original variational setting; that is, there should be one y value over each x value. We will deal with this restriction later on in section (4.1.1).

The two boundary vertices A and B partition the exterior edges into two sets which we call upper U and lower L. We introduce two additional faces, f_r and f_k, in the graph. The source face f_r is surrounded by edges in L and one additional outer edge e_r. The sink face f_k is constructed by edges in U and one additional outer face e_k. With these two added faces and edges, the new complex is still homeomorphic to B^2 (see Figure 3).

The following are simple definitions from *simplicial homology modulo 2* [7].

Definition 4.1. A *1-chain* in the graph is a union of some edges of the graph $\mathcal{C} = \cup e_i$. The *boundary* of a 1-chain $\mathcal{C}$, which is denoted as $\partial_1 \mathcal{C}$, is a union of vertices that are included in $\mathcal{C}$ an *odd* number of times. Similarly, a *2-chain* is a union of some faces of the graph $\mathcal{A} = \cup f_i$. The *boundary* $\partial_2 \mathcal{A}$ of a 2-chain $\mathcal{A}$ is a union of edges that are included in $\mathcal{A}$ an odd number of times.

For our problem, we are interested only in 1-chains that satisfy boundary conditions $\Gamma = \{A, B\}$.

Definition 4.2. Let us define a *valid curve* on the graph described above to be any 1-chain $\mathcal{C}$ with $e_r \notin \mathcal{C}$ and $e_k \notin \mathcal{C}$ such that $\partial_1 \mathcal{C} = \Gamma$. The *cost* of a valid curve is a sum of the weights of its edges.

The *minimal valid curve problem* is to find a valid curve with the minimal cost.

As it is shown in Appendix C, this is easily accomplished using a dual graph D (Figure 3,c). That is, we associate with each face f_i a dual vertex $\hat{v}_i$. We also associate a single source vertex $\hat{r}$ with a source face f_r and a single sink vertex $\hat{k}$ with a sink face f_r. With each edge e_i bounding two faces f_j and f_k, we associate a dual edge $\hat{e}_i$ that connects $\hat{v}_j$ and $\hat{v}_k$. We set the *capacity* of each dual edge the cost of the associated primal edge $c(\hat{e}_i) = c(e_i)$.

Definition 4.3. A *cut* is a partition of the vertices $\hat{v}$ into two sets $\hat{R}$ and $\hat{K}$ with $\hat{r} \in \hat{R}$ and $\hat{k} \in \hat{K}$. The *cost* of a cut $c(\hat{R}, \hat{K})$ is the sum of the capacities of the edges between $\hat{R}$ and $\hat{K}$.

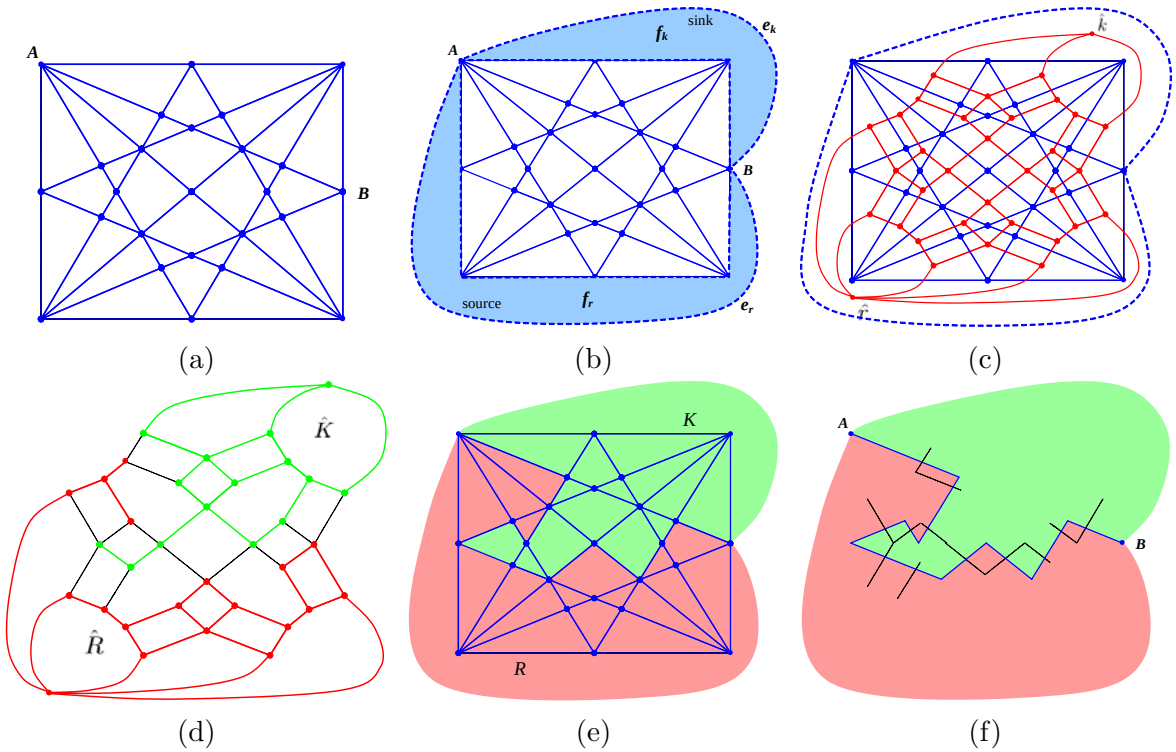


Figure 3: Duality for the minimum valid curve problem. (a) A primal graph with boundary conditions. (b) Source and sink faces and edges are added to the primal graph. (c) Dual graph is shown in red. (d) The minimal cut of the dual graph. The edges across the cut are shown in black. (e) A cut of the dual graph correspond to a partition of the primal graph. (f) The boundary of the (R, K) partition is a minimum valid curve on the primal graph. Every edge of the curve corresponds to an edge across the dual cut.

