
Colourful Linear Programming and Its Relatives

Author(s): Imre Bárány and Shmuel Onn

Source: *Mathematics of Operations Research*, Vol. 22, No. 3 (Aug., 1997), pp. 550-567

Published by: [INFORMS](#)

Stable URL: <http://www.jstor.org/stable/3690393>

Accessed: 19/10/2013 05:24

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at
<http://www.jstor.org/page/info/about/policies/terms.jsp>

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.



INFORMS is collaborating with JSTOR to digitize, preserve and extend access to *Mathematics of Operations Research*.

<http://www.jstor.org>

COLOURFUL LINEAR PROGRAMMING AND ITS RELATIVES

IMRE BÁRÁNY AND SHMUEL ONN

We consider the following *Colourful* generalization of Linear Programming: given sets of points $S_1, \dots, S_k \subset \mathbb{R}^d$, referred to as *colours*, and a point $b \in \mathbb{R}^d$, decide whether there is a *colourful* $T = \{s_1, \dots, s_k\}$ such that $b \in \text{conv}(T)$, and if there is one, find it. Linear Programming is obtained by taking $k = d + 1$ and $S_1 = \dots = S_{d+1}$. If $k = d + 1$ and $b \in \bigcap_{i=1}^{d+1} \text{conv}(S_i)$ then a solution always exists: we describe an efficient iterative approximation algorithm for this problem, that finds a *colourful* T whose convex hull contains a point ϵ -close to b , and analyze its real arithmetic and Turing time complexities. In contrast, we show that *Colourful Linear Programming* is strongly \mathcal{NP} -complete. We consider a class of linear algebraic relatives of *Colourful Linear Programming*, and give a computational complexity classification of the related decision and counting problems that arise. We also introduce and discuss the complexity of a hierarchy of (w_1, w_2) -Matroid-Basis-Nonbasis problems, and give an application of *Colourful Linear Programming* to the algorithmic problems of Tverberg's theorem in combinatorial geometry.

1. Introduction. The so-called *Carathéodory's theorem* says the following: if a set $S \subset \mathbb{R}^d$ of points in d -space contains a point b in its convex hull, then $b \in \text{conv}(T)$ for some subset $T \subseteq S$ containing at most $d + 1$ points. This statement allows to pose the problem of linear programming in the following way.

Linear programming problem. Given a finite set $S \subset \mathbb{Q}^d$ and a point $b \in \mathbb{Q}^d$, decide whether there is a subset $T \subseteq S$ of size at most $d + 1$ such that $b \in \text{conv}(T)$, and if there is one, find it.

Carathéodory's theorem admits a *colourful* generalization, due to Bárány (1982). To state it, we use the following terminology: given a family of sets $S_1, \dots, S_k \subset \mathbb{R}^d$, referred to as *colours*, a *colourful set* is a set $T = \{s_1, \dots, s_k\}$ where $s_i \in S_i$ for all i .

THEOREM 1.1. COLOURFUL CARATHÉODORY'S THEOREM. *If each of $d + 1$ given colours $S_0, \dots, S_d \subset \mathbb{R}^d$ in d -space contains the point b in its convex hull, then $b \in \text{conv}(T)$ for some *colourful set* $T = \{s_0, \dots, s_d\}$.*

A proof of Theorem 1.1 will be given in the next section. The specialization of this theorem to Carathéodory's is obtained when $S = S_0 = \dots = S_d$. The following algorithmic problem suggested by Theorem 1.1 is a natural generalization of the Linear Programming Problem.

Colourful linear programming problem. Given colours $S_1, \dots, S_k \subset \mathbb{Q}^d$ and a point $b \in \mathbb{Q}^d$, decide whether there is a *colourful* $T = \{s_1, \dots, s_k\}$ such that $b \in \text{conv}(T)$, and if there is one, find it.

The specialization of this problem to linear programming is obtained by taking $S = S_1 = \dots = S_{d+1}$.

In this article we study the computational complexity of this problem and its relatives in linear algebra, matroid theory, and combinatorial geometry. Our motivation is twofold.

Received September 5, 1995; revised June 4, 1996, September 6, 1996, and October 14, 1996.

AMS 1991 subject classification. Primary: 90C05; secondary: 90C10, 90C60, 68Q25.

OR/MS subject classification. Primary: Programming/Linear; Secondary: Programming/Integer, Analysis of algorithms/Computational complexity.

Key words. Linear programming, integer programming, computational complexity, combinatorial geometry, matroid, enumeration, permanent, polytope, Carathéodory's theorem, Tverberg's theorem, transversal, partition, #P, NP.

First, the determination of the complexity of the problem (particularly in the case covered by Theorem 1.1) is a very important theoretical issue: our Theorems 4.4 and 6.1 described below provide a step in that direction by giving contrasting positive and negative results on the complexity of the problem. Second, colourful linear programming has various applications. We discuss one such application to combinatorial geometry in §7 and briefly mention another in §8. A broader account of other applications and consequences of colourful linear programming will appear elsewhere. Related colourful conjectures and determinantal identities can be found in a recent article by Onn (1997).

An overview of the paper is as follows. In §§2 and 3 we provide a rather efficient approximation algorithm for colourful linear programming and analyze its real arithmetic complexity. In §4 we analyze the Turing time complexity for rational data. We also show how to use the approximation algorithm to solve colourful linear programming exactly. In §5 we give a computational complexity classification of a hierarchy of related decision and counting problems in linear algebra. We also introduce and discuss the complexity of a hierarchy of (w_1, w_2) -Matroid-Basis-Nonbasis problems. In §6 we show that colourful linear programming is strongly \mathcal{NP} -complete. In §7 we give an application of colourful linear programming to the algorithmic problem of Tverberg's theorem in combinatorial geometry. We conclude with a brief discussion.

Here is a more detailed description of the contents of the article and the main results. In §§2–4 we study an approximation algorithm for colourful linear programming. We concentrate on the case $k = d + 1$ and $b = 0 \in \bigcap_{i=1}^{d+1} \text{conv}(S_i)$ where a solution is guaranteed to exist, but needs to be found. Given an $\epsilon > 0$, the algorithm finds a colourful T which is ϵ -close to 0, that is, whose convex hull contains a point which is ϵ -close to 0. Interestingly, our algorithm specializes, in the case $S_1 = \dots = S_{d+1}$, to an algorithm of von Neumann for linear programming. Assuming that each S_i contains at most n points, that the points are normalized, and that a ball $B(0, \rho)$ is contained in $\bigcap_{i=1}^{d+1} \text{conv}(S_i)$, we obtain the following results on the real arithmetic complexity and, for rational data, on the running time on a Turing machine, respectively (see §§3–4 for the precise statements):

- **Theorem 3.1.** The number of real arithmetic operations taken by the algorithm to find a colourful set containing a point ϵ -close to 0 is $O(((nd + d^4)/\rho^2) \log(1/\epsilon))$.

- **Theorem 4.3.** For rational data, the algorithm finds a colourful set containing a point ϵ -close to 0 in time which is polynomial in the bit size L of the input points and in $\log(1/\epsilon)$ and $1/\rho$.

We also show that by suitably choosing ϵ it is possible to convert the approximation algorithm to an algorithm for an exact solution of colourful linear programming on rational data. We are grateful to a referee for his question which prompted us to derive the following result.

- **Theorem 4.4.** For normalized rational data there is an algorithm that finds a colourful set T containing 0 in its convex hull in time polynomial in the bit size L of the input points and in $1/\rho$.

In §§5 and 6 we give a computational complexity classification of a hierarchy of related problems. Colourful linear programming is equivalent to deciding if S_1, \dots, S_k admit a colourful T which is *positively dependent*. Replacing “positively” by “linearly” and “dependent” by “independent,” we get four decision and related counting problems. We have the following results, the hardness ones holding even if the number k of sets equals the dimension d .

- **Theorem 5.4.** All four counting problems are $\#P$ -complete.

- **Theorems 5.1 and 5.3.** The complexity of deciding the existence of a colourful set of each of the types above is given by the following table:

	Linearly	Positively
Dependent	<i>NP</i> -complete	<i>NP</i> -complete
Independent	Polynomial Time	

- **Theorem 6.1.** Colourful linear programming is strongly \mathcal{NP} -complete.

We also discuss the complexity of the following hierarchy of Basis-Nonbasis matroid problems, one for each pair (w_1, w_2) of positive numbers: it is the problem of deciding, given $w_1 + w_2$ matroids, whether there is a subset which is a basis in the first w_1 matroids but not in the others. Counting problems related to this hierarchy will be studied elsewhere, in an extension of Kleinschmidt and Onn (1996).

In §7 we turn to an application to a problem from combinatorial geometry motivated by Tverberg's theorem: given a set of points $S \subset \mathbb{Q}^d$ and a positive integer k , a colouring (partition) $S = \bigsqcup_{i=1}^k S_i$ such that $\bigcap_{i=1}^k \text{conv}(S_i) \neq \emptyset$ is sought. Based on a recent result by Sarkaria (1992), the following statement is derived (see §7 for the precise statement).

