

COMPLEXITY AND MODELING ASPECTS  
OF MESH REFINEMENT INTO  
QUADRILATERALS

by

ROLF H. MÖHRING  
MATTHIAS MÜLLER-HANNEMANN

No. 554/1997

# Complexity and Modeling Aspects of Mesh Refinement into Quadrilaterals\*

Rolf H. Möhring\*\*      Matthias Müller–Hannemann\*\*

May, 1997; revised August, 1997

## Abstract

We investigate the following mesh refinement problem: Given a mesh of polygons in three-dimensional space, find a decomposition into strictly convex quadrilaterals such that the resulting mesh is conforming and satisfies prescribed local density constraints.

The conformal mesh refinement problem is shown to be feasible if and only if a certain system of linear equations over  $GF(2)$  has a solution. To improve mesh quality with respect to optimization criteria such as density, angles and regularity, we introduce a reduction to a minimum cost bidirected flow problem. However, this model is only applicable, if the mesh does not contain branching edges, that is, edges incident to more than two polygons. The general case with branchings, however, turns out to be strongly  $\mathcal{NP}$ -hard. To enhance the mesh quality for meshes with branchings, we introduce a two-stage approach which first decomposes the whole mesh into components without branchings, and then uses minimum cost bidirected flows on the components in a second phase.

**Key Words.** Mesh generation, bidirected flows,  $\mathcal{NP}$ -completeness, mesh decomposition, computer-aided design

## 1 Introduction

For many years the finite element method has been a fundamental numerical analysis technique in the engineering community. In the field of computer-aided design, engineers often model their workpieces first in form of a coarse mesh of convex polygons in three-dimensional space which approximates the object's surface. However, in order to make a numerical analysis applicable, a suitable refinement of the coarse mesh is necessary.

A large amount of research has been done in the area of mesh refinement into triangles, see [Ho88], [BE95] and [BP97] for surveys. In contrast, there is much less work on quadrilaterals, although meshes which consist *solely* of quadrilaterals are more appropriate in many applications, such as torsion problems and crash simulations. A few studies have been published comparing the convergence characteristics and accuracy of meshes with triangle and quadrilateral elements, as well as of hexahedral versus tetrahedral elements. The interested reader is

---

\*An extended abstract of this paper appears in the Proceedings of the Eighth Annual International Symposium on Algorithms and Computation, ISAAC'97, Singapore.

\*\*Technische Universität Berlin, Fachbereich Mathematik, Sekr. MA 6-1, Straße des 17. Juni 136, 10623 Berlin, Germany, e-mail: {moehring,mhannema}@math.tu-berlin.de; URL: <http://www.math.tu-berlin.de/~mhannema> or <http://www.math.tu-berlin.de/~moehring>.

The second author was partially supported by the special program "Efficient Algorithms for Discrete Problems and Their Applications" of the Deutsche Forschungsgemeinschaft (DFG) under grant Mo 446/2-2; communicate with the second author.

referred to [CK92, Wei94, LL92, BPM<sup>+</sup>95]. Finite element textbooks ([ZT89], [Bra93]) include statements such as “... for reasons of better accuracy and efficiency, quadrilateral elements are preferred for two-dimensional meshes and hexahedral elements for three-dimensional meshes. This preference is clear in structural analysis and seems to also hold for other engineering disciplines.” [Bra93]. This is the background of our work, and therefore, in this paper, *refinement* of a mesh means decomposing each polygon into strictly convex quadrilaterals.

This paper addresses important modeling aspects of quadrilateral mesh refinement. Its intention is to show how these aspects can be captured within a combinatorial model and to elaborate on the consequences for the computational complexity of the resulting problems.

**Modeling aspects of mesh refinement into quadrilaterals.** We give a brief and informal description of the most important goals for mesh refinement to provide further background and to motivate the problems considered in this paper.

For use in the finite element method (or its variants) a refined mesh should have the following properties:

1. The finite element method derives a system of linear or non-linear equations from the mesh (i.e. assembles a stiffness matrix and the right hand vector) for the governing partial differential equations. To reflect continuity properties in a suitable way, the mesh is required to be *conformal*. In a *conformal* refinement of a mesh, any two distinct quadrilaterals which are not completely disjoint either share exactly one whole edge, or they have a single common vertex.
2. To achieve the required accuracy the refinement of a mesh has to be fine enough, but not too fine for reasons of efficiency. Hence, the mesh density has to be controlled.
3. Apart from the mesh density, mesh quality criteria depend upon the shape of the quadrilaterals. For numerical reasons in the finite element analysis, interior angles of quadrilaterals should neither be too small nor too large. There is no generally accepted, precise threshold, but one usually aims at generating quadrilaterals with no angles smaller than some given  $\alpha$  and no angles larger than some  $\beta$ . (In practice, one often uses as a rule of thumb values of  $\alpha = 30^\circ$  and  $\beta = 150^\circ$  [ZZHW91].)

For brevity, a conformal mesh refinement into strictly convex quadrilaterals will be called a *quadrangulation*.

The approximated surface need not be a two-manifold, i. e. the corresponding mesh model may contain *branchings* where more than two polygons meet, see Figure 4. Coping with non-manifold surfaces is an issue of crucial importance, as they appear in many practical examples. Moreover, quadrilateral surface meshing can be seen as a first step in hexahedral volume meshing (which still is an only partially solved problem with respect to both theory and practice). At least, it seems to be a promising approach to start a decomposition into hexahedra from a high quality quadrilateral surface mesh [Mit97]. Here, we want to point out that it might be advisable to decompose complicated solid models first into smaller, preferably convex, subdomains by insertion of internal polygons. This can be done either explicitly or implicitly. The latter approach, so-called *meshing by virtual decomposition*, has been introduced by [WMBS95]. In both variants, these additional polygons induce a number of branchings. However, this fits perfectly into our approach which mainly abstracts from geometry and essentially solves a combinatorial problem. In fact, we will introduce a simple and elegant method to ensure conformity between subdomains resulting from branchings.

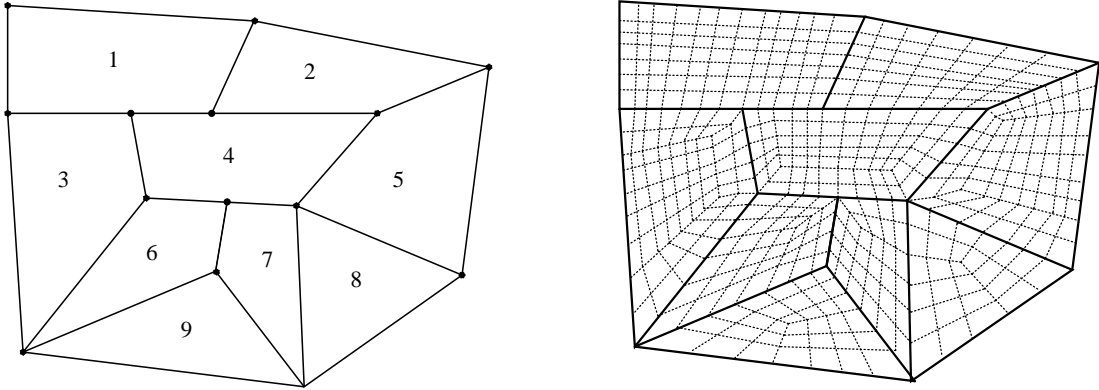


Figure 1: A small, planar artificial instance. Figure 2: A conformal refinement of Figure 1.

**Previous work.** Most work on conformal mesh refinements in the literature relies on a few classes of *templates* [TA93, MMW96]. A *template* (sometimes also called *meshing primitive* [TA93]) is a pattern which describes how a single polygon can be decomposed into quadrilaterals. Templates are designed to achieve good angles and aspect ratios heuristically. For example, the most important template for quadrangular polygons is a  $(p \times q)$ -grid, where  $p$  and  $q$  are variable. See the first three illustrations of Figure 13 for examples of widely used standard templates.

Tam and Armstrong [TA93] developed an integer programming based approach to conformal mesh refinement using the three standard templates. However, they did not consider the problem how to select an appropriate template for each macro element. In their approach, templates are preselected and fixed in advance (it is not clear by which heuristics this is done). The most important disadvantage of this approach based on general integer programming, however, is the lack of an efficient algorithm.

In [MMW96] the computational complexity of mesh refinement under template restrictions has been studied. In particular, it has been shown that even finding feasible solutions is strongly  $\mathcal{NP}$ -hard in this case. The same paper also introduces a heuristic approach for finding feasible solutions by a reduction to a sequence of bidirected flow subproblems (or, equivalently, to a sequence of  $b$ -matching problems). See the seminal paper of Edmonds [Edm67] or the monograph by Derigs [Der88] for an introduction to bidirected flows and  $b$ -matchings. Each of these bidirected flow problem can be solved efficiently in strongly polynomial time, but no guarantee can be given that the approach in [MMW96] finally succeeds in finding a feasible solution for the original problem.

Finding conformal mesh refinements with the minimum number of quadrilaterals has been shown to be  $\mathcal{NP}$ -hard in [MW97a]. In a subsequent work [MW97b], matching techniques have been used to approximate the minimum number of quadrilaterals. However, these methods do not take care of other criteria than the number of quadrilaterals.

Kannan and Soroker [KS92] have considered the very special problem to determine whether a single polygon can be refined into *parallelograms*. This problem can be solved in time  $O(n^2)$  for an  $n$ -vertex polygon using ordinary flow techniques.

Practical implementations often use advancing-front based heuristics, like paving [BS91], for decomposing single macro elements. Such techniques start from the boundary of a macro element and iteratively insert quadrilaterals from the boundary inward. Although these techniques seem to work quite well in practice, they are mere heuristics. Moreover, such approaches completely ignore the difficulties in ensuring conformity between macro elements.

**Our contribution.** In this paper, we generalize and improve previous results and work in several ways:

- For meshes with branchings, we show that the refinement problem is feasible if and only if a certain system of linear equations over  $GF(2)$  has a solution. From any solution to this system of linear equations we easily derive a feasible conformal mesh refinement. However, there seems to be no way to incorporate mesh quality optimization into this approach. This is not surprising as even optimizing the mesh density on the edges will be shown to be  $\mathcal{NP}$ -hard for meshes with branchings.
- We introduce a new bidirected flow model for meshes without branchings. Within that model finding a feasible mesh refinement now amounts to solving a *single* bidirected flow problem. This progress has been made possible by giving up the template restrictions imposed by the standard templates in [TA93, MMW96]. However, the new model contains the standard classes of templates as special cases. In practice, the coarse input mesh quite often consists of triangle- or quadrilateral-shaped macro elements, but our approach works for arbitrary convex polygons, too.
- We extend the bidirected flow model to a minimum cost bidirected flow problem where certain optimization criteria such as mesh density, interior angles and mesh regularity can also be handled.

