Alcom-FT Technical Report Series
ALCOMFT-TR-02-80

Approximation algorithms for the minimum test set problem

K.M.J. De Bontridder* B.V. Halldórsson†

M.M. Halldórsson‡§ C.A.J. Hurkens¶ J.K. Lenstra‖

R. Ravi‖ L. Stougie‖**

May 13, 2002

Abstract

In the minimum test set problem a set of items is given together with a collection of subsets of the items, called tests. A smallest set of tests is to be selected such that for every pair of items there is a test in the selection that contains exactly one of the two items. This problem is known to be NP-hard in general. We show that, unless P=NP, no algorithm can have an essentially better worst-case performance ratio than the greedy algorithm. In the special case where each test contains at most k items, we devise an $O(\log k)$-approximation algorithm.

Special attention is paid to the case in which each test contains only one or two items. A strong relation with a packing problem on a graph is established, showing immediately that also this special minimum test set problem is NP-hard, contradicting a claim in [Garey & Johnson 1979, page 222]. Furthermore we show APX-hardness of both problems. We derive performance guarantees for a series of local improvement heuristics.

---

*Institute of Information and Computing Sciences, Universiteit Utrecht, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands. e-mail: koendb@cs.uu.nl. Partially supported by the Future and Emerging Technologies programme of the EU under contract number IST-1999-14186 (ALCOM-FT).

†Celera Genomics, Rockville, MD, USA. e-mail: bjarni.halldorsson@celera.com. Work partially done while the author was a graduate student at Department of Mathematical Sciences, Carnegie Mellon University, where he was supported by a Merck Computational Biology and Chemistry Program Graduate Fellowship from the Merck Company Foundation.

‡Science Institute, University of Iceland, IS-107 Iceland. e-mail: mmh@hi.is.

§Iceland Genomics Corp. mmh@uvs.is

¶Department of Mathematics, Technische Universiteit Eindhoven, P.O.Box 513, 5600 MB Eindhoven, The Netherlands. e-mail: {wscor,jkl,leen}@win.tue.nl.

‖Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA, USA. e-mail: ravi@cmu.edu. Supported in part by subcontract No. 16082-RFP-00-2C in the area of "Combinatorial Optimization in Biology (XAXE)," Los Alamos National Laboratories and NSF grant CCR-0105548.

**Centre for Mathematics and Computer Science (CWI), P.O. Box 94079, 1090GB Amsterdam, The Netherlands. e-mail: stougie@cwi.nl.

The latter belong to the scarce results on performance guarantees of local search algorithms in the literature.

1 Prelude

The input of the *minimum test set problem* (MTS) consists of a set of *items*, $\{1, ..., n\}$, and a collection of *tests*, $T_1, ...T_m \subset \{1, ..., n\}$. A test T_l covers (differentiates) item pair $\{i, j\}$ if either $i \in T_l$ or $j \in T_l$ (but not both), i.e., $|T_l \cap \{i, j\}| = 1$. A family $\mathcal{T} \subset \{T_1, ..., T_m\}$ of tests is a *test set* if each of the $n(n-1)/2$ item pairs is covered by at least one test in $\mathcal{T}$. The objective is to find the test set of minimum cardinality τ^*.

The minimum test set problem arises naturally in identification problems: Given a set of individuals (database entries) and a set of binary attributes that may or may not occur in each individual, the goal is to find the minimal subset of attributes (a test set) such that each individual can be identified uniquely from the information the test set contains. In this way the incidence vector of any individual with the test set is a unique binary signature, which distinguishes the individual from any other individual. This problem is also known in the literature as the minimum test collection problem [Kann 1992, Garey & Johnson 1979], and arises commonly in fault analysis, medical diagnostics and pattern recognition (see, e.g., [Moret & Shapiro 1985]).

This paper is the work of two sets of researchers working independently. The first set of researchers was motivated over 20 years ago, by a request from the Wageningen University of Agriculture in The Netherlands concerning investigations of potato diseases. Each potato subspecies is vulnerable for a number of diseases. In order to investigate the effect of special treatments on the diseases a selection of potato subspecies had to be made such that any pair of diseases could be distinguished from one another. Here the diseases are the items and the potato subspecies are the tests.

The problem came to the attention of the second set of researchers following the work described in [Halldórsson et al. 2001], which proposed a new approach of using an array of antibodies that recognize and bind specifically to short peptide sequences (called epitopes) - such an epitope can distinguish proteins that contain this epitope from those that do not. Each antibody binds its cognate epitope by virtue of complementary three dimensional structure. To observe the occurrence of epitope binding, the epitopes are fluorescently tagged. The binding of antibodies in the chip to the given unidentified protein can then be measured using a fluorescence detector. Thus the final output is a binary indicator vector of dimension equal to the number of antibodies in the chip, each bit indicating whether or not the protein is bound to the corresponding antibody.

The proposal in [Halldórsson et al. 2001] is to generate a set of antibodies that recognize a set of epitopes that are shared by many proteins in such a way that the entire set of epitopes covers all possible proteins in the organism's proteome. Moreover, the set of antibodies will be selected such that each protein will be recognized by a unique subset of antibodies, i.e., each protein will have a unique signature of binding antibodies.

This problem can hence be viewed as a minimum test set problem where the items are the proteins and there is a test for each antibody, determined by the binding of the antibodies to the proteins.

In [Moret & Shapiro 1985] a strong relation between MTS and the well-known *set cover problem* has been pointed out. This relation shows immediately NP-hardness of MTS, a fact already shown in [Garey & Johnson 1979]. In the same paper the *greedy* algorithm for MTS is shown to have worst-case performance ratio $O(\log n)$, where n is the number of items. This has been made more precise to a $2 \ln n$-worst-case performance ratio in [Kann 1992].

For the sake of completeness, we review these results briefly in Section 2. In the same section we show that the worst-case performance ratio of the greedy algorithm is best possible by establishing a lower bound of $\Omega(\log n)$ on the worst-case performance ratio of any polynomial time approximation algorithm, unless $P = NP$.

In Section 3 we concentrate on instances of the problem in which each test contains at most k items. We note that this a commonly occurring restriction of MTS. For the protein identification problem introduced in [Halldórsson et al. 2001] the novelty of the approach is the utilization of antibodies that bind to a large subset of proteins. However, most currently known antibodies bind very specifically to protein fragments, justifying particular interest in MTS with small tests. In Section 3 we provide an $O(\log k)$-approximation algorithm for MTS with no more than k items per test. This improves on the straightforward $O(\log((n-k)k))$-approximation guarentee of the greedy algorithm (see [Moret & Shapiro 1985]).

In the even more special case with no more than 2 items per test, which we denote by MTS2, we prove a worst-case performance ratio of 11/8 for the natural greedy algorithm in Section 4.

MTS2 was claimed to be solvable in polynomial time in [Garey & Johnson 1979, page 222]. We disprove this claim by showing that MTS2 is NP-hard in Section 5. The NP-hardness follows from a strong relation between MTS2 and the problem of *packing paths of length 2* in a graph. In Section 6 we show that this path packing problem is APX-hard, showing that it highly unlikely that a polynomial time approximation scheme exists for this problem. As a result from the relations between the two problems APX-hardness of MTS2 follows immediately.

We also exploit the relation between MTS2 and the path packing problem in Section 7 in designing a series of local search heuristics. Each next heuristic in the series defines a larger neighborhood resulting in a lower worst-case performance ratio. We emphasize that the latter results concern worst-case performance analysis of local search heuristics, an interesting research subject, which is under-represented in the existing literature.

In [De Bontridder et al. 2002] branch and bound methods are developed to solve minimum test set problems exactly. Empirical analysis shows that these algorithms are quite fast on a set of practical and random instances.

2 The general MTS

MTS has the following natural REFORMULATION. Identify an element with each item pair and redefine each test as the set of item pairs (elements) that it covers. The objective is then to select a minimum cardinality collection of sets covering all elements. This problem is known in the literature as the *set cover problem* (SCP).

As a consequence, any MTS can be solved by methods for solving SCP. In particular, the greedy algorithm for SCP with worst-case performance ratio $1 + \ln N$ with N the number of elements of SCP, [Johnson 1972] and [Lovász 1975] , leads directly to a greedy algorithm for MTS with worst-case performance ratio $1+2\ln n$ with n the number of items of MTS [Kann 1992].

The above relation between MTS and SCP has been pointed out in [Moret & Shapiro 1985]. In the same paper the authors also show how to reduce SCP to MTS, yielding an alternative proof of NP-hardness of MTS to the one in [Garey & Johnson 1979], in which a reduction from 3-Dimensional Matching was used. The reduction is also used in [Moret & Shapiro 1985] to show that the above mentioned worst-case performance ratio of the greedy algorithm for MTS is tight and that a fully polynomial time approximation scheme does not exist unless P=NP.