- **Theorem 7.2.** The decision, counting, and search Tverberg colouring problems are polynomial time reducible to corresponding suitable problems of colourful linear programming.

We conclude with a brief discussion of another application to computational geometry.

Our Theorem 4.4, which guarantees an efficient algorithm for colourful linear programming when $k = d + 1$ and the convex hull of each colour contains a ball of positive volume, stands in contrast with our Theorem 6.1 which asserts that colourful linear programming is strongly \mathcal{NP} -complete. This makes us suggest the following question, which in the following form remains open, as an outstanding problem on the border line between tractable and intractable computational problems.

QUESTION 1.2. Given colours $S_0, \dots, S_d \subset \mathbb{Q}^d$ such that $0 \in \bigcap_{i=0}^d \text{conv}(S_i)$, is it possible to find a colourful $T = \{s_0, \dots, s_d\}$ such that $0 \in \text{conv}(T)$, which is guaranteed to exist, in polynomial time?

2. Approximating a colourful point. As we shall see in following sections, colourful linear programming is computationally hard. Therefore we first study the problem of *approximating* the colourful point. This will also lead to an exact solution in §4. So, we consider the problem of searching for a colourful set T such that $\text{conv}(T)$ contains a point which is ϵ -close to 0. In this section and the next two we describe and analyze a certain iterative pivoting algorithm for such an approximation. Interestingly, when $S_0 = \dots = S_d$, our algorithms essentially specializes to an algorithm of von Neumann for linear programming (see Dantzig 1992). So our algorithm can be regarded as a ‘‘colourful’’ refinement of von Neumann's algorithm.

If any of the input points is the origin then any colourful set containing it is a solution. So, without loss of generality we assume that no input point is 0 and that the input is *normalized*, that is, the norm of each point $s \in \bigcup_{i=0}^d S_i$ satisfies $1 \leq |s| \leq 2$. The reason we do not simply assume $|s| = 1$ is that we want to deal with rational data as well. Note that normalizing an arbitrary input is easy to do: with real arithmetic computation simply divide s by its norm $|s|$; with a Turing machine on rational data, first scale $s \in \mathbb{Q}^d$ to have integer components, and then divide it by the largest integer n satisfying $n^2 \leq |s|^2$. We shall extend the analysis to the situation where we know that a Euclidean ball $B(0, \rho)$ of radius ρ about 0 is contained in the convex hull of each monochromatic set S_i . For

positive ρ the convergence turns out to be much faster. So, we address the following problem.

Colourful point approximation problem. Given $\epsilon > 0$, $\rho \geq 0$, and normalized sets S_0, \dots, S_d of points in \mathbb{R}^d , each satisfying $B(0, \rho) \subset \text{conv}(S_i)$, find a colourful set whose convex hull contains a point x satisfying $|x| \leq \epsilon$.

In this section we describe an iterative algorithm for this problem and bound the number of iterations required. Each iteration involves a minimum norm computation over a polytope, which is a fairly heavy task. In the next section we describe a variant of the algorithm which avoids minimum norm computations, discuss the details of its efficient implementation, and analyze its real arithmetic complexity. In §4 we analyze the computational complexity on a Turing machine for rational data.

The following algorithm maintains in each intermediate iteration a colourful set $T_k = \{t_0, \dots, t_d\}$ and approximating point $x_k \in \text{conv}(T_k)$.

Algorithm 1.

- *Initialization.* Put $k = 1$. Pick an arbitrary colourful set $T_1 = \{t_0, \dots, t_d\}$. Let x_1 be the point of minimum norm in $\text{conv}(T_1)$.
- *Iteration.* If $|x_k| \leq \epsilon$ then stop and output T_k and x_k . Otherwise update the colourful set as follows: choose a colour i such that $x_k \in \text{conv}(T_k \setminus \{t_i\})$; choose a point $t \in S_i$ minimizing the inner product $\langle x_k, t \rangle$; update $t_i := t$ and let $T_{k+1} = \{t_0, \dots, t_d\}$ be the resulting new colourful set. Let x_{k+1} be the point of minimum norm in $\text{conv}(T_{k+1})$.

Increment k and proceed to the next iteration.

Note that in the k th iteration either $x_k = 0$ and the algorithm stops, or it is possible to choose a colour i to be exchanged: either $\text{conv}(T_k)$ is full dimensional and x_k lies on its boundary, or $\text{conv}(T_k)$ has affine dimension less than d ; in both cases x_k can be expressed as a convex combination of at most d points from T_k by the usual Carathéodory theorem. We now derive an upper bound on the number of iterations (Lemma 2.2 below), which turns out to be no worse than for von Neumann’s noncolourful algorithm (Dantzig 1992, Freund 1995).

PROPOSITION 2.1. *Let $\epsilon > 0$ and $0 \leq \rho \leq 1$, and let $S_0, \dots, S_d \subset \mathbb{R}^d$ be normalized sets of points, each satisfying $B(0, \rho) \subset \text{conv}(S_i)$. Then, when Algorithm 1 is applied, the following recursions hold while $x_k \neq 0$:*

$$(1) \quad \text{If } \rho = 0: \frac{1}{|x_{k+1}|^2} \geq \frac{1}{4} + \frac{1}{|x_k|^2}; \quad \text{If } \rho > 0: |x_{k+1}|^2 \leq \left(1 - \frac{\rho^2}{4}\right) |x_k|^2.$$

PROOF. Consider the k th iteration. The point $q = -\rho x_k / |x_k|$ lies in the ball $B(0, \rho)$ hence in $\text{conv}(S_i)$, and satisfies $\langle x_k, q \rangle = -\rho |x_k|$. Therefore, there must be a point in S_i , in particular the new point t_i chosen by the algorithm, which also satisfies $\langle x_k, t_i \rangle \leq -\rho |x_k|$. Let p be the point closest to the origin on the line spanned by x_k and t_i . So p is the projection of 0 onto that line, hence

$$p = \frac{\langle t_i - x_k, t_i \rangle x_k + \langle x_k - t_i, x_k \rangle t_i}{\langle t_i - x_k, t_i - x_k \rangle} \quad \text{and} \quad |p|^2 = \frac{|x_k|^2 |t_i|^2 - \langle x_k, t_i \rangle^2}{|x_k|^2 + |t_i|^2 - 2\langle x_k, t_i \rangle}.$$

Since $\langle x_k, t_i \rangle \leq -\rho |x_k|$ this gives

$$|p|^2 \leq \frac{(|t_i|^2 - \rho^2) |x_k|^2}{|t_i|^2 + 2\rho |x_k| + |x_k|^2}.$$

Since the input is normalized we have $|t_i| \leq 2$, and so we get for $\rho = 0$:

$$\frac{1}{|p|^2} \geq \frac{1}{|t_i|^2} + \frac{1}{|x_k|^2} \geq \frac{1}{4} + \frac{1}{|x_k|^2},$$

and for $\rho > 0$:

$$|p|^2 \leq \frac{(|t_i|^2 - \rho^2)|x_k|^2}{|t_i|^2} \leq \left(1 - \frac{\rho^2}{4}\right)|x_k|^2.$$

Now $\langle x_k, t_i \rangle \leq 0$ and p is on the line spanned by x_k and t_i , so p in fact lies on the line segment $[x_k, t_i] \subset \text{conv}(T_{k+1})$. Since x_{k+1} is defined as the point in $\text{conv}(T_{k+1})$ of smallest norm, we have $|x_{k+1}| \leq |p|$ and the proposition follows. \square

One consequence of the proposition is the following proof of the colourful Carathéodory theorem:

PROOF OF THEOREM 1.1. Let $S_0 \dots, S_d$ be sets of points in $\mathbb{R}^d \setminus \{0\}$ such that $0 \in \bigcap_{i=0}^d \text{conv}(S_i)$. Normalize the points by scaling: for any subset $T \subseteq \bigcup_{i=0}^d S_i$ this does not affect whether or not $0 \in \text{conv}(T)$. Now apply Algorithm 1. By Proposition 2.1 with $\rho = 0$, as long as $x_k \neq 0$ we have $|x_{k+1}| < |x_k|$ hence $x_{k+1} \neq x_k$. Since x_k is uniquely defined from T_k and there are only finitely many colourful sets, we will eventually have $0 = x_k \in \text{conv}(T_k)$ proving that the origin is in the convex hull of some colourful set. \square

Another consequence is the following estimate on the number of iterations of Algorithm 1.

