COMPUTING OPTIMAL MORSE MATCHINGS

MICHAEL JOSWIG AND MARC E. PFETSCH

Abstract. Morse matchings capture the essential structural information of discrete Morse functions. We show that computing optimal Morse matchings is $\mathcal{NP}$-hard and give an integer programming formulation for the problem. Then we present polyhedral results for the corresponding polytope and report on computational results.

1. Introduction

Discrete Morse theory was developed by Forman [9, 11] as a combinatorial analog to the classical smooth Morse theory. Applications to questions in combinatorial topology and related fields are numerous: e.g., Babson et al. [3], Forman [10], Batzies and Welker [4], and Jonsson [20].

It turns out that the topologically relevant information of a discrete Morse function f on a simplicial complex can be encoded as a (partial) matching in its Hasse diagram (considered as a graph), the *Morse matching* of f. A matching in the Hasse diagram is Morse if it satisfies a certain, entirely combinatorial, acyclicity condition. Unmatched k-dimensional faces are called *critical*; they correspond to the critical points of rank k of a smooth Morse function. The total number of non-critical faces equals twice the number of edges in the Morse matching. The purpose of this paper is to study algorithms which compute maximum Morse matchings of a given finite simplicial complex. This is equivalent to finding a Morse matching with as few critical faces as possible.

A Morse matching M can be interpreted as a discrete flow on a simplicial complex Δ. The flow indicates how Δ can be deformed into a more compact description as a CW complex with one cell for each critical face of M. Naturally one is interested in a most compact description, which leads to the combinatorial optimization problem described above. This way optimal (or even sufficiently good) Morse matchings of Δ can help to recognize the topological type of a space given as a finite simplicial complex. The latter problem is known to be undecidable even for highly structured classes of topological spaces, such as smooth 4-manifolds.

Optimization of discrete Morse matchings has been studied by Lewiner, Lopes, and Tavares [24]. Hersh [18] investigated heuristic approaches to the maximum Morse matching problem with applications to combinatorics. Morse matchings can also be interpreted as pivoting strategies for homology computations; see [21]. Furthermore, the set of all Morse matchings of a given simplicial complex itself has the structure of a simplicial complex; see [7].

Since its beginnings in Lovász' proof [25] of the Kneser Conjecture, combinatorial topology seeks to solve combinatorial problems with techniques from (primarily algebraic) topology. And, conversely, such applications to combinatorics frequently shed light on basic concepts in topology. Recent years saw a still growing influx of ideas from differential geometry to the subject. Besides discrete Morse theory this concerns, e.g., various notions of discrete curvature which have been applied to problems in computational geometry and computer graphics. This is paralleled in the

Date: 8/23/2004.
2000 *Mathematics Subject Classification.* Primary 90C27; Secondary 06A07, 52B99, 57Q05, 57R70.
Key words and phrases. Discrete Morse function, Morse matching.
Both authors are (partially) supported by the DFG Research Center “Matheon”.

development of a discrete differential geometry. Here the combinatorial cores of known phenomena in differential geometry are investigated, while keeping a steady eye on applications in computer graphics and mathematical physics; see, e.g., Pinkall and Polthier [28], Bobenko and Suris [5]. We firmly believe that —via discrete Morse theory— techniques from combinatorial optimization are highly relevant to these topics. Our contribution is a first step.

The paper is structured as follows. First we show that computing optimal Morse matchings is $\mathcal{NP}$-hard. This issue had been addressed previously by Lewiner, Lopes, and Tavares [24], but we believe that their argument has a minor gap. Then we give an integer programming formulation for the problem. The formulation consists of two parts: one for the matching conditions and one for the acyclicity constraints. This turns out to be related to the acyclic subgraph problem studied by Grötschel, Jünger, and Reinelt [15]. We derive polyhedral results for the corresponding polytope. In particular, we give two different polynomial time algorithms for the separation of the acyclicity constraints. The paper closes with computational results.

Like most of discrete Morse theory, also most of our results extend to arbitrary finite CW-complexes. We stick to the simplicial setting, however, to simplify the presentation.

2. Discrete Morse Functions and Morse Matchings

We will first introduce discrete Morse functions as developed by Forman [9]. Chari [6] showed that the essential structure of discrete Morse functions is captured by so-called Morse matchings. It turns out that this latter formulation directly leads to a combinatorial optimization problem in which one wants to maximize the size of a Morse matching.

We first need some notation. Let Δ be a *(finite abstract) simplicial complex*, i.e., a set of subsets of a finite set V with the following property: if $F \in \Delta$ and $G \subseteq F$, then $G \in \Delta$; in other words, Δ is an independence system with ground set V. In the following we will ignore $\varnothing$ as a member of Δ. The elements in V are called *vertices* and the sets in Δ are called *faces*. The *dimension* of a face F is $\dim F := |F| - 1$. Let $d = \max\{\dim F \,:\, F \in \mathcal{F}\}$ be the dimension of Δ. We often write i-faces for i-dimensional faces. Let $\mathcal{F}$ be the set of faces of Δ and let $f_i = f_i(\Delta)$ be the number of faces of dimension $i \geq 0$. The maximal faces with respect to inclusion are called *facets* and 1-faces are called edges. The complex Δ is *pure*, if all facets have the same dimension. For $F, G \in \Delta$, we write $F \prec G$ if $F \subset G$ and $\dim F = \dim G - 1$, i.e., "$\prec$" denotes the covering relation in the Boolean lattice. The *graph* of Δ is the (abstract) graph on V in which two vertices are connected by an edge if there exists a 1-face containing both vertices. Throughout this paper we assume that Δ is *connected*, i.e., its graph is connected. This is no loss of generality since the connected components can be treated separately.

The *size* of Δ is defined as the coding length of its face lattice, i.e., if Δ has n faces, then $\operatorname{size} \Delta = \mathcal{O}(n \cdot d \cdot \log n)$. Statements about the complexity of algorithms in the subsequent sections are always with respect to this notion of size.

A function $f : \Delta \to \mathbb{R}$ is a *discrete Morse function* if for every $G \in \Delta$ the sets

$$\{F : F \prec G,\ f(G) \leq f(F)\} \quad \text{and} \quad \{H : G \prec H,\ f(H) \leq f(G)\} \tag{1}$$

both have cardinality at most 1. The first set includes the faces covered by face G which are not assigned a lower value than G, while the second set includes the faces covering G which are not assigned a higher value. The face G is *critical* if both sets have cardinality 0. A simple example of a discrete Morse function can be obtained by setting $f(F) = \dim F$ for every $F \in \Delta$. With respect to this function every face is critical.

Discrete Morse functions are interesting because they can be used to deform a simplicial complex into a (smaller) CW-complex that has a cell for each critical face; see Section 3.