Because there is a one-to-one correspondence between the vertices of the dual graph and the faces of the primal graph, we can also denote the cut as a partition of the faces of the primal graph into two 2-chains R and K with $f_r \in R$, $f_k \in K$.

Theorem 4.4. *There is a one-to-one correspondence between the valid curves on the primal graph and cuts on the dual graph. The minimal valid curve problem over the primal undirected graph is equivalent to a minimal cut problem for the dual graph.*

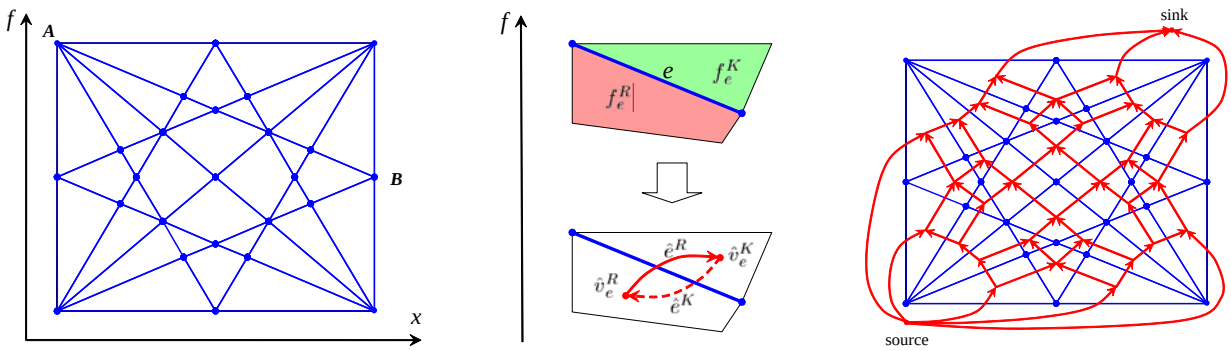
The proof of the theorem is given in the appendix C.
The minimal cut is a well-known problem that can be solved in polynomial time.

4.1.1 Additional constraints

In the previous section we considered a general problem of finding a minimal valid curve in G. Here we describe how to solve its restricted version where we want to find a minimal curve with some special properties. Our final purpose is to ensure that the chosen curve is a *function*.

Theorem (4.4) states that every valid curve $\mathcal{C}$ corresponds to a cut that is denoted as $(R_\mathcal{C}, K_\mathcal{C})$ in the primal graph and $(\hat{R}_\mathcal{C}, \hat{K}_\mathcal{C})$ in the dual graph. If the curve passes through an edge e, one of its adjacent faces belongs to the source 2-chain $R_\mathcal{C}$ and the other one belongs to the sink 2-chain $K_\mathcal{C}$. We specify an additional constraint to be satisfied when the curve passes through this edge.

Constraint 4.5. Let e to be an edge of the primal graph. We take one of its adjacent faces and denote it as f_e^R, the other one is denoted as f_e^K. We also denote corresponding vertices in the dual graph as $\hat{v}_e^R$ and $\hat{v}_e^K$. If $e \in \mathcal{C}$, we require that $f_e^R \in R_\mathcal{C}$ and $f_e^K \in K_\mathcal{C}$ (similarly, $\hat{v}_e^R \in \hat{R}_\mathcal{C}$, $\hat{v}_e^K \in \hat{K}_\mathcal{C}$).



(a) (b) (c)

Figure 4: Construction of the dual directed graph. (a) We want the minimal valid curve to be a function $f(x)$. (b) Constraints. If an edge e (top) belongs to the solution, the adjacent upper face belongs to a sink part of the cut and the adjacent lower face belongs to a source part of the cut. We enforce this condition by introducing two directed edges in the dual graph (bottom). The edge with the infinite weight is dashed. (c) The resulting directed dual graph (edges with infinite weights are not shown). Compare it with fig. 3c.

This condition can be specified independently for every edge (and it is quite possible that some face f belongs to a source 2-chain for one edge and belongs to the sink 2-chain for another edge). Note, that the constraints do not depend on the particular curve $\mathcal{C}$ and simply narrow the set of possible valid curves on the primal graph.

In order to satisfy constraints (4.5), instead of undirected dual graph, we create the *directed* dual graph $\hat{G}$. If the constraint (4.5) is specified for the primal edge e, then the corresponding edge $\hat{e}$ of the undirected dual graph is split into two directed edges $\hat{e}^R$ (from $\hat{v}_e^R$ to $\hat{v}_e^K$) and $\hat{e}^K$ (from $\hat{v}_e^K$ to $\hat{v}_e^R$). The capacities of the dual edges are defined as $c(\hat{e}^R) = c(e)$ and $c(\hat{e}^K) = \infty$. If the constraint is not specified for e, then the capacities are set as $c(\hat{e}^R) = c(\hat{e}^K) = c(e)$.

Definition 4.6. The cost of a cut $c(\hat{R}, \hat{K})$ in $\hat{G}$ is the sum of the capacities of the edges *from* $\hat{R}$ *to* $\hat{K}$.

Theorem 4.7. *Let $\mathcal{C}$ be a valid curve on the primal graph with constraint (4.5) defined for every edge, $(\hat{R}, \hat{K})$ be a corresponding cut of the dual directed graph. The curve $\mathcal{C}$ satisfies the constraints if and only if the cost of the cut is finite, $c(\hat{R}, \hat{K}) < \infty$. Therefore, min-cut of the dual directed graph corresponds to the curve that has the minimal weight and satisfies all of the constraints (4.5).*

Proof. There is a one-to-one correspondence between valid curves and cuts.