Take an instance of SCP with N elements and a collection $S_1, ...S_M$ of sets. Replace each element j by a pair of items, a male $\mathbf{m}_j$ and a female $\mathbf{f}_j$, thus creating a set of $n = 2N$ items for MTS. For each set S_i we define a test $T_i = \{\mathbf{m}_j : j \in S_i\}$. Thus, a set cover $\mathcal{S}$ corresponds to a *partial* test set $\mathcal{T} := \{T_i | S_i \in \mathcal{S}\}$, which differentiates all pairs $\{\mathbf{m}_j, \mathbf{f}_k\}$, $j = 1, \ldots, N$, $k = 1, \ldots, N$ but none of the pairs $\{\mathbf{f}_j, \mathbf{f}_k\}$, $j = 1, \ldots, N$, $k = 1, \ldots, N$, $j \neq k$.

Therefore, an extra collection of tests $\mathcal{T}^C$ is constructed, together covering all pairs of female items. It is not too difficult to see that a family of $\lceil \log_2 N \rceil$ *even splitting tests* is sufficient to achieve this. By including in these tests not only the female items but also their male partners, i.e., including $\mathbf{m}_j$ in a test if $\mathbf{f}_j$ is in the test, these tests differentiate all pairs of items except the N pairs $\{\mathbf{m}_j, \mathbf{f}_j\}$, $j = 1, \ldots, N$. All tests in $\mathcal{T}^C$ are essential since no other tests differentiate pairs of female items. It is easy to see that $\mathcal{S}$ is a set cover if and only if $\mathcal{T} \cup \mathcal{T}^C$ is a test set. The following lemma is immediate.

Lemma 2.1 *An instance of SCP has a solution of size σ if and only if its reduction to MTS described above has a solution of size $\tau = \sigma + \lceil \log_2 N \rceil$.*

An example of this reduction is depicted in Figure 1. We use it to show that the worst-case performance ratio of the greedy algorithm for MTS is essentially best possible unless P=NP, given the inapproximability results for SCP.

[Lund and Yannakakis 1994] showed that SCP cannot be approximated within $c \ln N$ for any $c \leq \frac{1}{4}$ unless NP$\subset$DTIME($N^{poly \ln N}$). [Feige 1998] improved on this result by showing that SCP cannot be approximated within $(1 - \epsilon) \ln N$ unless NP$\subset$DTIME($N^{\log \log N}$). Finally, [Arora and Sudan 1997] showed that SCP cannot be approximated within $o(\log N)$ unless P=NP.

$$
\begin{array}{c|ccc|}
 & S_1 & S_2 & S_3 \\
\hline
1 & 1 & 0 & 0 \\
2 & 0 & 1 & 0 \\
3 & 1 & 1 & 1 \\
4 & 1 & 0 & 1 \\
\hline
\end{array}
\quad \Longrightarrow \quad
\begin{array}{c|cc||ccc|}
 & \multicolumn{2}{c}{\mathcal{T}^c} & T_1 & T_2 & T_3 \\
\hline
\mathbf{m}_1 & 1 & 1 & 1 & 0 & 0 \\
\mathbf{m}_2 & 1 & 0 & 0 & 1 & 0 \\
\mathbf{m}_3 & 0 & 1 & 1 & 1 & 1 \\
\mathbf{m}_4 & 0 & 0 & 1 & 0 & 1 \\
\hline
\mathbf{f}_1 & 1 & 1 & 0 & 0 & 0 \\
\mathbf{f}_2 & 1 & 0 & 0 & 0 & 0 \\
\mathbf{f}_3 & 0 & 1 & 0 & 0 & 0 \\
\mathbf{f}_4 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
$$

Figure 1: Reduction from SCP to MTS

Let I be an instance of SCP with N elements, and $M(I)$ the instance of MTS defined by the above reduction. Lemma 2.1 shows that the size of the optimal solution of $M(I)$ is $\lceil \log_2 N \rceil$ larger than the size of the optimal solution of I. Hence, this reduction does not guarantee hardness of $o(\log n)$-approximation of MTS directly. To achieve this, we make multiple copies of the SCP instance, and apply the reduction to the copied instance. The same trick was used in [Moret & Shapiro 1985] to show that a fully polynomial time approximation scheme does not exist unless P=NP.

Consider the following approximation algorithm for SCP using one call to an approximation algorithm for MTS.

Algorithm 2.1

1 *Given instance I of SCP, create k copies of I, yielding instance I^k of SCP*

2 *Reduce I^k to an instance $M(I^k)$ of MTS in the way described before.*

3 *Solve $M(I^k)$ using approximation algorithm A for MTS.*

4 *Extract a solution to I^k from the solution to $M(I^k)$.*

5 *Split the solution of I^k into k solutions, one for each of the copies of I.*

6 *Select the best of the k solutions as solution for I .*

Clearly, this is a polynomial time algorithm if the number k of copies made of the SCP instance is bounded by a polynomial function of the number of elements of the instance and A is a polynomial time algorithm for solving MTS. We use σ^* to denote the size of an optimal solution of SCP.

Theorem 2.1 *MTS with n items is hard to approximate in polynomial time within $o(\log n)$ unless P=NP.*

Proof. Suppose there exists a polynomial time algorithm A that can approximate MTS within $o(\log n)$. Apply Algorithm 2.1 to instance I of SCP choosing k, the number of copies created, equal to $\lceil \log N \rceil$.

The MTS instance $M(I^k)$ created in Algorithm 2.1 has $2kN$ entries and therefore the performance guarantee of A ensures that the size of the solution returned by A is at most $o(\log(2kN))\tau^*(M(I^k))$, where $\tau^*(M(I^k))$ is the size of the optimal test set of $M(I^k)$. Lemma 2.1 implies that $\tau^*(M(I^k)) = \lceil \log_2 kN \rceil + \sigma^*(I^k)$. Therefore, at least one of the copies of I must have a solution of size no more than $o(\log(2kN))\frac{1}{k}(\lceil \log_2 kN \rceil + \sigma^*(I^k)) = o(\log(kN))(\frac{1}{k}\lceil \log_2 kN \rceil + \sigma^*(I))$. Inserting $k = \lceil \log N \rceil$ in this bound yields a bound of $o(\log N)\sigma^*(I)$, which contradicts the results of [Arora and Sudan 1997] and proves the theorem. □

A similar proof referring to [Feige 1998] holds for a slight refinement of the previous theorem.

Theorem 2.2 *If there exists some $\epsilon > 0$ such that a polynomial time algorithm can approximate MTS within $(1-\epsilon)\ln n$, with n the number of items then NP$\subset$DTIME($n^{\log\log n}$).*

3 MTS with tests of size at most k

In case each of the tests in MTS contains at most k items (MTSk) a straightforward application of the transformation to SCP in [Moret & Shapiro 1985] would give a worst-case performance ratio for the greedy algorithm of $O(\log(k(n-k)) = O(\log n)$, where n is again the total number of items in the MTSk. Instead, we devise an $O(\log k)$-approximation algorithm.

Before describing the algorithm we digress to the concept of partial test sets. Any partial test set $\mathcal{U}$ defines an *equivalence relation* on the set of items. Given $\mathcal{U}$, two items are equivalent if there is no test in $\mathcal{U}$ that differentiates them. This equivalence relation induces a partition of the set of items into subsets $E_1^{\mathcal{U}}, E_2^{\mathcal{U}}, \ldots, E_p^{\mathcal{U}}$, for some p, such that any two items in any subset are equivalent whereas any two items from different subsets are differentiated by a test in $\mathcal{U}$.

First we prove a relation between the MTS problem and the SCP obtained by identifying each item in MTS with an element of SCP, and each test in MTS with a set of SCP. Notice that the resulting SCP is not a reduction of the MTS problem as before.

Lemma 3.1 *Take I as an instance of MTS and as an instance of SCP. Then $\sigma^*(I) \leq \tau^*(I) + 1$.*

Proof. By the definition of MTS all item pairs must by covered. Hence, in any test set of MTS all items, except for at most one, must be present in some test of the test set, i.e., any feasible solution to MTS is a set cover for all but at most one of the items. □

Algorithm 3.1

Phase I *(Cover) Given an MTS instance I, find a set cover of the items with tests, identifying items with elements and tests with sets, using the greedy algorithm for SCP. This yields a solution $\mathcal{S}(I)$. If this set cover is a test set the algorithm stops. Otherwise, $\mathcal{S}(I)$ is a partial test set and we go to the next phase.*

Phase II *Create a new instance of SCP by identifying an element for each pair of items in I that is not differentiated by $\mathcal{S}(I)$ and defining a set for each test not in $\mathcal{S}(I)$ containing the item pairs that are differentiated by the test and are not differentiated by $\mathcal{S}(I)$. Solve this instance of SCP by the greedy algorithm and add the corresponding tests to $\mathcal{S}(I)$, yielding a test set for the MTS instance.*

Theorem 3.1 *The performance guarantee of Algorithm 3.1 for MTSk is $O(\ln k)$.*

PROOF. The set cover $\mathcal{S}(I)$ found in Phase I of the algorithm, has size $\sigma^G(I)$, satisfying $\sigma^G(I) \leq (1 + \ln k)\ \sigma^*(I)$ [Johnson 1972] [Lovász 1975] . Applying Lemma 3.1 yields $\sigma^G(I) = O(\ln k)\tau^*(I)$.