Consider the *Hasse diagram* $H = (\mathcal{F}, A)$ of Δ, that is, a directed graph on the faces of Δ with an arc $(G, F) \in A$ if $F \prec G$; note that the arcs lead from higher to lower dimensional faces. Let $M \subset A$ be a matching in H, i.e., each face is incident to at most one arc in M. Let $H(M)$ be the directed graph obtained from H by reversing the direction of the arcs in M. Then M is a *Morse matching* of Δ if $H(M)$ does not contain directed cycles, i.e., is acyclic (in the directed sense).

Chari [6] observed the following relation between Morse functions and Morse matchings. Let f be a discrete Morse function and let M be the set of arcs $(G, F) \in A$ such that $f(G) \leq f(F)$, i.e., f is not decreasing on these arcs. A simple proof shows that at most one of the sets in (1) can have cardinality one; see Chari [6]. This shows that M is a matching. Since the order given by f can be refined to a linear ordering of the faces of Δ, the directed graph $H(M)$ is in fact acyclic and therefore a Morse matching. To construct a discrete Morse function from a Morse matching, compute a linear ordering extending $H(M)$ (which is acyclic) and then number the faces consecutively in the reverse order.

Although we loose the concrete numbers attached to the faces when going from a discrete Morse function f to the corresponding Morse matching M, we do not loose the information about critical faces: Critical faces of f are exactly the unmatched faces of M. Hence, by maximizing $|M|$ we minimize the number of critical faces of f. In fact, the number of critical faces is $|\mathcal{F}| - 2\,|M|$. For $0 \leq j \leq d$, let $c_j = c_j(M)$ be the number of critical faces of dimension j and let $c(M)$ be the total number of critical faces.

It seems helpful to briefly describe the case of Morse matchings for a one-dimensional simplicial complex Δ. Then Δ represents the incidences of a graph G. A Morse matching M of Δ matches edges with nodes of G. Let $\tilde{G}$ be the following oriented subgraph of G: take all edges which are matched in M and orient them towards its matched node. Since M is a matching, this construction is well defined and the in-degree of each node is one. The acyclicity property shows that $\tilde{G}$ contains no directed cycles and hence is a branching, i.e., the underlying graph is a forest and each (weakly) connected component has a unique root. Therefore, the Morse matchings on a graph G are in one-to-one correspondence with orientations of subgraphs of G which are branchings.

Generalizing this idea, Lewiner, Lopes, and Tavares [24] developed a heuristic for computing optimal Morse matchings. This heuristic computes maximum Morse matchings for combinatorial 2-manifolds, i.e., Morse matchings with maximal cardinality. However, for general simplicial complexes this problem is $\mathcal{NP}$-hard, see Section 4.

3. Properties of Morse Matchings

In this section we briefly review some important properties of Morse matching which we need in the sequel.

Let F be a facet of Δ and let G be a facet of F, which is not contained in any other facet of Δ. The operation of transforming Δ to $\Delta \setminus \{F, G\}$ is called a *simplicial* or *elementary collapse*. We will simply use collapse in the following.

Proposition 3.1 (Forman [9]). *Let Δ be a simplicial complex and Σ a subcomplex of Δ. Then there exists a sequence of collapses from Δ to Σ if and only if there exists a discrete Morse function such that $\Delta \setminus \Sigma$ contains no critical face.*

Forman [9] also proved the following result, which describes one of the most interesting features of Morse matchings:

Theorem 3.2. *Let Δ be a simplicial complex and M be a Morse matching on Δ. Then Δ is homotopy equivalent to a CW-complex containing a cell of dimension i for each critical face of dimension i.*

We refer to Munkres [27] for more information on CW-complexes. By this result we can hope for a compact representation of the topology of Δ (up to homotopy) by computing a Morse matching with few critical faces. This is the main motivation for the combinatorial optimization problem studied in this paper.

Let K be a field and let $\beta_j = \beta_j(K)$ to be the Betti number for dimension j over K for Δ; see again Munkres [27] for details. Forman [9] proved the following bounds on the number of critical faces c_j of a Morse matching M:

Theorem 3.3 (*Weak Morse inequalities*)**.** *Let K be a field, Δ be a simplicial complex, and M a Morse matching for Δ. We have*

$$c_j \geq \beta_j \qquad \text{for all } j = 0, \dots, d \tag{2}$$

and

$$c_0 - c_1 + c_2 - \cdots + (-1)^d c_d = \beta_0 - \beta_1 + \beta_2 - \cdots + (-1)^d \beta_d. \tag{3}$$

The Betti numbers over $\mathbb{Q}$ and finite fields can easily be obtained in polynomial time (in the size of Δ), by computing the ranks of the boundary matrices for each dimension. Although harder to compute (see Iliopoulos [19]), the homology over $\mathbb{Z}$ can be used to choose among the finite fields or $\mathbb{Q}$, in order to obtain the strongest form of the Morse inequalities (3.3).

4. Hardness of Optimal Morse Matchings

In this section we prove $\mathcal{NP}$-hardness of the problem to compute a maximum Morse matching, i.e., to find a Morse matching M with maximal cardinality. As we saw previously, this is equivalent to minimize the number of critical faces.

We want to reduce the following *collapsibility problem*, introduced by Eğecioğlu and Gonzalez [8], to the problem of finding an optimal Morse matching: Given a connected pure 2-dimensional simplicial complex Δ, which is embeddable in $\mathbb{R}^3$ and an integer k, decide whether there exists a subset $\mathcal{K}$ of the facets of Δ with $|\mathcal{K}| \leq k$ such that there exists a sequence of collapses which transforms $\Delta \backslash \mathcal{K}$ to a 1-dimensional complex. Eğecioğlu and Gonzalez proved that this collapsibility problem is strongly $\mathcal{NP}$-complete. Using Proposition 3.1, this results reads in terms of discrete Morse theory:

Theorem 4.1. *Given a connected pure 2-dimensional simplicial complex Δ, which is embeddable in $\mathbb{R}^3$, and a nonnegative integer k, it is $\mathcal{NP}$-complete in the strong sense to decide whether there exists a Morse matching with at most k critical 2-faces.*

When k is fixed, we can try all possible sets $\mathcal{K}$ of size at most k and then decide whether the resulting complex is collapsible to a 1-dimensional complex in polynomial time. Therefore we let k be part of the input.

We need the following construction. Consider a Morse matching M for a simplicial complex Δ, with $\dim \Delta \geq 1$. Let $\Gamma(M)$ be the graph obtained from the graph of Δ by removing all edges (1-faces) matched with 2-faces. Note that $\Gamma(M)$ contains all vertices of Δ.

Lemma 4.2. *The graph $\Gamma(M)$ is connected.*

Proof. Without loss of generality we assume that $\dim \Delta \geq 2$. Otherwise $\Gamma(M)$ coincides with the graph of Δ, which is connected (recall that Δ is connected).

