On the Integral Carathéodory Property

Winfried Bruns

CONTENTS

1. Introduction
2. Deciding UHC
3. Deciding ICP
4. The Search
5. Computational Issues
Acknowledgments
References

In this note we document the existence of a finitely generated rational cone that is not covered by its unimodular Hilbert subcones, but satisfies the integral Carathéodory property. We explain the algorithms that decide these properties and describe our experimental approach that led to the discovery of the examples.

1. INTRODUCTION

Let $C \subset \mathbb{R}^d$ be a finitely generated rational cone, i.e., the set of all linear combinations $a_1x_1 + \cdots + a_nx_n$ of rational vectors $x_1, \ldots, x_d$ with coefficients from $\mathbb{R}_+$. We can of course assume that $x_i \in \mathbb{Z}^d$, $i = 1, \ldots, n$. In this note a *cone* is always supposed to be rational and finitely generated. Moreover, we will assume that C is *pointed*: if $x, -x \in C$, then $x = 0$. Finally, it is tacitly understood that C has full dimension d.

The monoid $M(C) = C \cap \mathbb{Z}^d$ is finitely generated by Gordan's lemma (for example, see [Bruns and Gubeladze 07, Section 2.A]). Since C is pointed, M is a *positive* monoid, so that 0 is the only invertible element in $M(C)$.

It is not hard to see that $M(C)$ has a unique minimal system of generators, which we call its *Hilbert basis*, denoted by $\mathrm{Hilb}(M(C))$ or simply $\mathrm{Hilb}(C)$. It consists of those elements $z \neq 0$ of $M(C)$ that have no decomposition $z = x + y$ in $M(C)$ with $y, z \neq 0$.

We want to discuss combinatorial conditions on $\mathrm{Hilb}(C)$ expressing that C or $M(C)$ is covered by certain "simple" subcones or submonoids, respectively. To this end we define a *u-subcone* of C to be a subcone generated by vectors $x_1, \ldots, x_d \in \mathrm{Hilb}(C)$ that form a basis of the group $\mathbb{Z}^d$. In particular, $x_1, \ldots, x_d$ are linearly independent, and if just this weaker condition is satisfied, then the cone S generated by $x_1, \ldots, x_d$ is called an *f-subcone*. In this case we let $\Gamma(S)$ denote the subgroup of $\mathbb{Z}^d$ generated by $x_1, \ldots, x_d$, and $\Sigma(S)$ the submonoid of $\mathbb{Z}^d$ generated by $x_1, \ldots, x_d$. Note that

2000 AMS Subject Classification: Primary: 52B20

Keywords: Affine monoid, lattice polytope, unimodular covers, integral Carathéodory property

1058-6458/2007 $0.50 per page
Experimental Mathematics **16**:3, page 359

S is a u-subcone if and only if $\Gamma(S) = \mathbb{Z}^d$, or equivalently, $\Sigma(S) = S \cap \mathbb{Z}^d$.

One says that C satisfies (UHC) if C is the union of its u-subcones. The letter U stands for *unimodular*, H reminds us of the condition that the generators of the u-subcones belong to $\mathrm{Hilb}(C)$, and C simply stands for *cover*.

A weaker condition than (UHC) is the *integral Carathéodory property* (ICP). One says that C has (ICP) if every element of $M(C)$ can be written as a linear combination of at most d elements $x_i \in \mathrm{Hilb}(C)$ with *integral* nonnegative coefficients a_i. The terminology is motivated by Carathéodory's theorem: let $y_1, \ldots, y_m$ be a minimal system of generators of the cone C; then every element $y \in C$ is a linear combination $y = a_1 y_{i_1} + \cdots + a_d y_{i_d}$ with nonnegative real coefficients.

Both (UHC) and (ICP) can be formulated more generally for positive affine monoids $M \subset \mathbb{Z}^d$. However, it is easy to see that every monoid M satisfying (UHC) is given in the form $M = \mathbb{R}_+ M \cap \mathbb{Z}^d$. By a theorem of Bruns and Gubeladze [Bruns and Gubeladze 99, Theorem 6.1], the same is true if M satisfies (ICP), provided the group $\mathrm{gp}(M)$ generated by M equals $\mathbb{Z}^d$. In [Bruns and Gubeladze 99] it is also shown that (ICP) is equivalent to the formally stronger condition that M is the union of its submonoids $\Sigma(S)$. (There, this condition is called (FHC).) The equivalence is crucial for our note, and therefore we reproduce the statement and its proof in Theorem 3.2.

While we view (UHC) and (ICP) as structural properties of (normal) affine monoids, these properties were first discussed in the context of integer programming: see [Cook et al. 86, Sebő 90].

