

Relaxation of 3-partition instances

Sebastiaan J.C. Joosten^{a,c} and Hans Zantema^{b,c}

^aDepartment of Applied Mathematics, University of Twente, Enschede, The Netherlands

^bDepartment of Computer Science, TU Eindhoven, Eindhoven, The Netherlands

^cInstitute for Computing and Information Sciences, Radboud University Nijmegen, Nijmegen, The Netherlands

February 19, 2013

Abstract

The 3-partition problem admits a straightforward formulation as a 0-1 Integer Linear Program (ILP). We investigate problem instances for which the half-integer relaxation of the ILP is feasible, while the ILP is not. We prove that this only occurs on a set of at least 18 elements, and in case of 18 elements such an instance always contains an element of weight ≥ 10 . These bounds are sharp: we give all 14 instances consisting of 18 elements all having weight ≤ 10 . Our approach is based on analyzing an underlying graph structure.

1 Introduction

Let $3k$ natural numbers be given, possibly containing duplicates, can we partition them into k triples all having the same sum? This problem is called the 3-partition problem (3-PART), and is well-known to be NP-complete [Garey, 1975]. A straightforward approach to deal with this problem is the following: first determine the set \mathcal{C} of all *candidate sets*, that is, all sets of three elements from the given set of numbers having sum N/k where N is the sum of all given numbers. Now 3-PART can be reformulated as finding k of these candidate sets in such a way that every element occurs exactly once in a chosen candidate set. A solution of 3-PART now consists of a mapping $f : \mathcal{C} \rightarrow \{0, 1\}$ such that

$$\sum_{C \ni a} f(C) = 1$$

for every of the numbers a , where the sum runs over all $C \in \mathcal{C}$ containing the number a . Observe that in this way 3-PART has been expressed as feasibility of an integer linear program (ILP). If we extend the range of f to the full interval $[0, 1]$ rather than the two elements $\{0, 1\}$, then we obtain a linear program (LP) which is known to be polynomially solvable. This implies, assuming $P \neq NP$, that there must be 3-PART instances that are feasible to the LP, but for which 3-PART has no solution. Finding such an instance was stated as an open problem in [Kern and Qiu, 2011]. In this paper we present and investigate such instances.

Most of the solutions we found were half-integral, that is, satisfying $f : \mathcal{C} \rightarrow \{0, \frac{1}{2}, 1\}$. Instances of 3-PART having a solution are called *feasible*, therefore instances that are not feasible but admit a half-integral solution we call *nearly-feasible*. Two of our main results are

- Every nearly-feasible instance has $k \geq 6$.
- Every nearly-feasible instance with $k = 6$, contains a number being at least 10.

Tightness of both bounds follows from showing that the 18 numbers

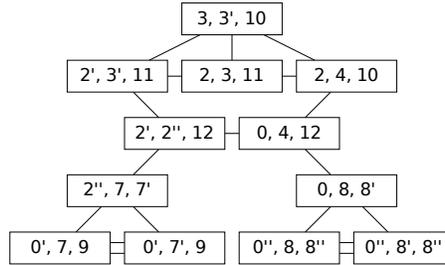
$$0, 0, 1, 1, 1, 2, 2, 2, 4, 4, 4, 5, 5, 5, 8, 8, 10, 10$$

forms a nearly-feasible instance; in fact we will present all nearly-feasible instances for which $k = 6$ elements and all numbers are ≤ 10 .

The analysis of nearly-feasible instances is guided by its underlying graph structure. A half-integral solution $f : \mathcal{C} \rightarrow \{0, \frac{1}{2}, 1\}$ easily implies a selection of $2k$ elements of \mathcal{C} in which each of the $3k$ numbers is chosen exactly twice: if $f(C) = 1$ the triple C is chosen twice, if $f(C) = \frac{1}{2}$ then C is chosen once, and if $f(C) = 0$ then C is not chosen at all. Now for the half-integral solution the underlying undirected multigraph is defined to have $\{C \in \mathcal{C} \mid f(C) = 1\}$ as its set of nodes, and for two nodes C, C' the number of edges between C and C' is defined to be the size of $C \cap C'$. For instance, on the 18 numbers

$$0, 0', 0'', 2, 2', 2'', 3, 3', 4, 7, 7', 8, 8', 8'', 9, 10, 11, 12$$

in which primes are added to distinguish numbers with the same value, a half-integral solution is represented by the following multigraph



In the half-integral solution we have $f(C) = \frac{1}{2}$ for all C being a node of this graph, and $y_C = 0$ for all other C . Note that this indeed yields a half-integral solution since every of the 18 numbers occurs exactly twice in a triple corresponding to a node in the graph, while the edges in the graph connect the nodes with such a common occurrence.