Suppose that $\Gamma(M)$ is disconnected. Let N be the set of nodes in a connected component of $\Gamma(M)$, and let C be the set of *cut edges*, that is, edges of Δ with one vertex in N and one vertex in its complement. Since Δ is connected, C is not empty. By definition of $\Gamma(M)$, each edge in C is matched to a unique 2-face.

Consider the directed subgraph D of the Hasse diagram consisting of the edges in C and their matching 2-faces. The standard direction of arcs in the Hasse diagram (from the higher to the lower dimensional faces) is reversed for each matching pair of M, i.e., D is a subgraph of $H(M)$.

We construct a directed path in D as follows. Start with any node of D corresponding to a cut edge e_1. Go to the node of D determined by the unique 2-face τ_1 to which e_1 is matched to. Then τ_1 contains at least one other cut edge e_2, otherwise e_1 cannot be a cut edge. Now iteratively go to e_2, then to its unique matching 2-face τ_2, choose another cut edge e_3, and so on. We observe that we obtain a directed path $e_1, \tau_1, e_2, \tau_2, \dots$ in D, i.e., the arcs are directed in the correct direction.

Since we have a finite graph at some point the path must arrive at a node of D which we have visited already. Hence, D (and therefore also $H(M)$) contains a directed cycle, which is a contradiction since M is a Morse matching. □

Now pick an arbitrary node r and any spanning tree of $\Gamma(M)$ and direct all edges away from r. This yields a maximum Morse matching on $\Gamma(M)$; see end of Section 2. It is easy to see that replacing the part of M on $\Gamma(M)$ with this matching yields a Morse matching. This Morse matching has only one critical vertex (the root r). Note that every Morse matching contains at least one critical vertex; this can be seen from the Morse inequalities Theorem 3.3 or directly by observing that we get a directed cycle in the Hasse diagram if every vertex is matched to an edge. Furthermore, the total number of critical faces can only decrease, since we computed an optimal Morse matching on $\Gamma(M)$. The number of critical i-faces for $i \geq 2$ stays the same. We have thus proved the following, which is also implicit in Forman [9].

Corollary 4.3. *Let M be a Morse matching on Δ. Then we can compute a Morse matching M' in polynomial time which has exactly one critical vertex and the same number of critical faces of dimension 2 or higher as M, such that $c(M') \leq c(M)$.*

We can now prove the hardness result.

Theorem 4.4. *Given a simplicial complex Δ and a nonnegative integer c, it is strongly $\mathcal{NP}$-complete to decide whether there exists a Morse matching with at most c critical faces, even if Δ is connected, pure, 2-dimensional, and can be embedded in $\mathbb{R}^3$.*

Proof. Clearly this problem is in $\mathcal{NP}$. So let (Δ, k) be an input for the collapsibility problem. We claim that there exists a Morse matching with at most k critical 2-faces if and only if there exists a Morse matching with at most $g(k) := 2(k+1) - \chi(\Delta)$ critical faces altogether. Here, $\chi(\Delta) = \beta_0 - \beta_1 + \cdots + (-1)^d \beta_d$ is the Euler characteristic, which can be computed in polynomial time; see Section 3. Hence g is a polynomial-time computable function. Using Theorem 4.1 then finishes the proof.

So assume that M is a Morse matching on Δ with at most k critical 2-faces. We use Corollary 4.3 to compute a Morse matching M', in polynomial time, such that $c_0(M') = 1$, $c_2(M') = c_2(M)$, and $c(M') \leq c(M)$. By the Morse equation of Theorem 3.3, we have $c_1(M') = c_2(M') + 1 - \chi(\Delta)$. Since $c(M') = c_0(M') + c_1(M') + c_2(M')$ it follows that

$$c_2(M) = c_2(M') = \tfrac{1}{2}(c(M') + \chi(\Delta))) - 1. \tag{4}$$

Solving for $c(M')$, it follows that M' has at most $2(k+1) - \chi(\Delta)$ critical faces altogether.

Conversely, assume that there exists a Morse matching M with at most $g(k)$ critical faces. Computing M' as above, we obtain by (4), that

$$c_2(M) = c_2(M') \leq \tfrac{1}{2}(g(k) + \chi(\Delta))) - 1 = k,$$

which completes the proof. □

Lewiner, Lopes, and Tavares [24] showed that it is $\mathcal{NP}$-hard to compute an optimal Morse matching with exactly one critical vertex. They claimed it for the general case, but we do not see an argument similar to Lemma 4.2 above.

Since there exists a Morse matching with at most c critical faces if and only if there exists a Morse matching of size at least $\frac{1}{2}(|\mathcal{F}| - c)$, we proved:

Corollary 4.5. *Let Δ be as in Theorem 4.4 and m be a nonnegative integer. Then it is $\mathcal{NP}$-complete in the strong sense to decide whether there exists a Morse matching of size at least m.*

We do not know about the complexity status for this problem with m fixed.

Eğecioğlu and Gonzalez [8] even proved that the collapsibility problem is as hard to approximate as the set covering problem. In particular, the collapsibility problem cannot be approximated better than within a logarithmic factor in polynomial time, unless $\mathcal{P} = \mathcal{NP}$. Using this, Lewiner, Lopes, and Tavares [24] claimed that the problem to compute a Morse matching minimizing the number of critical faces is hard to approximate. However, the function g used in the proof above is not "approximation preserving" and we do not see how the non-approximability result carries over.

Similarly, the problem to approximate the size of a Morse matching seems to be open.

5. An IP-Formulation

In this section we introduce an integer programming formulation for the problem to compute a Morse matching of maximal size. It can easily be extended to arbitrary weights.

We use the following notation. We depict vectors in bold font. Let $\boldsymbol{e}_i$ be the ith unit vector and let $\mathbb{1}$ be the vector of all ones. For any vector $\boldsymbol{x} \in \mathbb{R}^n$ and $I \subseteq \{1, \dots, n\}$ we define

$$\boldsymbol{x}(I) := \sum_{i \in I} x_i.$$

Furthermore, for $S \subseteq \{1, \dots n\}$, $\boldsymbol{I}(S) \in \mathbb{R}^n$ denotes the incidence vector of S.

For a node v in a directed graph, let $\delta(v)$ be the arcs incident to v, i.e., the arcs having v as one of their endnodes. For a subset $A' \subseteq A$, we denote by $N(A')$ the nodes incident to at least one arc in A'. Throughout this article, all directed or undirected cycles are assumed to be *simple*, i.e., without node repetitions.

For ease of notation, we consider the Hasse diagram H as directed or undirected depending on the context; we will explicitly use *directed* when we refer to the directed version.