As mentioned above, minimum cost bidirected flow problems are equivalent to minimum cost perfect  $b$ -matchings and can, therefore, be solved in strongly polynomial time, as shown by Anstee [Ans87] and Edmonds (cf. Gerards' survey on matching [Ger95]). The asymptotic running time of strongly polynomial algorithms for these problems are dominated by strongly polynomial algorithms for ordinary minimum cost flow problems. Thus, the best strongly polynomial time bound for minimum cost bidirected flow is  $O((m \log n)(m + n \log n))$ , where  $m$  denotes the number of edges and  $n$  the number of vertices of the underlying graph [AMO93]. The number of edges and vertices in the auxiliary graph of the bidirected flow model is linear in the number of original polygons and mesh edges.

- To improve the mesh quality of meshes with branchings, we introduce mesh decomposition into homogeneous components and combine both approaches: In a first phase, we determine the subdivision numbers for all branching edges by solving a system of linear equations over  $GF(2)$ . Afterwards, in a second phase, we solve a minimum cost bidirected flow problem for each homogeneous component with fixed subdivision numbers on the branching edges. For real-world instances the number of branching edges is usually relatively small in comparison with the number of polygons. This empirical observation intuitively explains why one should hope that the proposed combined approach is likely to achieve a reasonably good overall mesh quality. In fact, an extensive computational study [MH97] shows that an implementation of the approaches described in this paper simultaneously leads to a very good mesh quality with respect to density, interior angles, edge lengths and aspect ratio.

**Structure of the paper.** In Section 2 we give a precise definition of our mesh model and present a general, but very simple approach to conformal mesh refinement. We also elaborate on the relation of quadrangulations and solving systems of equations over  $GF(2)$ . Then, in Section 3, we consider meshes without branchings. For such meshes, we develop step by step a bidirected flow model which can be used to optimize quadrangulations with respect to a number

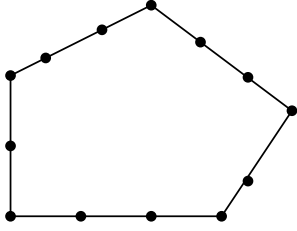


Figure 3: A convex polygon with five segments and thirteen edges and vertices.

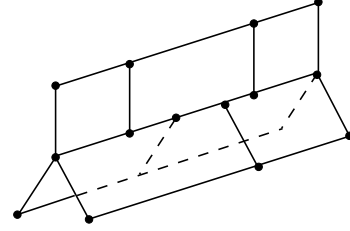


Figure 4: A small mesh with five branching edges.

of mesh quality criteria. Finally, in Section 4, we present a decomposition approach for meshes with branchings. The latter approach combines the simplicity of finding feasible solutions by solving a systems of equations over  $GF(2)$  with the more sophisticated bidirected flow approach for the optimization of the mesh quality.

## 2 A simple approach for conformal mesh refinement

**Preliminaries.** A *polygon* is a region in the plane or, more generally, of a smooth surface in the three-dimensional space, bounded by a finite, closed sequence of straight line segments (*edges*). The endpoints of the line segments or curves are the *vertices*. A polygon is *simple* if its edges do not cross each other, and *convex* if the internal angle at each vertex is at most  $\pi$ . A vertex of a convex polygon is a *corner* if its internal angle is strictly less than  $\pi$ . An *interval* of a polygon  $P$  is a path of edges on its boundary. A *segment*  $S$  is an interval between two successive corners of  $P$  (see Figure 3).

A *mesh* is a set of openly disjoint, convex and simple polygons, the so-called *macro elements*. The macro elements are convex, but not necessarily strictly convex. Let  $\mathcal{M} = \{P_1, P_2, \dots, P_q\}$  be the set of polygons of the mesh.

Two macro elements are *neighbored* if they have points of the boundary in common which are not corners. These neighborhood relationships induce an undirected graph  $G = (V, E)$ , which is embedded on the surface approximated by the mesh. More precisely,  $V$  consists of the vertices of the polygons. If a vertex of a polygon also belongs to the interior of a side of another polygon, it subdivides this side. Hence, we may identify common intervals of neighbored sides of polygons with each other, and  $E$  consists of these intervals after identification. (In the small example of Figure 1, the quadrilateral-shaped macro element number 4 has 6 neighborhood relations and edges: it is neighbored with elements 1,2,5,7,6, and 3. Macro elements number 5 and number 7 are not neighbored as they have only one corner in common.)

Note that the graph  $G$  of a mesh need not be planar; for example, a mesh approximating the surface of a torus has genus one. Even more, a mesh may contain *branching edges*, that is, edges incident to more than two polygons (Figure 4). We call a mesh *homogeneous* if it does not contain branching edges.

For an edge  $e_i \in E$ , let  $E_i$  be the set of all those polygons which contain  $e_i$ . A *combinatorial description* of a mesh consists of the graph  $G$  and the hypergraph  $H = (\mathcal{M}, \{E_1, \dots, E_m\})$  with vertex set  $\mathcal{M}$  and edge set  $\{E_1, \dots, E_m\}$ . We will often identify a mesh with its combinatorial description.

For the refinement of a single polygon into quadrilaterals we allow to insert new vertices in the interior of the polygon and on the boundary. The *subdivision number* of edge  $e \in E$ , denoted by  $x_e$ , is the number of additional vertices which are placed on edge  $e$  in a refinement.

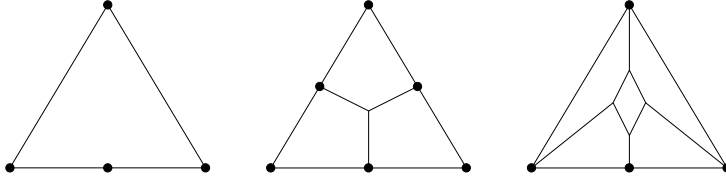


Figure 5: A triangular-shaped polygon with four vertices (left); a *refinement*, which places two extra vertices on the boundary (middle); and a *decomposition*, where no additional vertices on the boundary are allowed (right).

In the mesh refinement problem, the polygons cannot be refined independently since we have to ensure that the mesh is conformal. Hence, we carefully distinguish between conformal refinements of polygons, where vertices can be inserted at arbitrary positions, and conformal decompositions (see Figure 5 for an example). By a *conformal decomposition* of a single polygon we will always mean the variant which does not allow to subdivide edges but to place vertices into the *interior* of the polygon.

A crucial insight is the following well-known [Joe95, MW97a], but important characterization of those polygons which can be decomposed into strictly convex quadrilaterals:

**Lemma 2.1** *A simple polygon  $P$  admits a conformal decomposition into strictly convex quadrilaterals if and only if the number of vertices of  $P$  is even.*

The necessity of this evenness condition follows directly from a counting argument, and, as a side-effect of the proof of Proposition 3.1 below, the sufficiency will be made explicit below in a constructive way (for convex polygons).

An easy way to model local density control is by means of lower and upper constraints on the number of additional subdivision points for each edge. Hence, we assume that we are given *density constraints* of the following kind:

For an edge  $e \in E$ , the subdivision number  $x_e$  is at least  $\ell_e$  and at most  $u_e$ :  
 $\ell_e \leq x_e \leq u_e$  (*lower and upper edge capacities*,  $(\ell_e, u_e)$ -capacities, for short).

More sophisticated ways to control local mesh density will be dealt with in Section 3. An edge with equal upper and lower capacities is a *fixed edge*, otherwise it is *free*. A conformal mesh refinement is *feasible* if it respects the density constraints.

**A simple approach to quadrangulations.** As we know from Lemma 2.1, it suffices to determine the subdivision numbers such that all polygons become even. Then we can always complete the quadrangulation of all polygons.

For each edge  $e \in E$ , let  $\tilde{x}_e$  be some fixed integer in the interval  $[\ell_e, u_e]$ . Then define for each polygon  $P \in \mathcal{M}$  the “parity number”  $b_P := \sum_{e \in P} (\tilde{x}_e + 1) \bmod 2$ .

For each edge  $e \in E$ , let  $y_e$  be a 0/1-integer variable, but fix  $y_e \equiv 0$ , if  $u_e - \ell_e = 0$ .

Now, if we take for each polygon  $P$  an equation of the form  $\sum_{e \in P} y_e = b_P$ , this defines a system of linear equations over  $GF(2)$  which one can easily solve. Clearly, any solution  $\tilde{y}$  to this system immediately makes all polygons even, if we set

$$x_e := \begin{cases} \tilde{y}_e + \tilde{x}_e & \text{if } \tilde{x}_e < u_e \\ -\tilde{y}_e + \tilde{x}_e & \text{otherwise.} \end{cases}$$

Note that the solvability of this system of equations over  $GF(2)$  does not depend on the choice of  $\tilde{x}$ .

A system of linear equations over  $GF(2)$  is simply solved by Gaussian elimination. No care has to be taken for numerical stability over  $GF(2)$ .

This shows that finding subdivision numbers such that all polygons become even is not harder than solving a system of linear equations over  $GF(2)$ . The converse is also true, as we will show below.

**Theorem 2.2** *The determination of subdivision numbers for a mesh with branchings and edge capacities such that a feasible conformal refinement is possible (or proving its non-existence) is as hard as solving a system of linear equations over  $GF(2)$ , where the encoding length of each problem is linear in the size of the other.*

**Proof:** In the above remarks we have already seen how to reduce the determination of the subdivision numbers to the solution of a system of linear equations over  $GF(2)$ . Hence, it only remains to show how to transform any system of linear equations over  $GF(2)$  to a mesh refinement problem of linear size in the encoding length for the system of equations.

Therefore, let  $A = (a_{ij})$  be some  $m \times n$  0/1-matrix,  $x \in \{0, 1\}^n$ ,  $b \in \{0, 1\}^m$ , and  $Ax = b$  the linear system of equations. Without loss of generality we may assume that each column and each row of  $A$  contains at least one non-zero entry. Denote by  $|a_i| := \sum_{j=1}^n a_{ij}$  the number of non-zero entries in the  $i$ -th row of  $A$ .

The transformation to a mesh refinement instance works as follows. For row  $i$  of  $A$  we create a polygon  $P_i$  with  $p_i$  segments where all segments are single edges. These polygons are disjoint, the coupling between them is described below.

If  $b_i$  and  $|a_i|$  have the same parity, we choose  $p_i := |a_i| + 2$ , otherwise we set  $p_i := |a_i| + 3$ . The first  $|a_i|$  edges get lower capacities  $\ell_i = 0$  and upper capacities  $u_i = 1$ , all others are *fixed to zero*, i.e.  $\ell_i = u_i = 0$ . (The fixed edges are used to ensure that  $p_i \geq 3$  for each polygon.)