For solutions with minimal k we observe that $f(C) = 1$ will not occur. Hence, the number of nodes of the underlying multigraph will be exactly $2k$. Our bound $k \geq 6$ is obtained by proving that for $k \leq 5$ no multigraph with $2k$ nodes exists satisfying some basic graph properties derived from this underlying structure. It turns out that for $k = 6$ there are exactly 14 such multigraphs up to isomorphism. This raises the question whether all of these 14 multigraphs can be realized as nearly-feasible instance. The answer is positive: for each of these 14 multigraphs we give a

corresponding nearly-feasible instance having only this multigraph as the underlying structure for a corresponding half-integral solution.

Many of our results are based on computer support. In particular, apart from using Haskell as a running language we made use of the following tools.

- **Yices**, a Satisfiability Modulo Theory (SMT) solver, was used to ensure that the nearly-feasible instances have no solution to 3-PART. Moreover, it was used to find the 14 nearly-feasible instances corresponding to the 14 multigraphs. Details about **Yices** can be found in [Dutertre and de Moura, 2006].
- **genbg**, a bipartite graph generator from the **nauty** package, was used to generate cubic multigraphs up to isomorphism. Details can be found in [McKay, 2009].
- The simplex algorithm from the GLPK package was used to find LP solutions. Details on the simplex algorithm are found in standard textbooks such as [Karloff, 1991].

The paper is organized as follows. In Section 2 the basic notions are introduced. In Section 3, the underlying graph structure is analyzed, yielding the lower bound $k \geq 6$ for nearly-feasible instances. Moreover, for all 14 possible multigraphs for $k = 6$, corresponding instances are given. In Section 4, minimal values for the case $k = 6$ are investigated. We conclude in Section 5, in which we also relate our problem to variants like the 3-dimensional matching problem.

2 Definition and notation

As sketched in the introduction, the 3-partition problem (3-PART) is the decision-problem whether a given set of elements with integer weights can be partitioned into triples all having the same sum of weights. Here we prefer to speak about elements with weights rather than only numbers in order to be able to distinguish between copies of the same number. This 3-PART problem was proven to be NP-complete by Garey [Garey, 1975].

More precisely, for $k \in \mathbb{N}$ a *3-PART* instance is a set A of $3k$ elements, each of a certain weight determined by a weight function $w : A \rightarrow \mathbb{N}$. For giving an instance, we write $w(a_1), \dots, w(a_{3k})$ for a_1, \dots, a_{3k} being the elements of A . The objective is to partition A over k sets of 3 elements each, where for each of these sets the sum of the weights is equal.

Since we know k and the total sum of all sets, we also know the sum per set $c = \sum_{a \in A} w(a)/k$. For $i \in \mathbb{N}$ write $\mathcal{P}_i(A)$ for the set of all subsets of A having exactly i elements. For any set $C \subseteq A$ we write $w(C) = \sum_{a \in C} w(a)$.

Any set $C \in \mathcal{P}_3(A)$ satisfying $w(C) = c$ is called a *candidate set*. The set of all candidate sets is denoted by \mathcal{C} : $\mathcal{C} = \{C \in \mathcal{P}_3(A) \mid w(C) = c\}$. Note that $|\mathcal{C}| \leq \binom{k}{3} \in O(k^3)$, so calculating \mathcal{C} is polynomial.

A solution of 3-PART is defined to be a selection of candidate sets represented by a mapping

$$f : \mathcal{C} \rightarrow \{0, 1\} \text{ such that: } \sum_{C \ni a} f(C) = 1 \text{ for all } a \in A, \quad (1)$$

in which the sum runs over all $C \in \mathcal{C}$ containing a .

In this way the NP-hard problem 3-PART is expressed as an integer linear program (ILP).

