PARETO OPTIMA OF MULTICRITERIA INTEGER LINEAR PROGRAMS

JESÚS A. DE LOERA, RAYMOND HEMMECKE, AND MATTHIAS KÖPPE

Abstract. We settle the computational complexity of fundamental questions related to multicriteria integer linear programs, when the dimensions of the strategy space and of the outcome space are considered fixed constants. In particular we construct:

1. polynomial-time algorithms to exactly determine the number of Pareto optima and Pareto strategies;
2. a polynomial-space polynomial-delay prescribed-order enumeration algorithm for arbitrary projections of the Pareto set;
3. an algorithm to minimize the distance of a Pareto optimum from a prescribed comparison point with respect to arbitrary polyhedral norms;
4. a fully polynomial-time approximation scheme for the problem of minimizing the distance of a Pareto optimum from a prescribed comparison point with respect to the Euclidean norm.

1. Introduction

Let $A = (a_{ij})$ be an integral $m \times n$-matrix and $\mathbf{b} \in \mathbb{Z}^m$ such that the convex polyhedron $P = \{\, \mathbf{u} \in \mathbb{R}^n : A\mathbf{u} \leq \mathbf{b} \,\}$ is bounded. Given k linear functionals $f_1, f_2, \ldots, f_k \in \mathbb{Z}^n$, we consider the *multicriterion integer linear programming problem*

$$\begin{aligned} \text{vmin} \quad & \big(f_1(\mathbf{u}), f_2(\mathbf{u}), \ldots, f_k(\mathbf{u})\big) \\ \text{subject to} \quad & A\mathbf{u} \leq \mathbf{b} \\ & \mathbf{u} \in \mathbb{Z}^n \end{aligned} \tag{1}$$

where vmin is defined as the problem of finding all Pareto optima and a corresponding Pareto strategy. For a lattice point $\mathbf{u}$ the vector $\mathbf{f}(\mathbf{u}) = \big(f_1(\mathbf{u}), \ldots, f_k(\mathbf{u})\big)$ is called an *outcome vector*. Such an outcome vector is a *Pareto optimum* for the above problem if and only if there is no other point $\tilde{\mathbf{u}}$ in the feasible set such that $f_i(\tilde{\mathbf{u}}) \leq f_i(\mathbf{u})$ for all i and $f_j(\tilde{\mathbf{u}}) < f_j(\mathbf{u})$ for at least one index j. The corresponding feasible point $\mathbf{u}$ is called a *Pareto strategy*. Thus a feasible vector is a Pareto strategy if no feasible vector can decrease some criterion without causing a simultaneous increase in at

Date: Revision: 1.29 − Date: 2007/07/09 22:49:05 .

J. A. De Loera gratefully acknowledges support from NSF grant DMS-0608785.

M. Köppe was supported by a Feodor Lynen Research Fellowship from the Alexander von Humboldt Foundation.

least one other criterion. For general information about the multicriteria problems see, e.g., [8, 12].

In general multiobjective problems the number of Pareto optimal solutions may be infinite, but in our situation the number of Pareto optima and strategies is finite. There are several well-known techniques to generate Pareto optima. Some popular methods used to solve such problems include, e.g., weighting the objectives or using a so-called global criterion approach (see [6]). In abnormally nice situations, such as multicriteria *linear* programs [9], one knows a way to generate all Pareto optima, but most techniques reach only some of the Pareto optima.

The purpose of this article is to study the sets of *all* Pareto optima and strategies of a multicriterion integer linear program using the algebraic structures of generating functions. The set of Pareto points can be described as the formal sum of monomials

$$\sum \{\, \mathbf{z}^{\mathbf{v}} :\ \mathbf{u} \in P \cap \mathbb{Z}^n \text{ and } \mathbf{v} = \mathbf{f}(\mathbf{u}) \in \mathbb{Z}^k \text{ is a Pareto optimum} \,\}. \tag{2}$$