If $\mathcal{C}$ satisfies the constraints (4.5), the weight of the corresponding cut is finite because all the edges from $\hat{R}$ to $\hat{K}$ are finite by construction.

If the cut has a finite weight, every edge $\hat{e}$ from $\hat{R}$ to $\hat{K}$ is finite. Therefore, every edge e of the corresponding curve satisfies the condition, otherwise $c(\hat{e})$ would be infinite. □

4.1.2 Duality for minimum functions

Thus far, the particular embedding of the planar primal graph has not been important. In our application, the primal graph represents a grid which has been used to discretize the univariate variational problem (section 3.1). In particular, this primal graph is embedded in the rectangular region $[0, 1] \times [0, 1]$ and every vertex graph has two coordinates $v_i = (x_i, y_i)$. One of the boundary vertices has its first coordinate equal to 0, the other one has its first coordinate equal to 1. The edges of the primal graph (except for $e_{\hat{r}}$ and $e_{\hat{k}}$) are straight line segments.

Only some valid curves on this graph will correspond to (piece-wise linear) functions $f : [0, 1] \mapsto [0, 1]$. We wish to add our additional constraints in such a way as to restrict our solution to this subset.

One can think of every piece-wise linear function f as dividing $[0,1] \times [0,1]$ into two regions, one of them is "below" the function and the other, above. In our cut solution, we want the region below the function to be the source-connected 2-chain R_f.

Therefore, for every edge e of the primal graph we introduce an additional condition. If the edge is belongs to the output function f, we require the face below the edge to belong to R_f and the edge above f to belong to K_f. This is exactly a constraint (4.5) described in the previous section.

Formally, the additional constraints are defined as follows. Every primal edge e_i is a segment of a line $y = a_i x + b_i$ [5]. The adjacent face in the half-plane $y < a_i x + b_i$ is denoted as $f^R_{e_i}$, the face in the half-plane $y > a_i x + b_i$ is denoted as $f^K_{e_i}$. The constraint for e_i is defined as (4.5). The construction of the directed dual graph for these constraints is described in (4.1.1). Example of the dual graph can be seen on the figure (4).

Theorem 4.8. *The curve $\mathcal{C}$ is a function if and only if for every edge it satisfies the constraint defined above.*

Proof. If the curve is a function, it satisfies the constraints by definition. Indeed, all the faces below the function are in the source 2-chain, all the faces above the function are in the sink 2-chain.

Let us now prove that if a curve satisfies the constraints above, it is a function, i.e. the curve $\mathcal{C}$ defines a unique mapping $f : \Omega \mapsto [0,1]$. Let's prove it by contradiction. If the mapping is not unique in some point $x \in \Omega$, there exist two edges $e_1, e_2 \in \mathcal{C}$ and two points on these edges with the same first coordinate $(x, y_1) \in e_1$, $(x, y_2) \in e_2$. Without loss of generality we can assume that $y_1 < y_2$ and there are no other edge $e_3 \in \mathcal{C}$ and $(x, y_3) \in e_3$, such that $y_1 < y_3 < y_2$.

By definition the face $f^K_{e_1}$ is located above the edge e_1 and belongs to a sink 2-chain $K_{\mathcal{C}}$. Similarly, the face $f^R_{e_2}$ is located below the edge e_2 and belongs to a sink 2-chain $R_{\mathcal{C}}$. Let us consider the set F of all faces intersected by an open segment $((x, y_1), (x, y_2))$. On the one hand, this segment does not intersect any edges of $\mathcal{C}$ and therefore either $F \subset R_{\mathcal{C}}$ or $F \subset K_{\mathcal{C}}$. On the other hand, $f^K_{e_1} \in F$ and $f^R_{e_2} \in F$, which is a contradiction. Therefore, the mapping is unique. □

Corollary 4.9. *The minimal function problem on the grid can be reduced to the minimal cut problem for the directed dual graph defined above.*

4.1.3 Relaxed boundary conditions

So far we considered the strict boundary conditions (2.2); in the univariate case it can be stated as $f(0) = A$, $f(1) = B$. In fact, by a very easy modification we can allow the relaxed boundary conditions (2.3); in the univariate case it can be stated as $A_1 \le f(0) \le A_2$, $B_1 \le f(1) \le B_2$.

To enforce these constraints (see figure 5), we remove from the dual graph all edges (shown in dotted red) that are dual to exterior edges (shown in green) that are within the region allowed by the relaxed boundary condition.

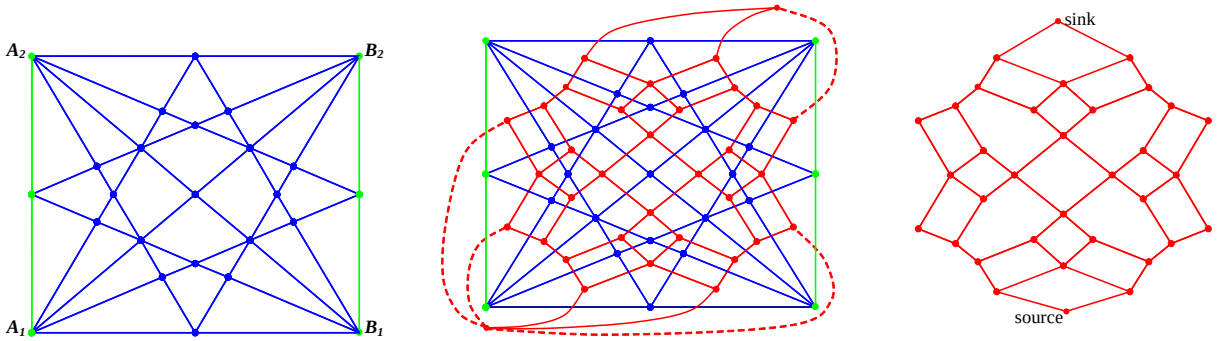
The removal of these dual edges is equivalent to assigning a zero cost to their associated primal edges. This allows the boundary of the function to "slide" up and down in the allowed region with no added cost. It can be easily seen that after this modification there is a one-to-one correspondence between functions that satisfy the relaxed boundary conditions and cuts in the dual graphs.