As described above the partial test set $\mathcal{S}(I)$ produced by Phase I partitions the set of items into subsets of pairwise non-covered items, denoted $E_1^{\mathcal{S}}, \ldots, E_s^{\mathcal{S}}$. The non-covered item pairs form the elements in a SCP-instance, that we denote $C(I_R)$. Each test $T \notin \mathcal{S}(I)$ is converted into a set $S = \cup_{t=1}^{s}\{\{i,j\}|i \in T \cap E_t^{\mathcal{S}(I)}, j \in T\backslash E_t^{\mathcal{S}(I)}\}$. Notice that each subset $E_t^{\mathcal{S}}(I)$ contains no more than k items, because each item is in some test of $\mathcal{S}(I)$ and therefore differentiated from at least $n-k$ other items. This together with the fact that each test T contains no more than k items, implies that each set S in $C(I_R)$ contains at most $k(k-1)$ elements. Again applying the result in [Johnson 1972] and [Lovász 1975] and Lemma 3.1, the size $\sigma^G(C(I_R))$ of the greedy set cover on $C(I_R)$ satisfies $\sigma^G(C(I_R)) \leq (1+\ln k(k-1))\sigma^*(I_R) = O(\ln k)\tau^*(I_R) = O(\ln k)\tau^*(I)$.

The result follows by combining the contributions from the two phases.

□

4 Greedy for MTS2

A special case of MTSk is obtained if each test has no more than 2 items (MTS2). First we argue that without loss of generality we may assume that instances of MTS2 only contain tests of size exactly 2.

Let $\mathbf{T}$ denote the set of available tests. Let $\mathcal{T}$ denote a minimal test set.

Suppose that we have k items not contained in any test, ($k \in \{0,1\}$), and l items $s_1, \ldots, s_l$, with s_t contained only in test $\{s_t\} \in \mathbf{T}$, and m pairs of items $(i_1, j_1), \ldots, (i_m, j_m)$ with the property that, for $t = 1, \ldots, m$, $\{i_t, j_t\} \in \mathbf{T}$,

$\{i_t\} \in \mathbf{T}$, possibly $\{j_t\} \in \mathbf{T}$, and no other test contains i_t or j_t, and assume $k+l+m > 0$. Then $\mathcal{T}$ contains, without loss of generality, the first $k+l+2m-1$ tests from $\{s_1\}, \{s_2\}, \ldots, \{s_l\}$, $\{i_1\}, \{i_1, j_1\}$, $\{i_2\}, \{i_2, j_2\}$, $\ldots$, $\{i_m\}, \{i_m, j_m\}$, leaving one item isolated.

Each item i not involved in the above case has the property that (a) $\exists \{i,j\} \in \mathbf{T}$ and (b) $\forall \{i,j\} \in \mathbf{T}$, $\exists \{k,l\} \in \mathbf{T}$, such that $|\{k,l\} \cap \{i,j\}| = 1$.

Without loss of generality we may now assume that $\mathcal{T}$ does not contain singleton tests except the ones mentioned above. For suppose $\mathcal{T}$ does contain another singleton test $\{i\}$. As $\mathcal{T}$ is minimal, it does not contain two tests $\{i,j\}$ and $\{i,k\}$. If $\mathcal{T}$ contains no test $\{i,\cdot\}$, replace $\{i\}$ by any test $\{i,j\} \in \mathbf{T}$, which exists by (a). If by this action i and j become indistinguishable (j was apparently left isolated), or if $\mathcal{T}$ already contains test $\{i,j\}$, replace $\{i\}$ by the corresponding $\{k,l\} \in \mathbf{T}$, see (b).

By eliminating from the original instance all $k+l+2m$ items involved, and the tests that contain them, eliminating all other singleton tests, and adding one isolated item if $k+l+m > 0$, we obtain an equivalent instance of MTS2, with only tests of size two.

Assuming that MTS2 has only tests of size two, as we will do from now on, it can be formulated as a graph optimization problem. Create for each item in MTS2 a vertex of the graph G, and for each test in MTS2 an edge of G between the two vertices corresponding to the two items in the test. This allows for the following characterization of test sets in terms of a graph.

Lemma 4.1 *Given a graph $G = (V,E)$, a subset of the edges $\mathcal{E} \subset E$ is a test set if and only if the subgraph $\mathcal{G} = (V, \mathcal{E})$ has no isolated edges and at most one isolated vertex.*

Proof. $\mathcal{E}$ is a test set if and only if for each edge $(i,j) \in \mathcal{E}$ there exists an adjacent edge $(i,k) \in \mathcal{E}$ or $(l,j) \in \mathcal{E}$. One vertex, u say, may be isolated since, by being the only isolated vertex, for any other vertex $v \in V$, there exists an edge incident to v and not to u in $\mathcal{E}$. □

We assume from now on that the instances of MTS2 that we consider are all feasible.

Now, we consider the greedy algorithm on the graph representation of MTS2. Let n be the number of vertices (items) and m be the number of edges (tests). Each edge covers $2(n-2)$ item pairs. The greedy algorithm will choose the first edge (i,j) arbitrarily, covering all $2(n-2)$ item pairs $\{i,k\}$ and $\{j,k\}$, for all $k \neq i,j$. After this, each other edge (s,t) that is not adjacent to (i,j) will cover $2(n-2)-4$ item pairs, since the pairs $\{s,i\}$, $\{s,j\}$, $\{t,i\}$, and $\{t,j\}$ have already been covered by the edge (i,j).

An edge (i,u) (or (j,u)) adjacent to (i,j) will cover only an extra $n-3+1 = n-2$ item pairs. The vertex u is differentiated from all others except i and j, from which it was already differentiated by (i,j). The vertex i (or j) was already differentiated from all other vertices, except from j (or i) from which it is differentiated now.

Repeating the argument, we deduce that greedy will first start to create a matching in the graph. After having constructed a matching with k edges, addition of a next edge to the matching covers $2(n-2)-4k$ new item pairs, whereas an edge adjacent to an edge already in the matching covers $n-2k$ new item pairs. An edge adjacent to two edges already in the matching covers only 2 new item pairs.

In what we call it's first phase, the greedy algorithm will construct a maximal matching or an almost perfect matching, leaving 2 or 3 vertices unmatched. The set of edges in this matching is called $\mathcal{E}_1$.

A similar simple counting argument shows that in the following phase, the greedy algorithm selects edges that are adjacent to only one edge in $\mathcal{E}_1$, thus creating paths of two edges in the graph, until this is no longer possible, or only one vertex is left that is not incident to any of the selected edges. The set of edges selected in this phase is called $\mathcal{E}_2$. After this second phase, the graph $G_2 = (V, \mathcal{E}_1 \cup \mathcal{E}_2)$ consists of a maximal packing of paths with two edges, and isolated edges and vertices.

In the third phase edges are selected that connect isolated vertices to any of the paths in G_2. Such an edge covers a number of item pairs equal to the number of isolated vertices minus 1, whereas an edge connecting two isolated edges in G_2 covers 2 item pairs, and an edge connecting an isolated edge to a path in G_2 covers only 1 new item pair. The third phase therefore stops as soon as 2 vertices are left that remain isolated. The set of edges selected in this phase is called $\mathcal{E}_3$. The graph $G_3 = (V, \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_3)$ consists of trees containing at least 3 vertices, isolated edges and at most 2 isolated vertices.

In the fourth phase edges are selected that connect two isolated edges in G_3, constituting the set $\mathcal{E}_4$. The subgraph at the end of this phase is G_4.

Finally, in the fifth phase edges are selected that connect the remaining isolated edges and at most one isolated vertex to paths in G_4, constituting the set $\mathcal{E}_5$.

Let τ^G and τ^* denote the size of a greedy and an optimal test set, respectively.