LEMMA 2.2. *Let $\epsilon > 0$ and $0 \leq \rho \leq 1$, and let $S_0, \dots, S_d \subset \mathbb{R}^d$ be normalized sets of points, each satisfying $B(0, \rho) \subset \text{conv}(S_i)$. Then we have the following upper bounds on the number of iterations performed by Algorithm 1 to find a colourful point which is within distance ϵ from the origin:*

$$(2) \quad \text{If } \rho = 0: \left\lceil \frac{4}{\epsilon^2} \right\rceil = O\left(\frac{1}{\epsilon^2}\right); \quad \text{If } \rho > 0: 1 + \left\lceil \frac{16}{\rho^2} \log \frac{2}{\epsilon} \right\rceil = O\left(\frac{1}{\rho^2} \log \frac{1}{\epsilon}\right).$$

PROOF. For $\rho = 0$, let $I = \lceil 4/\epsilon^2 \rceil$. Summation of the expressions in (1) for $k = 1, \dots, I - 1$ gives

$$\frac{1}{|x_I|^2} \geq \frac{1}{4} \left(\left\lceil \frac{4}{\epsilon^2} \right\rceil - 1 \right) + \frac{1}{|x_1|^2} \geq \frac{1}{4} \left\lceil \frac{4}{\epsilon^2} \right\rceil \geq \frac{1}{\epsilon^2},$$

which proves the claim in this case.

For $\rho > 0$, let $t = 4/(4 - \rho^2) > 1$ and let $I = 1 + \lceil (2/\log t) \log(2/\epsilon) \rceil$. Then $(I - 1) \log t \geq 2 \log(2/\epsilon)$, so

$$\left(\frac{1}{t}\right)^{I-1} \leq \left(\frac{\epsilon}{2}\right)^2,$$

and so by Proposition 2.1,

$$|x_I|^2 \leq \left(\frac{1}{t}\right)^{I-1} \cdot |x_1|^2 \leq \epsilon^2.$$

Since $\rho \leq 1$ we have

$$\log t = \log\left(\frac{4}{4 - \rho^2}\right) = \log\left(1 + \frac{\rho^2}{4 - \rho^2}\right) \geq \left(\frac{1}{2}\right) \frac{\rho^2}{4 - \rho^2} > \frac{\rho^2}{8},$$

so $2/\log t < 16/\rho^2$. This proves the claim in this case as well. \square

3. A variant of the algorithm and its arithmetic complexity. Finding a point x_{k+1} of minimum norm in $\text{conv}(T_{k+1})$ in the k th iteration of Algorithm 1 is a heavy task which involves the minimization of a quadratic form, and can be solved only approximately. To avoid this, we present a variant of the algorithm in which only linear algebraic computations (such as solutions of systems of linear equations and computations of determinants) will be required. Such computations can be easily carried out in strongly polynomial time (i.e., polynomial number of arithmetic operations and polynomial time in Turing computations on rational data) via Gaussian elimination. To arrive at such a variant of the algorithm, we reconsider the choice of the new point x_{k+1} in the k th iteration step of the algorithm. If x_{k+1} is chosen to be any point of $\text{conv}(T_{k+1})$ which is expressible as a convex combination of at most d elements, then the next iteration can be carried out—namely a colour can be found in which a new point can replace the old one. If x_{k+1} is chosen to be any point of $\text{conv}(T_{k+1})$ satisfying $|x_{k+1}| \leq |p|$, where p is the projection of 0 to the line segment $[x_k, t_i]$ as in the proof of Proposition 2.1, then the proposition remains true. So the bound on the number of iterations in Lemma 2.2 will hold for any variant of Algorithm 1 in which, at each iteration k , the point x_{k+1} is chosen so as to satisfy these two properties. This suggests the following variant of Algorithm 1. In the k th iteration, the set T_{k+1} is constructed as before. Then the point p is computed. If $\text{conv}(T_{k+1})$ happens to contain p on its boundary or is not full dimensional, then p is expressible as a convex combination of d or fewer points from T_{k+1} , so p itself can be taken as the next point x_{k+1} . Otherwise, $\text{conv}(T_{k+1})$ is a d -simplex containing p in its interior. In this case it is easy to compute the at most $d + 1$ intersection points $\alpha_j p$ of the line spanned by p and 0 with the hyperplanes $\text{aff}(T_{k+1} \setminus \{t_j\})$ spanned by the facets of $\text{conv}(T_{k+1})$. It is then possible to either conclude that $0 \in \text{conv}(T_{k+1})$ or to identify the boundary point αp of $\text{conv}(T_{k+1})$ where the simplex is stabbed by the line from 0 to p . This boundary point can then be taken to be the next point x_{k+1} .

Here is a more formal description of this algorithm. In each intermediate iteration it maintains a colourful set $T_k = \{t_0, \dots, t_d\}$, an approximating point $x_k \in \text{conv}(T_k)$, and a coefficient vector λ^k expressing x_k as a convex combination $x_k = \sum_{j=0}^d \lambda_j^k t_j$ of T_k with at least one $\lambda_i^k = 0$. The precise implementation details of the various steps are included in the proof of Theorem 3.1 below.

Algorithm 2.

- *Initialization.* Put $k = 1$. Pick an arbitrary colourful set $T_1 = \{t_0, \dots, t_d\}$. Let $\lambda^1 = (1, 0, \dots, 0)$ and $x_1 = \sum_{i=0}^d \lambda_i^1 t_i = t_0$.

- *Iteration.* If $|x_k| \leq \epsilon$ then stop and output $T_k, x_k,$ and λ^k . Otherwise proceed as follows:

- (1) Update the colourful set: choose i such that $\lambda_i^k = 0$; choose a point $t \in S_i$ minimizing the inner product $\langle x_k, t \rangle$; modify $t_i := t$ and let $T_{k+1} = \{t_0, \dots, t_d\}$ be the resulting new colourful set.

- (2) Compute the projection point p of 0 onto the line segment $[x_k, t_i]$ expressed as a convex combination $p = \sum_{j=0}^d \lambda_j t_j$ of T_{k+1} . If $p = 0$ or some $\lambda_j = 0$ then go to Step 5.

- (3) If T_{k+1} is affinely dependent then modify λ so that some λ_j becomes zero while $p = \sum_{j=0}^d \lambda_j t_j$ is maintained, and go to Step 5.

- (4) Determine the point αp where the line directed from 0 to p enters $\text{conv}(T_{k+1})$. If $\alpha \leq 0$ then let $p := 0 \in \text{conv}(T_{k+1})$, whereas if $\alpha > 0$ let $p := \alpha p$. Express p as a convex combination $p = \sum_{j=0}^d \lambda_j t_j$ of T_{k+1} .

- (5) Put $\lambda^{k+1} := \lambda$ and $x_{k+1} := p$.

Increment k and proceed to the next iteration.

The following theorem establishes the correctness and the real arithmetic complexity of this algorithm. The proof includes a description of the implementation details. The analysis is for the case of positive ρ , but can easily be adopted to the case $\rho = 0$ as well. The theorem gives a bound on the number of arithmetic operations performed on normalized data, which depends polynomially on the dimension, the number of points, and $\log(1/\epsilon)$.

THEOREM 3.1. *Given $\rho > 0$, $\epsilon > 0$, and normalized sets $S_0, \dots, S_d \subset \mathbb{R}^d$, each satisfying $|S_i| \leq n$ and $B(0, \rho) \subset \text{conv}(S_i)$, the number of real arithmetic operations taken by Algorithm 2 to find a colourful set containing a point ϵ -close to 0 is*

$$O\left(\frac{nd + d^4}{\rho^2} \log \frac{1}{\epsilon}\right).$$

PROOF. First we show that the algorithm is correct and maintains the bound on the number of iterations of Lemma 2.2. Consider the k th iteration loop. In Step 1, since at least one $\lambda_i^k = 0$, it is possible to construct the new colourful set T_{k+1} . Next, consider Step 2: the point p and the vector λ can be computed as follows. Let

$$\lambda_i = \frac{\langle x_k - t_i, x_k \rangle}{|t_i - x_k|^2} \quad \text{and} \quad \lambda_j = \frac{\langle t_i - x_k, t_i \rangle}{|t_i - x_k|^2} \lambda_j^k \quad (j \neq i).$$

Then, since $\langle x_k, t_i \rangle \leq 0$, the point

$$p = \sum_{j=0}^d \lambda_j t_j = \frac{\langle t_i - x_k, t_i \rangle x_k + \langle x_k - t_i, x_k \rangle t_i}{|t_i - x_k|^2}$$

is the projection of 0 onto the segment $[x_k, t_i]$, expressed as a convex combination of T_{k+1} (see proof of Proposition 2.1). If now $p = 0$, or $\lambda_j = 0$ for some j , then x_{k+1} can be taken to be p and it is possible to proceed to the next iteration. Otherwise, in Step 3, using Gaussian elimination it is possible to check if T_{k+1} is affinely dependent, and if it is, find a nontrivial affine relation $\sum_{j=0}^d \mu_j t_j = 0$ with $\sum_{j=0}^d \mu_j = 0$. It is then possible to choose a suitable multiplier δ so that $\lambda := \lambda + \delta \mu$ remains nonnegative and some λ_j becomes zero, and then proceed to the next iteration. If Step 4 is reached, then $\text{conv}(T_{k+1})$ is a full dimensional simplex containing $p \neq 0$ in its interior. To determine the point αp where the line $\text{lin}(p)$ directed from 0 to p enters $\text{conv}(T_{k+1})$, proceed as follows. For $j = 0, \dots, d$, this line intersects the hyperplane $\text{aff}(T_{k+1} \setminus \{t_j\})$ if and only if the determinant

$$\Delta = \det \begin{bmatrix} 0 & 1 & \cdots & 1 & 1 & \cdots & 1 \\ p & t_0 & \cdots & t_{j-1} & t_{j+1} & \cdots & t_d \end{bmatrix}$$

is nonzero, in which case the intersection point is $\alpha_j p$ where

$$\alpha_j = -\frac{1}{\Delta} \det[t_0 \cdots t_{j-1} \quad t_{j+1} \cdots t_d].$$