4.2 Minimal cost valid surface

The generalization of the shortest path problem to higher dimensions is straightforward[6]. Here we show its application in 3D. In the discrete spatial minimal cost surface problem a three-dimensional spatial complex

[5] We can avoid the vertical edges in two ways. The first one is to "shake" all vertices of the primal graph by some small random value to make sure that there are no vertical edges in the graph. The second one is to assign infinite weight with both of the dual edges.

[6] Unlike the minimal curve problem, which can be solved efficiently even for non-planar graphs, the minimal cost surface problem can only be solved efficiently for a complex that is embedded in three dimensional space. Without this restriction, the minimal cost two dimensional function problem becomes NP-Complete by reduction to min-cost planar triangulation [15].



(a) (b) (c)

Figure 5: Enforcing relaxed boundary conditions. (a) Exterior edges that are within the region allowed by the relaxed boundary condition are shown in green. (b) In the dual graph, the edges that intersect the green edges are removed (removed edges are dashed). (c) Resulting dual graph.

G is homeomorphic to B^3 and consists of vertices v_i, edges e_i, faces f_i, and cells c_i. Associated with each face f_i is a cost w_i. Also input is a closed boundary 1-chain Γ, which is on the exterior of the complex and which is homeomorphic to a circle S^1.

The boundary Γ partition the exterior faces of the complex into two sets which we call upper U and lower L. These sets are both disks, due to the Jordan Curve Theorem. Following the recipe of section (4.1), we attach a source and a sink cell c_r and c_k to the complex. The source cell c_r is bounded by the faces of L and one additional outer face f_r. The sink cell c_k is bounded by the faces of U and one additional outer face f_k. This new complex is still homeomorphic to the ball B^3.

For our problem, we are interested only in 2-chains that satisfy the boundary condition Γ.

Definition 4.10. A *valid surface* in the spatial complex is any 2-chain $\mathcal{A}$ with $f_r \notin \mathcal{A}$ and $f_k \notin \mathcal{A}$ such that $\partial_2 \mathcal{A} = \Gamma$.

Note that under this definition, our "surface" can include handles, have disconnected components, or even be non-manifold.

The *minimal valid surface problem* is to find a valid surface with minimal cost.

Omitting the intermediate steps that were described in details for the univariate case, we construct the directed dual graph as follows. That is, we associate with each cell c_i a dual vertex $\hat{v}_i$. A source cell c_r corresponds to a source dual vertex $\hat{r}$, a sink cell c_k corresponds to a sink dual vertex $\hat{k}$. Each non-exterior face f_i is a segment of a plane $z = a_i x + b_i y + d_i$ and has two adjacent cells. One of them, $c^r_{f_i}$, is below the plane, the other one, $c^k_{f_i}$ is above the plane. In the dual directed graph, with each face f_i we associate two dual edges. One of them, $\hat{e}^r_i$, goes from $\hat{v}^r_{f_i}$ to $\hat{v}^k_{f_i}$, the other one, $\hat{e}^k_i$, goes the opposite direction. The capacities of the directed edges are set as $c(\hat{e}^r_i) = w(f)$, $c(\hat{e}^k_i) = \infty$.

Theorem 4.11. *The minimal function problem on the grid can be reduced to the minimal cut problem for the directed dual graph defined above.*

The proof of this theorem is identical to that of the univariate case, except that is based on 3-chains and their properties in a complex homeomorphic to B^3.

For the relaxed boundary case, we have and upper and lower boundary curve, Γ_1 and Γ_2. In the graph construction we remove from the dual graph all edges that are dual to exterior primal *faces* that are within the region allowed by the relaxed boundary condition.

5 Implementation and results

We represent our spatial structure using a BSP-tree data structure. In this tree, the root node is associated with the volume of the entire bounding 3D cube. Random planes entered into the tree one by one. As each

plane is entered, each node whose associated volume is intersected by the plane is split into two "child" nodes representing the associated halves of the parent node's volume. At the end of this construction, each leaf of the tree represents a cell in the spatial complex. The nice property of this representation is that the leaves that are intersected by a hyperplane can be located hierarchically, because the hyperplane intersects the child cell in the binary tree only if it intersects the parent cell.

Integrals over each of the faces (3.2) are computed analytically or evaluated numerically by Monte-Carlo methods. We also used the elegant implementation of the min-cut algorithm written by Kolmogorov [1].

For problems with a fixed boundary condition Γ we force a small number of the planes (usually, less than 10% of the total number of the planes) to go through the piece-wise linear segments of the boundary constraints. In other words, each segment of the boundary becomes a pencil of planes; the orientations of the planes are chosen randomly.

As we have proven above, as the number of the planes goes to infinity, the discrete surface will approach the continuous minimal surface. Unfortunately, our computing constraints currently limit us to the use of only few hundreds of planes. In order to deal practically with these limitations we have explored the use of a number of heuristic-extensions.

Relaxation. One obvious extension which we call *relaxation* is to use the output of the our combinatorial optimization as the starting guess for a numerical iterative method. Because the numerical method only has to discretize the 2D domain, it can be discretized to a much higher resolution. The goal of this step is to refine the answer to a high-resolution, highly accurate local minima that is nearby to the combinatorial global optimum. For our numerical method, we employed a multi-resolution gradient descent method. The multi-resolution hierarchy was made up of three grids with sizes 20x20, 40x40, and 80x80. A finite differencing approach was used to derive the discrete optimization problem.