It was asked by Sebő [Sebő 90] whether every cone C has (ICP) or (UHC), and he proved that (UHC) holds if $d \leq 3$. He actually proved a stronger statement: C has a triangulation by u-subcones. A counterexample to (UHC) in dimension 6, called C_{10} in the following, was found by Bruns and Gubeladze [Bruns and Gubeladze 99], and then verified to violate (ICP), too, in cooperation with Henk, Martin, and Weismantel [Bruns et al. 99]. Despite the existence of the counterexample, one can fairly say, at least heuristically, that almost all cones satisfy (UHC).

It remained an open problem whether (UHC) is strictly stronger than (ICP). In this note we want to document the existence of cones that satisfy (ICP) but fail (UHC), explain the algorithms that decide (UHC) and (ICP), and describe the experimental approach that led to the discovery of the examples.

All our experiments seem to indicate that C_{10} is the core counterexample to (ICP) and (UHC). In fact, all counterexamples to these properties that have been found contain it. It would be very desirable indeed to clarify the situation in dimensions 4 and 5.

```
UNICOVER(D,n)
 1  for i ← n to N
 2  do
 3     if D ⊂ U_i
 4       then return
 5     if int(D) ∩ int(U_i) ≠ ∅
 6       then (D_1,D_2) ← SPLIT(D,U_i)
 7            UNICOVER(D_1,i)
 8            UNICOVER(D_2,i)
 9            return
10  OUTPUT( D not u-covered )
11  return

MAIN()
 1  Create the list U_1,...,U_N of u-subcones of C
 2  UNICOVER(C,1)
```

FIGURE 1. An algorithm deciding UHC.

2. DECIDING UHC

Let us say that $x \in C$ is *u-covered* if it is contained in a u-subcone. A subset of C is u-covered if each of its elements is u-covered. Using this simple terminology, we can describe an algorithm deciding (UHC); see Figure 1.

In the algorithm we use a function named SPLIT (see Figure 2). It decomposes D along a support hyperplane H of a u-subcone U such that $D \cap H^> \neq \varnothing$ as well as $D \cap H^< \neq \varnothing$. Such a hyperplane does indeed exist if $\mathrm{int}(D) \cap \mathrm{int}(U) \neq \varnothing$, but $D \not\subset U$. The cones produced are $D_1 = D \cap H^+$ and $D_2 = D \cap H^-$. (The open half-spaces determined by H are denoted by $H^>$ and $H^<$, and H^+ and H^- are the corresponding closed half-spaces.)

It is easy to see that the algorithm terminates: there are only finitely many hyperplanes by which we split subcones. Therefore only finitely many subcones can be created.

In order to check the correctness of the algorithm, observe that at each step in the **for** loop in UNICOVER none of the u-subcones $U_1, \ldots, U_{i-1}$ of C intersects the interior of D. In fact, the index i is increased only if $\mathrm{int}(D) \cap \mathrm{int}(U_i) = \varnothing$. (In the

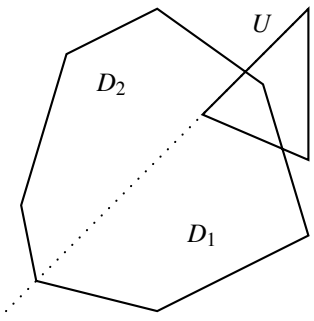

FIGURE 2. The function SPLIT.

recursive call, the index i that has been reached at the parent level starts the **for** loop at the child level.)

Thus, when the loop terminates with $i = N+1$, none of the u-subcones of C intersects the interior of D. So D, and consequently C, indeed contains a vector that is not u-covered.

Conversely, if C contains such a vector x, then on each level of the recursion tree one finds a subcone D containing x, and for such D the condition $D \subset U_i$ can never be satisfied. Therefore there exists an end node of the recursion tree at which the loop is left with $i = N+1$.

The pseudocode in Figure 1 is a somewhat simplistic sketch of the actual implementation, since it is necessary to cope with substantial memory requirements. For example, the list $U_1, \dots, U_N$ is not produced a priori, but extended whenever necessary and always kept as small as possible. Moreover, all allocated memory is recycled carefully within the program.

Instead of starting the covering algorithm with the full cone C, the actual implementation uses the output of a preprocessor that computes several triangulations $\Delta_1, \dots, \Delta_t$ of C. The input to UNICOVER (and also to CARADEC below) is the list of intersections $D_1 \cap \dots \cap D_t$, where D_i is a nonunimodular simplicial cone in Δ_i.