Each free edge of the polygon  $P_i$  corresponds one-by-one to a non-zero entry  $a_{ij}$  of the  $i$ -th row. By construction, the equation corresponding to row  $i$  is fulfilled over  $GF(2)$  for any subdivision which makes the polygon  $P_i$  even if  $x_j$  equals the subdivision number of the edge corresponding to  $a_{ij}$  for each non-zero entry of row  $i$ .

To make sure that the  $x_j$  have to be chosen in a consistent way we add the following construction. For each non-zero entry  $a_{ij}$  we create a quadrilateral  $Q_{ij}$ . One of its edges corresponds to the variable  $x_j$  and gets a lower capacity of 0 and an upper capacity of 1, and another edge which is not adjacent to the first is identified with the edge of polygon  $P_i$  which corresponds to  $a_{ij}$ . The two remaining edges of each such quadrilateral  $Q_{ij}$  are fixed to zero. Finally, all edges corresponding to the same  $x_j$  are identified to constitute a branching edge. Obviously, each feasible refinement immediately yields a feasible solution of our system of equations.  $\square$

Note that the subsequent embedding phase can be done in linear time (linear in the number of subdivision points) using the decomposition algorithm in [MW97a]. The obvious advantage of this approach is its simplicity. The disadvantage, however, is that there seems to be no way to incorporate optimization of the mesh quality. Angle control and a preference of the mesh structure seem to be impossible by this approach.

Only density control can be achieved to a certain extent, if we initially choose  $\tilde{x}_e$  as the desired subdivision number. But we will be forced to change the subdivision number for all those edges which have a non-zero entry in the solution of our system of equations. In fact, it might happen that a large portion of all edges attains non-zero values in such a solution. We also note that we cannot hope to find a solution of such a system with a minimum number of non-zero entries, as this problem is  $\mathcal{NP}$ -hard.



**Theorem 2.3** *For a mesh with branchings and desired subdivision numbers it is strongly  $\mathcal{NP}$ -hard to find a feasible refinement such that the sum of deviations from the desired subdivision numbers over all edges is minimized.*

**Proof:** *Exact Cover by 3-Sets* is the following problem. Given a ground set  $U = \{u_1, \dots, u_{3m}\}$  and a family  $S_1, S_2, \dots, S_n$  of subsets of  $U$ , each of cardinality three, is there a subfamily of  $m$  subsets that covers  $U$ ? This problem is known to be strongly  $\mathcal{NP}$ -complete [GJ79].

The reduction from *Exact Cover by 3-Sets* to an instance of the mesh refinement problem is as follows. Given an instance of *Exact Cover by 3-Sets*, denote by  $k_i$  the total number of appearances of  $u_i$  within the subsets  $S_j$ . For each element  $u_i$  we define a “ground set polygon”  $P_i$  with  $p_i \geq \max\{k_i, 3\}$  segments, where  $p_i$  is chosen odd and all segments are just single edges. These polygons are pairwise disjoint. The first  $k_i$  edges of such a polygon get  $(0, 1)$ -capacities, all others are fixed to zero.

For each subset  $S_j$  we create a (branching) edge  $e_{S_j}$  with  $(0, 1)$ -capacities. Such an edge becomes a branching edge when we connect it by quadrilaterals to the three “ground set polygons” which correspond to the elements contained in  $S_j$ . More precisely, if  $u_i$  is contained in  $S_j$ , we introduce a quadrilateral which is incident to  $e_{S_j}$  and one of the  $k_i$  free edges of  $P_i$  (in such a way that these two edges are not adjacent and no edge of  $P_i$  becomes a branching edge). The two remaining edges of such a quadrilateral are fixed to zero.

As each  $P_i$  is an odd polygon, at least one of its free edges has to be subdivided in a feasible refinement. A subdivision of such an edge enforces the subdivision of the branching edge  $e_{S_j}$  to which  $P_i$  is connected by some quadrilateral. Hence, in any feasible refinement all elements  $u_i$  are covered by at least one  $S_j$ , if we interpret subdividing as covering.

If the desired subdivision number is zero for all edges, then it is easy to see that an exact cover corresponds to a feasible refinement with a total deviation of  $4m$ , and any other feasible refinement has a strictly larger deviation.  $\square$

### 3 Optimizing meshes without branchings

The simple approach from the previous section just finds feasible solutions. We continue with the more ambitious task to find special solutions which fulfill certain optimization criteria. For meshes without branchings, we first show in Subsection 3.1 the correspondence of quadrangulations and feasible solutions of certain bidirected flow problems.

Afterwards, we show that a number of mesh optimization criteria can be incorporated into this model and solved efficiently by a single minimum cost bidirected flow problem.

Moreover, we show that we can also model the following more general kind of density constraints. In a cooperation with engineers from a German company, the SFE Gesellschaft für Strukturanalyse in Forschung und Entwicklung mbH, Berlin, we have been asked to have a strict and local density control not only on a per edge basis, but also to control the density for a whole segment. This is a non-trivial distinction, as, typically, the input mesh is not conformal (see Figure 1 for an example). Therefore, we would like to model both requirements simultaneously. As this can be done without problems for meshes without branchings, we use two different kinds of *density constraints* for that purpose:

- I. For an edge  $e \in E$ , the subdivision number  $x_e$  is at least  $\ell_e$  and at most  $u_e$ :  
 $\ell_e \leq x_e \leq u_e$  (*lower and upper edge capacities*,  $(\ell_e, u_e)$ -capacities, for short).
- II. For a segment  $S$  of a polygon, the sum of the subdivision numbers over all edges of  $S$  is at least  $\ell_S$  and at most  $u_S$ :  $\ell_S \leq \sum_{e \in S} x_e \leq u_S$  (*lower and upper segment capacities*).

### 3.1 Mesh refinement as a bidirected flow problem

**Bidirected flows.** Bidirected flow problems can be defined in several (equivalent) ways (see [Edm67, Der88]), we will henceforth use the following setting:

Let  $\tilde{G} = (\tilde{V}, \tilde{E})$  be an undirected graph (loops and parallel edges allowed), and for  $\tilde{e} \in \tilde{E}$  let  $u_{\tilde{e}} \geq \ell_{\tilde{e}} \geq 0$  be the *upper* and *lower capacity* of edge  $\tilde{e}$ . Given an integer vector  $x \in \mathbb{Z}^{\tilde{E}}$  and a subset  $F \subseteq \tilde{E}$  we abbreviate  $x(F) := \sum_{\tilde{e} \in F} x_{\tilde{e}}$ .

For each vertex  $\tilde{v} \in \tilde{V}$ , there is a *vertex capacity*  $b_{\tilde{v}} \in \mathbb{Z}$ . Vertices with  $b_{\tilde{v}} = 0$  are called *transshipment vertices*.

For each vertex  $\tilde{v} \in \tilde{V}$ , the set of all incident edges is partitioned into two parts,  $A_1(\tilde{v})$  and  $A_2(\tilde{v})$ , where  $A_2(\tilde{v})$  may be the empty set. We define  $\delta(A_i(\tilde{v}))$  as the set of non-loop edges in  $A_i(\tilde{v})$ , and  $\gamma(A_i(\tilde{v}))$  as the set of loops within  $A_i(\tilde{v})$ , for  $i = 1, 2$ .

An integer weighting  $x \in \mathbb{Z}^{\tilde{E}}$  of all edges in  $\tilde{E}$  is a *feasible bidirected flow* if and only if

- (1)  $\ell_{\tilde{e}} \leq x_{\tilde{e}} \leq u_{\tilde{e}}$  for each edge  $\tilde{e} \in \tilde{E}$  (*edge capacity constraints*),
- (2)  $x(\delta(A_1(\tilde{v}))) + 2x(\gamma(A_1(\tilde{v}))) - x(\delta(A_2(\tilde{v}))) - 2x(\gamma(A_2(\tilde{v}))) = b_{\tilde{v}}$   
for each vertex  $\tilde{v} \in \tilde{V}$  (*flow conservation constraints*).

The *imbalance*  $imb(\tilde{v})$  of a vertex  $\tilde{v}$  is the absolute amount by which the flow conservation constraint is violated, i.e.  $imb(\tilde{v}) := |b_{\tilde{v}} - x(\delta(A_1(\tilde{v}))) - 2x(\gamma(A_1(\tilde{v}))) + x(\delta(A_2(\tilde{v}))) + 2x(\gamma(A_2(\tilde{v})))|$ .

Note that a bidirected flow is a proper generalization of the usual flow definition for a directed graph, where the orientation of the edges induces the bipartition  $A_1(\tilde{v}), A_2(\tilde{v})$  in a simple way, namely into incoming and outgoing edges for each vertex  $\tilde{v}$ .

**Single polygons.** For a single polygon  $P$  with  $p$  segments we now define a small bidirected flow problem on a graph  $\tilde{G}_P = (\tilde{V}_P, \tilde{E}_P)$ . See Figure 6 for an example of a polygon with five segments. Let us suppose that we are given some nonnegative integer  $N_i$  for each segment  $S_i$ , such that the segment  $S_i$  has to be subdivided into  $N_i + 1$  edges in the refinement. (Clearly, the  $N_i$  should be consistent with the given segment capacities.)

The vertex set  $\tilde{V}_P$  consists of vertices  $\tilde{v}_i$  and  $\tilde{w}_i$  corresponding to each segment  $S_i$ , and of one additional vertex  $\tilde{v}_c$  ( $c$  for central), if  $p$  is odd.

The edge set  $\tilde{E}_P = \tilde{E}^{in} \cup \tilde{E}^{cap} \cup \tilde{E}^{par}$  consists of

- the set  $\tilde{E}^{in}$  which contains all pairs  $(\tilde{v}_i, \tilde{v}_j)$ , for  $i, j = 1, \dots, p$ , including the loops,
- the set  $\tilde{E}^{cap}$  which contains an edge  $(\tilde{v}_i, \tilde{w}_i)$  for each segment  $S_i$ , and
- the set  $\tilde{E}^{par}$  which contains an edge  $(\tilde{v}_i, \tilde{v}_c)$  for each segment  $S_i$ , if  $p$  is odd, and is empty otherwise.

The edges in  $\tilde{E}^{in}$  are “internal” edges of the polygon, those in  $\tilde{E}^{cap}$  will be used to model the segment capacity constraints, and edges in  $\tilde{E}^{par}$  are used to ensure the “correct parity” of the subdivision if the polygon initially is odd. We assign the following capacities to the edges:

- For an edge  $\tilde{e} \in \tilde{E}^{in}$ , we set  $\ell_{\tilde{e}} := 0$  and  $u_{\tilde{e}} := +\infty$ .
- For an edge  $\tilde{e} = (\tilde{v}_i, \tilde{w}_i) \in \tilde{E}^{cap}$  which corresponds to segment  $S_i$  of  $P$ , we set  $\ell_{\tilde{e}} := \ell_{S_i}$  and  $u_{\tilde{e}} := u_{S_i}$ .