For each j , if $\Delta \neq 0$ then compute α_j by the expression above, whereas if $\Delta = 0$ then set $\alpha_j = \infty$. Let $\alpha = \max\{\alpha_j : 0 \leq j \leq d, \alpha_j < 1\}$. Then, since $p = 1 \cdot p$ is in the interior of $\text{conv}(T_{k+1})$, the desired point where $\text{lin}(p)$ enters $\text{conv}(T_{k+1})$ is αp . Now, if $\alpha \leq 0$ then 0 is in the line segment $[\alpha p, p]$ hence in $\text{conv}(T_{k+1})$, and p will be modified to $p := 0$. If $\alpha > 0$, then αp is in $[0, p]$ hence closer to 0 than p , and p will be modified to p

$:= \alpha p$. Since $\text{conv}(T_{k+1})$ is a simplex in this situation, the unique vector λ which expresses p as a convex combination of T_{k+1} is given by

$$\lambda = \begin{bmatrix} 1 & \cdots & 1 \\ t_0 & \cdots & t_d \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ p \end{bmatrix}.$$

If $p = 0$ then the algorithm will terminate in the following iteration, whereas if $p \neq 0$ then p is a boundary point of $\text{conv}(T_{k+1})$ so some λ_i is zero and it is possible to proceed to the next iteration.

The number of arithmetic operations in each iteration can be bounded as follows. In Step 1, the inner product $\langle x_k, t \rangle$ is computed and compared for $|S_i| \leq n$ points t , which involves $O(nd)$ operations. Step 2 involves $O(d^2)$ arithmetic operations. The work in Step 3 is dominated by the Gaussian elimination, which takes $O(d^3)$ operations. Step 4 involves $O(d)$ determinant computations and one matrix inversion, each done again by Gaussian elimination. So this step involves $O(d^4)$ operations. Thus, the total number of arithmetic operations per iteration is $O(nd + d^4)$. By Lemma 2.2, for positive ρ the number of iterations is $O((1/\rho^2) \log(1/\epsilon))$, which gives the bound stated in the theorem. \square

4. Computational complexity for rational data. In this section we discuss the time complexity on a Turing machine when the input consists of rational points, and the possible conversion of the approximation algorithm to one that finds a genuine colourful set containing 0 in its convex hull.

The bit size L of the input points $\cup_{i=0}^d S_i \subset \mathbb{Q}^d$ is the total number of binary bits needed to encode all coordinate values of all the points. Note that $L \geq d \cdot \sum_{i=0}^d |S_i| > d^2$. We start with the following two propositions.

PROPOSITION 4.1. *Let $V = \{v_1, \dots, v_m\}$ be a set of points in \mathbb{Q}^{d-1} of bit size L . If $0 \notin \text{conv}(V)$ then the norm of any $x \in \text{conv}(V)$ is larger than 2^{-3dL} .*

PROOF. Suppose $0 \notin \text{conv}(V)$ and let x^* be the point of minimum norm in $\text{conv}(V)$. By Carathéodory's theorem there is a subset $U = \{u_1, \dots, u_k\} \subset V$ of affinely independent points such that x^* is in the relative interior of $\text{conv}(U)$. Thus, x^* equals the orthogonal projection of 0 onto $\text{aff}(U)$. To express it let $\mathbf{1}$ be the vector of all 1 in \mathbb{Q}^k and let A be the full column rank matrix

$$A = \begin{bmatrix} 1 & \cdots & 1 \\ u_1 & \cdots & u_k \end{bmatrix}.$$

We claim that $x^* = p$ where p is defined by

$$\begin{bmatrix} p_0 \\ p \end{bmatrix} := \frac{1}{\mathbf{1}^T(A^T A)^{-1} \mathbf{1}} \cdot A(A^T A)^{-1} \mathbf{1}.$$

Indeed, $p_0 = 1$, so p is an affine combination of the u_i hence lies in $\text{aff}(U)$, and

$$A^T \cdot \begin{bmatrix} p_0 \\ p \end{bmatrix} = \frac{1}{\mathbf{1}^T(A^T A)^{-1} \mathbf{1}} \cdot \mathbf{1},$$

which shows that the value of $\langle u_i, p \rangle$ is the same for all i , so $p - 0$ is orthogonal to $\text{aff}(U)$.

Now, the bit size of A (which is the total number of bits needed to encode all of its entries) is at most L , so the common denominator of the entries of A is a positive integer

q satisfying $q \leq 2^L$, and qA is an integer matrix whose entries are of absolute value at most 2^L . Let B be the adjoint matrix of $q^2(A^T A)$. The entries of B are proper minors of $q^2(A^T A)$, so by Hadamard's inequality they are integers of absolute value at most $d^{(3/2)d} \cdot 2^{2dL}$, and so $\mathbf{1}^T B \mathbf{1} \leq d^{(3/2)d+2} \cdot 2^{2dL}$. Now $(A^T A)^{-1}$ is a scalar multiple of B , and so

$$\begin{bmatrix} 1 \\ x^* \end{bmatrix} = \frac{1}{q \mathbf{1}^T B \mathbf{1}} \cdot (qA)B \mathbf{1}.$$

Therefore each coordinate of x^* is of the form p/D for some integer p , with

$$D = q \cdot (\mathbf{1}^T B \mathbf{1}) \leq d^{(3/2)d+2} \cdot 2^{(2d+1) \cdot L} < 2^{3dL}.$$

By assumption $x^* \neq 0$ hence has a nonzero coordinate x_i^* . Therefore, as claimed,

$$|x^*| \geq |x_i^*| \geq \frac{1}{D} > 2^{-3dL}.$$

□

PROPOSITION 4.2. *Let $V = \{v_1, \dots, v_m\}$ be a set of points in \mathbb{Q}^{d-1} of bit size L . If the interior of $\text{conv}(V)$ contains the origin 0 then in fact it contains the ball $B(0, 2^{-3dL})$.*

PROOF. Suppose to the contrary that 0 is in the interior of $\text{conv}(V)$ but there is a point x on the boundary of $\text{conv}(V)$ with $|x| \leq 2^{-3dL}$. Let U be a subset of V such that $\text{conv}(U)$ is a face of $\text{conv}(V)$ which contains x . Since 0 is interior it does not lie in $\text{conv}(U)$. Since the bit size of $U \subset V$ is at most L , Proposition 4.1 implies $|x| > 2^{-3dL}$ which is a contradiction. □

We now proceed to discuss the complexity of Algorithm 2 for rational data. We need to make sure that the bit size of the numbers involved in the computations throughout the algorithm remains polynomially bounded in the size of the input. This is achieved by replacing Step 5 in the iteration loop of Algorithm 2 by the following *rounding step*:

Rounding step for rational data: let $D = \lceil 32 \cdot d \cdot 2^{6(d+1)L} / \epsilon \rceil$. Put $x_{k+1} := \sum_{j=0}^d \lambda_j^{k+1} t_j$, where

$$\lambda_j^{k+1} := \frac{\lceil D \lambda_j \rceil}{\sum_{r=0}^d \lceil D \lambda_r \rceil} \quad (j = 0, \dots, d).$$

We now show that with the rounding step, the running time of Algorithm 2 on a Turing machine when applied to normalized rational input, is pseudopolynomial in ρ and polynomial in the rest of the data.