3. DECIDING ICP

Let us first fix some terminology that parallels that for (UHC). An element $x \in C \cap \mathbb{Z}^d$ is *f-covered* if it belongs to one of the monoids $\Sigma(S)$, where S is an f-subcone; and a subset of C is f-covered if each of its elements is f-covered.

The following lemma contains the basic criterion by which we can check that C is f-covered. In the theorem following it, we will then see that this property is equivalent to (ICP).

Lemma 3.1.

(a) *Let $G_1, \dots, G_n$ be subgroups of $\mathbb{Z}^d$, and let N be a residue class of $\mathbb{Z}^d$ modulo $G_1 \cap \dots \cap G_n$. Then $N \subset G_1 \cup \dots \cup G_n$ if and only if $N \cap (G_1 \cup \dots \cup G_n) \neq \varnothing$.*

(b) *Let $S_1, \dots, S_n$ be f-subcones of C, each containing the d-dimensional subcone D of C. If every residue class of $\mathbb{Z}^d$ modulo $\Gamma(S_1) \cap \dots \cap \Gamma(S_n)$ meets $\Gamma(S_1) \cup \dots \cup \Gamma(S_n)$, then D is f-covered.*

(c) *Let D be a d-dimensional subcone of C with the following property: for every f-subcone S either $D \subset S$ or $\operatorname{int}(D) \cap \operatorname{int}(S) = \varnothing$. Furthermore, let G_D be the intersection of the groups $\Gamma(S)$ and $S \supset D$, with H_D their union. Then D is f-covered if and only if every residue class of $\mathbb{Z}^d$ modulo G_D meets H_D.*

Proof: (a) Suppose that $N \cap (G_1 \cup \dots \cup G_n) \neq \varnothing$, and let x be an element in the intersection, $x \in G_i$. The subgroup $G' = G_1 \cap \dots \cap G_n$ is contained in G_i, and so $N = x + G' \subset G_i$. The converse implication is trivial.

(b) Let $x \in D \cap \mathbb{Z}^d$. It follows from (a) that $x \in \Gamma(S_i)$ for some i. But $x \in S_i$, too. Therefore $x \in \Gamma(S) \cap S_i = \Sigma(S_i)$. (At this point we use that the generators of S_i are linearly independent.)

(c) It remains to show only the necessity of the condition. For this we need only observe that every residue class N of $\mathbb{Z}^d$ modulo G_D meets $\operatorname{int}(D)$. By hypothesis on D, an element $x \in N \cap \operatorname{int}(D)$ is f-covered if and only if $x \in \Sigma(S)$ for some f-subcone S containing D. □

We include the next theorem and its proof for the convenience of the reader. It is a simplified version of [Bruns and Gubeladze 99, Theorem 6.1], whose proof contains the crucial ideas for the algorithm deciding (ICP).

Theorem 3.2. *Let $M \subset \mathbb{Z}^d$ be a positive affine monoid such that $\operatorname{gp}(M) = \mathbb{Z}^d$. If M satisfies* (ICP)*, then $M = \mathbb{R}_+M \cap \mathbb{Z}^d$, and every element of M is f-covered.*

Proof: We dissect $C = \mathbb{R}_+M$ along all the support hyperplanes of the cones spanned by linearly independent vectors $x_1, \dots, x_d$ of $\operatorname{Hilb}(M)$ into *elementary* subcones. Set $\bar{M} = \mathbb{R}_+M \cap \mathbb{Z}^d$ and choose $x \in \bar{M}$. Suppose that x has no representation as a linear combination $x = a_1y_1 + \dots + a_dy_d$, with $a_1, \dots, a_d \in \mathbb{Z}_+$ and $y_1, \dots, y_d \in \operatorname{Hilb}(M)$ linearly independent. The element x belongs to one of the elementary subcones D, and as in the proof of the lemma it follows that there exists a finite-index subgroup G of $\mathbb{Z}^d$ such that no element z of $(x+G) \cap \operatorname{int}(D)$ has a representation $a_1y_1 + \dots + a_dy_d$ with $a_1, \dots, a_d \in \mathbb{Z}_+$ and $y_1, \dots, y_d \in \operatorname{Hilb}(M)$ linearly independent.