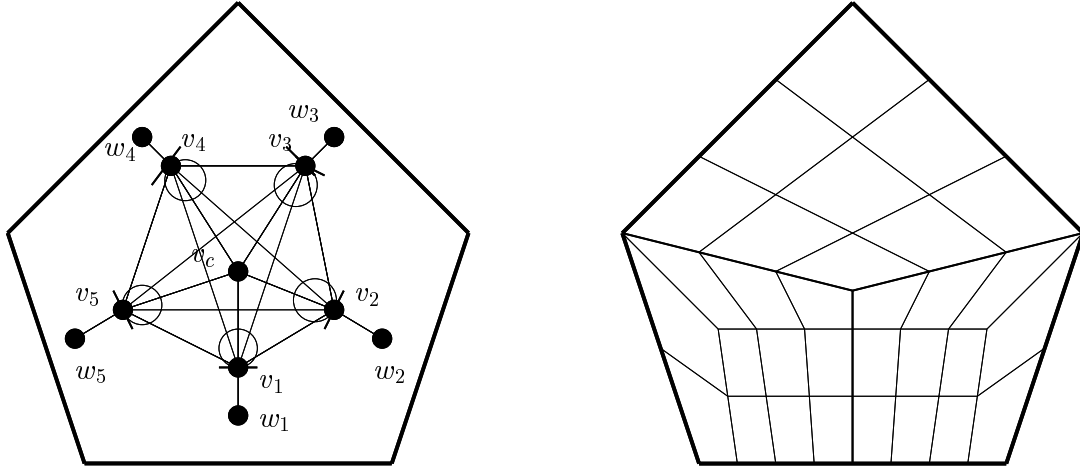


Figure 6: Example: the bidirected flow graph for a polygon with 5 segments and a possible refinement. A short solid line through a vertex is used to indicate a non-trivial bipartition of its adjacency list into two parts.

- For an edge  $\tilde{e} \in \tilde{E}^{par}$ , we set  $\ell_{\tilde{e}} := 0$  and  $u_{\tilde{e}} := 1$ .

The bipartition of the vertices is as follows: For each vertex  $\tilde{v}_i$ , the incident edges are partitioned into the sets  $\tilde{E}^{in} \cup \tilde{E}^{par}$  and  $\tilde{E}^{cap}$ . For each vertex  $\tilde{w}_i$  and for  $\tilde{v}_c$  (if the latter vertex exists), all incident edges are in  $A_1(\cdot)$ , and  $A_2(\cdot)$  is empty.

We define  $b_{\tilde{v}_c} = 1$ , if  $p$  is odd, and all other vertices  $\tilde{v}_i$  are transshipment vertices. Finally, let  $b_{\tilde{w}_i} = N_i$  for the given nonnegative integers  $N_i$ . This completes the definition of a bidirected flow problem for the polygon  $P$ .

**Proposition 3.1** *The bidirected flow problem for the polygon  $P$  as defined above has the following properties:*

- (1) *If the polygon  $P$  admits a feasible conformal refinement such that the segment  $S_i$  consists of exactly  $N_i + 1$  edges in that refinement, then there is a feasible bidirected flow.*
- (2) *For any feasible flow  $x$ , there is a conformal refinement of  $P$  into strictly convex quadrilaterals such that exactly  $x_{(\tilde{v}_i, \tilde{v}_j)}$  disjoint paths go from the interior of segment  $S_i$  to the interior of segment  $S_j$ , and the segment  $S_i$  consists of  $N_i + 1$  edges in the refinement, for  $i = 1, \dots, p$ , and all segment capacity constraints are fulfilled.*
- (3) *If there is a feasible flow, there is also a feasible flow  $x'$  such that at most  $p$  edges within the set  $\tilde{E}^{in}$  have a non-zero flow value.*

**Proof:** Suppose first that the polygon  $P$  admits a feasible conformal refinement such that each segment  $S_i$  consists of exactly  $N_i + 1$  edges. Then, by Lemma 2.1,  $\sum_{i=1}^p (N_i + 1)$  must be even. If  $p$  is odd, this means that  $\sum_i N_i$  is odd, and this sum is even otherwise. We construct a feasible flow by the following procedure:

For all edges  $(\tilde{v}_i, \tilde{w}_i) \in \tilde{E}^{cap}$ , we set  $x_{(\tilde{v}_i, \tilde{w}_i)} := N_i$ , which clearly fulfills the capacity constraints, as the conformal refinement of  $P$  with  $N_i + 1$  edges on segment  $S_i$  is feasible by assumption. Initialize all other edges with a zero flow value. At this stage, the imbalance for the transshipment vertices  $\tilde{v}_i$  is  $imb(\tilde{v}_i) = |0 - N_i|$ , for all  $i = 1, \dots, p$ .

Figure 7: Step 1: “Quadrangulation of  $P'$ .”

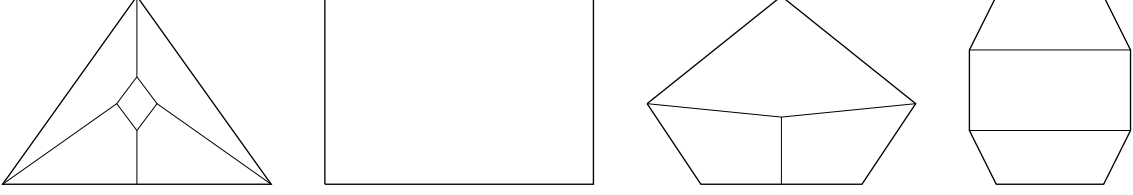


Figure 8: Step 2: “Routing of the flow on the loops.”

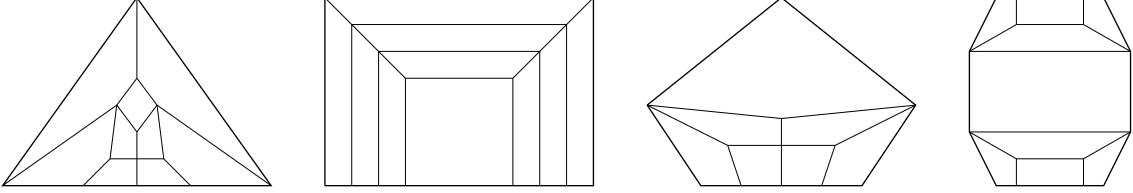
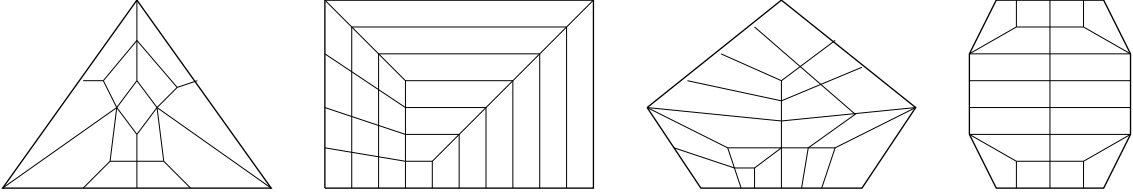


Figure 9: Step 3: “Routing of the flow on all other edges.”



If  $p$  is odd, there must be some index  $k$  such that  $N_k$  is strictly positive, as  $\sum_i N_i$  is odd. In this case, we take one such index and set  $x_{(\tilde{v}_k, \tilde{v}_c)} := 1$ .

This makes the total imbalance  $imb^{tot} := \sum_{i=1}^p imb(\tilde{v}_i)$  even, regardless of the parity of  $p$ . As long as there are two vertices  $\tilde{v}_i$  and  $\tilde{v}_j$  with a non-zero imbalance, we set  $x_{(\tilde{v}_i, \tilde{v}_j)} := \min\{imb(\tilde{v}_i), imb(\tilde{v}_j)\} > 0$ . As a result, at least one more vertex fulfills its flow conservation constraint. Observe that this always preserves the invariant that the total imbalance  $imb^{tot}$  is even. Hence, we end up at a point where at most one vertex  $\tilde{v}_i$  may be imbalanced, but  $imb(\tilde{v}_i)$  is even. Thus we may finally use the loop at  $\tilde{v}_i$  and set  $x_{(\tilde{v}_i, \tilde{v}_i)} := imb(\tilde{v}_i)/2$  to get a feasible bidirected flow. This proves property (1), but also (3), as no more than  $p$  edges in  $\tilde{E}^{in}$  can receive a non-zero flow value in this procedure.

To establish property (2), we have to show how to build a conformal refinement when a feasible bidirected flow  $x$  is given.

This is done in several steps: In the first step, we take a “geometric copy”  $P'$  of  $P$ , where we regard only the corners  $v_1, \dots, v_p$  of  $P$  (in counterclockwise order) as vertices of  $P'$  and take the whole segments of  $P$  as edges for  $P'$ . We decompose  $P'$  into strictly convex quadrilaterals. If the number of segments is even, this can be done by the insertion of edges between corners. More precisely, as long as  $P'$  is not completely decomposed into quadrilaterals, we choose corners  $v_i, v_{i+3}$  and add an edge  $(v_i, v_{i+3})$ . If  $P'$  has an odd number of segments, the feasible flow contains exactly one edge  $(\tilde{v}_i, \tilde{v}_c)$  with a flow of one unit. Vertex  $\tilde{v}_i$  corresponds to segment  $S_i$  which is spanned by  $v_i$  and  $v_{i+1}$ . If this segment does not contain a non-corner vertex of  $P$ , we place one subdivision point somewhere on this segment, otherwise we choose such a non-corner vertex. Call this original vertex or the new subdivision point  $w$ . If  $p \geq 5$ , insert a new vertex  $w'$  somewhere in the interior of the convex region spanned by  $w, v_{i-1}$ , and  $v_{i+2}$ , and insert the edges  $(w', w), (w', v_{i-1}), (w', v_{i+2})$ . This decomposes  $P'$  into two strictly convex quadrilaterals and some smaller polygon  $P''$ , which is now even. Hence we can complete the decomposition of

$P'$  into quadrilaterals in the same way as for a polygon with an even number of segments. The case  $p = 3$  is an exception, but can be treated as depicted in Figure 7.