THEOREM 4.3. *Given $\rho > 0$, rational $\epsilon > 0$ and rational normalized sets $S_0, \dots, S_d \subset \mathbb{Q}^d$ of bit size L , each satisfying $B(0, \rho) \subset \text{conv}(S_i)$, the time taken by Algorithm 2 with the rounding step to find a colourful set containing a point ϵ -close to 0 is polynomial in L , $\log(1/\epsilon)$, and $1/\rho$.*

PROOF. First, we show that the bound of Lemma 2.2 on the number of iterations remains valid. Consider the k th iteration of the algorithm, and let λ and $p = \sum_{j=0}^d \lambda_j t_j$ be as computed in this iteration before entering the rounding step. Now apply the rounding step and obtain x_{k+1} and λ^{k+1} . Note that the rounded λ^{k+1} is nonnegative and satisfies $\sum_{j=0}^d \lambda_j^{k+1} = 1$, so expresses x_{k+1} as an exact convex combination of the t_j . Now, for each i the origin is in the interior of $\text{conv}(S_i)$ so by Proposition 4.2 we have $B(0, 2^{-3(d+1)L}) \subset \text{conv}(S_i)$. Let $\bar{\rho} = \max\{\rho, 2^{-3(d+1)L}\}$. Then

$$D = \left\lceil \frac{32 \cdot d \cdot 2^{6(d+1)L}}{\epsilon} \right\rceil \geq \frac{32 \cdot d}{\epsilon \bar{\rho}^2} \quad \text{and} \quad 1 - \frac{\bar{\rho}^2}{8} \geq \sqrt{1 - \frac{\bar{\rho}^2}{4}},$$

so

$$D \geq \frac{4d}{\epsilon \left(1 - \sqrt{1 - \frac{\bar{\rho}^2}{4}}\right)}.$$

Since $B(0, \bar{\rho}) \subset \text{conv}(S_i)$ for all i , we obtain by Proposition 2.1 that

$$\begin{aligned} |x_{k+1}| &\leq \left| \frac{1}{D} \sum_{j=0}^d \lceil D\lambda_j \rceil t_j \right| = \left| \sum_{j=0}^d \lambda_j t_j + \sum_{j=0}^d \frac{1}{D} \left(\lceil D\lambda_j \rceil - D\lambda_j \right) t_j \right| \\ &\leq |p| + \frac{1}{D} \sum_{j=0}^d |t_j| \leq \left(\sqrt{1 - \frac{\bar{\rho}^2}{4}} \right) |x_k| + \frac{1}{D} d \cdot 2 \\ &\leq \left(\sqrt{1 - \frac{\bar{\rho}^2}{4}} \right) |x_k| + \frac{1 - \sqrt{1 - \frac{\bar{\rho}^2}{4}}}{2} \epsilon \leq \frac{1 + \sqrt{1 - \frac{\bar{\rho}^2}{4}}}{2} |x_k|. \end{aligned}$$

So $|x_{k+1}|$ is a constant fraction of $|x_k|$. The recursion (1) deduced in Proposition 2.1 can be replaced by the recursion $|x_{k+1}| \leq (1 + \sqrt{1 - \bar{\rho}^2}/u)/2 |x_k|$, and an analysis similar to that of Lemma 2.2 shows that number of iterations is $O((1/\bar{\rho}^2) \log(1/\epsilon)) = O((1/\rho^2) \times \log(1/\epsilon))$.

Next, we show that the running time per iteration is polynomially bounded in the data. Since the number of arithmetic operations per iteration can be bounded as in the proof of Theorem 3.1, it is enough to show that the bit size of the numbers appearing throughout the iteration remains polynomially bounded in the size of the input. The key point is that the rounding step makes sure that, at the end of the $(k - 1)$ th iteration, the numerator of each λ_j^k is a positive integer not exceeding D , and its denominator is a positive integer not exceeding dD . Since $\log D = O(\log d + \log(1/\epsilon) + dL)$, this guarantees that the bit size of the vector λ^k and the point $x_k = \sum_{j=0}^d \lambda_j^k t_j$ used at the beginning of the k th iteration are $O(d^2L + d \log(1/\epsilon))$. Therefore, the data manipulated within each iteration, which consists only of the original data and λ^k and x_k , has size polynomially bounded in the input size. So the growth in size of numbers is not accumulated from one iteration to the next. Now the work in the iteration consists of standard computations in Steps 1 and 2, plus $O(d)$ Gaussian eliminations in Steps 3 and 4. Since Gaussian elimination is polynomial time implementable (see Schrijver 1986), we conclude that the running time per iteration is polynomial in L and $\log(1/\epsilon)$. \square

Finally, we show that if ϵ is specified to be small enough then the colourful set found by Algorithm 2 contains 0 in its convex hull, and so provides an exact solution of the colourful linear programming problem. We are grateful to a referee for his question which prompted us to derive the following result.

THEOREM 4.4. *There is an algorithm that, given $\rho > 0$ and rational normalized sets S_0, \dots, S_d in \mathbb{Q}^d of bit size L , each satisfying $B(0, \rho) \subset \text{conv}(S_i)$, finds in time polynomial in L and $1/\rho$ a colourful set T containing 0 in its convex hull.*

PROOF. Let L be the bit size of the input $\cup_{i=0}^d S_i$, and define $\epsilon = 2^{-3(d+1)L}$. Apply Algorithm 2 with the rounding step. By Theorem 4.3 the algorithm stops in time polynomial in L , $\log(1/\epsilon) = O(dL)$, and $1/\rho$, and outputs a colourful set T whose convex hull contains a point x of norm $|x| \leq \epsilon$. By Proposition 4.1 above, $\text{conv}(T)$ contains the origin as well, and so T is an exact solution of the colourful linear programming problem. \square

5. Colourful linear algebra and its complexity. A set $S = \{s_1, \dots, s_n\}$ of vectors in \mathbb{R}^d is *positively dependent* if it has a nontrivial linear dependency $\sum_{i=1}^n \mu_i s_i = 0$ with all μ_i nonnegative. Since this is equivalent to $0 \in \text{conv}(S)$, Carathéodory’s theorem now says that if $S \subset \mathbb{R}^d$ is positively dependent then it has a subset of size at most $d + 1$ which also is. This is in analogy with the trivial statement for a linearly dependent set. Given now a family of colours $S_0, \dots, S_d \subset \mathbb{R}^d$, the colourful Carathéodory theorem says the following: if each member S_i of the family is positively dependent, then the family admits a positively dependent colourful set $T = \{s_0, \dots, s_d\}$. Again, this is in analogy with the trivial statement for linear dependence.

We now consider several algorithmic problems that arise, and that turn out to be hard for the linear-dependence case as well. For each problem, we shall distinguish a **Decision** problem, a **Search** problem, and a **Counting** problem. Here we shall restrict attention to the Turing machine computation model, and to the field \mathbb{Q} of rational integers. However, these problems have obvious analogs over any field \mathcal{F} and any type of \mathcal{P} -dependency (i.e., when the coefficients in the linear relation are required to come from a fixed subset $\mathcal{P} \subset \mathcal{F}$) and could be studied under other models of computation such as by Blum, Shub, and Smale (1989).

Colourful set problem. Given are k, d , and a family of nonempty colours $S_1, \dots, S_k \subset \mathbb{Q}^d$.

- **Decide** if the family has a linearly (positively) dependent (independent) colourful set.
- **Find** such a colourful set if one exists.
- **Count** the number of such colourful sets.

Note that the positive-dependence version is the same as the colourful linear programming problem. Also note that if $k = d + 1$ then *any* colourful set is linearly dependent, so the linear-dependence version of the problem becomes trivial. If $k = d + 1$ and each S_i is positively dependent, the positive-dependence version of the *decision* problem also becomes trivial by Theorem 1.1. So the following statement is sharp.

THEOREM 5.1. *Deciding linearly (positively) dependent colourful sets is \mathcal{NP} -complete, even if $k = d$ and each S_i is itself positively dependent.*

PROOF. We prove the statements for the linear and positive versions simultaneously. Clearly both problems are in the complexity class \mathcal{NP} . We reduce the \mathcal{NP} -complete problem of *Partition* (see Garey and Johnson 1979) to each. Given positive integers a_1, \dots, a_d , let for $i = 2, \dots, d$,

$$u_1 = a_1 e_1 + \sum_{i=2}^d e_i, \quad v_1 = -a_1 e_1 + \sum_{i=2}^d e_i,$$

$$\text{and } u_i = a_i e_1 - e_i, \quad v_i = -a_i e_1 - e_i,$$

where e_1, \dots, e_d denote the standard unit vectors in \mathbb{Q}^d , and let $S_i = \{u_i, v_i\}$ for all i . It is now easy to verify that there is a linearly dependent colourful set if and only if there is a positively dependent one if and only if there is a partition $I \uplus J$ of $[d] = \{1, \dots, d\}$ such that $\sum_{i \in I} a_i = \sum_{j \in J} a_j$: first, if $I \uplus J$ is such a partition then $\sum_{i \in I} u_i + \sum_{j \in J} v_j$

$= 0$, so $\{u_i : i \in I\} \cup \{v_j : j \in J\}$ is a linearly and positively dependent colourful set. Conversely, let $T = \{s_1, \dots, s_d\}$ be a colourful set admitting a nontrivial linear relation $\sum_{j=1}^d \lambda_j s_j = 0$. For $i = 2, \dots, d$, by considering the i th coordinate of the equation $0 = \sum_{j=1}^d \lambda_j s_j$ we find that $0 = \lambda_1 - \lambda_i$. Therefore $\lambda_1 = \dots = \lambda_d$ and so $\sum_{i=1}^d s_i = 0$ as well. So T is also positively dependent, and the sets $I = \{i : s_i = u_i\}$ and $J = \{j : s_j = v_j\}$ form a partition of $[d]$ with the desired property. To prove that the problem remains \mathcal{NP} -complete if each S_i is itself positively dependent, simply take $S_i = \{u_i, v_i, -u_i, -v_i\}$. Now, if $\{s_1, \dots, s_d\}$ is a linearly dependent colourful set with $\sum_{i=1}^d \lambda_i s_i = 0$ then, flipping the sign of both s_i and λ_i if necessary, we find another linearly dependent colourful set with $s_i \in \{u_i, v_i\}$ for all i , and proceed as above to show that it is also positively dependent and to construct a suitable partition of $[d]$. \square

It would be interesting to determine the complexity of deciding the existence of a positively dependent colourful set when $k = d + 1$ but the S_i are not necessarily positively dependent.