We split H into d levels $H_0 = (\mathcal{F}^0, A_0)$, $\dots$, $H_{d-1} = (\mathcal{F}^{d-1}, A_{d-1})$, where H_i denotes the level of the Hasse diagram between faces of dimension i and $i+1$. Then A is the disjoint union of $A_0, \dots, A_{d-1}$ and $\mathcal{F}^{i-1} \cap \mathcal{F}^i$ consists of the faces of dimension i. Recall that the arcs in the Hasse diagram are directed from the higher to the lower dimensional faces.

Let $M \subset A$ be a Morse matching of Δ. By definition, its incidence vector $\boldsymbol{x} = \boldsymbol{I}(M) \in \{0,1\}^A$ satisfies the *matching inequalities*:

$$\boldsymbol{x}(\delta(F)) \leq 1 \quad \forall\, F \in \mathcal{F}. \tag{5}$$

Now assume that for some $M \subseteq A$ there exists a directed cycle D in $H(M)$. Then in D "up" and "down" arcs alternate; see for example Figure 1. In particular, the size of D is always even. Hence, $\frac{1}{2}|D|$ arcs are contained in M, i.e., are reversed in $H(M)$. Furthermore we note:

Observation. Let $M \subset A$ be a matching. If D is a directed cycle in $H(M)$, the edges in D can only belong to one level H_i ($i \in \{0, \dots, d-1\}$), i.e., $\{\dim F \,:\, F \in N(D)\} = \{i, i+1\}$.

Putting these arguments together we obtain: If M is acyclic, $\boldsymbol{x} = \boldsymbol{I}(M)$ satisfies the following *cycle inequalities*:

$$\boldsymbol{x}(C) \leq \tfrac{1}{2}|C| - 1 \quad \forall\, C \in \mathcal{C}_i,\ i = 1, \dots, d-1, \tag{6}$$

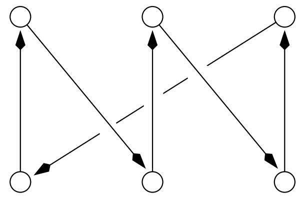

Figure 1: Example for a directed cycle of size 6; at least three arcs with reversed orientation (pointing "up") are necessary to close a 6-cycle in the Hasse diagram of a simplicial complex.

where $\mathcal{C}_i$ are the cycles in H_i.

Conversely, it is easy to see that every $\boldsymbol{x} \in \{0,1\}^A$ which fulfills inequalities (5) and (6) is the incidence vector of a Morse matching. Hence, we arrive at the following IP formulation for the problem to find a maximum Morse matching:

$$\begin{array}{llll} (\text{MaxMM}) & \max & \mathbb{1}^{\mathrm{T}}\boldsymbol{x} & \\ & \text{s.t.} & \boldsymbol{x}(\delta(F)) \leq 1 & \forall\, F \in \mathcal{F} \\ & & \boldsymbol{x}(C) \leq \frac{1}{2}|C| - 1 & \forall\, C \in \mathcal{C}_i,\ i = 1,\dots,d-1 \\ & & \boldsymbol{x} \in \{0,1\}^A. & \end{array}$$

We define the corresponding polytope:

$$P_M = \operatorname{conv}\big\{\boldsymbol{x} \in \{0,1\}^A \,:\, \boldsymbol{x} \text{ satisfies (5) and (6)}\big\}.$$

Let M be a Morse matching and $\boldsymbol{x} = \boldsymbol{I}(M)$ be its incidence vector. Then $F \in \mathcal{F}$ is a critical face with respect to M if and only if it is unmatched by M, i.e., $\boldsymbol{x}(\delta(F)) = 0$. Hence, the total number of critical faces is:

$$c(M) = \sum_{F \in \mathcal{F}} \big(1 - \sum_{a \in \delta(F)} x_a\big) = |\mathcal{F}| - 2 \sum_{a \in A} x_a = |\mathcal{F}| - 2\,\mathbb{1}^{\mathrm{T}}\boldsymbol{x}, \tag{7}$$

since every arc is incident to exactly two nodes. Using this formula one can easily switch between the number of critical faces and the number of arcs in a Morse matching.

The LP relaxation of MaxMM can be strengthened by using the weak Morse inequalities (2) of Theorem 3.3. Applying (7), this yields the following *Betti inequality* for dimension i:

$$\sum_{F : \dim F = i} \big(1 - \sum_{a \in \delta(F)} x_a\big) \geq \beta_i \quad \Leftrightarrow \quad \sum_{F : \dim F = i} \sum_{a \in \delta(F)} x_a \leq f_i - \beta_i. \tag{8}$$

Observe that we can choose the field in Theorem 3.3 to employ the Morse inequalities in their strongest form.

Note. The cycle inequalities (6) are similar to the cycle inequalities for the acyclic subgraph problem (ASP); see Jünger [22], and Grötschel, Jünger, and Reinelt [15]. The separation problem for (6), however, is more complicated than the corresponding problem for ASP; see Section 5.2.

Furthermore, there is a similarity to the relation between the ASP and the linear ordering problem (see Reinelt [29], and Grötschel, Jünger, and Reinelt [14]): an alternative formulation for our problem can be obtained by adding matching inequalities to the linear ordering formulation; this directly models discrete Morse functions as linear orderings of the faces. Since this formulation is based on the relation between faces, it leads to quadratically many variables in the number of faces; therefore we have opted for the above formulation, at the cost of having to solve the separation problem for the cycle inequalities; see Section 5.2.

Figure 2: Example for a non-monotone behavior of acyclic matchings. The directed graph on the right, obtained from the left graph by reversing the dashed arcs, is acyclic. However, if the top arc is set to its original orientation, the graph is not acyclic anymore. This shows that subsets of acyclic matchings are not necessarily acyclic.

5.1. **Facial Structure of P_M.** It is easy to see that P_M is a full dimensional polytope and $x_a \geq 0$ defines a facet for every $a \in A$. Furthermore, P_M is monotone, since every subset of a Morse matching is a Morse matching. It is well known that this implies that every facet defining inequality $\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{x} \leq \beta$ not equivalent to the non-negativity inequalities fulfills: $\boldsymbol{\alpha} \geq 0$, $\beta > 0$; see Hammer, Johnson, and Peled [17].

Interestingly, if we generalize Morse matchings to acyclic matchings for arbitrary graphs, the collection of such acyclic matchings is not necessarily monotone anymore; see the example in Figure 2. Therefore, the structure of the generalized problem is likely to be more complicated.