If we extend the range of f to the real interval $[0, 1]$, so in this formulation we replace $f : \mathcal{C} \rightarrow \{0, 1\}$ by $f : \mathcal{C} \rightarrow [0, 1]$, we obtain a linear program (LP). Since LP is well-known to be

polynomially solvable, this implies, assuming $P \neq NP$, that there must be 3-PART instances that have a solution to the LP, but are still not feasible. Finding such an instance was stated as an open problem in [Kern and Qiu, 2011]. We found several such 3-PART instances with no solutions to the ILP, but having LP solutions. Most of them are half-integral solutions, that is, the range of f is contained in $\{0, \frac{1}{2}, 1\}$. Such instances without integer but having half-integral solution we will call *nearly-feasible*. In order to avoid writing fractions, we multiply the result by 2. In this way every half-integral solution coincides with a solution of the following ILP problem: find a mapping

$$g : \mathcal{C} \rightarrow \{0, 1, 2\} \text{ such that: } \sum_{C \ni a} g(C) = 2 \text{ for all } a \in A, \quad (2)$$

in which again the sum runs over all $C \in \mathcal{C}$ containing a .

Definition. We say that a 3-PART instance is nearly-feasible if \mathcal{C} is such that the problem (2) has a solution, while the problem (1) does not.

3 Nearly-feasible instances with minimal k

In this section, we prove that every nearly-feasible instance has $k \geq 6$ and show that this bound is tight. For this, we construct a multigraph (V, E) corresponding to any solution g of (2) defined by

$$V = \{C \in \mathcal{C} \mid g(C) = 1\}, \quad E(\{C, C'\}) = |C \cap C'| \quad (3)$$

So the vertices are the candidate sets for which $g(C) = 1$; since (1) has no solution by definition of nearly-feasible, there are such candidate sets and $V \neq \emptyset$. The number of edges between two such vertices is given by the number of elements the two candidate sets have in common. This (multi)graph is called the *solution graph* of g .

The following lemma investigates basic properties of solution graphs for k being minimal.

Lemma 1. Let g be a solution to a nearly-feasible instance with minimal k , and (V, E) the solution graph of g . Then:

1. (V, E) is cubic, that is: $\forall C \in V. \sum_{C' \neq C} E(\{C, C'\}) = 3$.
2. (V, E) is connected
3. $|V| = 2k$
4. The candidate sets corresponding to any $k - 1$ vertices are not pairwise disjoint

Proof. Let A be the elements of the original 3-PART instance.

We first prove part 1 of the lemma. By (2), $\sum_{C \ni a} g(C) = 2$. For $C \in V$ we have $g(C) = 1$, so for every $a \in C$ there is exactly one $C' \neq C$ with $a \in C'$ and $g(C') = 1$. So, for $C \in V$: $\sum_{C' \neq C} |C \cap C'| = |C| = 3$. Hence $\forall C \in V. \sum_{C' \neq C} E(\{C, C'\}) = 3$.

To see that (V, E) is connected, we use minimality of k . Assume (V, E) is not connected. Let V' be any connected component. Note that $A - \bigcup V'$ with the original weight-function is a 3-PART instance. It must have a solution to (2), since we can take the original values in g . This instance has a smaller k (since V' is non-empty), so it cannot be nearly-feasible. Therefore, $A - \bigcup V'$ must have a solution to (1). Using the same argument, $\bigcup V'$ must also have a solution to (1). However,

then A must have a solution to (1) too (by combining the two solutions), contradicting that the original 3-PART instance is nearly-feasible. This proves part 2 of the lemma.

Every candidate set contains three elements, so $|V| \cdot 3 = \sum_{C \in \mathcal{V}} |C|$. As we have seen in the proof of part 1, every element occurs in exactly two candidate sets: $3|V| = 2|A|$. By definition of a 3-PART instance, $|A| = 3k$. Hence $|V| = 2k$.

To prove part 4, assume there are $k - 1$ such disjoint candidate sets. As the total number of elements is $3k$, there are exactly three elements a_1, a_2, a_3 not in these $k - 1$ disjoint candidate sets. As the weight of each of the $k - 1$ disjoint candidate sets is c and the total weight of all $3k$ elements is $c \cdot k$, we conclude that $w(a_1) + w(a_2) + w(a_3) = c$, so $\{a_1, a_2, a_3\}$ is a candidate set. Define f by

$$f(C) = \begin{cases} 1 & C \text{ is one of the } k - 1 \text{ disjoint candidate sets or } C = \{a_1, a_2, a_3\} \\ 0 & \text{otherwise.} \end{cases}$$

By construction f is a solution to (1), contradicting the assumption that the instance is nearly-feasible. \square