Related is the following hardness statement.

THEOREM 5.2. *Given two sets $S_1, S_2 \in \mathbb{Q}^d$, it is \mathcal{NP} -complete to decide whether there is a positively dependent set T of size d with $|T \cap S_1| = |T \cap S_2|$.*

PROOF. By reduction from *exact partition* (Garey and Johnson 1979). Given positive integers a_1, \dots, a_d , let u_i and v_i be as in the proof of Theorem 5.1, and let $S_1 = \{u_1, \dots, u_d\}$ and $S_2 = \{v_1, \dots, v_d\}$. Then it is easy to verify that there is a set T as desired if and only there is a partition $I \uplus J$ of $[d]$ such that $|I| = |J|$ and $\sum_{i \in I} a_i = \sum_{j \in J} a_j$. \square

Note that by taking $d/2$ copies of each of S_1 and S_2 above, this implies again the hardness of deciding positively dependent colourful sets.

In contrast with Theorem 5.1 we have the following statement observed together with M. Loeb.

THEOREM 5.3. *Deciding linearly independent colourful sets can be done in polynomial time.*

PROOF. Let V be the disjoint union of the S_i (so a point of \mathbb{Q}^d appearing in several S_i will have several copies in V). Define two matroids M_1, M_2 on V as follows: M_1 will be the matroid of linear dependencies on V , while M_2 will be the matroid whose bases are all colourful sets. Then a k -subset of V is a linearly independent colourful set if and only if it is independent in both M_1 and M_2 , and so the decision (and search) problem reduce to 2-matroid intersection, which can be done in polytime. \square

This proof raises some questions about matroids, which we discuss later on. But before that, we show that counting is hard for all four variants of the problem.

THEOREM 5.4. *Counting linearly (respectively, positively) dependent (respectively, independent) colourful sets is $\#P$ -complete.*

PROOF. All problems are clearly in $\#P$. We now reduce the $\#P$ -complete problem of computing the *permanent* of a $\{0, 1\}$ -matrix (see Valiant 1979) to each. Let $A = (A_{i,j})$ be such a matrix of size $d \times d$, and for $i = 1, \dots, d$ define $S_i = \{e_j : A_{i,j} = 1\}$. Then

$$\begin{aligned} \text{perm}(A) &:= \sum \left\{ \prod_{i=1}^d A_{i,\pi(i)} : \pi \text{ permutation on } [d] \right\} \\ &= \sum \{1 : \pi \text{ permutation, } e_{\pi(1)} \in S_1, \dots, e_{\pi(d)} \in S_d\} \\ &= \#\{\text{linearly independent colourful sets of } S_1, \dots, S_d\} \\ &= \prod_{i=1}^d |S_i| - \#\{\text{linearly dependent colourful sets of } S_1, \dots, S_d\}. \end{aligned}$$

This proves the hardness of counting linearly independent and linearly dependent transversals. For the positive analogs, define sets T_0, \dots, T_d in \mathbb{Q}^{d+1} by

$$T_0 = \left\{ -\sum_{j=1}^d e_j \right\}, \quad T_i = \{ e_j : A_{i,j} = 1 \} \cup \{ e_{d+1} \}, \quad i = 1, \dots, d.$$

It is easy to see that a d -tuple of vectors v_1, \dots, v_d forms a linearly independent transversal of S_1, \dots, S_d if and only if v_1, \dots, v_d together with $-\sum_{j=1}^d e_j$ forms a positively dependent transversal of T_0, T_1, \dots, T_d , so

$$\begin{aligned} \text{perm}(A) &= \#\{\text{positively dependent transversals of } T_0, \dots, T_d\} \\ &= \prod_{i=0}^d |T_i| - \#\{\text{positively independent transversals of } T_0, \dots, T_d\}. \end{aligned}$$

□

The linear variants of the colourful set problem form a special case of a general hierarchy of *Matroid Basis-Nonbasis* problems which we now introduce. Given are a pair $w = (w_1, w_2)$ of nonnegative integers and matroids $M_1, \dots, M_{w_1+w_2}$ of the same rank d , defined on the same set. A d -subset of the elements is a w -set if it is a basis in each of the first w_1 matroids and is not a basis in each of the last w_2 matroids. Note that, for complexity considerations, one has to specify the way in which the matroids are presented—e.g., by an independent set oracle, or by a matrix when the matroids are linear.

Matroid basis-nonbasis problem. Given are a pair $w = (w_1, w_2)$ of nonnegative integers, positive integers d and n , and $w_1 + w_2$ matroids of rank d defined on the same set of n elements.

- **Decide** if there exists a w -set.
- **Find** a w -set if one exists.
- **Count** the number of w -sets.

The decision and search problems are polynomial time solvable for $w = (1, 0)$ by the so-called greedy algorithm and for $w = (2, 0)$ by the 2-matroid intersection algorithm mentioned before, even if the matroids are given by oracles. For $w = (3, 0)$ the decision problem is known to be \mathcal{NP} -complete. The complexity for $w = (0, 1)$, which includes as a special case the problem of checking if n given points in d -space are in general position, is unknown. Theorem 5.1 shows that for $w = (1, 1)$ the decision problem is \mathcal{NP} -complete. It would be interesting to settle the complexity of these problems restricted to special classes of matroids.

Another interesting problem related to the colourful set problem concerns common zeros of systems of quadratic forms. Identify \mathbb{Q}^d with the vector subspace of linear forms in the algebra of polynomials $\mathbb{Q}[x_1, \dots, x_d]$ in the obvious way. Let \mathcal{F} be any field extension of \mathbb{Q} (possibly $\mathcal{F} = \mathbb{Q}$). A quadratic form is *simple* (of rank 1) if it is the product $u \cdot v$ of two linear forms $u, v \in \mathbb{Q}^d$.

THEOREM 5.5. *Deciding if a system of rational quadratic forms have a common non-trivial zero over \mathcal{F} is NP-complete, even if the number of forms equals the number of indeterminates and each form is simple.*

PROOF. By reduction from *Partition*: construct d sets $S_i = \{u_i, v_i\}$, where $u_i, v_i \in \mathbb{Q}^d$, as in the proof of Theorem 5.1. Now, the system $u_1 \cdot v_1, \dots, u_d \cdot v_d$ of simple quadratic forms in $\mathbb{Q}[x_1, \dots, x_d]$ admit a common nontrivial zero $\lambda \in \mathcal{F}^d$ if and only if

λ is in the orthogonal complement in \mathbb{Q}^d of some colourful set of the S_i , i.e., if and only if the S_i admit a linearly dependent colourful set. \square

6. Strong hardness of colourful linear programming. Here we show that the colourful linear programming problem is *strongly* \mathcal{NP} -complete (see Garey and Johnson 1979). Thus, even a pseudo-polynomial time algorithm (of running time polynomial in the unary representation of the data—see Garey and Johnson 1979 for the exact definition) does not exist unless $\mathcal{P} = \mathcal{NP}$.

THEOREM 6.1. *Given colours $S_1, \dots, S_d \subset \mathbb{Q}^d$ such that $0 \in \bigcap_{i=1}^d \text{conv}(S_i)$, it is strongly \mathcal{NP} -complete to decide whether there is a colourful $T = \{s_1, \dots, s_d\}$ such that $0 \in \text{conv}(T)$.*

PROOF. The proof is by reduction from 3-satisfiability. Let $f = c_1 \wedge \dots \wedge c_m$ be a given Boolean function in 3-conjunctive normal form on variables x_1, \dots, x_n , where each c_i is a clause of the form $c_i = l_{i_1} + l_{i_2} + l_{i_3}$, with each $l_r \in \{x_r, \bar{x}_r\}$ being a literal.

Let $k = d = n + m$, and let $\{e_{c_1}, \dots, e_{c_m}, e_{x_1}, \dots, e_{x_n}\}$ denote the standard basis of \mathbb{Q}^d , where the first m unit vectors correspond to clauses and the last n to variables, so that a typical point in \mathbb{Q}^d is denoted by $v = (v_{c_1}, \dots, v_{c_m}, v_{x_1}, \dots, v_{x_n})$.

We now construct sets of points T_1, \dots, T_m and S_1, \dots, S_n in \mathbb{Q}^d , corresponding again to clauses and variables, respectively, so that

$$0 \in \bigcap_{i=1}^m \text{conv}(T_i) \bigcap_{j=1}^n \text{conv}(S_j),$$

and such that there is a colourful T with $0 \in \text{conv}(T)$ if and only if f is satisfiable (here a colourful set is one that contains one point from each T_i and one from each S_j).

For each $j = 1, \dots, n$ we construct below three vectors v^j, \bar{v}^j , and u^j , and we let $S_j = \{v^j, \bar{v}^j, u^j\}$. For each $i = 1, \dots, m$ we construct below five vectors $w^{i,1}, w^{i,5}, w^{i,9}, y^i$, and z^i , and we let $T_i = \{w^{i,1}, w^{i,5}, w^{i,9}, y^i, z^i\}$.