We have the following two results:

Proposition 5.1. *The matching inequalities $\boldsymbol{x}(\delta(F)) \leq 1$ define facets of P_M for $F \in \mathcal{F}$, except if $|\delta(F)| = 1$, i.e., F is a vertex.*

Proof. Let F be a face with $|\delta(F)| > 1$ (note that $|\delta(F)| = 0$ does not occur). We can assume that $A = \{a_1, \dots, a_k, a_{k+1}, \dots, a_m\}$, where $\delta(F) = \{a_1, \dots, a_k\}$. For $i = k+1, \dots, m$, observe that a_i cannot be adjacent to every arc in $\delta(F)$: since $|\delta(F)| > 1$, a_i would either be incident to at least two nodes of the same dimension or to two nodes whose dimensions are two apart, which is impossible. Therefore, choose $p(i) \in \{1, \dots, k\}$ such that a_i and $a_{p(i)}$ are not adjacent. It follows that $\boldsymbol{e}_i + \boldsymbol{e}_{p(i)} \in P_M$. Then

$$\boldsymbol{e}_1, \dots, \boldsymbol{e}_k, \boldsymbol{e}_{k+1} + \boldsymbol{e}_{p(k+1)}, \dots, \boldsymbol{e}_m + \boldsymbol{e}_{p(m)}$$

are affinely independent and fulfill $\boldsymbol{x}(\delta(F)) = 1$. $\square$

It follows that the inequalities $x_a \leq 1$, $a \in A$, never define facets.

Theorem 5.2. *The cycle inequalities* (6) *define facets of P_M.*

Proof. We extend the corresponding proof by Jünger [22] for the ASP.

Let C be a cycle in H. Without loss of generality assume that $A = \{a_1, \dots, a_k, a_{k+1}, \dots, a_m\}$, where $C = (a_1, \dots, a_k)$ and k is even. We will construct affinely independent feasible vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_k, \boldsymbol{v}_{k+1}, \dots, \boldsymbol{v}_m$ satisfying the cycle inequality corresponding to C with equality.

Let $C_1 = \{a_1, a_3, \dots, a_{k-1}\}$ and $C_2 = \{a_2, a_4, \dots, a_k\}$. Hence C_1 and C_2 are the "up" and "down" arcs in C.

Define

$$\boldsymbol{v}_i = \begin{cases} \boldsymbol{I}(C_1 \setminus \{a_i\}) & \text{if } a_i \in C_1 \\ \boldsymbol{I}(C_2 \setminus \{a_i\}) & \text{if } a_i \in C_2 \end{cases} \qquad \text{for } i = 1, \dots, k.$$

Hence, for $i = 1, \dots, k$ we have $\boldsymbol{v}_i(C) = \frac{k}{2} - 1$.

For $i = k+1, \dots, m$, consider $a_i = \{u, v\} \notin C$. We have four cases.

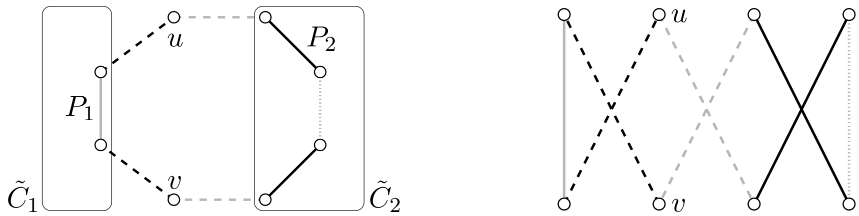

Figure 3: Illustration of the first case in the proof of Theorem 5.2. The sets P_1 and P_2 are shown by continuous lines. The edges in C_1 are drawn gray and hence $P_1 \subset C_1$; edges in C_2 are drawn black. The dashed edges incident to u and v are not considered. The right hand side shows the graph embedded in the Hasse diagram.

- $\triangleright$ $u, v \in N(C)$: Let $\tilde{C} := C \setminus \big(\delta(u) \cup \delta(v)\big)$. We have that $|\tilde{C}| = k - 4$ (since there exist no odd cycles) and $\tilde{C}$ splits into two odd nonempty parts $\tilde{C}_1$ and $\tilde{C}_2$, which are both paths. Let $k_1 := |\tilde{C}_1|$ and $k_2 := |\tilde{C}_2|$; k_1 and k_2 are odd, since u and v are on opposite sides of the bipartition. We choose a subset $P_1 \subset \tilde{C}_1$ by taking every second arc in order to get $|P_1| = \frac{k_1+1}{2}$; similarly we choose $P_2 \subset \tilde{C}_2$ with $|P_2| = \frac{k_2+1}{2}$. By construction either $P_i \subset C_1$ or $P_i \subset C_2$ and either $P_i \cap C_2 = \varnothing$ or $P_i \cap C_1 = \varnothing$ for $i = 1, 2$. An easy calculation shows that $|P_1 \cup P_2| = \frac{k}{2} - 1$. See Figure 3 for an illustration of this case. Then define $\boldsymbol{v}_i := \boldsymbol{I}(P_1 \cup P_2 \cup \{a_i\})$.
- $\triangleright$ $u \notin C,\ v \in C$: Here we define $\boldsymbol{v}_i := \boldsymbol{I}(C_1 \setminus \delta(v) \cup \{a_i\})$.
- $\triangleright$ $u \in C,\ v \notin C$: Define $\boldsymbol{v}_i := \boldsymbol{I}(C_1 \setminus \delta(u) \cup \{a_i\})$.
- $\triangleright$ $u, v \notin C$: Choose any $a \in C_1$ and define $\boldsymbol{v}_i := \boldsymbol{I}(C_1 \setminus \{a\} \cup \{a_i\})$.

It is easy to check in each case that $\boldsymbol{v}_i \in P_M$ and that $\boldsymbol{v}_i(C) = \frac{k}{2} - 1$.

It can be shown that the m vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$ are affinely independent, which concludes the proof. $\square$

The separation problem for the cycle inequalities is discussed in the next section.

5.2. **Separating the Cycle Inequalities.** Of course, there are exponentially many cycle inequalities (6). Hence we have to deal with the separation problem for these inequalities.

For the separation problem, we can assume that we are given $\boldsymbol{x}^* \in [0,1]^A$, which satisfies all matching inequalities (5). We consider the separation for each graph H_i in turn, $i = 0, \dots, d-1$. The problem is to find an undirected cycle C in H_i such that

$$\boldsymbol{x}^*(C) > \tfrac{1}{2}|C| - 1$$

or conclude that no such cycle exists. In the next sections we describe two methods to solve this separation problem in polynomial time.

5.2.1. *Undirected Shortest Path with Conservative Weights.* A usual trick to solve the above separation problem is to apply an affine transformation and obtain a shortest cycle problem. The transformation suitable for our needs is $\boldsymbol{x}' = \frac{1}{2}\mathbb{1} - \boldsymbol{x}$, which yields:

$$\boldsymbol{x}(C) \leq \tfrac{1}{2}|C| - 1 \qquad \Leftrightarrow \qquad \boldsymbol{x}'(C) \geq 1.$$

The separation problem can now be solved as follows: compute a shortest cycle in H_i with respect to the weights $\frac{1}{2}\mathbb{1} - \boldsymbol{x}^*$. If its weight is at most 1, this cycle yields a violated cycle inequality, otherwise no such cycle exists.

However, the weights can be negative and we have to rule out negative cycles in order to apply polynomial time methods from the literature; that is, we want the weights to be *conservative*.