To prove the main theorem of this section, we investigated the graphs with the properties of Lemma 1 using `genbg` from the `nauty`-package to generate connected cubic graphs. There are, up to isomorphism, 509 connected cubic multigraphs on 12 points. We generated connected cubic graphs on fewer points as well. The generated graphs were then used as input for a custom Haskell program that tested the graph for independent sets. These computations were performed within seconds. This yielded the following lemma:

Lemma 2. *Every connected cubic multigraph on $2k$ points has an independent set of size $k - 1$ for $k \leq 5$. There are, up to isomorphism, 14 connected cubic multigraphs on 12 points with no independent set of size 5.*

This yields our first main theorem:

Theorem 1. *Every nearly-feasible instance has $k \geq 6$.*

Proof. Immediate by Lemma 2 and Lemma 1. \square

To show that the bound in Theorem 1 is tight, a nearly-feasible instance consisting of $3k = 18$ elements should be created. According to Lemma 2 it should have a solution of which the solution graph is one of the 14 indicated multigraphs, one of which was presented in the introduction. We searched for much stronger requirements: for each of the 14 multigraphs we searched for an instance for which the 12 candidate sets corresponding to the vertices in the graph are the only candidate sets, that is, every other triple of elements has a sum unequal to c . Scaling the problem to having sum 1, this means that for every of the 18 edges we introduce a real variable, and for every node of the graph we require that the sum of the three variables corresponding to the three adjacent edges equal 1, while for all other triples of variables we require that the sum is unequal to 1. In this way we have 12 equalities and $\binom{18}{3} - 12 = 804$ inequalities on the 18 variables. Note that this constraint problem is not a linear program due to these inequalities containing \neq rather than \leq . Instead we used a tool for Satisfiability Modulo Theory (SMT), using the theory of linear inequalities. In this format not only conjunctions of linear inequalities can be expressed as in linear programming, but also any combination of negations, conjunctions and disjunctions of linear inequalities. So our combination of equalities and inequalities fits in this format. For solving these problems we used

the SMT-solver `Yices`. Surprisingly, for each of the 14 multigraphs this yielded a solution, again found within seconds. Scaled back to integer values, the found solutions are shown in figure 1.

The multigraphs in this figure are drawn as described by (3). Every vertex is shown as a box with three elements in it. Elements are indicated by their weights, with a ' added to distinguish between elements when necessary.

4 Nearly-feasible instances with low weights

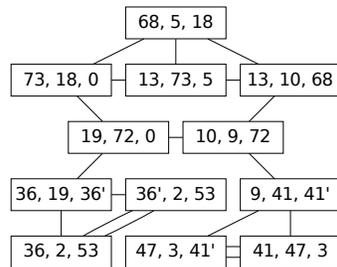
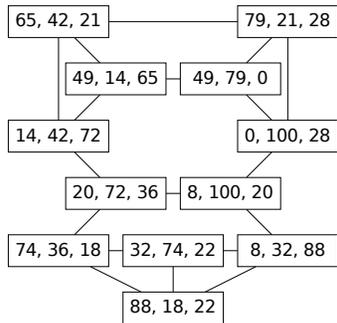
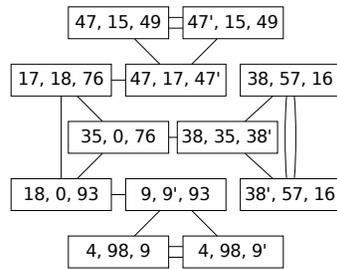
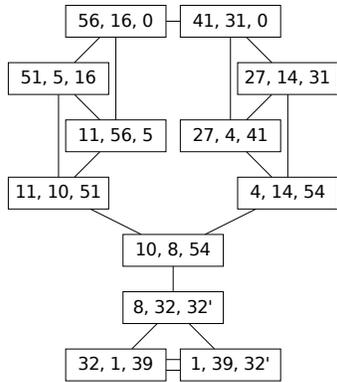
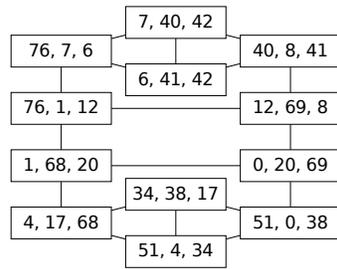
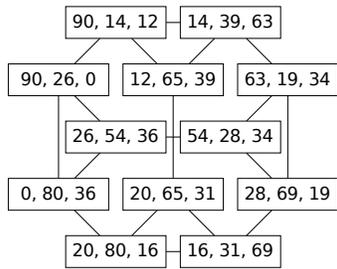
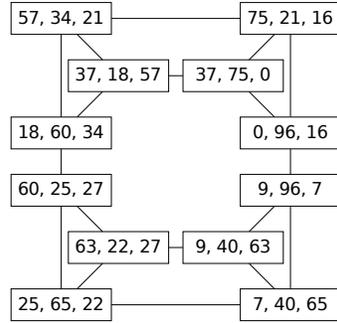
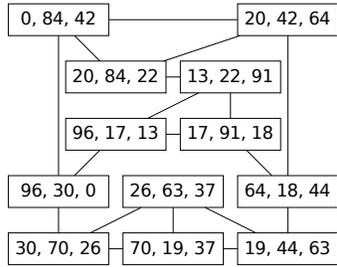
In this section, we prove that every nearly-feasible instance with $k = 6$ contains a number being at least 10, and we prove that this bound is tight. Moreover, we give all nearly-feasible instances with $k = 6$ and the highest weight is 10. The method by which we achieved this result is by enumerating all 3-PART instances with 18 elements and all weights ≤ 10 , and checking whether (1) does not hold and (2) holds. Testing whether millions of instances have solutions to the problems (1) and (2) is slow, however. Therefore, we have used the following observations to reduce the number of tests to be done.