If there is a flow value of  $x_{(\tilde{v}_i, \tilde{v}_j)}$  units on edge  $(\tilde{v}_i, \tilde{v}_j)$ , we want to construct so many disjoint paths between segment  $S_i$  and  $S_j$ . We construct them path by path in the following way: We first place two (disjoint) end-vertices, one on segment  $S_i$ , the other on segment  $S_j$  of  $P'$ . (For loops  $(\tilde{v}_i, \tilde{v}_i)$  these two segments coincide.) As long as there are non-corners on  $S_i$  or  $S_j$  which have not been used as end-vertices before, we choose them as the end-vertices, in order to guarantee that, finally, all non-corners of  $P$  will be used. Otherwise we may select any position disjoint from previously determined vertices. In every case, each end-vertex is incident to a quadrilateral of the current decomposition of  $P'$ . If both end-vertices are incident to the same quadrilateral of  $P'$  we can directly use one of the possibilities shown in Figure 10 to embed the path between these vertices and iterate.

Otherwise, consider the dual graph of the decomposition  $G'$  of  $P'$  we have constructed so far and some dual path between the two quadrilaterals incident to the two end-vertices.

Such a path induces a chain of quadrilaterals where two adjacent quadrilaterals share exactly one edge, which we subsequently subdivide by the insertion of a new vertex for each such edge. Now each quadrilateral on the dual path has two new vertices, but we can rebuild a conformal decomposition into strictly convex quadrilaterals by using one of the possibilities for each of these quadrilaterals as shown in Figure 10.

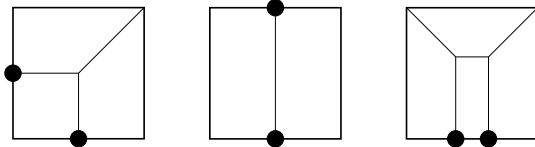


Figure 10: The different possibilities (up to symmetry) to complete the decomposition of a quadrilateral with two additional vertices on the boundary.

It is easy to see that one can select the end-vertices and dual paths in such a way, that all constructed paths corresponding to the flow on edge  $(\tilde{v}_i, \tilde{v}_j)$  are disjoint. This establishes property (2).  $\square$

**Meshes without branchings.** We now extend our definition of a bidirected flow instance for a single polygon to an instance for a whole mesh. Let  $G = (V, E)$  be the undirected graph of a mesh without branching edges.

The underlying graph  $\tilde{G} = (\tilde{V}, \tilde{E})$  of the bidirected flow instance is built up using the graphs  $\tilde{G}_P = (\tilde{V}_P, \tilde{E}_P)$  as defined for single polygons as subgraphs. Roughly speaking, two such subgraphs are connected by a “dual edge” if the corresponding polygons share an edge in  $G$ .

The vertex set  $\tilde{V}$  is the union of

- the vertex sets  $\tilde{V}_P$  of all polygons  $P$ ,
- all vertices among  $V$  which are non-corners of some polygon, and
- one special vertex  $\tilde{v}_{out}$ .

The edge set  $\tilde{E}$  contains the following edges:

- We take the union of the edges sets  $\tilde{E}_P$  over all polygons  $P$ , and the edge capacities as in  $\tilde{G}_P$ .

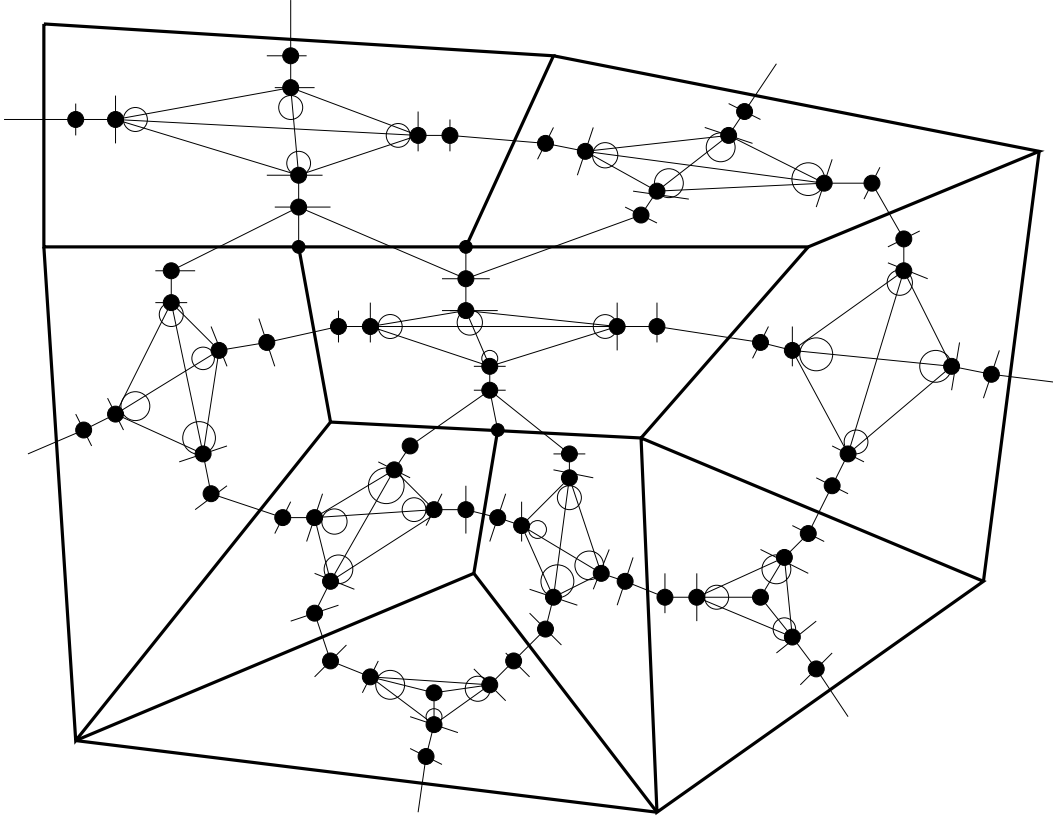


Figure 11: The graph  $\tilde{G} = (\tilde{V}, \tilde{E})$  of the bidirected flow problem for an artificial instance. (The special vertex  $\tilde{v}_{out}$  is omitted in this figure.)

- Let  $e \in E$  be an edge which belongs to two polygons, say to  $P$  and  $Q$ , and suppose that  $\tilde{w}_i \in \tilde{V}_P$  and  $\tilde{w}_j \in \tilde{V}_Q$  are the vertices which correspond to the segments of  $P$  and  $Q$  to which  $e$  belongs. For each such edge  $e$ , we introduce a “dual” edge  $\tilde{e} = (\tilde{w}_i, \tilde{w}_j)$ , with edge capacities equal to that of  $e$  in  $G$ .
- If an edge  $e \in E$  belongs only to one polygon, say to  $P$ , and  $\tilde{w}_i \in \tilde{V}_P$  is the vertex which corresponds to the segment of  $P$  to which  $e$  belongs, then we introduce a “dual” edge  $\tilde{e} = (\tilde{w}_i, \tilde{v}_{out})$ , with edge capacities equal to that of  $e$  in  $G$ .
- For each polygon  $P$  and each vertex  $v \in V \cap P$  which is not a corner, we introduce an edge  $\tilde{e} = (\tilde{w}_i, v)$ , where  $v$  lies on the segment of  $P$  to which  $\tilde{w}_i \in \tilde{V}_P$  corresponds. Such an edge gets identical lower and upper capacities  $\ell_{\tilde{e}} := u_{\tilde{e}} := 1$ .
- A loop  $\tilde{e} = (\tilde{v}_{out}, \tilde{v}_{out})$  with capacities  $\ell_{\tilde{e}} := 0$  and  $u_{\tilde{e}} := +\infty$ .

The bipartition of the vertices is as follows: For each vertex  $\tilde{w}_i \in \tilde{V}_P$ , we put exactly all incident edges  $\tilde{e} \in \tilde{E} \setminus \tilde{E}_P$  into  $A_2(\tilde{w}_i)$ . For each vertex  $v \in V$ , all incident edges are in  $A_1(v)$ , and  $A_2(v)$  is empty. All incident edges to  $\tilde{v}_{out}$  are in  $A_1(\tilde{v}_{out})$  except  $(\tilde{v}_{out}, \tilde{v}_{out})$  which belongs to  $A_2(\tilde{v}_{out})$ .

All vertices  $\tilde{w}_i \in \tilde{V}_P$  which correspond to segments of  $P$  become transshipment vertices (in contrast to the case of single polygons). For each vertex  $\tilde{v} \in \tilde{V}$ , let  $b_{\tilde{v}}$  be equal to the number of incident edges in  $\tilde{G}$ . Finally, let  $b_{\tilde{v}_{out}} := 0$  if  $\sum_{\tilde{v} \in \tilde{V} \setminus \tilde{v}_{out}} b_{\tilde{v}}$  is even, and  $b_{\tilde{v}_{out}} := 1$ , otherwise.

**Theorem 3.2** *There exists a feasible conformal refinement for the homogeneous mesh  $G$  if and only if the bidirected flow problem as defined above has a feasible bidirected flow.*

**Proof:** Suppose first that there is a feasible conformal refinement for the mesh  $G$  with edge  $e \in E$  having a subdivision number of  $x_e$ . Then the assignment  $x_{\tilde{e}} := x_e$  for each dual edge  $\tilde{e}$  of  $e$  respects the edge capacities. All edges in  $\tilde{E}$  incident to vertices of  $V$  have a fixed flow value of one unit in any feasible flow, hence there is no choice for the assignment of these edges. The assignment of flow values we determined so far induces for each polygon of  $G$  a bidirected flow problem as in Proposition 3.1. Clearly, the feasible conformal refinement for the whole mesh induces a feasible conformal refinement for each individual polygon, and so by Proposition 3.1 there is a feasible bidirected flow for all polygons. Notice, finally, that we can always fulfill the flow balance constraint of  $\tilde{v}_{out}$  by an appropriate flow assignment to the loop edge  $(\tilde{v}_{out}, \tilde{v}_{out})$ .

Conversely, it is easy to see how to convert a feasible bidirected flow into a feasible conformal mesh refinement. We just notice that the restriction of the flow for the whole problem to the individual polygons yields conformal refinements for the polygons again by Proposition 3.1. Conformality for the whole mesh is guaranteed if we subdivide each edge  $e$  of  $G$  exactly  $x_{\tilde{e}}$  times. By definition of the flow problem, all edge and segment capacities are fulfilled for the mesh.  $\square$

### 3.2 Optimizing the mesh quality

The task to find special solutions which fulfill certain optimization criteria instead of just feasible solutions leads to *minimum cost bidirected flow problems*. Let  $\tilde{G} = (\tilde{V}, \tilde{E})$  be the graph of a bidirected flow instance where a cost  $c_{\tilde{e}}$  is associated with every edge  $\tilde{e}$ . Then the *minimum cost bidirected flow problem* seeks for a feasible bidirected flow with minimum cost  $\sum_{\tilde{e} \in \tilde{E}} c_{\tilde{e}} x_{\tilde{e}}$ . We will consider three kinds of mesh quality criteria:

- control over the mesh density,
- avoidance of too small or too large angles, and
- “regularity” of the overall mesh structure.