Theorem 4.1 $\tau^G/\tau^* \leq 11/8$*, and this bound is tight.*

PROOF. We notice that the edges that are isolated at the start of the 4th phase were already isolated at the end of the 2nd phase. Thus, reversing the order of the 4th and the 3rd phase does not alter the outcome of the greedy algorithm. The graph $G_4' = (V, \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_4)$ consists of components that are paths of length 3 and 2, isolated edges and isolated vertices. We denote their number by, respectively, c_4, c_3, c_2, and c_1, where the indices indicate the number of vertices in the components. In the 3rd and the 5th phase, all isolated edges and all but one of the isolated vertices in G_4' are connected to one of the paths in G_4'. Therefore, the size of the resulting greedy test set is

$$\tau^G = 3c_4 + 2c_3 + c_2 + (c_2 + c_1 - 1) = 3c_4 + 2c_3 + 2c_2 + c_1 - 1. \tag{1}$$

Given Lemma 4.1, it is not so difficult to see (and indeed will be shown in

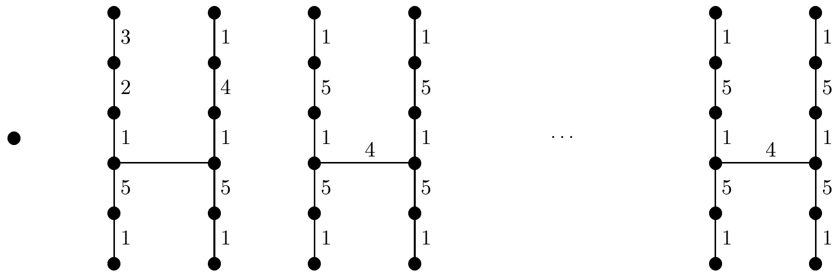

Figure 2: Worst-case instance for the greedy algorithm

Section 5) that

$$\tau^* \geq \frac{2}{3}(|V| - 1).$$

Clearly, $|V| - 1 = 4c_4 + 3c_3 + 2c_2 + c_1 - 1$. Thus,

$$\tau^* \geq \frac{2}{3}(4c_4 + 3c_3 + 2c_2 + c_1 - 1). \tag{2}$$

To obtain yet another lower bound on the size of an optimal test set, consider again the graph G'_4. Each of the isolated edges and each of the isolated vertices of this graph, except one, needs an adjacent edge. Moreover, each pair of isolated edges or vertices cannot be combined by one extra edge into a path of length 2 or 3, otherwise this would have been done already in either the 2nd or the 4th phase. Since we assume that the instances are feasible and non-trivial, we know that, also in an optimal solution, the isolated edges and isolated vertices of G'_4 do not belong to the same path of length 2. Hence,

$$\tau^* \geq 2c_2 + c_1 - 1. \tag{3}$$

Addition of $\frac{9}{8}(2)$ and $\frac{2}{8}(3)$ yields

$$\frac{11}{8}\tau^* \geq 3c_4 + \frac{9}{4}c_3 + 2c_2 + c_1 - 1 \geq \tau^G, \tag{4}$$

where we used (1) for the latter inequality.

To show that the bound is asymptotically tight consider the graph depicted in Figure 2. The graph consists of k equal components on 12 vertices each, and one isolated vertex. The numbers displayed at the edges indicate the phases of the greedy heuristic in which the edges are selected. The size of the resulting test set is $\tau^G = 11k - 1$. It is easy to see that $\tau^* = 8k$, since each component can be partitioned into four paths of length 2. Thus,

$$\lim_{k\to\infty} \frac{\tau^G}{\tau^*} = \frac{11}{8}.$$

□

5 MTS2 and path packing

Consider again the graph representation of MTS2. We will exploit the relation of MTS2 to another graph optimization problem concerning packing paths of two adjacent edges (paths of length 2) in a graph.

Definition 5.1 PACKING PATHS OF LENGTH 2 (PP2). Given a graph, find the maximum number of vertex disjoint paths of length 2.

For the sake of the relation to MTS2, we use a slight diversion of this path packing problem.

Definition 5.2 NEARLY PACKING PATHS OF LENGTH 2 (NPP2). Given a graph, find the maximum number of vertex disjoint paths of length 2, leaving at least 1 vertex isolated.

NP-hardness of this problem follows easily from the NP-completeness of the related decision problem PARTITIONING a graph INTO PATHS OF LENGTH 2 [Kirkpatrick & Hell 1978] (see also [Garey & Johnson 1979, page 193]).

¿From now on we will assume that the instances of MTS2 that we consider are non-trivial, next to being feasible. We call an instance trivial if the corresponding graph is a collection of paths of length 2, in which case the edge set is the only feasible test set.

A consequence of Lemma 4.1 is the following property of the subgraph defined by a minimal test set.

Lemma 5.1 *Let $G = (V, E)$ be a graph representing an instance of MTS2 and let $\mathcal{E}$ be a minimal test set. Then the components of the subgraph $\mathcal{G} = (V, \mathcal{E})$ consist of trees of at least 2 edges and at most 1 isolated vertex.*

PROOF. By Lemma 4.1, $\mathcal{G}$ contains no isolated edges and at most 1 isolated vertex. Therefore, it is sufficient to show that $\mathcal{G}$ does not contain any cycle. Minimality of $\mathcal{E}$ implies immediately that $\mathcal{G}$ does not contain any parallel edges. Thus, any cycle in $\mathcal{G}$ has at least 3 edges. Suppose there is such a cycle in $\mathcal{G}$, and that it contains the edges $\{u, i\}$, $\{i, j\}$, and $\{j, v\}$, (u may be equal to v). If we remove $\{i, j\}$ from $\mathcal{E}$ then we only have to be concerned if i is still differentiated from u and j is still differentiated from v. This is ensured by the fact that every vertex of the cycle has degree at least 2 in $\mathcal{G}$. Indeed, $\mathcal{E}\backslash\{i, j\}$ is again a test set, which contradicts the minimality of $\mathcal{E}$. □

The lemma implies that every component of the subgraph representing a test set contains a path of length 2 and no cycle. This leads to the following relation between test sets and path packings. Since we consider only non-trivial MTS2 problems, we consider only problems in which there exist feasible test sets leaving one vertex isolated.

Lemma 5.2 *Let $G = (V, E)$ be a given graph, and let $\mathcal{E}$ be a minimal test set of size τ having one isolated vertex, then there exists a path packing of size $\pi = |V| - 1 - \tau$.*

Proof. Suppose that the subgraph $\mathcal{G} = (V, \mathcal{E})$ consists of $k + 1$ components of which one is an isolated vertex. Then $\tau = |V| - 1 - k$, since, by Lemma 5.1, $\mathcal{G}$ doesn't contain cycles. From the same lemma we know that the components are trees with at least 2 edges. Therefore, from each of the components we can select a path of length 2. This gives a path packing of size $\pi = k = |V| - 1 - \tau$. $\square$

Conversely, a similar relation holds. In the following lemma a maximal path packing is a path packing such that after its deletion from the graph only isolated edges and vertices remain.

Lemma 5.3 *Let $G = (V, E)$ be a given graph, and let $\mathcal{P}$ be a maximal (near) path packing of size π, then there exists a test set of size $\tau = |V| - 1 - \pi$.*

Proof. Consider the graph $\mathcal{G} = (V, \mathcal{P})$. Notice that the 3π vertices incident to the packing $\mathcal{P}$ are differentiated from one another pairwise and from all $|V| - 3\pi$ isolated vertices of $\mathcal{G}$. Then $\mathcal{P}$ can be extended to a test set by consecutively connecting all but one of the isolated vertices by an edge to components started from the paths of length 2 in $\mathcal{P}$. This is possible since we assume that the instances are feasible. In this way $|V| - 3\pi - 1$ edges to are added to $\mathcal{P}$, resulting in a test set of size $\tau = 2\pi + |V| - 1 - 3\pi = |V| - 1 - \pi$. $\square$

In particular the extension procedure described in the above proof produces a minimal test set, given a maximal path packing. The two lemmas together imply directly a relation between optimal solution values of MTS2 and NPP2.

Lemma 5.4 *Given a graph $G = (V, E)$, an optimal path packing of size π^* and an optimal test set of size are related by τ^*, $\pi^* = |V| - 1 - \tau^*$.*

This relation shows immediately that MTS2 is NP-hard since NPP2 is NP-hard. We notice that this contradicts the statement in [Garey & Johnson 79, page 222], saying that MTS2 is solvable in polynomial time.

Lemma 5.5 *If there exists an approximation algorithm A for NPP2, producing a maximal path packing of size π^A, with worst-case performance ratio $\pi^*/\pi^A \leq \rho$ then there exists an approximation algorithm A' for MTS2, producing a test set of size $\tau^{A'}$, having worst-case performance ratio $\tau^{A'}/\tau^* \leq \frac{3\rho-1}{2\rho}$.*

Proof. The proof follows directly from the arguments in the proofs of Lemmas 5.3 and 5.4.