The crucial point is that $\bar{M} \setminus M$ is contained in the union of finitely many hyperplanes [Bruns and Gubeladze 07, Section 2.B], and the same applies to all elements of M that are linear combinations of at most d linearly dependent elements of $\operatorname{Hilb}(M)$. But $(x+G) \cap \operatorname{int}(D)$ is not contained in the union of finitely many hyperplanes, and so must contain elements of M. This is impossible if M satisfies (ICP). □

For the algorithm deciding (ICP) we have to enrich our data structure by those components that have shown up in the proof of the lemma. Subcones are replaced by triples $(D, G, \mathscr{R})$, where D is a subcone of C, G is a finite-index subgroup of $\mathbb{Z}^d$, and $\mathscr{R}$ is a list of residue classes in $\mathbb{Z}^d/G$. In $\mathscr{R}$ each residue class is represented by a single vector that belongs to it, and

```
CARADEC(D, G, ℛ, n)
 1  for i ← n to N
 2  do
 3     if D ⊂ S_i
 4       then ℛ' ← ∅
 5            for (x ∈ ℛ, y ∈ G/(G∩Γ(S_i)))
 6            do
 7               if x+y ∉ Γ(S_i)
 8                 then ℛ' = ℛ' ∪ {x+y}
 9            G ← G∩Γ(S_i),   ℛ ← ℛ'
10            if ℛ = ∅
11              then return
12     if D ⊄ S_i and int(D)∩int(S_i) ≠ ∅
13       then (D_1, D_2) ← SPLIT(D, S_i)
14            CARADEC(D_1, G, ℛ, i)
15            CARADEC(D_2, G, ℛ, i)
16            return
17  OUTPUT( D not f-covered )
18  return

MAIN()
1   Create the list S_1, ..., S_N of f-subcones of C
2   CARADEC(C, Z^d, {0}, 1)
```

FIGURE 3. An algorithm deciding ICP.

in the algorithm (see Figure 3) the loop

$$\textbf{for } \big(x \in \mathscr{R},\ y \in G/(G\cap\Gamma(S_i))\big)$$

runs over all elements of $\mathscr{R} \times G/(G\cap\Gamma(S_i))$.

Again it is clear that the algorithm terminates after finitely many steps: the number of hyperplanes that we can use to split subcones of C is still finite (though larger than for (UHC)).

The crucial point for `caradec` is that at each step in the loop the f-cones $S_1,\dots,S_N$ satisfy the following conditions:

1. for each $j \le i-1$ either $S_j \supset D$ or $\operatorname{int}(S_j)\cap\operatorname{int}(D) = \varnothing$;
2. G is the intersection of all groups $\Gamma(S_j)$, $j \le i-1$, for which $D \subset S_j$;
3. $\mathscr{R}$ is the list of those residue classes in $\mathbb{Z}^d/G$ that are not contained in the union of the groups $\Gamma(S_j)$, $j \le i-1$, $D \subset S_j$.

We have only to check that these conditions remain satisfied when S_i is tested against D. To this end, let $S_{k_1},\dots,S_{k_m}$ be those among $S_1,\dots,S_{i-1}$ that contain D.

If $\operatorname{int}(S_i)\cap\operatorname{int}(D) = \varnothing$, then $D \not\subset S_i$, and this case is done.

If $\operatorname{int}(S_i)\cap\operatorname{int}(D) \ne \varnothing$, but $D \not\subset S_i$, then i is not increased (!) and all three conditions are inherited by both D_1 and D_2: among the S_j, $j \le i-1$, exactly $S_{k_1},\dots,S_{k_m}$ contain D_1 or D_2, simply because $S_j \supset D_1$ or $S_j \supset D_1$ implies $\operatorname{int}(S_j)\cap\operatorname{int}(D) \ne \varnothing$, and so $S_j \supset D$.

But if $S_i \supset D$, the bookkeeping is also correct. Evidently G is replaced by the correct group $G\cap\Gamma(S_i)$. Next observe that all residue classes of $G\cap\Gamma(S_i)$ that are contained in residue classes modulo G not appearing in $\mathscr{R}$ remain in $\Gamma(S_{k_1})\cup\cdots\cup\Gamma(S_{k_m})$. On the other hand, those that refine elements of $\mathscr{R}$ must belong to $\Gamma(S_i)$ to be in $\Gamma(S_{k_1})\cup\cdots\cup\Gamma(S_{k_m})\cup\Gamma(S_i)$. The correctness of the algorithm follows now immediately from Lemma 3.1.

The greatest hurdle is the lists $\mathscr{R}$ of residue classes, which usually become extremely long already in dimension 6. Moreover, along each branch of the recursion tree, several of them must be kept in memory. (This problem cannot be eliminated by a nonrecursive implementation.)

The growth of the list $\mathscr{R}$ can be estimated. Set

$$e = \#\big(G/(G\cap\Gamma(S_i))\big) \quad \text{and} \quad e' = \#(\mathbb{Z}^d/\Gamma(S_i)).$$

Each element $x \in \mathscr{R}$ is involved in e vectors $x+y$. At most one of them lies in $\Gamma(S_i)$, since the vectors y belong to distinct residue classes modulo $\Gamma(S_i)$. Therefore

$$\#(\mathscr{R}') \ge (e-1)\#(\mathscr{R}).$$

If the elements of $\mathscr{R}$ are randomly distributed over the residue classes of $\mathbb{Z}^d$ modulo $\Gamma(S_i)$, then the expected share of vectors $x+y \in \Gamma(S_i)$ drops to $1/e'$.