Plane selection. Because we use random grids, a natural way to use the method is to run the discrete algorithm several times and to pick up the solution with the smallest energy. We can use this idea to develop the following extension, which we call *plane selection*. With this extension we run our discrete algorithm several times for a single problem. After each iteration, we keep the planes that have been included in the current minimum, drop the planes that are not included in the minimum, and replace these with new random planes. At each iteration, the energy must necessarily not increase. Of course, plane selection can be coupled with relaxation.

Adaptive Refinement The last extension, to adaptively refine the grid by only subdividing cells that lie near to the current solution. More specifically, we run the algorithm a number of times and compute an average solution as well as deviation from this average. From these data we construct an upper and lower boundary ("envelope") of the desired answer. In the succeeding iterations we add more random planes, but all the cells above (resp. below) the envelope are merged with the source (resp. sink).

5.1 Simple examples

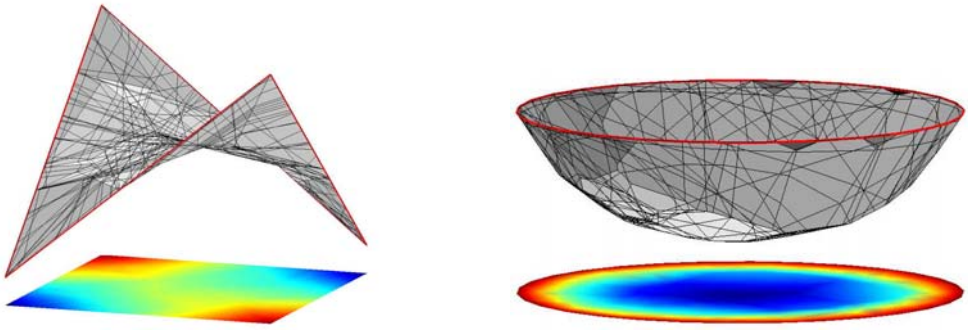
We first show how our algorithm behaves on some simple classical variational problems. Next we show how our algorithm works on a problem that has numerous local minima.

Our first example is the famous problem of finding the function of the minimal area that "spans" a given boundary curve. Such a function $f(x, y)$ minimizes the integral

$$F_1(f) = \iint_\Omega \sqrt{1 + f'_x(x,y)^2 + f'_y(x,y)^2}\, dx\, dy. \tag{5.1}$$

Figure 6a shows the result when using a saddle boundary configuration.

A standard elasticity problem is the deformation of a membrane in a gravity field. Such a membrane function $f(x, y)$ minimizes the integral



(a) (b)

Figure 6: Results of the plane selection algorithm. Boundary conditions are shown in red. (a) Minimal area surface for the saddle boundary conditions, 150 random planes. (b) Membrane in the gravity field, 200 random planes.

$$F_2(f) = F_1(f) - \iint_\Omega k(x,y) f(x,y) g \, dx \, dy \tag{5.2}$$

In Figure 6b, we show a solution with a constant density $k(x,y) = c$, though nothing changes in our method if we use any other distribution of the density or gravity field.

The space and time costs of our algorithm can be seen in the table (2). In this experiment a cubic cell Φ was subdivided by N_{planes} random planes. We measured time of the construction of the cell tree, time of the min-cut computation, number of the total cells in the cell tree, and total memory usage. The experiments were performed at Athlon 1.5GHz machine with 2GB of memory. From the table, we observe that the major bottlenecks of the algorithm are memory usage and min-cut computation time. In theory the worst case for n planes has n^3 memory complexity and $n^6 \log n$ time complexity. From experiments with various sized problems, we observed that in practice the algorithm had $n^{2.7}$ memory complexity and $n^{4.6}$ time complexity.

N_{planes}	$T_{cell\ tree}$, sec.	$T_{min-cut}$, sec.	N_{cells}, thousands	Memory used, Mb
40	0.5	0.5	11	3
80	4	14	107	35
120	14	95	375	123
160	34	380	903	300
200	68	1350	1810	596
240	117	3400	3150	1040

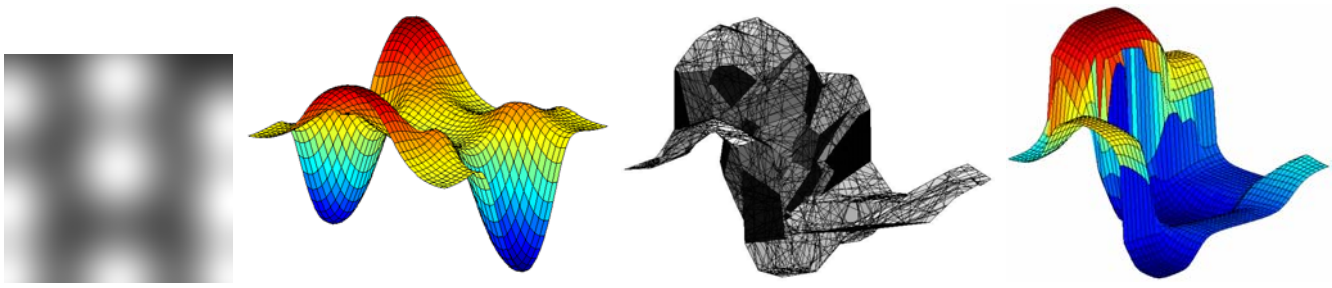
Table 2: Experimental time and memory usage by the minimal surface algorithm. N_{planes} is a number of the planes used for the unit cube subdivision. $T_{cell\ tree}$ and $T_{min-cut}$ are the times used for the construction of the cell tree and computation of the min-cut. N_{cells} is the number of the cells in the cell tree.