**Density control.** To get control over the density of a mesh, one can use a convex cost function for each dual edge with a minimum at the desired density  $d_{\tilde{e}}$ . The deviation from the desired density is punished with increasing costs. As we allow only integer flows, the cost functions can be assumed to be piecewise linear. For practical purposes, it will often suffice to use piecewise linear functions with only two slopes, as depicted in Figure 12.

This defines a minimum convex cost bidirected flow problem. The standard way to transform such a problem to an ordinary linear minimum cost bidirected flow problem is to replace each edge  $\tilde{e}$  by as many copies as there are slopes (cf. [AMO93, pages 551ff.]). So if there are  $p$  slopes with cost coefficients  $c_{\tilde{e}}^k$  and breakpoints at  $d_{\tilde{e}}^k$ , the  $k$ -th copy  $\tilde{e}^k$  gets edge capacities  $[0, d_{\tilde{e}}^k - d_{\tilde{e}}^{k-1}]$  and a cost coefficient of  $c_{\tilde{e}}^k$ . (Here, we assume  $d_{\tilde{e}}^0 = \ell_{\tilde{e}}$  and  $d_{\tilde{e}}^p = u_{\tilde{e}}$ .)

For ease of exposition, we will discuss the next two optimization criteria only for polygons with four segments.

**Angle control and mesh structure.** Recall from Proposition 3.1 that, for a polygon with four segments, we may assume that we have a feasible flow with at most four edges with a non-zero flow within the set  $\tilde{E}^{in}$ . Furthermore, there is a conformal refinement such that exactly  $x_{(\tilde{v}_i, \tilde{v}_j)}$

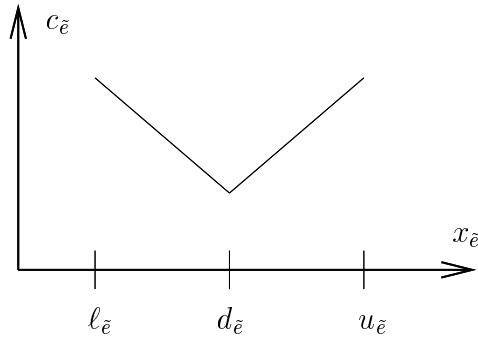


Figure 12: Piecewise linear convex cost functions for density control.

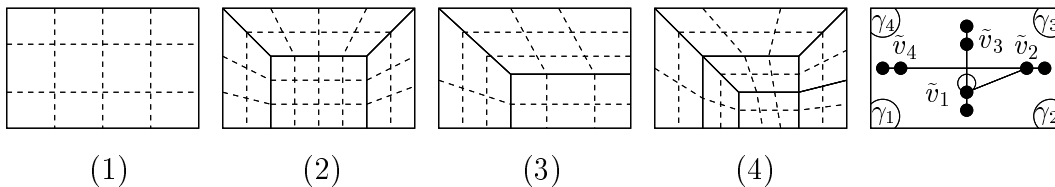


Figure 13: Illustration of the four templates for polygons with four segments. Solid lines are mandatory for each template (up to rotation and symmetry), whereas the number of dashed lines may vary. The rightmost figure shows the corresponding subgraph of the bidirected flow instance with only those edges displayed which may have a non-zero flow value.

disjoint paths go from the interior of segment  $S_i$  to the interior of segment  $S_j$ . Hence, for an appropriate relabeling of the segments, conformal subdivisions of polygons with four segments can be assumed to be of the form as shown in Figure 13. The first three possibilities are the *standard templates* used in [MMW96, TA93], whereas the last template is new and generalizes all standard templates.

The shape of the quadrilaterals in the refinement is closely related to the choice of the template which determines the refinement of the macro element. As a rule of thumb, the more the polygon looks like a trapezoid, the better template (2) will be, the more it looks like a kite, the better template (3) will be. In all other cases, template (1) is likely to be the best.

Moreover, template (1) tends to produce a fairly regular mesh. “Regularity of the mesh structure” is a mesh quality criterion which seemingly cannot be fully formalized. But the heuristical rule to prefer template (1) often achieves practical results which reflects such a goal quite satisfactorily [MMW96].

Hence, we would like to refine as many polygons by template (1) as possible (among those polygons which have no very small or large angle), or more generally, we would like to maximize the number of macro elements which are refined to some preferred template, for a given preference order for each individual polygon of our instance. Unfortunately, such a goal is intractable.

**Theorem 3.3** *Given a feasible homogeneous mesh instance, it is strongly  $\mathcal{NP}$ -hard to find a solution where the number of macro elements with four segments which are refined according to the  $(m \times n)$ -grid template (i.e. template (1) in Figure 13) is maximized.*

**Proof:** The proof of Theorem 2.4 in [MMW96] directly carries over to this more general case.  $\square$



So what can we hope for? As the mesh refinement problem only allows to insert new vertices and edges, no sharp input angle can be erased. Hence, we can only try to avoid the creation of *new* angles smaller than  $\alpha$ . On the other hand, we can try to enforce the splitting of an angle larger than  $\beta$ .

Observe that, for the given templates in Figure 13, a non-zero flow on edge  $(\tilde{v}_1, \tilde{v}_2)$  induces the splitting of the angle  $\gamma_4$ , and a non-zero flow on the loop  $(\tilde{v}_1, \tilde{v}_1)$  induces the splitting of both  $\gamma_4$  and  $\gamma_3$ .

There are two possibilities to modify our bidirected flow instance for these purposes: First, we can change the edge capacities of “internal edges”  $\tilde{E}^{in}$  of a polygon: In the pure feasibility problem all these edges have a lower capacity of zero and a large upper bound (“plus infinity”). Hence, we can enforce to use an edge, if we set the lower capacity to one, or we can forbid an edge completely, if we set the upper capacity to zero. The disadvantage of this approach is, that such a modification may cause the infeasibility of the bidirected flow problem.

A second (and not so restrictive) way is to assign costs to these edges in order to make them more attractive or unattractive.

Using these ideas we can express our preferences for the choice of the chosen template for each polygon. So if we prefer a realization of template (1) for some polygon, we assign high cost coefficients to all internal edges but  $(\tilde{v}_1, \tilde{v}_3)$  and  $(\tilde{v}_2, \tilde{v}_4)$ , which get zero cost. Or, for example, if the angle  $\gamma_4$  should be split by an application of template (3), we either raise the lower bound for edge  $(\tilde{v}_1, \tilde{v}_2)$  to one or make its cost coefficient negative.

Hence, in summary, we can use local information about the geometry of the unrefined polygon and so can model our preferences of the choice of an appropriate template and enforce or forbid the splitting of macro element angles by these modifications.

Similar ideas work for polygons with a different number of segments than four. We give an example for a polygon with three segments: If we set  $b_c := 3$  (rather than  $b_c := 1$ ) for the vertex  $\tilde{v}_c$  of such a polygon, the parity is still even for every feasible flow, but we can use a refinement which does not necessarily splits all angles, as in the leftmost refinement in Figure 7. In contrast, we can use one single interior vertex and connect this vertex to one subdivision point on each segment to yield a quadrangulation in step 1. We should also note that this is in fact the standard way to treat polygons with three segments [MMW96].

## 4 Quadrangulations of meshes with branchings

**A decomposition approach for meshes with branchings.** In Section 2 we gave a general and simple procedure to find just feasible quadrangulations for meshes with branchings. But optimization of the mesh quality is highly desirable for meshes with branchings, too.

For these reasons, we introduce a mesh decomposition into homogeneous components. This allows us to combine the general approach from Section 2 and the bidirected flow approach for homogeneous components into a two-phase approach: In a first phase, we determine the subdivision numbers for all branching edges by solving a modified system of linear equations over  $GF(2)$  which we explain below. Afterwards, in a second phase, we solve a minimum cost bidirected flow problem for each homogeneous component where the flow values for all branching edges are predetermined from the first phase.

For all instances from real-world applications we know of, the number of branchings is relatively small in comparison with the number of all polygons. Hence, there is reason to assume that not too many edges have a prescribed flow value in the second phase and that solving the minimum cost bidirected flow problem is worth its effort.

Recall from the Introduction that a combinatorial description of a mesh consists of a graph

$G = (V, E)$  and a hypergraph  $H = (\mathcal{M}, \{E_1, \dots, E_m\})$ .

Depending on the number of polygons they belong to, the edges of  $G$  can be partitioned into sets  $E^1, E^2$ , and  $E^{\geq 3}$ . The set  $E^1$  contains the *boundary edges*, i.e. edges which belong to exactly one polygon, the set  $E^{\geq 3}$  contains all branching edges, and  $E^2$  all remaining edges. A *non-branching path* in  $H$  is a path between two polygons  $P_1, P_2 \in \mathcal{M}$  which contains only hyperedges of cardinality two, i.e. hyperedges corresponding to edges in  $E^2$ . Being connected by a non-branching path whose edges are all free is an equivalence relation on the set of polygons. Its equivalence classes are exactly the *homogeneous components* of the mesh decomposition.

Such a *decomposition into homogeneous components* of a mesh with branchings can be obtained by the following procedure which “splits” all branching edges and fixed edges: A branching edge which is incident to  $p$  polygons is replaced by  $p$  copies, once for each polygon. The copied edges are treated as boundary edges, and they get the same capacities as the original ones. All fixed edges in  $E^2$  are replaced in the same way by two copies, once for each incident polygon. Let  $G_1, \dots, G_k$  be the resulting homogeneous components, and  $\mathcal{M}_1, \dots, \mathcal{M}_k$  the sets of polygons contained in these components. The decomposition into homogeneous components can be easily implemented by a straightforward depth or breadth first search.

For a homogeneous component, we may define a certain variant of the dual graph, denoted by  $G^D = (V^D, E^D)$ . Herein,  $V^D$  is the set of all polygons of the component plus one additional vertex for the region outside the component. For each edge  $e \in E^2$  between two polygons, there is an edge  $e^D \in E^D$  connecting these two polygons. Analogously, for each boundary edge  $e \in E^1$ , there is an edge  $e^D$  that connects the polygon incident to  $e$  with the boundary vertex introduced for this component.

Later we will need the following result:

**Lemma 4.1** *Let  $G = (V, E)$  be a homogeneous mesh, where the subdivision numbers  $x_e$  are fixed for all boundary edges and  $\sum_{e \in E^1} (x_e + 1)$  is even, but all other edges are free. Then  $G$  has a feasible conformal refinement.*

**Proof:** For each edge  $e \in E \setminus E^1$ , choose some integer  $\tilde{x}_e$  within the interval  $[\ell_e, u_e]$ , and for all other edges  $e \in E^1$  let  $\tilde{x}_e = x_e$ .