Our main theoretical result states that, under the assumption that the number of variables is fixed, we can compute in polynomial time a compact expression for the huge polynomial above, thus *all* its Pareto optima can in fact be counted exactly. The same can be done for the corresponding Pareto strategies when written in the form

$$\sum \{\, \mathbf{x}^{\mathbf{u}} :\ \mathbf{u} \in P \cap \mathbb{Z}^n \text{ and } \mathbf{f}(\mathbf{u}) \text{ is a Pareto optimum} \,\}. \tag{3}$$

Theorem 1. *Let $A \in \mathbb{Z}^{m\times n}$, a d-vector $\mathbf{b}$, and linear functions $f_1, \dots, f_k \in \mathbb{Z}^n$ be given. There are algorithms to perform the following tasks:*

(i) *Compute the generating function (2) of all the Pareto optima as a sum of rational functions. In particular we can count how many Pareto optima are there. If we assume k and n are fixed, the algorithm runs in time polynomial in the size of the input data.*

(ii) *Compute the generating function (3) of all the Pareto strategies as a sum of rational functions. In particular we can count how many Pareto strategies are there in P. If we assume k and n are fixed, the algorithm runs in time polynomial in the size of the input data.*

(iii) *Generate the full sequence of Pareto optima ordered lexicographically or by any other term ordering. If we assume k and n are fixed, the algorithm runs in polynomial time on the input size and the number of Pareto optima. (More strongly, there exists a* polynomial-space polynomial-delay prescribed-order enumeration algorithm*.)*

In contrast it is known that for non-fixed dimension it is #P-hard to enumerate Pareto optima and NP-hard to find them [7, 13]. The proof of Theorem 1 parts (i) and (ii) will be given in section 2. It is based on the theory of rational generating functions as developed in [1, 2]. Part (iii) of Theorem 1 will be proved in section 3.

For a user that knows some or all of the Pareto optima or strategies, a goal is to select the "best" member of the family. One is interested in selecting one Pareto optimum that realizes the "best" compromise between the individual objective functions. The quality of the compromise is often measured by the distance of a Pareto optimum $\mathbf{v}$ from a user-defined comparison point $\hat{\mathbf{v}}$. For example, often users take as a good comparison point the so-called *ideal point* $\mathbf{v}^{\text{ideal}} \in \mathbb{Z}^k$ of the multicriterion problem, which is defined as

$$v_i^{\text{ideal}} = \min\{\, f_i(\mathbf{u}) : \mathbf{u} \in P \cap \mathbb{Z}^n \,\}.$$

The criteria of comparison with the point $\hat{\mathbf{v}}$ are quite diverse, but some popular ones include computing the minimum over the possible sums of absolute differences of the individual objective functions, evaluated at the different Pareto strategies, from the comparison point $\hat{\mathbf{v}}$, i.e.,

$$f(\mathbf{u}) = |f_1(\mathbf{u}) - \hat{v}_1| + \dots + |f_k(\mathbf{u}) - \hat{v}_k|, \tag{4a}$$

or the maximum of the absolute differences,

$$f(\mathbf{u}) = \max\bigl\{|f_1(\mathbf{u}) - \hat{v}_1|, \dots, |f_k(\mathbf{u}) - \hat{v}_k|\bigr\}, \tag{4b}$$

over all Pareto optima $(f_1(\mathbf{u}), \dots, f_k(\mathbf{u}))$. Another popular criterion, sometimes called the *global criterion*, is to minimize the sum of relative distances of the individual objectives from their known minimal values, i.e.,

$$f(\mathbf{u}) = \frac{f_1(\mathbf{u}) - v_1^{\text{ideal}}}{|v_1^{\text{ideal}}|} + \dots + \frac{f_k(\mathbf{u}) - v_k^{\text{ideal}}}{|v_k^{\text{ideal}}|}. \tag{4c}$$

We stress that if we take any one of these functions as an objective function of an integer program, the optimal solution will be a non-Pareto solution of the multicriterion problem (1) in general. In contrast, we show in this paper that by encoding Pareto optima and strategies as a rational function we avoid this problem, since we evaluate the objective functions directly on the space of Pareto optima.