1. Every permutation of an instance is also an instance. Therefore, we only generate instances where the weights form an increasing sequence.
2. Adding a constant to all weights of an instance yields an instance. Therefore, we only generate instances where 0 is the weight of some element.
3. The sum of all weights must be a multiple of 6, since only then c is integer.
4. If m is the highest weight in an instance in which the sum per set is c , replacing weight $w(i)$ with $m - w(i)$ for all elements i creates another instance in which the sum per set is $3m - c$. Therefore, we only generate instances where $2c \leq 3m$.
5. In a nearly-feasible instance of 18 elements, every element occurs in at least 2 candidate sets. Therefore, we only proceed with instances for which this holds.

In this way, 701827 instances remained to be checked, among which we are interested in instances for which (2) has a solution and (1) has not. If (2) has a solution, so does the LP-relaxation of (1). Therefore, we next run simplex to determine whether this LP-relaxation has a solution for all remaining instances. Most of them turn out to have no LP-solution, by which 197110 instances remain. By inspecting the LP-solutions much more instances can be removed. In case only values 0 and 1 occur in the solution, then it is a solution of (1), by which the instance fails. Moreover, if both $\frac{1}{2}$ and 1 occur, and for the rest only 0 occurs, then assuming that (1) has a solution, the elements in candidate sets C with $f(C) = 1$ may be removed yielding a nearly-feasible instance with $k < 6$, contradicting Theorem 1. So all instances with LP solutions having values in $\{0, \frac{1}{2}, 1\}$ in which 1 occurs do not need to be checked. For the remaining instances we checked (1), and only 7 remained for which (1) has no solution. For all of these in the LP solution only the values 0 and $\frac{1}{2}$ occurred, all yielding a solution of (2), hence proving the instance is nearly-feasible. This full computation took eight hours on a 1.6GHz netbook.

The resulting 7 nearly-feasible instances are:

- 0,0,1,1,1,2,2,2,4,4,4,5,5,5,8,8,10,10



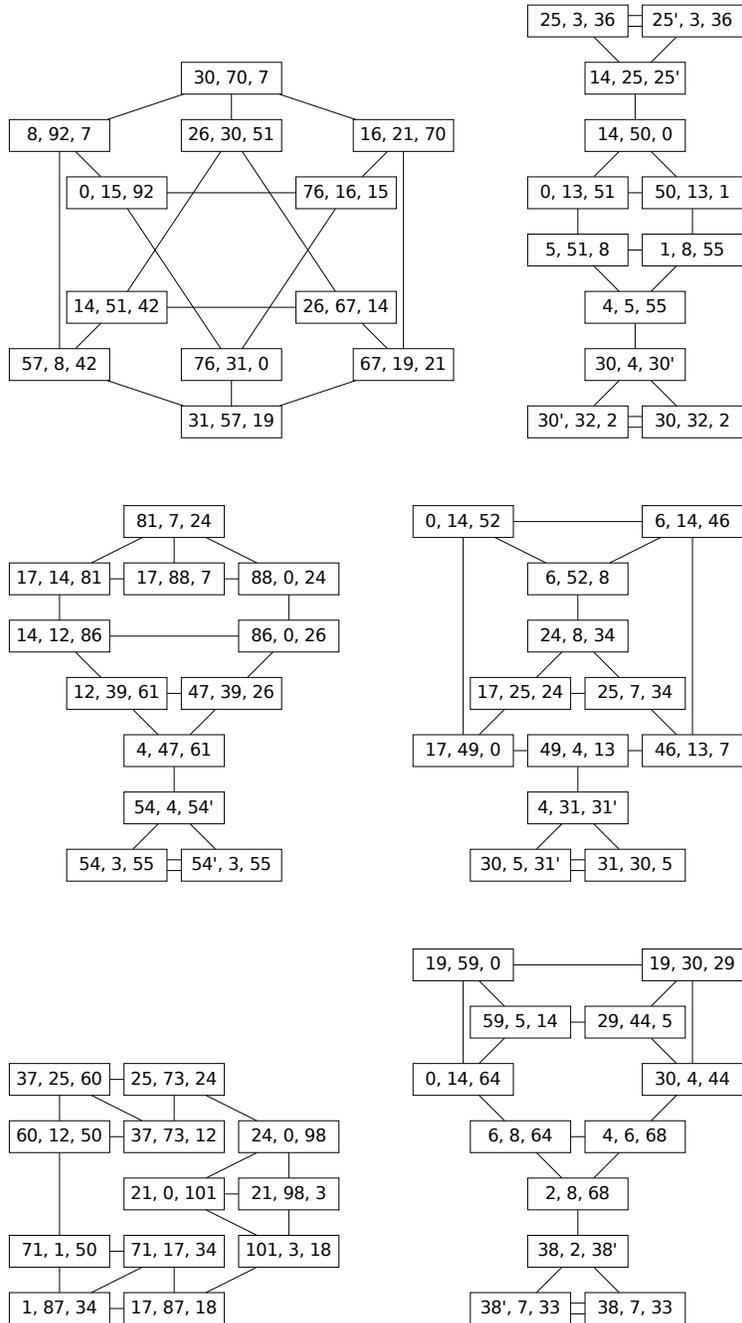


Figure 1: 14 nearly feasible instances

- 0,0,1,1,1,2,3,3,4,4,4,4,4,6,6,9,10,10
- 0,0,1,1,2,2,2,2,4,4,4,5,5,5,8,8,9,10
- 0,1,1,1,2,2,2,4,4,4,4,4,5,5,5,8,10,10
- 0,1,1,1,2,2,3,3,3,4,4,4,4,6,6,6,10,10
- 0,1,1,1,2,3,3,3,4,4,4,4,4,6,6,10,10
- 0,1,1,2,2,2,2,4,4,4,4,4,5,5,5,8,9,10

By using the trick stated in number 4, we also obtain:

- 0,0,2,2,5,5,5,6,6,6,8,8,8,9,9,9,10,10
- 0,0,1,4,4,6,6,6,6,6,7,7,8,9,9,9,10,10
- 0,1,2,2,5,5,5,6,6,6,8,8,8,8,9,9,10,10
- 0,0,2,5,5,5,6,6,6,6,6,8,8,8,9,9,9,10
- 0,0,4,4,4,4,6,6,6,7,7,7,8,8,9,9,9,10
- 0,0,4,4,4,6,6,6,6,6,7,7,7,8,9,9,9,10
- 0,1,2,5,5,5,6,6,6,6,6,8,8,8,8,9,9,10

So these and all its permutations are the only nearly-feasible instances of 18 elements with weights ≤ 10 . In particular, we proved the following.

Theorem 2. *Every nearly-feasible instance with $k = 6$, has an element with weight 10 or higher.*

In the first seven nearly-feasible instances, candidate sets have a sum of $c = 12$. In the latter seven, $c = 18$.

Apart from this we have verified that there are no feasible instances with $k = 6$ featuring candidate sets with a sum of $c = 11$. This was also done using the approach mentioned here. Hence in the sense of c , the first seven instances are minimal as well.

5 Conclusions and related work

The following two theorems about nearly-feasible instances of 3-PART have been proven:

- Every nearly-feasible instance has $k \geq 6$. (Theorem 1)
- Every nearly-feasible instance with $k = 6$, has an element with weight 10 or higher. (Theorem 2)

We gave all 14 instances with $k = 6$ and the highest weight 10, showing both bounds to be tight.