Consider the graph $G = (V, E)$ and let A' be the algorithm that extends the path packing found by A to a test set, as described in the proof of Lemma 5.3. Then according to Lemma 5.3, $\tau^{A'} + \pi^A = |V| - 1$. Lemma 5.4 says that

$\tau^* + \pi^* = |V| - 1$. Combining these two equalities and inserting the worst-case performance ratio for A, we obtain

$$\tau^{A'} \leq \tau^* + \left(\frac{\rho - 1}{\rho}\right)\pi^*. \tag{5}$$

Clearly, $\pi^* \leq \frac{1}{3}(|V|-1)$. Therefore, again using Lemma 5.4, $\tau^* = |V|-1-\pi^* \geq 3\pi^* - \pi^* = 2\pi^*$. Insertion in (5) yields the desired result. $\square$

6 APX-hardness

In this section we show that PP2 is APX-hard. This follows from a reduction of 3-dimensional matching with each element occurring in at most 3 triples and a result presented in [Kann 1991].

Definition 6.1 3-DIMENSIONAL MATCHING (3DM). Given a set C of m triples $c_i \in X \times Y \times W$, with X, Y, and W disjoint sets having n elements each, find the largest cardinality matching $C' \subset C$, i.e., a set $C' \subset C$ such that no element of X, Y, and W is in more than one triple of C'.

This problem is shown to be APX-hard in [Kann 1991].

Definition 6.2 3-DIMENSIONAL MATCHING-3 (3DM3). This is the 3DM problem in which each element of X, Y, and W occurs in at most three triples of C.

Lemma 6.1 *There exists a constant $\epsilon > 0$ such that it is NP-hard to determine whether an instance $G = (V, E)$ of PP2 has a packing of $\frac{|V|}{3}$ or of size $(1-\epsilon)\frac{|V|}{3}$.*

PROOF. Given an instance of 3DM we create the graph G_P as an instance of PP2 with vertices

- γ_0, γ_1 for each $\gamma \in X \cup Y \cup W$,
- c_x, c_y, c_w for each $c \in C$,

and edges

- (γ_0, γ_1) for all $\gamma \in X \cup Y \cup W$.
- (γ_1, c_γ) for all $c \in C$ and $\gamma \in c$.
- $(c_x, c_y), (c_y, c_w)$ for all $c \in C$.

This reduction is depicted in Figure 3.

We will show that G_P contains $m + 2n$ vertex disjoint paths of length 2 if and only if there exists a matching of size n in the 3DM-instance.

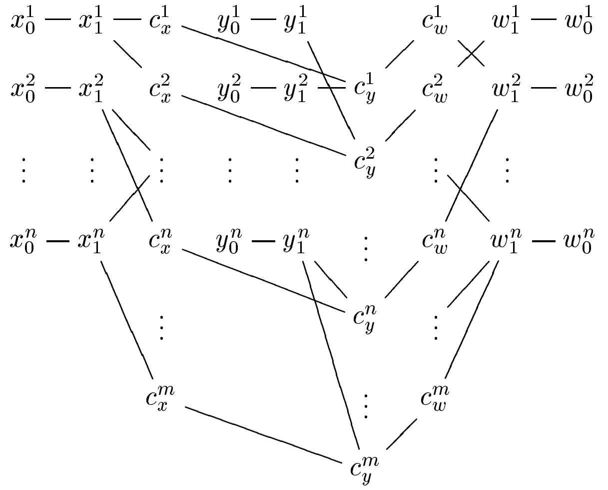

Figure 3: Reduction of three dimensional matching to PP2. Each node γ^i in the ground sets X,Y,W is copied to two nodes, γ_0^i, γ_1^i and three nodes c_x, c_y, c_w are constructed for each hyperedge. The node γ_1^i is connected to the node c_γ^j if γ^i occurs in the hyperedge c^j.

If the 3DM-instance has a matching C' of size n, then G_P contains the the paths (x_0, x_1, c_x^j), (y_0, y_1, c_y^j), (w_0, w_1, c_w^j) for all triples $c^j \in C'$ and the paths (c_x^j, c_y^j, c_w^j) for all triples $c^j \in C \backslash C'$, giving a total number of $3n + (m - n) = m + 2n$ paths.

Now, let μ^* be the size of the largest matching of the 3DM-instance and let $\mathcal{P}$ be an optimal solution of PP2 on G_P with π_P^* paths.

We call a path of type $(\gamma_0, \gamma_1, c_\gamma)$ an *e(lement)-path* and a path of type (c_x, c_y, c_w) a *t(riple)-path*. It is easy to see that other types of paths in any optimal path packing can always be replaced by e-paths. We now bound the number of t-paths in $\mathcal{P}$. Let m_0, m_1, m_2, m_3 be the number of triples in C intersected by, respectively, $0, 1, 2, 3$ e-paths in $\mathcal{P}$.

$$\pi_P^* \leq m_0 + m_1 + 2m_2 + 3m_3 = m + m_2 + 2m_3, \tag{6}$$

noticing that $m = m_0 + m_1 + m_2 + m_3$. Since there are $3n$ e-paths, $3m_3 + 2m_2 + m_1 \leq 3n$. Moreover, clearly, $m_3 \leq \mu^*$. Hence,

$$\pi_P^* \leq m + \frac{3}{2}(n - \mu^*) + 2\mu^* = m + \frac{3}{2}n + \frac{1}{2}\mu^*. \tag{7}$$

The above holds for any 3DM-problem. Now let us switch to 3DM3, and let ϵ_1 be such that it is NP-hard to determine whether 3DM3 has matching of size n or a matching of size $(1 - \epsilon_1)n$. Using (7) it must be NP-hard to determine whether $\pi_P^* = m + 2n$ or $\pi_P^* = m + 2n - \frac{\epsilon_1 n}{2}$. Noticing, that for 3DM3 $m = n$,

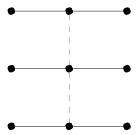

Figure 4: Worst-case instance for H_0

we see that it is NP-hard to determine whether $\pi_P^* = 3n$ or $\pi_P^* = 3n - \frac{\epsilon_1}{6}3n$. Taking $\epsilon = \frac{\epsilon_1}{6}$ finishes the proof. □

We immediately have the following corollary.

Corollary 6.1 *PP2 and NPP2 are APX-hard.*

APX-hardness of MTS2 now follows from Lemmas 5.4 and 6.1.

Theorem 6.1 *MTS2 is APX-hard.*

7 Local search for MTS2

In this section we concentrate on approximation algorithms for NPP2, and extend the NPP2 solution to a test set in the way described in the previous section. The worst-case performance ratio's for the corresponding test set algorithms follow through Lemma 5.5 from those for NPP2. We present a series of local improvement heuristics, with the greedy solution for NPP2, which we denote by H_0, as a basis.

7.1 The greedy algorithm for NPP2

The greedy algorithm for NPP2 selects in each iteration an arbitrary path of length 2 from the graph that remains after deleting all previously selected paths of length 2 and their adjacent edges. The algorithm stops if in the remaining graph no path of length 2 exists or only 3 vertices remain uncovered.

A bad example is given by the graph depicted in Figure 4. The greedy algorithm may selects only 1 path of length 2 whereas 3 is optimal, implying that the worst-case performance ratio is at least 3. We show that this is indeed a worst case example. We denote π^{H_0} as the greedy solution value, and, as usual π^* as the optimal solution value.

Theorem 7.1 *The greedy algorithm for NPP2 has worst-case performance ratio* $\pi^*/\pi^{\mathrm{H}_0} \leq 3$.

Proof. Notice that any path of length 2 in the greedy algorithm intersects at most 3 paths of length 2 in the optimal solution. Moreover, by maximality of the greedy solution, the optimal solution either contains no path of length 2 that does not intersect with a greedy path of length 2, or the greedy solution left exactly 3 vertices uncovered that form a path of length 2. In the latter case the greedy path packing has the same size as the optimal one. From these observations the theorem follows. $\square$

By Lemma 5.5 we obtain the worst-case performance ratio of the algorithm that extends the greedy path packing to a test set.

Corollary 7.1 *The extension of the greedy path packing to a test set has performance ratio* $4/3$. $\square$

7.2 A series of local search heuristics

In this section we propose a series of local improvement heuristics. Each next one in the series defines a larger search neighborhood, requires more computing time, and has strictly better worst-case performance ratio. We notice that the results are examples of worst case performance guarantees for local search algorithms that are very scarce in the combinatorial optimization literature.

The k-th heuristic in the series, which we denote by $\mathrm{H_k}$ starts from a maximal path packing (found by the greedy algorithm) and the graph obtained after deleting the path packing and its adjacent edges. Iteratively, it takes any set of at most k paths of length 2 from the path packing, and reinsert it in the graph. Then a complete enumeration is done to search for $k+1$ paths of length 2 in this graph. If successful these $k+1$ paths are selected, increasing the size of the path packing by 1. The procedure is repeated, until no set of k paths in the current packing allows for such an improvement.