We call a vertex  $v^D$  of  $G^D$  (i.e. a polygon of  $G$ ) *odd (even)* if  $\sum (\tilde{x}_e + 1)$  is odd (even), where the sum is taken over all edges  $e \in E$  which are in one-to-one correspondence to dual edges  $e^D \in E^D$  incident to  $v^D$ . As the sum over all boundary edges of the subdivision numbers plus one, namely  $\sum_{e \in E^1} (\tilde{x}_e + 1)$ , is even, the dual graph  $G^D$  has an even number of odd vertices which correspond to polygons of the mesh.

By Lemma 2.1, we have to modify  $\tilde{x}$  such that all polygons become even. Hence, we look for a subgraph  $G^d$  of  $G^D$  such that a vertex is odd in  $G^d$  if and only if it is odd in  $G^D$ .

To achieve this task, let  $T$  be a spanning tree of  $G^D$  without the boundary vertex and let  $F$  be the set of all edges of  $T$  whose deletion divides  $T$  into two subtrees with an odd number of odd vertices each.

Then we change the subdivision number by one unit for each edge of  $G$  that corresponds to an edge of  $F$ . It is easy to see that each polygon becomes even by that modification. Just note that for each vertex  $v^D$  the set of incident edges with respect to  $T$  induces by subtree containment a partition of all odd vertices. If  $v^D$  is even, then there is an even number of induced subtrees with an odd number of odd vertices, and we take by construction exactly so many edges into  $F$ . A similar argument works for the case that  $v^D$  is even.

As all edges in  $F$  are free, we can do a change by one unit without violating the edge capacities.  $\square$

Note that the proof is not only constructive, but also yields a linear-time algorithm (linear in the size of the combinatorial description of the input mesh, i.e. in the number of its vertices

and edges) for determining subdivision numbers of a feasible refinement of homogeneous meshes with edge but without segment capacities.

Let  $H' = (\mathcal{M}', \{E'_1, \dots, E'_{m'}\})$  be the hypergraph which we obtain from  $H = (\mathcal{M}, \{E_1, \dots, E_m\})$  if we

- (1) delete all hyperedges  $E_i$  where  $e_i$  is fixed, then
- (2) contract all those hyperedges of degree two which correspond to free edges in  $E^2$ , and finally,
- (3) identify all parallel hyperedges, i.e. we keep only one hyperedge for all edges  $e \in E$  which are incident to exactly the same set of polygons.

Observe that there is a one-to-one correspondence between the vertices of  $H'$  and the homogeneous components  $G_1, \dots, G_k$  of  $G$ .

We will now define a smaller system of equations over  $GF(2)$  based on  $H'$ . We start similar as in the first approach above. Again, let  $\tilde{x}_e$  be some fixed integer in the interval  $[\ell_e, u_e]$ , for each edge  $e \in E$ , and define for each polygon  $P \in \mathcal{M}$  the parity number  $b_P := \sum_{e \in P} (\tilde{x}_e + 1) \bmod 2$ .

These parity numbers are aggregated to parity numbers for each homogeneous component, by setting  $b_{G_i} := \sum_{P \in \mathcal{M}_i} b_P \bmod 2$ , for  $i = 1, \dots, k$ .

For each hyperedge  $E'_j \in H'$ , let  $y_{E'_j}$  be a 0/1-integer variable. For each homogeneous component  $G_i$  which is not incident to a free boundary edge in  $G$ , we take an equation of the form

$$\sum_{E'_j: E'_j \cap G_i \text{ odd}} y_{E'_j} = b_{G_i},$$

that is, we sum over those hyperedges which are derived from a branching edge with an odd number of incidences with the homogeneous component  $G_i$ . Notice, that it might happen that some set  $\{E'_j : E'_j \cap G_i \text{ odd}\}$  is empty (because of the deletion of fixed edges, for example). In that case, we define the sum over the empty set to be zero. This defines our system of linear equations over  $GF(2)$ .

Suppose that this system has a solution  $\tilde{y}$ . For each hyperedge  $E'_i$  with  $\tilde{y}_{E'_i} = 1$ , take one corresponding branching edge  $e = e_i$  (there may be a choice if  $E'_i$  resulted from parallel hyperedges of a branching), and set  $x_e := \tilde{x}_e \pm 1$  (such that the edge capacities remain fulfilled). For all other branching edges, set  $x_e := \tilde{x}_e$ . This completes the first phase. In the second phase, we fix the subdivision values  $x_e$  for all branching edges and then solve the bidirected flow problems for each homogeneous component separately. We claim that each bidirected flow problem is feasible if we have a feasible solution  $\tilde{y}$  in the first phase. This is summarized in the following theorem:

**Theorem 4.2** *There exists a feasible conformal refinement for a mesh  $G$  with branchings if and only if the two-phase approach as described above yields a feasible solution.*

**Proof:** If the two-phase approach yields a feasible solution, then each homogeneous component of our mesh has a feasible conformal refinement by Theorem 3.2, and as all branching edges are subdivided before the treatment of the homogeneous components, this yields an overall conformal mesh refinement.

For the other direction, let us suppose that the mesh has a feasible conformal refinement with  $x_e$  subdivision points on edge  $e$ . By Lemma 2.1, each polygon is even, thus we have

$0 = \sum_{e \in P} (x_e + 1) \pmod 2$ . Summing up over all polygons of a homogeneous component, we obtain

$$0 = \sum_{P \in G_i} \sum_{e \in P} (x_e + 1) \pmod 2.$$

By definition of the parity numbers, we also have

$$b_{G_i} = \sum_{P \in G_i} b_P = \sum_{P \in G_i} \sum_{e \in P} (\tilde{x}_e + 1) \pmod 2.$$

Combining both equations yields

$$b_{G_i} = \sum_{P \in G_i} b_P = \sum_{P \in G_i} \sum_{e \in P} (\tilde{x}_e - x_e) \pmod 2.$$

As the sum on the right hand side of this equation includes all edges in  $E^2$  twice, the branching edges with an even number of incidences with  $G_i$  appear an even number of times, and fixed edges do not contribute to the sum, this equation holds if and only if

$$b_{G_i} = \sum_{e \in G_i \setminus E^2, e \cap G_i \text{ odd}, e \text{ free}} (\tilde{x}_e - x_e) \pmod 2.$$

Suppose further that  $E'_i \in H'$  has been derived from the branching edges  $e_{1_i}, \dots, e_{s_i}$  which are incident to the same set of polygons. If we now group the edges on the right-hand side of the last equation by containment in the same branching and if we set

$$y_{E'_i} := \left( \sum_{j=1_i}^{s_i} (\tilde{x}_{e_j} - x_{e_j}) \right) \pmod 2,$$

this clearly defines a solution to our system of equations over  $GF(2)$  in the first phase.

It remains to show that we have a feasible solution to all bidirected flow instances in the second stage, for any solution  $\tilde{y}$  of the first. Hence, let  $\tilde{y}$  a solution of our system of equations. For each hyperedge  $E'_i$  with  $\tilde{y}_{E'_i} = 1$ , take one corresponding branching edge  $e$ , and set  $x_e := \tilde{x}_e \pm 1$ . For all other edges, set  $x_e := \tilde{x}_e$ . With such an assignment, the boundary of each homogeneous component  $G_i$  fulfills the precondition of Lemma 4.1. Hence, there is a feasible refinement for each homogeneous component, and by Theorem 3.2 the corresponding bidirected flow problem is feasible, too.  $\square$

**Intractability of quadrangulations with segment capacities.** Finally we show that the feasible conformal mesh refinement problem is highly intractable in the general case with branchings and segment capacities:

**Theorem 4.3** *The feasible conformal mesh refinement problem with segment capacities is strongly  $\mathcal{NP}$ -hard.*

**Proof:** The problem is in  $\mathcal{NP}$  because a feasible solution, if it exists, can obviously be checked for feasibility in polynomial time. To show  $\mathcal{NP}$ -completeness we shall give a reduction from 3SAT [GJ79]: We are given a Boolean formula consisting of  $m$  clauses  $C_1, \dots, C_m$  (in conjunctive normal form) and involving the variables  $y_1, \dots, y_n$ , where each clause contains three literals. Is the formula  $C_1 \cdot C_2 \cdot \dots \cdot C_m$  satisfiable?

Let  $F = C_1 \cdot C_2 \cdot \dots \cdot C_m$  be a formula of this type. As in many  $\mathcal{NP}$ -completeness proofs, the construction of an instance of our problem requires the design of special-purpose components:

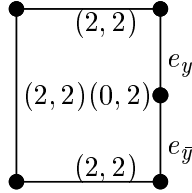


Figure 14: Representation of a variable.

For each variable  $y \in \{y_1, \dots, y_n\}$  we define a corresponding macro element with segment capacities (the pair of lower and upper capacities is indicated in parentheses, for each segment) as shown in Figure 14.

The negation of a variable is indicated by a bar. Two edges of such a macro element have a special meaning: If  $e_y$  is subdivided in a refinement, this corresponds to the truth assignment  $y := \text{TRUE}$ , otherwise, if  $e_{\bar{y}}$  is subdivided, this induces the assignment  $y := \text{FALSE}$ . Obviously, in any feasible refinement either  $e_y$  or  $e_{\bar{y}}$  is subdivided exactly once, but not both.

The clauses are represented by macro elements in the following way: For a clause  $C \in \{C_1, \dots, C_m\}$  we define a macro element with four segments and segment capacities as shown in Figure 15. The three edges  $e_{1,C}$ ,  $e_{2,C}$  and  $e_{3,C}$  are intended to correspond to the three literals of the clause. It is no problem to find a feasible refinement for this type of macro elements, but in any case there must be at least one out of the three edges  $e_{1,C}$ ,  $e_{2,C}$  and  $e_{3,C}$  which is subdivided for each clause.

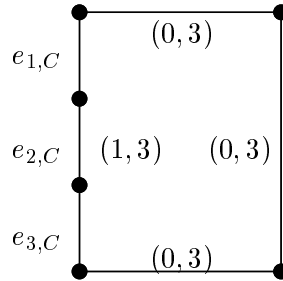


Figure 15: Representation of a clause.

Finally, we connect variables with clauses in order to make the truth assignments of all variables and clauses consistent. To this end, edge  $e_{k,C}$  of clause  $C$  is connected by a “channel” of macro elements with  $e_{y_k}$  if the  $k$ -th literal of  $C$  is  $y_k$ , or with  $e_{\bar{y}_k}$  if it is  $\bar{y}_k$ , cf. Figure 16. The subdivision numbers of certain edges of the channel are fixed (to some small constant integer  $s$ ) such that the subdivision numbers for  $e_{y_k}$  and  $e_{k,C}$  have the same parity in any feasible refinement.