Let A (respectively, \bar{A}) be the m by n incidence matrix of clauses and variables (respectively, negated variables), namely

$$A_{i,j} = \begin{cases} 1 & \text{if } x_j \in c_i, \\ 0 & \text{otherwise,} \end{cases} \quad \bar{A}_{i,j} = \begin{cases} 1 & \text{if } \bar{x}_j \in c_i, \\ 0 & \text{otherwise.} \end{cases}$$

Now let

$$v^1 = \sum_{i=1}^m (3A_{i,1} - \bar{A}_{i,1})e_{c_i} + \sum_{j=1}^n e_{x_j}, \quad \bar{v}^1 = \sum_{i=1}^m (-A_{i,1} + 3\bar{A}_{i,1})e_{c_i} + \sum_{j=1}^n e_{x_j},$$

and for $j = 2, \dots, n$,

$$v^j = \sum_{i=1}^m (3A_{i,j} - \bar{A}_{i,j})e_{c_i} - e_{x_j}, \quad \bar{v}^j = \sum_{i=1}^m (-A_{i,j} + 3\bar{A}_{i,j})e_{c_i} - e_{x_j}.$$

For $i = 1, \dots, m$ and $r = 1, 5, 9$ let

$$w^{i,r} = -re_{c_i} - \frac{1}{m} e_{x_1}, \quad y^i = 15e_{c_i} + 50e_x$$

Finally let $u^j = -(v^j + \bar{v}^j)$ and $z^i = -(w^{i,1} + w^{i,5} + w^{i,9} + y^i)$ so that S_j and T_i are positively dependent for all i and j .

Now, any satisfying assignment for f gives a positively dependent colourful set as follows. For $j = 1, \dots, n$ let

$$s^j = \begin{cases} v^j & \text{if } x_j = 1, \\ \bar{v}^j & \text{if } x_j = 0. \end{cases}$$

Since the assignment is satisfying, we have $r_i := \sum_{j=1}^n s^j_{c_i} \in \{1, 5, 9\}$, and we let $t^i = w^{i,r_i}$ for $i = 1, \dots, m$. It is then easy to verify that $\sum_{i=1}^m t^i + \sum_{j=1}^n s^j = 0$, so that $T = \{t^1, \dots, t^m, s^1, \dots, s^n\}$ is positively dependent.

For the converse, suppose $T = \{t^1, \dots, t^m, s^1, \dots, s^n\}$ is a colourful set and μ_i and λ_j are nonnegative numbers, not all zero, such that $a = 0$ where $a := \mu_i + \sum_{j=1}^n \lambda_j s^j$. By considering the various equations $a_{c_i} = 0$ and $a_{x_j} = 0$ for the various coordinates of $a = 0$, we will be able to exhibit a satisfying assignment for f . First, note that some λ_j must be nonzero: if not, then $0 = a_{c_i} = \mu_i t^i_{c_i}$, so whenever $\mu_i \neq 0$ there must hold $t^i_{c_i} = 0$ so $t^i = z^i$. But then $0 = a_{x_1} = \sum_{i=1}^m \mu_i z^i_{x_1}$ which would imply $\mu_i = 0$ for all i . Next, note that moreover $\lambda_1 = \lambda_2 = \dots = \lambda_n \neq 0$, and $s^j = u^j$ either for all j or for no j : this follows by the claim just proved and by considering the equations $a_{x_j} = 0$ for $j = 2, \dots, n$.

We now proceed to show that $s^j \in \{v^j, \bar{v}^j\}$ for all j . Suppose, indirectly, that this is not the case. Then, by the statement above, we have $s^j = u^j$ for all j . Write $\lambda = \lambda_1 = \dots = \lambda_n$. Then, for all i ,

$$0 = a_{c_i} = \sum_{l_j \in c_i} \lambda u^j_{c_i} + \mu_i t^i_{c_i} = 3\lambda \cdot (-2) + \mu_i t^i_{c_i},$$

so we must have $t^i_{c_i} > 0$, so $t^i = y^i$ and $\mu_i = 6\lambda/y^i_{c_i} = 6\lambda/15 = 2\lambda/5$. Thus,

$$0 = a_{x_1} = \lambda u^1_{x_1} + \sum_{i=1}^m \mu_i y^i_{x_1} = -2\lambda + m \frac{2}{5} \lambda \cdot 50 = (20m - 2)\lambda > 0,$$

which is impossible. Therefore indeed $s^j \in \{v^j, \bar{v}^j\}$, and we can define a variable assignment

$$x_j = \begin{cases} 1 & \text{if } s^j = v^j, \\ 0 & \text{if } s^j = \bar{v}^j. \end{cases}$$

It remains to show that this assignment is satisfying for f , namely that each clause c_i has at least one literal $l_j = 1$. Observe first that for all i we have $0 = a_{c_i} = \sum_{l_j \in c_i} \lambda s^j_{c_i} + \mu_i t^i_{c_i}$. Now $\sum_{l_j \in c_i} \lambda s^j_{c_i} \in \{-3\lambda, \lambda, 5\lambda, 9\lambda\}$ so is nonzero. Therefore, we must have $t^i_{c_i} \neq 0$, and so $t^i \neq z^i$. The claim will be established once we prove that $t^i \neq y^i$ either, which will imply that $t^i_{c_i} < 0$ so that we get $\sum_{l_j \in c_i} \lambda s^j_{c_i} \in \{\lambda, 5\lambda, 9\lambda\}$ which implies that clause c_i is satisfied.

Let I_1, I_2 be the partition of $\{1, \dots, m\}$ given by

$$I_1 = \{i: t^i \neq y^i\}, \quad I_2 = \{i: t^i = y^i\},$$

so we need to show that $I_2 = \emptyset$. Now, for any i we have $\sum_{l_j \in c_i} \lambda s^j_{c_i} \leq 9\lambda$. If $i \in I_1$ so that $t^i = w^{i,r}$, we have $t^i_{c_i} = -r$, so $0 = a_{c_i} = \sum_{l_j \in c_i} \lambda s^j_{c_i} + \mu_i t^i_{c_i}$ implies that $\mu_i r \leq 9\lambda$ so $\mu_i \leq 9\lambda/r \leq 9\lambda/1 = 9\lambda$. Thus,

$$0 = a_{x_1} = \lambda s_{x_1}^1 + \sum_{i=1}^m \mu_i t_{x_1}^i = \lambda + \sum_{i \in I_1} \mu_i \cdot \left(-\frac{1}{m}\right) + \sum_{i \in I_2} \mu_i \cdot 50 \geq \lambda - |I_1| \frac{9\lambda}{m} + 50 \sum_{i \in I_2} \mu_i,$$

so

$$\sum_{i \in I_2} \mu_i \leq \frac{\lambda}{50} \left(\frac{9|I_1|}{m} - 1\right) \leq \frac{8\lambda}{50} < \frac{1}{5} \lambda.$$

Thus, for all $i \in I_2$ we have $\mu_i \leq \sum_{i \in I_2} \mu_i < (1/5)\lambda$, and since $\mu_i t_{c_i}^i = \mu_i y_{c_i}^i = \mu_i \cdot 15$, we obtain $0 \leq \mu_i t_{c_i}^i < 3\lambda$. But then, for $i \in I_2$,

$$0 = a_{c_i} = \sum_{l_j \in c_i} \lambda s_{c_i}^j + \mu_i t_{c_i}^i$$

implies that $\sum_{l_j \in c_i} \lambda s_{c_i}^j \leq 0$ so we must have $\sum_{l_j \in c_i} \lambda s_{c_i}^j = -3\lambda$, so

$$0 = a_{c_i} = -3\lambda + \mu_i t_{c_i}^i < -3\lambda + 3\lambda = 0,$$

impossible. We conclude that indeed $I_2 = \emptyset$ and we are done. \square

7. Tverberg colourings. A *Tverberg k -colouring* of a set S of points in \mathbb{R}^d is a k -colouring of S such that the convex hulls of the monochromatic subsets have a point in common, i.e., a partition $S = \bigsqcup_{i=1}^k S_i$ such that the intersection $\bigcap_{i=1}^k \text{conv}(S_i)$ is nonempty. Tverberg (1966) proved the following, so-called *Tverberg’s theorem*.

THEOREM 7.1. *Any set of more than $(k - 1)(d + 1)$ points in \mathbb{R}^d has a Tverberg k -colouring.*

This theorem will follow from the proof of Theorem 7.2 below. Note that for $k = 2$, it reduces to the fact that any set of more than $d + 1$ points in \mathbb{R}^d is affinely dependent (*Radon’s theorem*): a 2-partition is given by the signs of the coefficients forming an affine dependency.