It is clear that a solution found by $\mathrm{H_{k+1}}$ can not be improved locally by $\mathrm{H_k}$. Thus, the worst-case performance ratio of $\mathrm{H_{k+1}}$ will be at most that of $\mathrm{H_k}$.

Lemma 7.1 *For fixed k, $\mathrm{H_k}$ runs in polynomial time.*

Proof. The number of iterations is $O(|V|)$, since each iteration raises the size of the path packing by 1, and any path packing has size no more than $\lfloor(|V|-1)/3\rfloor$.

Each iteration starts with at most $\lfloor(|V|-1)/3\rfloor$ paths of length 2, and therefore the number of possible combinations of k such paths is bounded by $\binom{\lfloor(|V|-1)/3\rfloor}{k}$.

The search for an improvement given a set of k paths in the current solution can be done by completely enumerating all possible combinations of $k+1$ paths of length 2 from the set of all possible paths of length 2 in the graph. The latter is bounded by $\binom{|V|}{3}$.

These three observations complete the proof. $\square$

We notice that the running time of $\mathrm{H_k}$ is not polynomial in k.

7.3 Performance guarantees for local search heuristics

We establish worst-case performance ratio's of the heuristics $\mathrm{H}_0, \ldots, \mathrm{H}_4$. We introduce the notation $V(G)$ for the vertices in graph or subgraph G. Consider the graph $G = (\{u\} \cup V(\mathcal{H}) \cup V(\mathcal{O}), E(\mathcal{H}) \cup E\mathcal{O}))$, with u an isolated vertex, which we will disregard from now on, $\mathcal{H}$ a collection of paths of length 2, which can not be improved within G by the heuristic H_k, and $\mathcal{O}$ another collection of paths of length 2 of G.

¿From now on we will use simply the word *path* for path of length 2, and we distinguish between $\mathcal{H}$-paths and $\mathcal{O}$-paths. Clearly, the worst-case performance ratio of heuristic H_k is obtained on a graph G with the ratio $\frac{|\mathcal{O}|}{|\mathcal{H}|}$ as large as possible. To determine upper bounds on this ratio for $\mathrm{H}_0, \ldots, \mathrm{H}_4$ we formulate five linear programming problems.

For each vertex in $V(\mathcal{H})$ we define a label (vertices in $V(\mathcal{O}\backslash\mathcal{H})$ remain unlabeled). All vertices $v \in V(\mathcal{H}\backslash\mathcal{O})$ receive label 0. All other vertices are as well on an $\mathcal{O}$-path as on a $\mathcal{H}$-path, and their label depends on how the $\mathcal{O}$-path intersects the collection of $\mathcal{H}$-paths.

Let p be an $\mathcal{O}$-path. If p intersects 3 $\mathcal{H}$-paths, then each $v \in p$ receives label 4.

If p intersects 2 $\mathcal{H}$-paths, then either of the following two cases occurs. One $\mathcal{H}$-path contains two vertices of p, both getting label 5, whereas the remaining vertex of p gets label 3. In the other case both $\mathcal{H}$-paths contain only one vertex of p (the remaining vertex is unlabeled), in which case the middle vertex of p gets label 2 and the border vertex of p gets label 3. A little thought should make it clear that the case in which the middle vertex remains unlabeled is dominated by the previous case.

If p intersects only one $\mathcal{H}$-path then $v \in p$ receives label 1 if $|\mathcal{H} \cap p| = 1$. If $|\mathcal{H} \cap p| = 2$, then v receives label 6, and if $|\mathcal{H} \cap p| = 3$ v receives label 7. However, the last case will not occur in a worst-case graph, and the last but one case is always dominated by a construction of the first case. We disregard these two cases from now on.

Denoting by V_i the set of vertices with label i, $i = 0, 1, \ldots, 5$, the number of $\mathcal{O}$-paths can be expressed as

$$|V_1| + \frac{1}{2}|V_2| + \frac{1}{2}|V_3| + \frac{1}{3}|V_4| + \frac{1}{4}|V_5|. \tag{8}$$

Let an $\alpha\beta\gamma$-path be a $\mathcal{H}$-path with the middle vertex labeled β and the border vertices labeled α and γ, with $\alpha \leq \gamma$, $0 \leq \alpha, \beta, \gamma \leq 5$. If $f(\alpha\beta\gamma)$ is the fraction of $\alpha\beta\gamma$-paths among the $\mathcal{H}$-paths, then, for $i = 0, \ldots, 5$,

$$|V_i| = \sum_{(\alpha\beta\gamma)} |\mathcal{H}| f(\alpha\beta\gamma)(1_i(\alpha) + 1_i(\beta) + 1_i(\gamma)), \tag{9}$$

with $1_i(x) = 1$ if $x = i$, and 0 otherwise. Substituting (9) in (8) and dividing the result by $|\mathcal{H}|$ yields a formulation of the objective of finding a graph with highest possible ratio $\frac{|\mathcal{O}|}{|\mathcal{H}|}$.

The objective function is to be maximized under the restrictions that all fractions are non-negative and add up to 1:

$$\sum_{(\alpha\beta\gamma)} f(\alpha\beta\gamma) = 1. \tag{10}$$

An extra restriction follows from the definition of the labels 2, 3, and 5. For any vertex with label 3 there must be either a vertex with label 2 or two vertices with label 5. Therefore we add the equality

$$|V_2| + \frac{1}{2}|V_5| = |V_3|. \tag{11}$$

The solution of this first LP-problem which we denote by LP_0 gives an upper bound on the ratio ρ_0 of the greedy heuristic H_0. The solution is 3, with $f(111) = 1$ and all other fractions equal to 0. The bound is matched with a worst-case example depicted in Figure 4.

Since a 111-path can be improved by any $\mathrm{H_k}$, $k \geq 1$, such paths should get a 0-fraction in the solution of the LP-problems for $k \geq 1$. However, there are more path types to be excluded, even for $\mathrm{H_1}$ (think e.g. of a 132-path). In order to facilitate definition of the proper restriction, we define the notion of a *black vertex*.

Define $N_{\mathcal{H}}(v)$ as the pair of vertices on the same $\mathcal{H}$-path as v. Similarly, $N_{\mathcal{O}}(v)$ denotes the pair of vertices on the same $\mathcal{O}$-path as v. A vertex v is black if $v \in V(\mathcal{H})$ and the subgraph induced by the vertex set $V(\mathcal{O}\backslash\mathcal{H}) \cup \{N_{\mathcal{H}}(v)\backslash N_{\mathcal{O}}(v)\}$ contains a path. A vertex being black depends only on the type of $\mathcal{H}$-path containing it. For instance 111-paths have 3 black vertices, of 212-paths the two vertices labeled 2 are black, but the vertex labeled 1 is not, 333-paths do not contain any black vertices. A black vertex threatens to create an improvement possibility. Specifically, if a black vertex exists with label 1 then $\mathrm{H_1}$ can improve $\mathcal{H}$, which must be excluded at this point. Let $BL(\mathcal{H})$ denote the set of black vertices. Thus, to create LP_1, we add to LP_0 the restriction that fractions of path-types that contain black vertices with label 1 are 0. We give the description of this constraint comprehensively in terms of vertex sets, since the description in terms of the fractions directly would be already rather lengthy here and descriptions would become even more lengthy in the next LP-formulations.

$$|BL(\mathcal{H}) \cap V_1| = 0. \tag{12}$$

The optimal solution value of LP_1 is 2, which is is matched with a worst-case example depicted in Figure 4. We emphasize here that we have not aimed at finding the LP-formulation that excludes all configurations on which an $\mathrm{H_1}$-improvement is possible. In this way we risk having an LP-formulation with an optimal value that is too high. However, if the optimal value is matched with an example in which this value is attained, then we know that we have not left out any constraints that are binding and non-redundant in the optimal solution

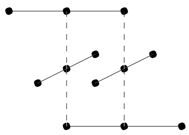

Figure 5: Worst-case instance for H_1

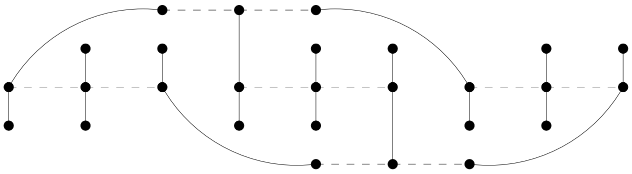

Figure 6: Worst-case instance for H_2

of the complete LP-formulation. It is important though that no configurations are excluded that should not have been excluded.