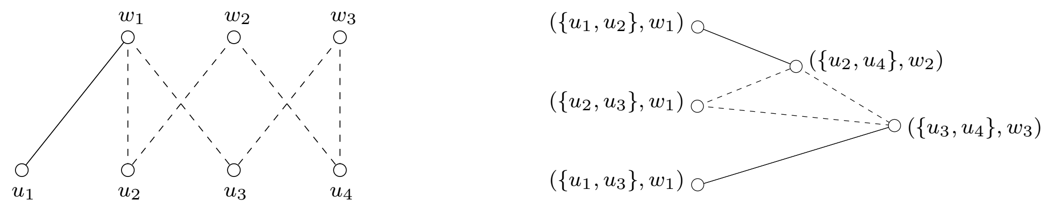

Figure 4: Example of the construction in Section 5.2.2. *Left*: original graph G. *Right:* constructed graph G'. The 6-cycle on the left corresponds to the 3-cycle on the right (both shown with dashed lines).

Lemma 5.3. *There exists no cycle of negative weight in H_i with respect to $\frac{1}{2}\mathbb{1}-\boldsymbol{x}^*$, for $0 \leq i \leq d-1$.*

Proof. Let $C = (a_1, \dots, a_k)$ be a cycle in H_i and let $F_1, \dots, F_k$ be the faces that are visited by C. Recall the $\boldsymbol{x}^*$ satisfies the matching inequalities. We obtain

$$\sum_{j=1}^{k} \sum_{a \in \delta(F_i) \cap C} x_a^* \;=\; 2 \sum_{a \in C} x_a^* \;=\; 2\,\boldsymbol{x}^*(C), \tag{9}$$

since each edge weight is counted twice in the first term. Applying the Matching inequalities (5) on the left hand side yields that $\boldsymbol{x}^*(C) \leq \frac{1}{2}k = \frac{1}{2}|C|$. Hence, the weight of C with respect to $\frac{1}{2}\mathbb{1} - \boldsymbol{x}^*$ can be bounded as follows

$$\sum_{a \in C} \big(\tfrac{1}{2} - x_a^*\big) = \tfrac{1}{2}|C| - \boldsymbol{x}^*(C) \geq 0,$$

which proves the lemma. □

We have now reduced the separation problem to finding a shortest cycle in a weighted undirected graph $G = (V, E)$ without negative cycles.

By using T-join techniques, one can compute a shortest path in an undirected graph without negative cycles in $\mathcal{O}(n_i(m_i + n_i \log n_i))$ time, where in this formula $n_i = |\mathcal{F}^i|$ and $m_i = |A_i|$; see Schrijver [30, Chapter 29]. It follows that a shortest cycle can be computed in $\mathcal{O}(m_i n_i(m_i + n_i \log n_i))$ time. Since $|A_i| \leq (i+2)n_i$, this leads to an $\mathcal{O}\big((d+1)^2 n^3 + (d+1)n^3 \log n\big)$ overall algorithm, where $n := |\mathcal{F}|$ is the number of faces and d is the dimension of the complex.

5.2.2. *Transforming the Graph.* Another method for the separation problem of cycle inequalities, which is easier to implement, works as follows.

Let $G = (U \mathbin{\dot\cup} W,\, E)$ be a bipartite graph, e.g., $G = H_i$ ($i \in \{0, \dots, d-1\}$), the i-th level of the Hasse diagram. Let $\ell : E \to \mathbb{R}_{\geq 0}$ be a length function for the edges of G. In the following we write $\ell(u, v) = \ell(v, u)$ for the length $\ell(\{u, v\})$.

We construct a graph $G' = (V', E')$ and lengths $\ell' : E' \to \mathbb{R}_{\geq 0}$ as follows; see Figure 4 for an example. The set of nodes of G' is

$$\big\{(\{u, u'\}, w) \,:\, \{u, u'\} \subseteq U,\ w \in W,\ \{u, w\} \in E,\ \{u', w\} \in E\big\}.$$

We have an edge between two nodes $(\{u_1, u_1'\}, w_1)$ and $(\{u_2, u_2'\}, w_2)$ if

$$\{u_1, u_1'\} \cap \{u_2, u_2'\} \neq \varnothing \quad \text{and} \quad w_1 \neq w_2.$$

The length of such an edge e' is defined by

$$\ell'(e') = \tfrac{1}{2}\big(\ell(u_1, w_1) + \ell(u_1', w_1) + \ell(u_2, w_2) + \ell(u_2', w_2)\big).$$

We now consider the relation of cycles in G and G'.

Lemma 5.4. *$C = (u_0, w_0, u_1, w_1, \ldots, w_{k-1}, u_1)$ is a cycle in G with $k > 1$ of length $\ell(C)$ if and only if*

$$C' = \big((\{u_0, u_1\}, w_0), (\{u_1, u_2\}, w_1), \ldots, (\{u_{k-1}, u_1\}, w_{k-1}), (\{u_0, u_1\}, w_0)\big)$$

is a cycle in G' with $\ell'(C') = \ell(C)$.

Proof. In the following we compute with indices modulo k.

First observe that C' is well defined: Each $(\{u_i, u_{i+1}\}, w_i)$ is a node of G', since $\{u_i, w_i\}$, $\{u_{i+1}, w_i\} \in E$. Furthermore,

$$\big\{(\{u_i, u_{i+1}\}, w_i), (\{u_{i+1}, u_{i+2}\}, w_{i+1})\big\} \in E'$$

since $w_i \neq w_{i+1}$ and $\{u_i, u_{i+1}\} \cap \{u_{i+1}, u_{i+2}\} \neq \varnothing$ (because $k \geq 1$). It is a cycle since $k \geq 2$.

The weight of C' can be calculated as follows:

$$\begin{aligned}
\ell'(C') &= \sum_{i=0}^{k} \ell\Big(\big\{(\{u_i, u_{i+1}\}, w_i), (\{u_{i+1}, u_{i+2}\}, w_{i+1})\big\}\Big) \\
&= \sum_{i=0}^{k} \tfrac{1}{2}\big(\ell(u_i, w_i) + \ell(u_{i+1}, w_i) + \ell(u_{i+1}, w_{i+1}) + \ell(u_{i+2}, w_{i+1})\big) \\
&= \sum_{i=0}^{k} \big(\ell(u_i, w_i) + \ell(u_{i+1}, w_i)\big) = \ell(C),
\end{aligned}$$

where the second to last equation follows since every edge in C occurs twice in the summation. □

The previous lemma does not cover cycles in G of length four. These do not occur for the case of $G = H_i$, since H_i is a level in the Hasse diagram of a *simplicial* complex. Moreover, cycles of length four can readily be detected in the construction of G' and handled accordingly (there is only a polynomial number of them).