5.2 Examples with many local minima

Here we experimented with a class of problems that has a huge number of local minimums and can hardly be solved in reasonable time by existing methods.

Our problems were created by starting with a 3D unit cube that has some given 3D density distribution $D(x,y,z)$ in it. Our problem was then to find the surface that breaks the cube into two pieces using minimal effort. In this case we must minimize the functional

$$\iint_\Omega D\left(x,y,f(x,y)\right) \sqrt{1 + f'_x(x,y)^2 + f'_y(x,y)^2} \, dx \, dy \tag{5.3}$$



(a) (b) (c) (d)

Figure 7: (a) A two-dimensional slice of the density function. Intensity of the color is proportional to the density. (b) Numerical optimization methods converge to the local minimum. (c) The minimal discrete surface using a grid of random planes. (d) Our method followed by numerical relaxation

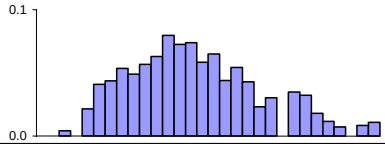
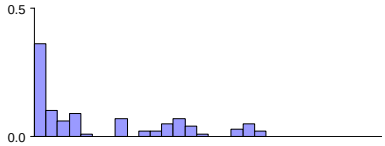
	time, *sec*	memory, *Mb*	energy histogram
a) local optimization, starting from random guess	13.5	3	
b) combinatorial optimization, 240 planes, + relaxation	3590	1040	
c) plane selection, 150 planes, 5 iterations, + relaxation	1365	200	

Table 3: $2 \cdot 10^6$ possible local minimum surfaces. (a) Local optimization with random starting guess. (b) Global combinatorial optimization with 240 planes. (c) Global optimization enhanced with a plane selection heuristic.

We only enforced the relaxed condition that the boundaries map to [0..1].

The distributions were created by summing up a number of 3D Gaussian blobs. We chose m locations over Ω, above each location we placed k Gaussian blobs at random heights. This results in $(k+1)^m$ locally optimal functions. The density distribution can be very complex; a 2D slice of such a distribution is shown in Figure (7,a).

In our experiments we compared our algorithm against a standard numerical optimization method. For the numerical method, we used a multi-resolution gradient descent method with random starting conditions. We terminated the iterations when the improvements were below 10^{-5}.

For our first experiment we constructed a density field with $m = 9$ and $k = 4$, there are about $2 \cdot 10^6$ possible local minimums for this example. The results of the experiment can be found in the table 3. In the rightmost column, we show a histogram of the minimal energies found by each method. We have shifted and the energies so that the leftmost column only counts solutions that are within epsilon of the absolutely smallest energy reported by any of the algorithms. (Since we do not have a closed form solution for this problem, we, of course, have no proof that this in-fact represents the global minimum.)

As we can see here, the local optimizer finds solutions with a range of energies, although it never finds the best one. The 240-plane combinatorial method, most often finds the best solution, but occasionally finds non-optimal ones. The use of plane selection with only 150 planes always finds the same solution at a fraction of the cost in both time and memory.

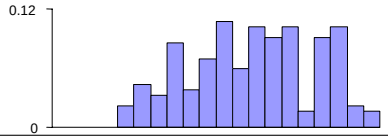
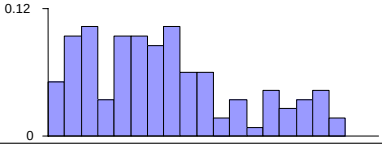
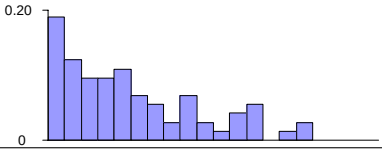
	time, *sec*	memory, *Mb*	energy histogram
a) local optimization, starting from random guess	13.5	3	
b) combinatorial optimization, 240 planes, + relaxation	3590	1040	
c) plane selection, 150 planes, 20 iterations, + relaxation	5450	200	
d) adaptive refinement, 150 planes, 3 iterations (5 plane selection iterations each), + relaxation	4120	290	

Table 4: Comparison of the different algorithms for the functional with two level of details.

All of the combinatorial solutions were followed by relaxation. The typical effect of this relaxation is to improve the quality of the result by about 1%. More specifically the quantity "(relaxed - unrelaxed)/relaxed" is on the order of 10^{-2}.

Next we experimented with a density field that had two levels of detail. In particular, we started with the functional from the previous experiment and randomly added the same number of Gaussian blobs, but these blobs were 5 times smaller than the original ones.

For the adaptive refinement extension, we ran three steps of refinement using an envelope offset of 1, 0.2, and 0.05 respectively. At every step we use 5 iterations of the plane selection extension to construct the best possible solution and the next envelope. This method always succeeded in finding the global minimum. These results are summarized in table 4.

6 Discussion

We are currently investigating various applications where our algorithms may be applicable. These include segmentation of volume data, and constructing iso-surfaces from contours. We hope that, combinatorial methods, such as the ones described here may find more applications and impact in the scientific computing community.

It would be nice to be able to solve variational problems of higher co-dimensions, such as minimal 2-surfaces in R^4, but it seems unlikely that such a generalization exists. Similarly, it seems difficult to generalize these methods for functionals that depend on the second derivative of the function.

In this paper we limited ourselves by constructing the grids as intersection of the hyper-planes. This is one of the reasons that time and space complexity of our method is so high. Every hyperplane that was added to the list intersected a lot of cells, most of which are located far form the solution. One of the possible directions for future research is to construct grids that can be refined locally in order to increase precision of the approximation without overcomplicating the grid structure. For example, in the planar case we investigated subdivision rules for triangular grids. Unfortunately, the simple subdivision rules (such as 1-to-4 split) do not ensure the necessary density properties of the resulting grids and more complicated rules (such as pinwheel grids [16]) are hard to implement and generalize to higher dimensions.