A remark in place here is that this theorem has an analogue over the integers. It says that for every k and d there is a finite number $t(k, d)$ such that any set of more than $t(k, d)$ points in $\mathbb{Z}^d \subset \mathbb{R}^d$ has an *integer k -colouring*, namely a k -colouring such that $\bigcap_{i=1}^k \text{conv}(S_i)$ contains an integer point. The determination of the numbers $t(k, d)$ is an outstanding open problem, even for $k = 2$. See Onn (1991) for the bounds $2^d < t(2, d) < d2^d$ and for related results.

Tverberg’s theorem raises the following algorithmic problems.

Tverberg colouring problem. Given are k, d , and a set S of points in \mathbb{Q}^d .

- **Decide** if the set admits a Tverberg k -colouring.
- **Find** a Tverberg k -colouring if one exists.
- **Count** the number of Tverberg k -colourings.

Recently, Sarkaria (1992) discovered that the colourful Carathéodory Theorem 1.1 implies Tverberg’s Theorem 7.1. Here we give a simplified variant of Sarkaria’s argument, and use it to reduce the algorithmic Tverberg colouring problem to the colourful linear programming problem.

THEOREM 7.2. *The decision (respectively, search, counting) variant of the Tverberg colouring problem is polynomial time reducible to the decision (respectively, search, counting) variant of the colourful linear programming problem.*

PROOF. Let k and $S = \{v_1, \dots, v_n\} \subset \mathbb{Q}^d$ be given. For $i = 1, \dots, k - 1$, let $f_i = e_i$ be the i th unit vector in \mathbb{Q}^{k-1} , and let $f_k = -\sum_{i=1}^{k-1} f_i$. Embed S in an open halfspace

in \mathbb{Q}^{d+1} , say replace each v_i by $\bar{v}_i = [v_i] \in \mathbb{Q}^{d+1}$. For $i = 1, \dots, n$ construct the following set T_i of k matrices in $\mathbb{Q}^{(d+1) \times (k-1)}$, regarded as points in $(k-1)(d+1)$ -space:

$$T_i = \{\bar{v}_i f_1^T, \dots, \bar{v}_i f_k^T\} \subset \mathbb{Q}^{(d+1) \times (k-1)}, \quad i = 1, \dots, n.$$

Now there is a natural bijection between colourful sets $\{t_1, \dots, t_n\}$ of T_1, \dots, T_n and k -colourings S_1, \dots, S_k of S : given a colourful set $\{t_1, \dots, t_n\}$, let

$$S_j = \{v_i : t_i = \bar{v}_i f_j^T\}, \quad j = 1, \dots, k.$$

We now show that a colourful set is positively dependent if and only if the corresponding colouring is Tverberg. Assume first that $\{t_1, \dots, t_n\}$ has a nontrivial positive dependency $\sum_{i=1}^n \mu_i t_i = 0$. We have

$$(3) \quad 0 = \sum_{j=1}^k \sum_{v_i \in S_j} \mu_i \bar{v}_i f_j^T = \sum_{j=1}^{k-1} \left(\sum_{v_i \in S_j} \mu_i \bar{v}_i - \sum_{v_i \in S_k} \mu_i \bar{v}_i \right) e_j^T.$$

Therefore,

$$(4) \quad \sum_{v_i \in S_j} \mu_i \bar{v}_i = \sum_{v_i \in S_k} \mu_i \bar{v}_i, \quad j = 1, \dots, k-1,$$

and in particular, by considering the $(d+1)$ th coordinate of the \bar{v}_i , we have

$$M := \sum_{v_i \in S_k} \mu_i = \sum_{v_i \in S_1} \mu_i = \dots = \sum_{v_i \in S_{k-1}} \mu_i, \quad M = \frac{1}{k} \sum_{i=1}^n \mu_i > 0.$$

We conclude that S_1, \dots, S_k is Tverberg, since

$$\frac{1}{M} \sum_{v_i \in S_k} \mu_i v_i \in \bigcap_{j=1}^k \text{conv}(S_j).$$

Conversely, if S_1, \dots, S_k is Tverberg with

$$\sum_{v_i \in S_1} \mu_i v_i = \dots = \sum_{v_i \in S_k} \mu_i v_i, \quad \mu_i \geq 0, \quad \sum_{v_i \in S_1} \mu_i = \dots = \sum_{v_i \in S_k} \mu_i = 1,$$

then (4) hence (3) hold, so $\{t_1, \dots, t_n\}$ is positively dependent. Thus, the Decision, Search, and Counting variants of the Tverberg colouring problem for S reduce to the corresponding variants of the colourful linear programming problem for T_1, \dots, T_n .

In particular, since the T_i lie in $(k-1)(d+1)$ -space and each is positively dependent by construction, the colourful Carathéodory Theorem 1.1 implies that if $n > (k-1)(d+1)$ then T_1, \dots, T_n have a positively dependent colourful set, hence S has a Tverberg k -colouring, which proves Tverberg’s Theorem 7.1. \square

8. Discussion. Tverberg’s Theorem 7.1 and the related algorithmic problems reduce to the Colourful Theorem 1.1 and to colourful linear programming. Similar reductions are possible for other theorems and related algorithmic problems. One example is that of finding a *weak ϵ -net*, a notion which has various applications in computational geometry: a set $T \subset \mathbb{R}^d$ is called a weak ϵ -net for a given set $S \subset \mathbb{R}^d$ if T intersects the convex hull of every ϵ -fraction of points of S . Alon, Bárány, Füredi, and Kleitman (1992) have shown that for every finite set $S \subset \mathbb{R}^d$ and $\epsilon > 0$, there exists a weak ϵ -net T of size at most $(d$

$+ 1)^{-(d+1)}\epsilon^{-(d+1)}$ (see Chazelle, Edelsbrunner, Grigni, Guibas, Sharir, and Welzl 1995 for improved bounds). This theorem was crucial in the solution of the Hadwiger-Debrunner (p, q) -problem by Alon and Kleitman (1992). It is possible to reduce this theorem and the algorithmic problem of finding a small ϵ -net to the colourful Carathéodory theorem and to colourful linear programming. A broader account of other applications of colourful linear programming will appear elsewhere. Related colourful conjectures and determinantal identities can be found in a recent article by Onn (1997).

Acknowledgment. The research of Imre Bárány was partially supported by Hungarian National Science Foundation no. 4296 and 016937. The research of Shmuel Onn was partially supported by the Alexander von Humboldt Stiftung, by the Fund for the Promotion of Research at the Technion, and by Technion VPR Fund no. 191–198. Shmuel Onn is grateful to the Alexander von Humboldt Stiftung for its support and to Peter Gritzmann for its hospitality in Universität Trier where part of this work was done. The authors thank Arkadi Nemirovski for very helpful discussions and suggestions and the referees for stimulating remarks and questions.

References

- Alon, N., I. Bárány, Z. Füredi, D. Kleitman (1992). Point selections and weak ϵ -nets for convex hulls. *Combin. Prob. Comp.* **1** 189–200.
- , D. Kleitman (1992). Piercing convex sets and the Hadwiger-Debrunner (p, q) -problem. *Adv. Math.* **96** 103–112.
- Bárány, I. (1982). A generalization of Carathéodory's theorem. *Discr. Math.* **40** 141–152.
- Blum, L., M. Shub, S. Smale (1989). On a theory of computation and complexity over the real numbers. *Bull. AMS* **21** 1–46.
- Chazelle, B., H. Edelsbrunner, M. Grigni, L. Guibas, M. Sharir, E. Welzl (1995). Weak ϵ -nets for convex sets. *Discr. Comp. Geom.* **13** 1–17.
- Dantzig, G. B. (1992). *An ϵ -precise feasible solution to a linear program in ϵ^{-2} iterations*, Report SOL/92/5, Systems Optimization Laboratory, Department of Operations Research, Stanford University.
- Freund, R. (1995). Personal communication.
- Garey, M. R., D. S. Johnson (1979). *Computers and Intractability*, W. H. Freeman, San Francisco.
- Kleinschmidt, P., S. Onn (1996). Signable posets and partitionable simplicial complexes. *Discr. Comp. Geom.* **15** 443–466.
- Onn, S. (1991). On the geometry and computational complexity of Radon partitions in the integer lattice. *SIAM J. Discr. Math.* **4** 436–447.
- (1997). A colorful determinantal identity, a conjecture of Rota, and Latin squares. *Amer. Math. Monthly* **104** 156–159.
- Sarkaria, K. S. (1992). Tverberg's theorem via number fields. *Israel J. Math.* **79** 317–320.
- Schrijver, A. (1986). *Theory of Linear and Integer Programming*, John Wiley, New York.
- Tverberg, H. (1966). A generalization of Radon's theorem. *J. London Math. Soc.* **41** 123–128.
- Valiant, L. G. (1979). The complexity of computing the permanent. *Theor. Comput. Sci.* **8** 189–201.

I. Bárány: Mathematical Institute of the Hungarian Academy of Sciences, P.O. Box 127, Budapest, 1364 Hungary; e-mail: barany@math-inst.hu

S. Onn: Operations Research, Faculty of Industrial Engineering and Management, Technion–Israel Institute of Technology, 32000 Haifa, Israel; onn@ie.technion.ac.il