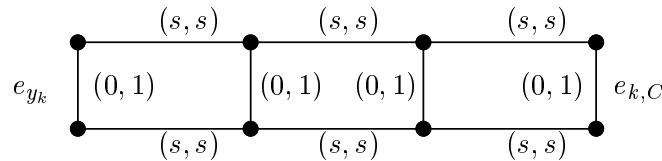


Figure 16: Connection between variables and clauses.

It is clear that there exists a feasible solution if and only if  $F$  is satisfiable, and the transformation is done in polynomial time. The problem is *strongly*  $\mathcal{NP}$ -complete, since all numbers we had to use in our reduction are small in the input length.  $\square$

## References

- [AMO93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows*, Prentice Hall, 1993.
- [Ans87] R. P. Anstee, *A polynomial algorithm for b-matching: An alternative approach*, Information Processing Letters **24** (1987), 153–157.
- [BE95] M. Bern and D. Eppstein, *Mesh generation and optimal triangulation*, Computing in Euclidean Geometry, 2nd Edition (D.-Z. Du and F. Hwang, eds.), World Scientific, Singapore, 1995, pp. 47–123.
- [BP97] M. Bern and P. Plassmann, *Mesh generation*, Handbook of Computational Geometry (J. Sack and J. Urrutia, eds.), Elsevier Science, 1997, to appear.
- [BPM<sup>+</sup>95] S. E. Benzley, E. Perry, K. Merkley, B. Clark, and G. Sjaardama, *A comparison of all hexagonal and all tetrahedral finite element meshes for elastic and elasto-plastic analysis*, Proceedings of the 4th International Meshing Roundtable, Sandia National Laboratories, Albuquerque, USA, 1995, pp. 179–191.
- [Bra93] J. R. Brauer, ed., *What every engineer should know about finite element analysis*, Marcel Decker Inc., 1993.
- [BS91] T. D. Blacker and M. B. Stephenson, *Paving: A new approach to automated quadrilateral mesh generation*, Int. J. Numer. Methods in Eng. **32** (1991), 811–847.
- [CK92] A. O. Cifuentes and A. Kalberg, *A performance study of tetrahedral and hexahedral elements in 3-D finite element structural analysis*, Finite Elements in Analysis and Design **12** (1992), 313–318.
- [Der88] U. Derigs, *Programming in networks and graphs*, Lecture Notes in Economics and Mathematical Systems, vol. 300, Springer-Verlag, Berlin, 1988.
- [Edm67] J. Edmonds, *An introduction to matching*, Lecture notes, University of Michigan, Ann Arbor, 1967.
- [Ger95] A. M. H. Gerards, *Matching*, (M. O. Ball et al., ed.), Handbooks in Operations Research and Management Science, vol. 7, North-Holland, 1995, pp. 135–224.
- [GJ79] M. R. Garey and D. S. Johnson, *Computers and intractability: A guide to the theory of  $\mathcal{NP}$ -completeness*, W. H. Freeman, 1979.
- [Ho88] K. Ho-Le, *Finite element mesh generation methods: a review and classification*, Computer-Aided Design **20** (1988), 27–38.
- [Joe95] B. Joe, *Quadrilateral mesh generation in polygonal regions*, Computer-Aided Design **27** (1995), 209–222.
- [KS92] S. Kannan and D. Soroker, *Tiling polygons with parallelograms*, Discrete Computational Geometry **7** (1992), 175–188.

- [LL92] S. H. Lo and C. K. Lee, *On using meshes of mixed element types in adaptive finite element analyses*, Finite Elements in Analysis and Design **11** (1992), 307–336.
- [MH97] M. Müller-Hannemann, *High quality quadrilateral surface meshing without template restrictions: A new approach based on network flow techniques*, to appear in Proceedings of the Sixth International Meshing Roundtable, Park City, Utah, 1997.
- [Mit97] S. A. Mitchell, *Choosing corners of rectangles for mapped meshing*, Proceedings of the 13th Annual ACM Symposium on Computational Geometry, Nice, France, ACM, 1997, pp. 87–93.
- [MMW96] R. H. Möhring, M. Müller-Hannemann, and K. Weihe, *Mesh refinement via bidirected flows: Modeling, complexity, and computational results*, Technical report No. 520/1996, Fachbereich Mathematik, Technische Universität Berlin, 1996, available via anonymous ftp from <ftp.math.tu-berlin.de>; *cd pub/Preprints/combi*; file *Report-520-1996.ps.Z*; to appear in the Journal of the ACM.
- [MW97a] M. Müller-Hannemann and K. Weihe, *Minimum strictly convex quadrangulations of convex polygons*, Proceedings of the 13th Annual ACM Symposium on Computational Geometry, Nice, France, ACM, 1997, pp. 193–202.
- [MW97b] M. Müller-Hannemann and K. Weihe, *Improved approximations for minimum cardinality quadrangulations of finite element meshes*, Technical report No. 559/1997, Fachbereich Mathematik, Technische Universität Berlin, 1997, available via anonymous ftp from <ftp.math.tu-berlin.de>; *cd pub/Preprints/combi*; file *Report-559-1997.ps.Z*; an extended abstract is to appear in Proceedings of the 5th Annual European Symposium on Algorithms, ESA'97, Graz, Austria.
- [TA93] T. K. H. Tam and C. G. Armstrong, *Finite element mesh control by integer programming*, Int. J. Numer. Methods in Eng. **36** (1993), 2581–2605.
- [Wei94] V. I. Weingarten, *The controversy of hex or tet meshing*, Machine Design (1994), 74–77.
- [WMBS95] D. R. White, L. Mingwu, S. E. Benzley, and G. D. Sjaardema, *Automated hexahedral mesh generation by virtual decomposition*, Proceedings of the 4th International Meshing Roundtable, Sandia National Laboratories, Albuquerque, USA, 1995, pp. 165–176.
- [ZT89] O. C. Zienkiewicz and R. L. Taylor, *The finite element method*, McGraw Hill, London, 1989.
- [ZZHW91] J. Z. Zhu, O. C. Zienkiewicz, E. Hinton, and J. Wu, *A new approach to the development of automatic quadrilateral mesh generation*, Int. J. Numer. Methods in Eng. **32** (1991), 849–866.

Reports from the group

**“Algorithmic Discrete Mathematics”**

of the Department of Mathematics, TU Berlin

- 559/1997** *Matthias Müller–Hannemann and Karsten Weihe*: Improved Approximations for Minimum Cardinality Quadrangulations of Finite Element Meshes
- 554/1997** *Rolf H. Möhring and Matthias Müller–Hannemann*: Complexity and Modeling Aspects of Mesh Refinement into Quadrilaterals
- 551/1997** *Hans Bodlaender, Jens Gustedt and Jan Arne Telle*: Linear-Time Register Allocation for a Fixed Number of Registers and no Stack Variables
- 550/1997** *Karell Bertet, Jens Gustedt and Michel Morvan*: Weak-Order Extensions of an Order
- 549/1997** *Andreas S. Schulz and Martin Skutella*: Random-Based Scheduling: New Approximations and LP Lower Bounds
- 542/1996** *Stephan Hartmann*: On the NP–Completeness of Channel and Switchbox Routing Problems
- 536/1996** *Cynthia A. Phillips, Andreas S. Schulz, David B. Shmoys, Cliff Stein, and Joel Wein*: Improved Bounds on Relaxations of a Parallel Machine Scheduling Problem
- 535/1996** *Rainer Schrader, Andreas S. Schulz, and Georg Wambach*: Base Polytopes of Series-Parallel Posets: Linear Description and Optimization
- 533/1996** *Andreas S. Schulz and Martin Skutella*: Scheduling–LPs Bear Probabilities: Randomized Approximations for Min–Sum Criteria
- 530/1996** *Ulrich H. Kortenkamp, Jürgen Richter–Gebert, Aravamathan Sarangarajan, and Günter M. Ziegler*: Extremal Properties of 0/1-Polytopes
- 524/1996** *Elias Dahlhaus, Jens Gustedt and Ross McConnell*: Efficient and Practical Modular Decomposition
- 523/1996** *Jens Gustedt and Christophe Fiorio*: Memory Management for Union-Find Algorithms
- 520/1996** *Rolf H. Möhring, Matthias Müller–Hannemann, and Karsten Weihe*: Mesh Refinement via Bidirected Flows: Modeling, Complexity, and Computational Results
- 519/1996** *Matthias Müller–Hannemann and Karsten Weihe*: Minimum Strictly Convex Quadrangulations of Convex Polygons
- 517/1996** *Rolf H. Möhring, Markus W. Schäffter, and Andreas S. Schulz*: Scheduling Jobs with Communication Delays: Using Infeasible Solutions for Approximation
- 516/1996** *Leslie A. Hall, Andreas S. Schulz, David B. Shmoys, and Joel Wein*: Scheduling to Minimize Average Completion Time: Off-line and On-line Approximation Algorithms



- 515/1996** *Christophe Fiorio and Jens Gustedt*: Volume Segmentation of 3-dimensional Images
- 514/1996** *Martin Skutella*: Approximation Algorithms for the Discrete Time-Cost Tradeoff Problem
- 509/1996** *Soumen Chakrabarti, Cynthia A. Phillips, Andreas S. Schulz, David B. Shmoys, Cliff Stein, and Joel Wein*: Improved Scheduling Algorithms for Minsum Criteria
- 508/1996** *Rudolf Müller and Andreas S. Schulz*: Transitive Packing
- 506/1996** *Rolf H. Möhring and Markus W. Schäffter*: A Simple Approximation Algorithm for Scheduling Forests with Unit Processing Times and Zero-One Communication Delays
- 505/1996** *Rolf H. Möhring and Dorothea Wagner*: Combinatorial Topics in VLSI Design: An Annotated Bibliography
- 504/1996** *Uta Wille*: The Role of Synthetic Geometry in Representational Measurement Theory
- 502/1996** *Nina Amenta and Günter M. Ziegler*: Deformed Products and Maximal Shadows of Polytopes
- 500/1996** *Stephan Hartmann and Markus W. Schäffter and Andreas S. Schulz*: Switchbox Routing in VLSI Design: Closing the Complexity Gap
- 498/1996** *Ewa Malesinska, Alessandro Panconesi*: On the Hardness of Allocating Frequencies for Hybrid Networks
- 496/1996** *Jörg Rambau*: Triangulations of Cyclic Polytopes and higher Bruhat Orders

Reports may be requested from:      S. Marcus  
 Fachbereich Mathematik, MA 6-1  
 TU Berlin  
 Straße des 17. Juni 136  
 D-10623 Berlin – Germany  
 e-mail: Marcus@math.TU-Berlin.DE

Reports are available via anonymous ftp from:      ftp.math.tu-berlin.de  
 cd pub/Preprints/combi  
 file Report-<number>-<year>.ps.Z