For these investigations, computer support was extensively used. As external tools we used graph generation software, an SMT-solver and a simplex routine. In addition, the observation of theorem 1 was needed to prove theorem 2.

We focused on nearly feasible instances, that is, the integer problem was relaxed to half-integer. However, one can also relax to allow other denominators than only 2. A logical generalization to (2) parametrized by $M \geq 2$ is: find a mapping

$$g : \mathcal{C} \rightarrow \{0, \dots, M\} \text{ such that: } \sum_{C \ni a} g(C) = M \text{ for all } a \in A, \quad (4)$$

The nearly-feasible instances presented until now turn out not to satisfy (4) for M odd. However, instances exist that have a solution to (4) for every $M \geq 2$, but not to (1) (or $M = 1$), for example: 0, 0, 0, 1, 1, 2, 3, 3, 4, 4, 4, 4, 4, 4, 6, 6, 9, 10, 11. This was checked by showing that (4) has a solution for both $M = 2$ and $M = 3$ (and not for $M = 1$); a solution to (4) for any $M > 3$ is obtained by taking a linear combination of the solutions for $M = 2$ and $M = 3$.

Also instances exist that have a solution to (4) for $M = 3$, but not for $M \leq 2$, for example: 0, 0, 1, 1, 2, 2, 4, 4, 5, 5, 5, 5, 5, 7, 7, 11, 13, 13. For such instances, the structure is not well understood, and it is unknown what the least size of such instances could be.

Our notion of nearly-feasible also applies to two other NP-complete problems taken from the book of Garey and Johnson [Garey and Johnson, 1979]. Every instance of 3-PART can be seen as an instance of exact cover by 3-sets. An exact cover by 3-sets instance is given by a collection \mathcal{C} of 3-element subsets of some set A with $|A| = 3k$. The instance is feasible if one can pick k of the 3-element subsets in \mathcal{C} such that every element in A is picked exactly once. The 3-Dimensional matching can be seen as a restricted version of this problem, in which \mathcal{C} must be a subset of $\{\{x, y, z\} | x \in X, y \in Y, z \in Z\}$, where X, Y and Z are disjoint sets that partition A . Such problems can be described as an ILP problem just like we did for 3-PART, and by relaxing it to half-integral solutions we obtain exactly the same notion of nearly-feasible instance as before. In this setting, nearly-feasible sets are much easier to find, and already nearly-feasible sets for $k = 2$ exists. For example, taking 6 elements $1, \dots, 6$ and four candidate sets $\{1, 3, 5\}, \{1, 4, 6\}, \{2, 3, 6\}, \{2, 4, 5\}$, then the full set is not the union of two of these sets, but by choosing all four candidate sets all elements are chosen exactly twice. So this example is a nearly-feasible set for $k = 2$ in this setting of exact cover by 3-sets. The instance is also a 3-Dimensional matching instance, since $X = \{1, 2\}, Y = \{3, 4\}$ and $Z = \{5, 6\}$ partitions A such that all candidate sets contain one element from each of the three sets.

References

- [Dutertre and de Moura, 2006] Dutertre, B. and de Moura, L. (2006). The YICES SMT solver. Tool paper at <http://yices.csl.sri.com/tool-paper.pdf>.
- [Garey, 1975] Garey, M. R. (1975). Complexity Results for Multiprocessor Scheduling under Resource Constraints. *SIAM Journal of Computing* 4, 397–411.
- [Garey and Johnson, 1979] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.
- [Karloff, 1991] Karloff, H. (1991). The Simplex Algorithm. In *Linear Programming Modern Birkhuser Classics* pp. 23–47. Birkhuser Boston.
- [Kern and Qiu, 2011] Kern, W. and Qiu, X. (2011). Improved taxation rate for bin packing games. In *First International ICST Conference on Theory and Practice of Algorithms in (Computer) Systems, TAPAS 2011, Rome, Italy, (Marchetti-Spaccamela, A. and Segal, M., eds), vol. 6595, of Lecture Notes in Computer Science* pp. 175–180, Springer Verlag, Berlin.
- [McKay, 2009] McKay, B. D. (2009). *nauty User’s Guide (Version 2.4)*. Australian National University ACT 0200, Australia.