Using spacial octrees (for 3D hyper-planes) is not as elegant and easy to implement as binary trees, but it is likely to increase the performance of the algorithm and better balance of the tree.

The dual graphs that arise from our algorithm are very sparse and have a special structure [7]. Exploiting these properties in MIN-CUT algorithm seems like a very interesting direction of future research.

7 Acknowledgements

We would like to thank Chris Buehler for help with development of the discrete duality construction, Ray Jones for help with the directed duality construction, and Tom Duchamp for pointing us to the literature on integral currents.

References

[1] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. In *EMMCVPR*, 2001.

[2] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *Proc. ICCV*, 2003.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *Proc. ICCV*, 1999.

[4] C. Buehler, S. J. Gortler, M. F. Cohen, and L. McMillan. Minimal surfaces for stereo. *ECCV*, pages 885–899, 2002.

[5] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, 2001.

[6] B. Dacorogna. *Direct Methods in the Calculus of Variations*. Springer-Verlag, 1989.

[7] P. J. Giblin. *Graphs, Surfaces and Homology*. Chapman and Hill, 1981.

[8] R. Glowinski. *Numerical Methods for Nonlinear Variational Problems*. Springer-Verlag, 1984.

[9] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2001.

[10] T. C. Hu. *Integer Programming and Network Flows*. Addison-Wesley, 1969.

[11] T. C. Hu, A. B. Kahng, and G. Robins. Optimal minimum-surface computations using network flow. *To appear in Mathematical Programming*.

[12] T. C. Hu, A. B. Kahng, and G. Robins. Solution of the discrete plateau problem. In *Proc. of the National Academy of Sciences*, volume 89, pages 9235–9236, 1992.

[13] D. B. Johnson. Parallel algorithms for minimum cuts and maximum flow in planar networks. *J. of the ACM*, 34(4):950–967, 1987.

[14] F. Morgan. *Geometric Measure Theory: a Beginner's Guide*. Academic Press, 1988.

[15] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1994.

[16] C. Radin and L. Sadun. The isoperimetric problem for pinwheel tilings. *Comm. Math. Phys*, 177:255–263, 1996.

[17] J. Sullivan. *A Crystalline Approximation Theorem for Hypersurfaces*. PhD thesis, Princeton University, 1990.

[7] For example, it can be easily shown that the dual graph of any hyperplane grid is bichromatic.

A Proof of the density theorem (3.5)

Before we begin to prove the theorem, we formulate technical lemma that we will use in the proof.

Lemma A.1. *Let $f(x) : [a, b] \mapsto \mathbb{R}$ be a twice differentiable function, $f(a) = A$, $f(b) = B$. Then there exists such a point $\bar{x} \in [a, b]$ that*

$$\frac{df}{dx}(\bar{x}) = \frac{B - A}{b - a}$$

Corollary A.2. *If $f(a) = A + \varepsilon_1$, $f(b) = B + \varepsilon_2$, then there exists such a point $\bar{x} \in [a, b]$ that*

$$\left| \frac{df}{dx}(\bar{x}) - \frac{B - A}{b - a} \right| \leq \frac{|\varepsilon_1| + |\varepsilon_2|}{b - a}$$

Theorem A.3. *Let us consider a continuous twice differentiable function $f(x) : [0, 1] \mapsto [0, 1]$ with bounded first and second derivative, $|f'(x)| \leq C_1$, $|f''(x)| \leq C_2$. For any $\varepsilon > 0$ there exists $g \in \mathcal{L}$ such that $\|f - g\| < \varepsilon$.*

Proof. Every piecewise linear function generated by the grid C_n can be coded by the second coordinate of its vertices

$$(dk_0, dk_1, \ldots, dk_n) \mid k_i \in \left\{0, 1, \ldots, n^2\right\},$$

where $d = 1/n^2$. In this notation the closest piece-wise linear approximation $g(x)$ of the function $f(x)$ will look like

$$\left(d\left[\frac{f(0)}{d}\right],\ d\left[\frac{f(h)}{d}\right],\ d\left[\frac{f(2h)}{d}\right],\ \ldots,\ d\left[\frac{f(nh)}{d}\right] \right)$$

where $[\cdot]$ denotes the integer part of a number and $h = 1/n$. By choosing $g(x)$ this way we make sure that $|f(hi) - g(hi)| \leq d/2$ for any $i = 0, 1, \ldots, n$.

Let us now assess the difference between f and g from above, using the fact that f has limited first and second derivatives.

$$\|f - g\| = \int_0^1 |f(x) - g(x)| + \left|f'(x) - g'(x)\right| \, dx \tag{A.-1}$$
$$\leq \max_{x \in [0,1]} (|f(x) - g(x)|) + \max_{x \in [0,1]} \left(\left|f'(x) - g'(x)\right|\right)$$

The difference $|f(x) - g(x)|$ in the points hi, $i = 0, 1, \ldots, n$ is less then $d/2$ and therefore

$$|f(x) - g(x)| \leq d/2 + C_1 h \tag{A.-2}$$

for all $x \in [0, 1]$.