To find an upper bound for ρ_2 we observe that any $\mathcal{O}$-path containing a black vertex labeled 3 and a black vertex labeled 2 or two black vertices labeled 5 (black vertices labeled 5 always come in pairs) gives rise to a H_2-improvement. Thus, for any $\mathcal{H}$-path p with a black vertex labeled 2 or two vertices labeled 5, there must exist a $|\mathcal{H}$-path q with a non-black vertex labeled 3, which is on the same $\mathcal{O}$-path as the black vertex. The other way round should also hold. Therefore, a constraint that enforces these situations does not only depend on a pair of path-types but also on how they are related through an $\mathcal{O}$-path. To define such a constraint we distinguish $\mathcal{H}$-paths of type 033, 303, 333, 032, 233, 334, 343, 234, 344, 434, 255, 355, 535, or 525 with a $^+$ or a $^-$ label. A path q of any of these types gets a $^+$ label if there is a black vertex $v \in V_2 \cup V_3 \cup V_5 \backslash V(q)$ with $N_{\mathcal{O}}(v) \cap V(q) \neq \emptyset$, and a $^-$ label otherwise. For these type of paths we also distinguish two variables $f(\alpha\beta\gamma^+)$ and $f(\alpha\beta\gamma^-)$ in our LP formulations. The non black vertices $v \in V_2 \cup V_3 \cup V_5$ contained in a $\mathcal{H}$-path with a $^-$ label are white. As a consequence an $\mathcal{O}$-path can not contain a black and a white vertex. All vertices in $V(\mathcal{H})$ that are not white or black are gray. We define $BL(\mathcal{H})$, $GR(\mathcal{H})$, and $WH(\mathcal{H})$ as the set of black, gray, and white vertices, respectively. We obtain the formulation LP_2 for bounding ρ_2, by adding the following constraints to LP_1:

$$|BL(\mathcal{H}) \cap V_2| + \frac{1}{2}|BL(\mathcal{H}) \cap V_5| \leq |GR(\mathcal{H}) \cap V_3|, \tag{13}$$

and

$$|BL(\mathcal{H}) \cap V_3| \leq |GR(\mathcal{H}) \cap V_2| + \frac{1}{2}|GR(\mathcal{H}) \cap V_5|. \tag{14}$$

The optimal solution value of LP_2 is $\frac{9}{5}$, which is matched by an example depicted in Figure 6.

In order to find an upper bound for ρ_3 we define LP_3 by adding three additional constraints to LP_2. The first and the second constraint are based on the following observation. If two gray vertices v and w labeled 2, 3, or 5 are on the same $\mathcal{H}$-path but not on the same $\mathcal{O}$-path, then $N_{\mathcal{O}}(v)$ and $N_{\mathcal{O}}(w)$ can not both contain a black vertex, unless these two vertices are contained in one $\mathcal{H}$-path q of type 202, 222, 232, 242, 323, or 223. If in this case q is of type 222, 232, 323, or 223, there can not be an $\mathcal{O}$-path containing the gray vertex contained in q and a black vertex. With $|\alpha\beta\gamma|$ we denote the number of $\mathcal{H}$-paths of type $\alpha\beta\gamma$. In Figure 7 we depict these situations. The solid and the dashed edges correspond to edges in $\mathcal{O}$ and $\mathcal{H}$, respectively. The relative sizes of configurations that are excluded in the above observations lead to the following two constraints, in which $|\alpha\beta\gamma|$ denotes the number of paths of type $\alpha\beta\gamma$.

$$\begin{aligned}&|BL(\mathcal{H}) \cap V_2| + \tfrac{1}{2}|BL(\mathcal{H}) \cap V_5| - |202| - |222| - |242| \\ &\leq |\mathcal{H}\text{-paths containing a gray vertex labeled 3}|,\end{aligned} \tag{15}$$

and

$$\begin{aligned}&|BL(\mathcal{H}) \cap (V_2 \cup V_3)| + \tfrac{1}{2}|BL(\mathcal{H}) \cap V_5| - |202| - |242| \\ &\leq |\mathcal{H}\text{-paths containing a gray vertex labeled 2, 3, or 5}|.\end{aligned} \tag{16}$$

The third constraint is based on the observation that an $\mathcal{O}$-path can not contain three black vertices labeled 4:

$$|BL(\mathcal{H}) \cap V_4| \leq 2 \cdot |GR(\mathcal{H}) \cap V_4|. \tag{17}$$

The optimal solution value of LP_3 is $\frac{11}{7}$, matched by an example depicted in Figure 8. The upper bound for ρ_4 is obtained by adding constraints based on the following two observations. First consider a $\mathcal{H}$-path containing at least two gray vertices labeled 2, 3, or 5 that are not contained in the same $\mathcal{O}$-path. By definition such a path has a $^+$ label and therefore contains a gray vertex v for which $N_{\mathcal{O}}(v) \cap BL(\mathcal{H}) \cap (V_2 \cup V_3 \cup V_5) \neq \emptyset$. Now observe that for each vertex $u \in (N_{\mathcal{H}}(v) \setminus N_{\mathcal{O}}(v)) \cap GR(\mathcal{H}) \cap (V_2 \cup V_3 \cup V_5)$ there can only be a $\mathcal{H}$-path p with a $^+$ label for which $V(p) \cap N_{\mathcal{O}}(u) \neq \emptyset$ if there is a $\mathcal{H}$-path q of type 202, 222, 232, 242, 323 or 223 containing a black vertex in $N_{\mathcal{O}}(v)$ and a black vertex w for which $V(p) \cap N_{\mathcal{O}}(w) \neq \emptyset$. If in this case q is of type 222, 232, 323, or 223, there can not be an $\mathcal{O}$-path containing the gray vertex contained in q and a black vertex.

If $N_{\mathcal{O}}(u)$ contains at least one black vertex, then a $\mathcal{H}$-path r of type 202, 222, 232, 242, 323 or 223 can also contain the black vertices in $N_{\mathcal{O}}(v)$ and $N_{\mathcal{O}}(u)$. If

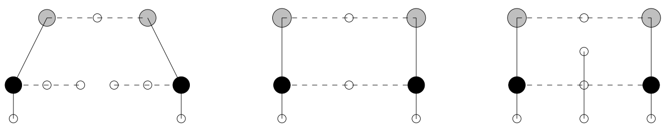

Figure 7: H_3 *improves the graphs on the left and on the right, but not the middle one.*

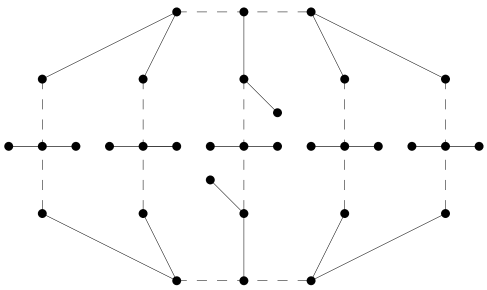

Figure 8: Worst-case instance for H_3

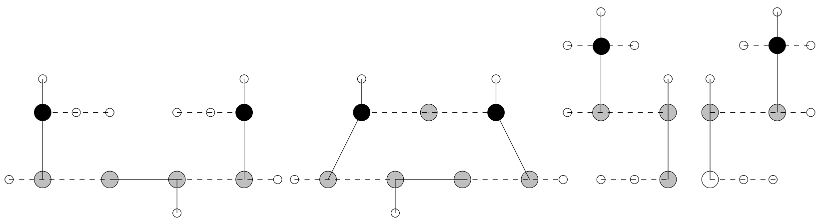

Figure 9: H_4 *improves the graph on the left, but not the other ones.*

in this case r is of type 222, 232, 323 or 223, then the gray vertex contained in r can not be contained in the same $\mathcal{O}$-path as a black vertex or a gray vertex contained in a path with a $^+$ label. In all other cases the gray vertices of type 2, 3, or 5 incident with a $\mathcal{H}$-path without a label can be contained in the same $\mathcal{O}$-path as vertex u. Vertex u can also be incident with an $\mathcal{O}$-path containing a white vertex; see Figure 9. Therefore, we add the following constraint to LP_3:

$$\begin{aligned}&2|BL(\mathcal{H}) \cap (V_2 \cup V_3)| + |BL(\mathcal{H}) \cap V_5| - |222| - |232| - |323| - |223|\\ &-3|242| - 3|202| \leq |\mathcal{H}\text{-paths with a }^+\text{ label}| - |333^+| + |434^+| + |344^+|\\ &+2|\mathcal{H}\text{-paths without a label containing a gray vertex labeled 2, 3, or 5}|\\ &+|\,WH(\mathcal{H}) \cap (V_2 \cup V_3)| + \tfrac{1}{2}|\,WH(\mathcal{H}) \cap V_5|.\end{aligned} \tag{18}$$