Instead of keeping the lists of residue classes in memory, one could alternatively try to follow the recursion tree along the whole list $S_1,\dots,S_N$, compute only G along each branch, and test the residue classes one by one only at the end nodes. However, this approach seems infeasible, since it derives no advantage from the case $e = 1$, which fortunately happens frequently and often stops the recursion before the end of $S_1,\dots,S_N$ is reached.

The list $S_1,\dots,S_N$ is actually scanned in growing order of the determinants of the S_i. This has turned out to be very effective, at least for those cones that satisfy (ICP). In fact, all cones in Table 3 with (ICP) are f-covered by simplicial subcones of determinant ≤ 2.

In addition to `CARADEC` we use a Monte Carlo approach for disproving (ICP). It reads the output of `UNICOVER`, computes a large number of vectors in the non-u-covered subcones of C, and tests whether they are f-covered.

Remark 3.3. `CARADEC` provides us with a precise measure for the failure of (ICP), namely the ratios $\#(\mathscr{R})/\#(G)$ at the end nodes of the recursion tree. For the cone C_{10} (Table 1), there is precisely one end node with $\mathscr{R} \ne \varnothing$, and the ratio is $32/15552 = 1/486$. The number of non-f-covered vectors in the Monte Carlo test confirms the ratio rather precisely.

For the cone C''_{15} (Table 3) there is again a single group with $\mathscr{R} \neq \varnothing$ and ratio

$$2,4468,480/2,286,144,000,000 \approx 1.07/10^6.$$

So the Monte Carlo test cannot be expected to be conclusive with fewer than 10^6 test vectors.

4. THE SEARCH

Let us recapitulate an important notion from [Bruns and Gubeladze 99]. An element x of $\mathrm{Hilb}(C)$ is called *destructive* if $H' = \mathrm{Hilb}(C) \setminus \{x\}$ is not the Hilbert basis of $\mathbb{R}_+H'$. We say that C is *tight* if every element of $\mathrm{Hilb}(C)$ is destructive. The crucial role of tight cones for (UHC) and (ICP) is illuminated by the following lemma [Bruns and Gubeladze 99, Corollary 2.3].

Lemma 4.1. *Let C be a cone that is a counterexample to* (UHC) *or* (ICP). *Suppose C is minimal first with respect to dimension and second with respect to $\#\mathrm{Hilb}(C)$. Then C is tight.*

Remark 4.2. Updating the information in [Bruns and Gubeladze 99], we mention that tight cones exist in all dimensions $d \geq 3$. The first 3-dimensional tight cone was found by P. Dueck. The smallest such cone found by the author has a Hilbert basis of 19 elements. The elements of the Hilbert basis in the extreme rays form a regular hexagon (with respect to the action of $\mathrm{GL}_3(\mathbb{Z})$), so that the cone has the dihedral group D_6 as its automorphism group. The regularity is an indication that it may be the smallest possible tight cone. (Here and in the following the automorphism group of a cone C is always understood to be the automorphism group of the monoid $C \cap \mathbb{Z}^d$.)

Our search for counterexamples has been based on the crucial Lemma 4.1. We produce a set of random vectors, consider them as the generating set of a cone C, and then use a program named SHRINK to remove nondestructive elements of $\mathrm{Hilb}(C)$ until a tight cone is reached (SHRINK is based on the same algorithm as NORMALIZ; see [Bruns et al. 98, Bruns and Koch 01]). Almost always, C shrinks to the 0-cone, but sometimes a nontrivial tight cone emerges. Then UNICOVER, and possibly CARADEC, are invoked.

When we started the search in spring 1998, we used cones over randomly generated lattice parallelepipeds. In May 1998 the search stopped with the counterexample C_{10}. Its Hilbert basis is shown in Table 1. The cone C_{10} has 27 support hyperplanes.

$z_1 = (0,1,0,0,0,0)$	$z_6 = (1,0,2,1,1,2)$
$z_2 = (0,0,1,0,0,0)$	$z_7 = (1,2,0,2,1,1)$
$z_3 = (0,0,0,1,0,0)$	$z_8 = (1,1,2,0,2,1)$
$z_4 = (0,0,0,0,1,0)$	$z_9 = (1,1,1,2,0,2)$
$z_5 = (0,0,0,0,0,1)$	$z_{10} = (1,2,1,1,2,0)$

TABLE 1. $\mathrm{Hilb}(C_{10})$.