Let us take a look on the interval $[hi, h(i + 1)]$ for some $0 \leq i \leq n - 1$. The derivative of the piecewise linear function $g(x)$ is constant on this interval and equal to $(g(h(i + 1)) - g(hi))/h$. We also know that at the ends of the interval the distance between $f(x)$ and $g(x)$ is less then $d/2$. Applying the corollary (A.2), we can say that there exists a point $\bar{x} \in [hi, h(i + 1)]$ such that

$$\left|f'(\bar{x}) - g'(\bar{x})\right| \leq \frac{d}{h}.$$

Using the fact that the second derivative is limited, we can conclude that

$$\left|f'(x) - g'(x)\right| \leq \frac{d}{h} + C_2 h. \tag{A.-2}$$

is true for all $x \in [hi, h(i + 1)]$ and therefore for all $x \in [0, 1]$.

Combining (A.-1), (A) and (A), and taking into account that $h = 1/n$, $d = 1/n^2$, we have

$$\begin{aligned}\|f - g\| &\le d/2 + C_1 h + d/h + C_2 h \\ &= \frac{1}{2n^2} + \frac{1}{n}\left(C_1 + C_2 + 1\right) \underset{n\to\infty}{\longrightarrow} 0.\end{aligned}$$

Thus, for any $\varepsilon > 0$ we can choose such an integer n and such a function $g \in \mathcal{M}(C_n) \subset \mathcal{L}$ that $\|f - g\| < \varepsilon$.

□

We can easily generalize this theorem for piece-wise differentiable functions.

Corollary A.4. *Let us consider a continuous **piece-wise** twice differentiable function $f(x) : [0,1] \mapsto [0,1]$ that has a finite number of pieces. The first and second derivative are defined at every internal point of these pieces and bounded, $|f'(x)| \le C_1$, $|f''(x)| \le C_2$. For any $\varepsilon > 0$ there still exists $g \in \mathcal{L}$ such that $\|f - g\| < \varepsilon$.*

Proof. We create the piece-wise linear function g the same way we did it in the previous theorem. The only difference from the previous proof is that now the function f has a finite number of points where the first and second derivatives are discontinuous. Let us consider one of these points $\hat{x}$ that is covered by k-th segment of the function g. Because the first derivative is bounded, we have

$$\int_{kh}^{(k+1)h} |f(x) - g(x)| + \left|f'(x) - g'(x)\right| \, dx \le (2 + C_1)h.$$

Therefore, the total difference introduced by these points of discontinuity goes to zero when h goes to zero. □

The proof of the theorem (3.5) now becomes trivial. We proved that for every function $f \in \mathcal{P}$ and any $\varepsilon > 0$ there still exists $g \in \mathcal{L}$ such that $\|f - g\| < \varepsilon$. Therefore, $\mathcal{L}$ is dense in $\mathcal{P}$.

B Proof of the density theorem (3.6)

Our investigation of the random grids will be based on the obvious fact that can be stated as follows.

Lemma B.1. *Let $f(x)$ be a piece-wise linear function with n linear segments, R_m be a random grid with $m \ge n$, and $\varepsilon > 0$. Then there is a non-zero probability $p(f, m) > 0$ that there exists such a function $g(x) \in \mathcal{M}(R_m)$ that $\|f - g\| < \varepsilon$.*

All the lines are generated independently, therefore we can consider the grid R_{2m} to be the intersection of two independent grids of the size m. Using the basics of the probability theory, we can conclude that for the conditions given in the lemma

$$p(f, 2m) \ge 2p(f, m) - p(f, m)^2.$$

We know that the sequence $p_{n+1} = p_n + p_n(1 - p_n)$ with the starting point $0 < p_1 < 1$ converges to 1. That is why

$$p(f, m) \underset{m\to\infty}{\longrightarrow} 1. \tag{B.0}$$

In other words, we can state the following corollary.

Corollary B.2. *Let $f(x)$ be a piece-wise linear function, and $\{R_m,\ m = 1, 2, \dots\}$ be a family of the random grids, and $\varepsilon > 0$. Then the probability that there exists such a function $g(x) \in \mathcal{L}_r$ that $\|f - g\| < \varepsilon$ is equal to 1.*

Corollary B.3. *Let $\{R_m,\ m = 1, 2, \dots\}$ be a family of the random grids. Then $\mathcal{L}_r$ is dense in $\mathcal{L}$ and therefore $\mathcal{L}_r$ is dense in $\mathcal{P}$.*

C Proof of the duality theorem (4.4)

In section 4.1 we already defined 1- and 2-chains and their boundaries.

Definition C.1. The chain is called *closed* if its boundary is the empty set.

The following lemmata reduce the minimum valid curve problem to one of planar partitioning.

Lemma C.2. *On our complex, there is a one-to-one correspondence between valid curves and closed 1-chains which include the edge e_r and do not include e_k.*

Proof. The correspondence between a 1-chain $\mathcal{C}$ satisfying the properties of the lemma and a valid curve $\mathcal{S}$ is $\mathcal{C} = \mathcal{S} \cup e_r$. □

Lemma C.3. *On any complex homeomorphic to B^2, there is a one-to-one correspondence between 2-chains and closed 1-chains. The relationship between the corresponding 2-chain $\mathcal{A}$ and 1-chain $\mathcal{C}$ can be expressed as $\partial_2 \mathcal{A} = \mathcal{C}$.*

Proof. The first homology group H_1 is defined as the quotient group: $H_1 = \frac{\ker(\partial_1)}{\mathrm{Im}(\partial_2)}$. In any complex homeomorphic to B^2, H_1 is the trivial group (see [7]), therefore $\ker(\partial_1) = \mathrm{Im}(\partial_2)$. □

Combining these two lemmas together, we establish the following correspondence.

Theorem C.4. *On our complex, there is a one-to-one correspondence between valid curves and 2-chains that include the face f_r and do not include the face f_k.*

Because there is a one-to-one correspondence between the faces of the primal graph and the vertices of the dual graph, there is also a one-to one correspondence between 2-chains that includes the face f_r and does not include the face f_k and cuts in the dual graph.

This proves theorem (4.4).