All of the above criteria (4) measure the distance from a prescribed point with respect to a *polyhedral norm*. In section 4, we prove:

Theorem 2. *Let the dimension n and the number k of objective functions be fixed. Let a multicriterion integer linear program (1) be given. Let a polyhedral norm $\|\cdot\|_Q$ be given by the vertex or inequality description of its unit ball $Q \subseteq \mathbb{R}^k$. Finally, let a prescribed point $\hat{\mathbf{v}} \in \mathbb{Z}^k$ be given.*

(i) *There exists a polynomial-time algorithm to find a Pareto optimum $\mathbf{v}$ of (1) that minimizes the distance $\|\mathbf{v} - \hat{\mathbf{v}}\|_Q$ from the prescribed point.*
(ii) *There exists a polynomial-space polynomial-delay enumeration algorithm for enumerating the Pareto optima of (1) in the order of increasing distances from the prescribed point $\hat{\mathbf{v}}$.*

Often users are actually interested in finding a Pareto optimum that minimizes the *Euclidean* distance from a prescribed comparison point $\hat{\mathbf{v}}$,

$$f(\mathbf{u}) = \sqrt{|f_1(\mathbf{u}) - \hat{v}_1|^2 + \dots + |f_k(\mathbf{u}) - \hat{v}_k|^2}, \tag{5}$$

but to our knowledge no method of the literature gives a satisfactory solution to that problem. In section 4, however, we prove the following theorem, which gives a very strong approximation result.

Theorem 3. *Let the dimension n and the number k of objective functions be fixed. There exists a fully polynomial-time approximation scheme for the problem of minimizing the Euclidean distance of a Pareto optimum of (1) from a prescribed comparison point $\hat{\mathbf{v}} \in \mathbb{Z}^k$.*

We actually prove this theorem in a somewhat more general setting, using an arbitrary norm whose unit ball is representable by a homogeneous polynomial inequality.

2. The rational function encoding of all Pareto optima

We give a very brief overview of the theory of rational generating functions necessary to establish Theorem 1. For full details we recommend [1, 2, 3, 5] and the references therein. In 1994 Barvinok gave an algorithm for counting the lattice points in $P = \{\, \mathbf{u} \in \mathbb{R}^n : A\mathbf{u} \le \mathbf{b} \,\}$ in polynomial time when the dimension n of the feasible polyhedron is a constant [1]. The input for Barvinok's algorithm is the binary encoding of the integers a_{ij} and b_i, and the output is a formula for the multivariate generating function

$$g(P; \mathbf{x}) = \sum_{\mathbf{u} \in P \cap \mathbb{Z}^n} \mathbf{x}^{\mathbf{u}}$$

where $\mathbf{x}^{\mathbf{u}}$ is an abbreviation of $x_1^{u_1} x_2^{u_2} \dots x_n^{u_n}$. This long polynomial with exponentially many monomials is encoded as a much shorter sum of rational functions of the form

$$g(P; \mathbf{x}) = \sum_{i \in I} \gamma_i \frac{\mathbf{x}^{\mathbf{c}_i}}{(1 - \mathbf{x}^{\mathbf{d}_{i1}})(1 - \mathbf{x}^{\mathbf{d}_{i2}}) \dots (1 - \mathbf{x}^{\mathbf{d}_{in}})}. \tag{6}$$

Barvinok and Woods in 2003 further developed a set of powerful manipulation rules for using these short rational functions in Boolean constructions on various sets of lattice points.

Throughout the paper we assume that the polyhedron $P = \{\, \mathbf{u} \in \mathbb{R}^n : A\mathbf{u} \le \mathbf{b} \,\}$ is bounded. We begin by recalling some useful results of Barvinok and Woods (2003):

Theorem 4 (Intersection Lemma; Theorem 3.6 in [3])**.** *Let ℓ be a fixed integer. Let S_1, S_2 be finite subsets of $\mathbb{Z}^n$. Let $g(S_1; \mathbf{x})$ and $g(S_2; \mathbf{x})$ be their generating functions, given as short rational functions with at most ℓ binomials in each denominator. Then there exists a polynomial time algorithm, which computes*

$$g(S_1 \cap S_2; \mathbf{x}) = \sum_{i \in I} \gamma_i \frac{\mathbf{x}^{\mathbf{c}_i}}{(1 - \mathbf{x}^{\mathbf{d}_{i1}}) \dots (1 - \mathbf{x}^{\mathbf{d}_{is}})}$$

with $s \le 2\ell$, where the γ_i are rational numbers, $\mathbf{c}_i, \mathbf{d}_{ij}$ are nonzero integer vectors, and I is a polynomial-size index set.