Secondly, notice that for a gray vertex v labeled 4 with $N_{\mathcal{O}}(v) \cap BL(\mathcal{H}) = 2$ there cannot exist a vertex $w \in N_{\mathcal{H}}(v) \cap V_3 \cap GR(\mathcal{H})$ with $N_{\mathcal{O}}(w) \cap (V_2 \cup V_5) \cap BL(\mathcal{H}) \neq \emptyset$, see Figure 10. The only exception is that there is a $\mathcal{H}$-path of type 224 that is incident with a vertex in $N_{\mathcal{O}}(v)$ and with a vertex in $N_{\mathcal{O}}(w)$. Notice that for a vertex v labeled 4 incident with a $\mathcal{H}$-path of type 344^+, 434^+, 334^+, or 343^+ we have $|N_{\mathcal{O}}(v) \cap BL(\mathcal{H})| \leq 1$ and as a consequence $|N_{\mathcal{O}}(v) \cap GR(\mathcal{H})| \geq 1$. We complete LP_4 by adding the following constraint:

$$\begin{aligned}&2 \cdot |BL(\mathcal{H}) \cap V_4| + 2 \cdot |BL(\mathcal{H}) \cap V_2| + |BL(\mathcal{H}) \cap V_5| - 2 \cdot |202|\\ &-2 \cdot |222| - 2 \cdot |232| - 2 \cdot |242| - 2 \cdot |224| \leq 4 \cdot |GR(\mathcal{H}) \cap V_4|\\ &+2 \cdot |\mathcal{H}\text{-paths incident with at least one gray vertex labeled 3 and no}\\ &\text{gray vertex labeled 4}| - 4 \cdot |344^+| - 4 \cdot |434^+| - |334^+| - |343^+|\end{aligned} \tag{19}$$

The optimal solution value of LP_4 is $\frac{3}{2}$, which coincides with the ratio in the example of Figure 11.

Of course, LP_4 could be extended to an LP-formulation for bounding ρ_5, but the results become more and more complicated. The analysis so far also did not shed any light on patterns that could be used in further investigations on higher order improvement heuristics. Therefore, we decided to stop here. The following table comprises the results on $\mathrm{H}_0, \ldots, \mathrm{H}_4$, and also of their extensions for the minimum test set problem with two items per test. The entries in the table for MTS2 are obtained by direct application of Lemma 5.5.

k	0	1	2	3	4
ρ_k	3	2	$\frac{9}{5}$	$\frac{11}{7}$	$\frac{3}{2}$
$\frac{3\rho-1}{2\rho}$	$\frac{4}{3}$	$\frac{5}{4}$	$\frac{11}{9}$	$\frac{13}{11}$	$\frac{7}{6}$

8 Postlude

It remains open to which value ρ_k will tend if k tends to ∞. However, we know from the APX-hardness of MTS2 established in Section 6 that it will not become arbitrarily close to 1.

As have been shown in Section 7.1 a direct analysis of ρ_0, the worst-case performance ratio of the greedy algorithm H_0 for NPP2, is more transparent than the one in Section 7.3. Direct analysis is also still an alternative for establishing ρ_1, but becomes rather complicated already for finding ρ_2, and we have not been able to use it for finding ρ_3. Still, one might hope to find a simpler argument for the worst-case analysis of the local search heuristics, since also the one using the LP-formulations becomes too complicated to proceed much further than ρ_5 or ρ_6.

In this respect the comparison to a similar problem is of interest.

Definition 8.1 PACKING TRIANGLES (PTP) Given a graph, find the maximum number of vertex disjoint *triangles* (cliques of size 3).

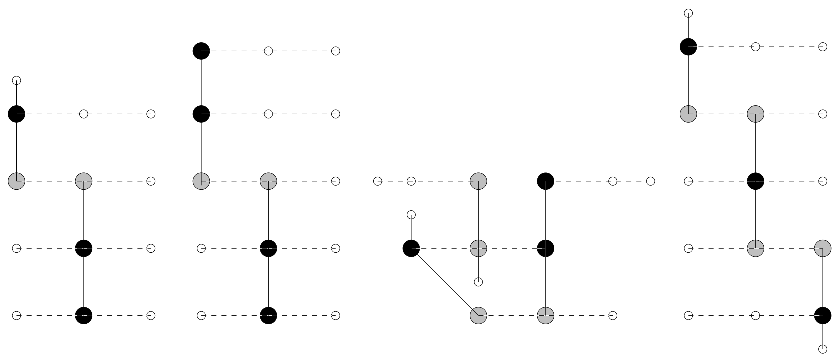

Figure 10: H_4 *improves the graph on the left, but not the other ones.*

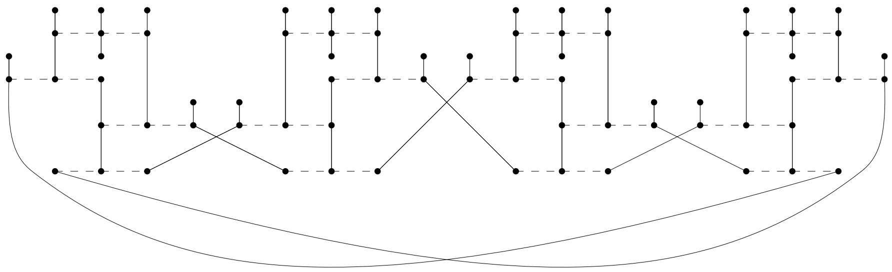

Figure 11: Worst-case instance for H_4

[Hurkens & Schrijver, 1989] presents worst-case performance ratios for a sequence of local search heuristics for PTP that are similar to the one we defined above (just replace every path of length 2 in the description by triangle).

We can show that these ratios form upper bounds on the worst-case performance ratios for the local search heuristics for NPP2. Using ϕ_k to denote the worst-case performance ratio of the k-th heuristic in the sequence for PTP, [Hurkens & Schrijver, 1989] give

$$\phi_k = \begin{cases} \frac{3\cdot 2^{\frac{1}{2}s+1}-3}{2\cdot 2^{\frac{1}{2}s+1}-3} & \text{if } s \text{ is even} \\ \frac{3\cdot 2^{\frac{1}{2}s+1}-2}{2\cdot 2^{\frac{1}{2}s+1}-2} & \text{if } s \text{ is odd} \end{cases},$$

which are, written out in a table,

k	0	1	2	3	4	5	6	7	8	$\cdots$	∞
ϕ_k	3	2	$\frac{9}{5}$	$\frac{5}{3}$	$\frac{21}{13}$	$\frac{22}{14}$	$\frac{45}{29}$	$\frac{46}{30}$	$\frac{93}{61}$	$\cdots$	$\frac{3}{2}$

These bounds, together with the observation that $\rho_k \leq \phi_k$, $k = 0, 1, \ldots$ and the worst-case examples in Figures 4, 5, and 6 prove the values for ρ_0, ρ_1, and ρ_2. However, from H_3 on the ratio's for the two problems start to diverge, and H_4 for NPP2 attains already a worst-case ratio of $\frac{3}{2}$, which is the best worst-case ratio for any H_k for PTP.

The difference in structure between NPP2 and PTP makes that the "clean" analysis for the PTP-heuristics cannot be used for the NPP2-heuristics.

References

[1] S. Arora and M. Sudan. Improved low degree testing and its applications. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, 1997, 485–495.

[2] K.M.J. De Bontridder, J.K. Lenstra, J.B. Orlin, L. Stougie. *Branch-and-bound algorithms for the minimum test set problem.* Technical Report SPOR-??, Technische Universiteit Eindhoven, 2002.

[3] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM 45*, 1998, 634–652.

[4] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* W.H. Freeman and Company, 1979.

[5] B.V. Halldórsson, J.S. Minden, and R. Ravi. PIER: Protein identification by epitope recognition. In N. El-Mabrouk, T. Lengauer, and D. Sankoff, editors, *Currents in Computational Molecular Biology 2001*, 2001, 109–110.

[6] C.A.J. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems. *SIAM Journal on Discrete Mathematics 2*, 1989, 68–72.

[7] D.S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences 9*, 1972, 256–278.

[8] V. Kann. *On the approximability of NP-complete optimization problems.* PhD thesis, Royal Institute of Technology, Stockholm, Sweden, May 1992.

[9] V. Kann Maximum bounded 3-dimensional matching is MAX SNP-complete. *Information Processing Letters*, 37:27-35, 1991.

[10] D.G. Kirkpatrick and P. Hell. On the complexity of a generalized matching problem. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, 1978, 240-245.

[11] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics 13*, 1975, 383–390.

[12] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM 41*, 1994, 960–981.

[13] B.M.E. Moret and H.D. Shapiro. On minimizing a set of tests. *SIAM Journal on Scientific and Statistical Computing 6*, 1985, 983–1003.