The reader should note that for the questions considered in this note we can always replace a given cone C by $\phi(C)$, where ϕ is an arbitrary transformation in $\mathrm{GL}_d(\mathbb{Z})$. In this sense, C stands for a class of cones that are isomorphic under an integral isomorphism of $\mathbb{R}^d$. We express this fact by speaking of different embeddings of a cone C.

While UNICOVER showed that C_{10} fails (UHC), it was then verified in cooperation with Henk, Martin, and Weismantel that C_{10} also fails (ICP) (CARADEC was not written before September 2006). See [Bruns and Gubeladze 99] and [Bruns et al. 99] for more information on C_{10}.

The automorphism group of C_{10} is remarkably large: it is the Frobenius group F_{20} of order 20, which acts transitively on $z_1,\dots,z_{10}$. (The group F_{20} is the semidirect product of $\mathbb{Z}_5$ with its automorphism group $\mathbb{Z}_5^* \cong \mathbb{Z}_4$.) From the embedding above one can see that at least the dihedral group $D_5 \subset F_{20}$ acts on C_{10}. All the remaining 10 automorphisms have order 4 and swap $\{z_1,\dots,z_5\}$ with $\{z_6,\dots,z_{10}\}$. Moreover, $z_1,\dots,z_{10}$ all lie in the hyperplane given by $-5\zeta_1 + \zeta_2 + \cdots + \zeta_6 = 1$. The convex hulls of $\{z_1,\dots,z_5\}$ and $\{z_6\dots,z_{10}\}$ are both simplices of dimension 4.

Remark 4.3. It was communicated to us by F. Santos that the lattice polytope spanned by $\mathrm{Hilb}(C_{10})$ is a projection of the Ohsugi–Hibi polytope [Ohsugi and Hibi 99]. The projection leads to the following description of $C_{10} \cap \mathbb{Z}^6$. Consider the complete graph K_5 and decompose it into two cycles of length 5 as shown in Figure 4.

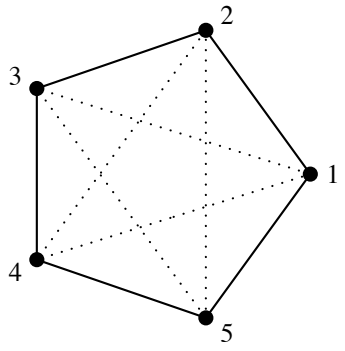

FIGURE 4. Cycle decomposition of K_5.

Now choose the incidence vectors $(1,1,0,0,0)$, etc., of the edges in the first cycle and prefix them with 0. Then prefix the incidence vectors $(1,0,1,0,0)$, etc., of the second cycle with 1. The resulting 10 vectors in $\mathbb{Z}^6$ generate a monoid M isomorphic to $C_{10} \cap \mathbb{Z}^6$. While this description is even more aesthetic than the one in Table 1, it has the disadvantage that $\mathrm{gp}(M)$ is of index 2 in $\mathbb{Z}^6$.

In the summer of 1998 a second counterexample C_{12} to (UHC) and (ICP) emerged. It has a Hilbert basis of 12 elements. We continued the search for two more years. The frustrating outcome was that C_{10} appeared over and over again, but no new counterexample showed up (and even C_{12} did not return until November 22, 2006).

The project was taken up again at the end of 2004 when our department installed a dual-processor Opteron system with very fast integer arithmetic. Nevertheless, the outcome of the search remained as disappointing as it had been before.

Finally, in August 2006 we did what should have been done long before, namely compare C_{12} with C_{10}: it turned out that $\mathrm{Hilb}(C_{12})$ (in an embedding that had to be found!) extends $\mathrm{Hilb}(C_{10})$ by two vectors. Relative to C_{10}, this finding explained why C_{12} fails (UHC) and (ICP), too: the extra u-subcones and f-subcones are not sufficient to cover all integral vectors in C_{10}. (It also shows that one cannot speed up shrinking by removing two vectors at a time.)

However, this not very surprising a posteriori insight made it suddenly clear that there might be many interesting objects in the vicinity of C_{10}. Especially when we approach C_{10} along a shrink path, why should the stronger property (UHC) not be lost before (ICP)? After a modification of SHRINK we also applied UNICOVER to the, for example, six last nontight approximations to C_{10}, and within hours many new non-(UHC) cones emerged. Several of them defeated all Monte Carlo attacks on (ICP). It became clear that CARADEC had to be implemented, and it indeed recognized many non-(UHC), but (ICP), cones.