To solve our separation problem, let $G = H_i$, $i \in \{0, \ldots, d-1\}$, and $\ell(e) = x_e^*$ for $e \in G$. Then we have $\ell'(e') \in [0, 1]$ for each $e' \in E'$, because of the matching inequalities. We now set $\tilde{\ell}(e') = 1 - \ell'(e')$ for $e' \in G'$ and hence $\tilde{\ell}(e') \in [0, 1]$. Let C be a cycle in G with at least six edges and C' be the corresponding cycle in G'. Note that $|C'| = \frac{1}{2}|C|$. We have:

$$\begin{aligned}
&\tilde{\ell}(C') = \sum_{e' \in C'} \tilde{\ell}(e') = \sum_{e' \in C'} (1 - \ell'(e')) < 1 \\
\Leftrightarrow \quad & \sum_{e' \in C'} \ell'(e') > |C'| - 1 \\
\Leftrightarrow \quad & \ell'(C') > |C'| - 1 \\
\Leftrightarrow \quad & \ell(C) > \tfrac{1}{2}|C| - 1 \qquad \text{(by Lemma 5.4).}
\end{aligned}$$

Hence, C violates the cycle inequality (6) if and only if $\tilde{\ell}(C') < 1$. Since $\tilde{\ell}(e') \geq 0$, we can use the Floyd-Warshall algorithm to solve the separation problem in time $\mathcal{O}\big(|V'|^3\big)$; see Korte and Vygen [23].

If $G = H_i$ and W is the part arising from the higher dimensional faces, we have $|V'| = \binom{i+2}{2}|W| = \binom{i+2}{2} f_{i+1}$. This leads to an $\mathcal{O}\big((d+1)^6 n^3\big)$ algorithm for separating cycle inequalities, which is roughly as fast as the method discussed in Section 5.2.1, but much easier to implement.

Table 1: Computational results of the branch-and-cut algorithm with separating cycle inequalities and Gomory cuts.

name	n	m	d	nodes	depth	time	β	c
solid_2_torus	24	42	2	1	0	0.00	2	2
simon2	31	60	2	1	0	0.00	1	1
projective	31	60	2	1	0	0.01	3	3
bjorner	32	63	2	1	0	0.05	2	2
nonextend	39	77	2	6	5	0.16	1	1
simon	41	82	2	1	0	0.18	1	1
dunce	49	99	2	385	10	2.62	1	3
c-ns3	63	128	2	349	10	3.47	1	3
c-ns	75	152	2	28	10	1.95	1	3
c-ns2	79	159	2	14	7	1.11	1	1
ziegler	119	310	3	1	0	0.01	1	1
gruenbaum	167	434	3	1	0	25.24	1	1
lockeberg	216	600	3	1	0	36.25	2	2
rudin	215	578	3	77	30	103.78	1	1
mani-walkup-D	392	1112	3	111	23	512.81	2	2
mani-walkup-C	464	1312	3	135	83	1658.02	2	2
MNSB	103	267	3	12	10	73.39	1	1
MNSS	250	698	3	292	110	750.36	2	2
CP2	255	864	4	230	80	558.14	3	3

6. Computational Results

In this section we report on computational experience with a branch-and-cut algorithm along the lines of Section 5. The C++ implementation uses the framework SCIP (Solving Constraint Integer Programs) by Achterberg, see [1]. It furthermore builds on `polymake`; see [12, 13]. As an LP solver we used CPLEX 9.0.

As the basis of our implementation we take the formulation of MaxMM in Section 5. Matching inequalities (5) and Betti inequalities (8) (together with variable bounds) form the initial LP. Cycle inequalities (6) are separated as described in Section 5.2.2. Additionally, Gomory cuts are added. As a branching rule we use *reliability branching* implemented in SCIP, a variable branching rule introduced by Achterberg, Koch, and Martin [2].

We implemented the following primal heuristic. First a simple greedy algorithm is run: We start with the empty matching $M = \varnothing$. We add arcs of the Hasse diagram to M in the order of decreasing value of the current LP solution as long as M stays an acyclic matching (which can easily be tested). Then the outcome is iteratively improved by a method described in Forman [9]: One searches for a unique path between two critical faces in $H(M)$. Such a path is alternating with respect to M. Then M can be augmented along the path (the new matching is the symmetric difference of M and the path). As is easily seen, this generates an acyclic matching, because the path is unique. This heuristic turns out to be extremely successful; see below.

We tested the implementation on a set of simplicial complexes collected by Hachimori; see [16] for more details. Additionally, we considered the following complexes: `CP2` (complex projective plane), `CP2+CP2` (connected sum of `CP2` with itself), `MNSB` and `MNSS` ((vertex) minimal non-shellable ball and sphere, respectively; see Lutz [26]).

All computational experiments were run on a 3 GHz Pentium machine running Linux. In the tables of computational results, n denotes the number of faces, m the number of arcs in the Hasse

Table 2: Computational results of the branch-and-cut algorithm without separation.

name	n	m	d	nodes	depth	time	β	c
solid_2_torus	24	42	2	1	0	0.00	2	2
simon2	31	60	2	1	0	0.01	1	1
projective	31	60	2	1	0	0.00	3	3
bjorner	32	63	2	1	0	0.01	2	2
nonextend	39	77	2	3	2	0.02	1	1
simon	41	82	2	4	3	0.02	1	1
dunce	49	99	2	168367	42	145.60	1	3
c-ns3	63	128	2	3665581	53	3940.40	1	3
c-ns	75	152	2	16625713	58	19359.69	1	3
c-ns2	79	159	2	4	3	0.03	1	1
ziegler	119	310	3	1	0	0.01	1	1
gruenbaum	167	434	3	21	20	0.68	1	1
lockeberg	216	600	3	1	0	0.05	2	2
rudin	215	578	3	81	80	3.18	1	1
mani-walkup-D	392	1112	3	107	100	2.00	2	2
mani-walkup-C	464	1312	3	1498	456	30.54	2	2
MNSB	103	267	3	1	0	0.01	1	1
MNSS	250	698	3	163	126	4.63	2	2
CP2	255	864	4	198	190	4.77	3	3
CP2+CP2	460	1592	4	5178	534	110.21	4	4

diagram (= number of variables), d the dimension, *nodes* the number of nodes in the branch-and-bound tree, *depth* the maximal depth in the tree, *time* the computation time in seconds, β is the lower bound obtained by adding all Betti inequalities (8), and c the number of critical faces in the optimal solution.

Our implementation could not solve the larger problems of Hachimori's collection in reasonable time: `bing`, `knot`, `poincare`, `nonpl_sphere`, and `nc_sphere`. In fact, for `poincare` we ran our code in different settings, each for about a week – without success.