Ironically, within a few weeks after we had given up our narrow-minded insistence on checking only tight cones, two new non-(UHC) such cones surfaced, both of them satisfying (ICP). They appear as C'_{12} and C_{15} in Table 2.

C_{12} :	$z'_{11} = (2,2,1,4,1,3)$	C_{15} :	$w_1 = (2,1,0,5,1,5)$
	$z'_{12} = (2,3,1,4,1,2)$		$w_2 = (1,0,-1,4,0,4)$
			$w_3 = (0,0,-1,1,0,1)$
C'_{12} :	$z''_{11} = (0,-1,2,-1,-1,2)$		$w_4 = (2,1,2,3,2,4)$
	$z''_{12} = (1,0,3,0,0,3)$		$w_5 = (1,1,0,3,1,2)$

TABLE 2. Additional vectors in $\mathrm{Hilb}(C_{12})$, $\mathrm{Hilb}(C'_{12})$, $\mathrm{Hilb}(C_{15})$.

	#Hilb	#Supp	ICP	Aut	flat
C_{10}	10	27	no	F_{20}	yes
C_{12}	12	39	no	$\mathbb{Z}_2 \times \mathbb{Z}_2$	yes
C'_{12}	12	40	yes	F_{20}	yes
C_{14}	14	34	yes	$\{\mathrm{id}\}$	yes
C'_{14}	14	39	no	$\{\mathrm{id}\}$	yes
C''_{14}	14	42	yes	$\mathbb{Z}_2$	no
C'''_{14}	14	44	yes	$\{\mathrm{id}\}$	yes
C_{15}	15	36	yes	$\{\mathrm{id}\}$	no
$C_{15'}$	15	36	yes	$\{\mathrm{id}\}$	yes
C''_{15}	15	44	no	$\{\mathrm{id}\}$	yes
C_{16}	16	49	no	$\mathbb{Z}_2$	yes
C'_{16}	16	36	no	$\mathbb{Z}_2$	yes
C_{19}	19	46	yes	$\{\mathrm{id}\}$	yes

TABLE 3. Tight non-(UHC) cones.

Since all these cones contain C_{10}, we list only the extra vectors that complement the Hilbert basis of C_{10} (using the embedding given in Table 1).

The most interesting example after C_{10} undoubtedly is C'_{12}, and not only because it satisfies (ICP). Its automorphism group—certainly invisible from the embedding given—is again the Frobenius group F_{20}. It is clear that F_{20} cannot act transitively on $\mathrm{Hilb}(C'_{12})$, which rather decomposes into an orbit of 10 elements and one of 2. However, this is by no means an extension of the action of F_{20} on C_{10}, since the orbit of two elements is $\{z_1, z_5\}$! Only a subgroup isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$ restricts to C_{10}, showing that there are five conjugate embeddings of C_{10} into C'_{12}, and each of them contains z_1, z_5.

The convex hulls of $\{z_1, \ldots, z_5, z''_{11}\}$ and $\{z_6 \ldots, z_{10}, z''_{12}\}$ are both bipyramids over a tetrahedron. The bipyramids are situated in the parallel hyperplanes with the equations $\zeta_1 = 0$ and $\zeta_1 = 1$. Both C_{12} and C'_{12} have their Hilbert bases in the hyperplane spanned by $\mathrm{Hilb}(C_{10})$, but this is not true for C_{15}.

Table 3 lists all 13 tight non-(UHC) cones of dimension 6 that had been found by February 2, 2007, including those mentioned already. In the last column we indicate whether the Hilbert basis is contained in a hyperplane.

The smallest non-(UHC) but (ICP) cone we have found has a Hilbert basis of 11 elements, and the largest has a Hilbert basis of 24 elements (most likely (ICP)). Like all the others, they are extensions of C_{10}, which seems to be the core obstruction to (ICP) and (UHC).

This finding is further supported by computations in dimension 7. All tight non-(UHC) (and partly non-(ICP)) cones of dimension 7 we have found inherit the failure of (UHC) or even (ICP) from C_{10}. We forgo including data similar to those in Table 3, since they do not add essentially new information.

While we have the implications (UHC) $\Longrightarrow$ (ICP) $\Longrightarrow$ $M = \mathbb{R}_+M \cap \mathbb{Z}^d$ (under the condition $\mathrm{gp}(M) = \mathbb{Z}^d$), it is now clear that the converse implications do not hold. However, it remains an open problem whether all cones in dimensions 4 and 5 have (ICP) or even (UHC).

5. COMPUTATIONAL ISSUES

All programs were written in C. The tight cones in Section 4 were found by the Opteron (O) system mentioned above. It runs Linux, and the executables were produced by the gcc compiler.

In the following we will also mention computations on two other systems, the author's Intel Core2 6600 (C2) with Windows XP and the DJGPP port of gcc, and the University of Osnabrück's Itanium (I) system with Linux and the Intel compiler icc. The machines (O) and (C2) are close to each other in speed; (I) is somewhat slower (for integer arithmetic), but has very large memory (32 GB).

Some of the equipment used in the 1998 computations is still accessible. This allowed us to measure the gain in speed by improved hardware: the factor is at least 40. Moreover, a better implementation of SHRINK yields an acceleration by a factor of at least 3. In other words, four months of the 1998 search take now a single day.

The number of cones shrunk by SHRINK per second depends very much on the parameters used for their creation. The performance for 6-dimensional cones generated by random 0-1 vectors, whose number varies between 6 and 26, is about 1000 per second on (O) or (C2). Cones over 5-dimensional parallelotopes of Euclidean volume ≤ 30 are shrunk at a rate of 0.6 per second. The output of tight cones is nevertheless comparable.

The cones in Section 4 are a light snack for UNICOVER. For example, the running time for C''_{14} on (C2) is 1.7 seconds. About 2.5 million vectors are created, but the list of vectors in memory simultaneously is bounded by $15,000$; C_{19} needs about 50 seconds on (C2).

While all the other programs use 32-bit arithmetic, CARADEC is set to 64-bit. It has no problem with all the cones mentioned, as long as they have (ICP), simply because in all cases they are f-covered by cones of determinant ≤ 2. The running time for C''_{14} on (C2) is 6.9 seconds.

Despite its sometimes enormous appetite for memory, CARADEC has also been successful for all the cones of Table 3 that lack (ICP), with the exception of C_{16} and C'_{16}.

It failed for these cones, though it was allowed 200 million vectors in memory.

The (ICP) property of C_{16} and C'_{16} was falsified by the Monte Carlo method with 1 million test vectors for each of the non-(UHC) subcones produced by UNICOVER (17.4 seconds on (C2) for C_{16}).

The longest successful run of CARADEC with a negative result was C''_{15}: 1.990 seconds on (I), 2.1 billion vectors, 110 million simultaneously. The Monte Carlo method does the job with 1 million vectors for the single non-(UHC) subcone in 2.7 seconds on (C2). See Section 3 for further data on C''_{15}.

ACKNOWLEDGMENTS

The author is very grateful to Joseph Gubeladze for inspiring discussions and to the Mathematisches Forschungsinstitut Oberwolfach, where the first steps of this project were taken during a joint visit within the MFO's RiP program.

REFERENCES

[Bruns and Gubeladze 99] W. Bruns and J. Gubeladze. "Normality and Covering Properties of Affine Semigroups." *J. Reine Angew. Math.* 510 (1999), 151–178.

[Bruns and Gubeladze 07] W. Bruns and J. Gubeladze. "Polytopes, Rings and K-Theory." In preparation. Preliminary version available online (http://www.math.uos.de/staff/phpages/brunsw/kripo.pdf).

[Bruns and Koch 01] W. Bruns and R. Koch. "Computing the Integral Closure of an Affine Semigroup." *Univ. Iagel. Acta Math.* 39 (2001), 59–70.

[Bruns et al. 99] W. Bruns, J. Gubeladze, M. Henk, A. Martin, and R. Weismantel. "A Counterexample to an Integer Analogue of Carathéodory's Theorem." *J. Reine Angew. Math.* 510 (1999), 179–185.

[Bruns et al. 98] W. Bruns, R. Koch, et al. "NORMALIZ: Computing Normalizations of Affine Semigroups." Available via anonymous ftp (ftp://ftp.math.uos.de/pub/osm/kommalg/software/), 1998.

[Cook et al. 86] W. Cook, J. Fonlupt, and A. Schrijver. "An Integer Analogue of Carathéodory's Theorem." *J. Comb. Theory, Ser. B* 40 (1986), 63–70.

[Ohsugi and Hibi 99] H. Ohsugi and T. Hibi. "A Normal $(0,1)$-Polytope None of Whose Regular Triangulations Is Unimodular." *Discrete Comput. Geom.* 21 (1999), 201–204.

[Sebő 90] A. Sebő. "Hilbert Bases, Carathéodory's Theorem, and Combinatorial Optimization." In *Integer Programming and Combinatorial Optimization*, edited by R. Kannan and W. Pulleyblank, pp. 431–456. Waterloo: University of Waterloo Press, 1990.

Winfried Bruns, Universität Osnabrück, FB Mathematik/Informatik, 49069 Osnabrück, Germany (wbruns@uos.de)

Received October 24, 2006; accepted December 13, 2006.