Table 1 shows the results of a computation where we separate cycle inequalities and Gomory cuts and run the heuristic every 10th level. At most seven separation rounds of cycle inequalities were performed at a node. We do not report results on the problems by Moriyama and Takeuchi in Hachimori's collection – they all could be solved within a second. The version with cut separation could not solve `CP2+CP2` within 90 minutes.

For most problems the bound obtained by adding Betti inequalities (8), as indicated in column "β", is tight. This means that the algorithm is done once an optimal solution is found. This usually happens very fast and shows that the heuristic is efficient. In fact, there are only three problems for which the bound is not tight and could be solved by our algorithm (`dunce`, `c-ns`, and `c-ns3`). These three problems are solved easily by the version with cut separation. In our problem set there exists no hard but still solvable problem with a "Betti bound" which is not sharp. We can therefore not estimate the limits of our implementation for these cases (`poincare` is the next larger problem of this kind with 1112 variables, but we could not solve it).

The tractability of problems with a tight "Betti bound" is supported by the results obtained by running the implementation without any separation; see Table 2. Only integer solutions are checked whether they are acyclic and the heuristic is run every 10th level. This essentially is a test of the performance of the primal heuristic. Indeed, all problems with tight "Betti bound" were

solved within a few seconds (CP2+CP2 and mani-walkup-C being the exception, but could be solved within two minutes). The results for the problems c-ns, c-ns3, and dunce show that the cycle inequalities and Gomory cuts are very effective in reducing the number of nodes in the tree and the computing time for problems where the "Betti bound" is not sharp.

Summarizing, we can say that our implementation can solve large instances with up to about 1500 variables if the bounds from the Betti numbers are tight and small instances with up to about 150 variables if the bounds are not tight. In all the instances computed so far, the topology of the spaces involved was known. In the future, we plan to apply our techniques to other cases.

Acknowledgment. We are indebted to Tobias Achterberg for his support of the implementation.

References

[1] T. Achterberg, *SCIP – a framework to integrate constraint and mixed integer programming*. ZIB-Report 04-19, 2004.
[2] T. Achterberg, T. Koch, and A. Martin, *Branching rules revisited*. ZIB-Report 04-13, 2004.
[3] E. Babson, A. Björner, S. Linusson, J. Shareshian, and V. Welker, *Complexes of not i-connected graphs*, Topology, 38 (1999), pp. 271–299.
[4] E. Batzies and V. Welker, *Discrete Morse theory for cellular resolutions*, J. Reine Angew. Math., 543 (2002), pp. 147–168.
[5] A. I. Bobenko and Y. B. Suris, *Integrable systems on quad-graphs*, Int. Math. Res. Not., (2002), pp. 573–611.
[6] M. K. Chari, *On discrete Morse functions and combinatorial decompositions*, Discrete Math., 217 (2000), pp. 101–113.
[7] M. K. Chari and M. Joswig, *Complexes of discrete Morse functions*, Discrete Math., to appear.
[8] Ö. Eğecioğlu and T. F. Gonzalez, *A computationally intractable problem on simplicial complexes*, Comut. Geom., 6 (1996), pp. 85–98.
[9] R. Forman, *Morse theory for cell-complexes*, Advances in Math., 134 (1998), pp. 90–145.
[10] R. Forman, *Morse theory and evasiveness*, Combinatorica, 20 (2000), pp. 489–504.
[11] ——, *A user's guide to discrete Morse theory*, Sém. Lothar. Combin., 48 (2002), pp. Art. B48c, 35 pp.
[12] E. Gawrilow and M. Joswig, polymake: *a framework for analyzing convex polytopes*, in Polytopes – Combinatorics and Computation, G. Kalai and G. M. Ziegler, eds., vol. 29 of DMV Seminar, Birkhäuser, Basel, 2000, pp. 43–74.
[13] ——, polymake: *Version 2.1.0*. http://www.math.tu-berlin.de/polymake, 2004. With contributions by T. Schröder and N. Witte.
[14] M. Grötschel, M. Jünger, and G. Reinelt, *A cutting plane algorithm for the linear ordering problem*, Oper. Res., 32 (1984), pp. 1195–1220.
[15] M. Grötschel, M. Jünger, and G. Reinelt, *On the acyclic subgraph polytope*, Math. Program., 33 (1985), pp. 28–42.
[16] M. Hachimori, *Simplicial complex library*. http://infoshako.sk.tsukuba.ac.jp/~hachi/math/library/index_eng.html, 2001.
[17] P. L. Hammer, E. L. Johnson, and U. N. Peled, *Facets of regular 0-1 polytopes*, Math. Program., 8 (1975), pp. 179–206.
[18] P. Hersh, *On optimizing discrete Morse functions*. Preprint, math.CO/0311270, 2003.
[19] C. S. Iliopoulos, *Worst-case complexity bounds on algorithms for computing the canonical structure of finite Abelian groups and the Hermite and Smith normal forms of an integer matrix*, SIAM J. Comput., 18 (1989), pp. 658–669.
[20] J. Jonsson, *On the topology of simplicial complexes related to 3-connected and Hamiltonian graphs*, J. Combin. Theory Ser. A, 104 (2003), pp. 169–199.
[21] M. Joswig, *Computing invariants of simplicial manifolds*. Preprint, math.AT/0401176, 2004.
[22] M. Jünger, *Polyhedral combinatorics and the acyclic subdigraph problem*, vol. 7 of Research and Exposition in Mathematics, Heldermann Verlag, Berlin, 1985.
[23] B. Korte and J. Vygen, *Combinatorial optimization. Theory and algorithms*, vol. 21 of Algorithms and Combinatorics, Springer, Berlin, 2nd ed., 2002.
[24] T. Lewiner, H. Lopes, and G. Tavares, *Towards optimality in discrete Morse theory*, Exp. Math., 12 (2003), pp. 271–285.

[25] L. Lovász, *Kneser's conjecture, chromatic number, and homotopy*, J. Combin. Theory Ser. A, 25 (1978), pp. 319–324.
[26] F. H. Lutz, *Small examples of non-constructible simplicial balls and spheres*, SIAM J. Discrete Math, 18 (2004), pp. 103–109.
[27] J. R. Munkres, *Elements of Algebraic Topology*, Addison-Wesley, Menlo Park CA, 1984.
[28] U. Pinkall and K. Polthier, *Computing discrete minimal surfaces and their conjugates*, Experiment. Math., 2 (1993), pp. 15–36.
[29] G. Reinelt, *The linear ordering problem: Algorithms and applications*, vol. 8 of Research and Exposition in Mathematics, Heldermann Verlag, Berlin, 1985.
[30] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, vol. 24 of Algorithms and Combinatorics, Springer, Berlin Heidelberg, 2003.

Michael Joswig, Institut für Mathematik, MA 6-2, TU Berlin, 10623 Berlin, Germany
E-mail address: joswig@math.tu-berlin.de

Marc E. Pfetsch, Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany
E-mail address: pfetsch@zib.de