The following theorem was proved by Barvinok and Woods using Theorem 4:

Theorem 5 (Boolean Operations Lemma; Corollary 3.7 in [3])**.** *Let m and ℓ be fixed integers. Let $S_1, S_2, \dots, S_m$ be finite subsets of $\mathbb{Z}^n$. Let $g(S_i;\mathbf{x})$ for $i = 1,\dots,m$ be their generating functions, given as short rational functions with at most ℓ binomials in each denominator. Let a set $S \subseteq \mathbb{Z}^n$ be defined as a Boolean combination of $S_1,\dots,S_m$ (i.e., using any of the operations $\cup$, $\cap$, $\setminus$). Then there exists a polynomial time algorithm, which computes*

$$g(S;\mathbf{x}) = \sum_{i\in I} \gamma_i \frac{\mathbf{x}^{\mathbf{c}_i}}{(1-\mathbf{x}^{\mathbf{d}_{i1}})\dots(1-\mathbf{x}^{\mathbf{d}_{is}})}$$

where $s = s(\ell, m)$ is a constant, the γ_i are rational numbers, $\mathbf{c}_i, \mathbf{d}_{ij}$ are nonzero integer vectors, and I is a polynomial-size index set.

We will use the *Intersection Lemma* and the *Boolean Operations Lemma* to extract special monomials present in the expansion of a generating function. The essential step in the intersection algorithm is the use of the *Hadamard product* [3, Definition 3.2] and a special monomial substitution. The Hadamard product is a bilinear operation on rational functions (we denote it by $*$). The computation is carried out for pairs of summands as in (6). Note that the Hadamard product $m_1 * m_2$ of two monomials m_1, m_2 is zero unless $m_1 = m_2$.

Another key subroutine introduced by Barvinok and Woods is the following *Projection Theorem*.

Theorem 6 (Projection Theorem; Theorem 1.7 in [3])**.** *Assume the dimension n is a fixed constant. Consider a rational polytope $P \subset \mathbb{R}^n$ and a linear map $T\colon \mathbb{Z}^n \to \mathbb{Z}^k$. There is a polynomial time algorithm which computes a short representation of the generating function $f\big(T(P\cap\mathbb{Z}^n);\mathbf{x}\big)$.*

One has to be careful when using earlier Lemmas (especially the Projection Theorem) that the sets in question are finite. The proof of Theorem 1 will require us to project and intersect sets of lattice points represented by rational functions. We cannot, in principle, do those operations for *infinite* sets of lattice points. Fortunately, in our setting it is possible to restrict our attention to finite sets.

Finally, one important comment. If we want to count the points of a lattice point set S, such as the set of Pareto optima, it would apparently suffice to substitute 1 for all the variables x_i of the generating function

$$g(S;\mathbf{x}) = \sum_{\mathbf{u}\in S} \mathbf{x}^{\mathbf{u}} = \sum_{i\in I} \gamma_i \frac{\mathbf{x}^{\mathbf{c}_i}}{(1-\mathbf{x}^{\mathbf{d}_{i1}})(1-\mathbf{x}^{\mathbf{d}_{i2}})\dots(1-\mathbf{x}^{\mathbf{d}_{in}})}$$

to get the specialization $|S| = g(S;\mathbf{x}=\mathbf{1})$. But this cannot be done directly due to the singularities in the rational function representation. Instead, choose a generic vector $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_n)$ and substitute each of the variables x_i by $\mathrm{e}^{t\lambda_i}$. Then we get

$$g(S, \mathbf{e}^{t\boldsymbol{\lambda}}) = \sum_{i \in I} \gamma_i \frac{\mathrm{e}^{t\langle \boldsymbol{\lambda}, \mathbf{c}_i \rangle}}{(1 - \mathrm{e}^{t\langle \boldsymbol{\lambda}, \mathbf{d}_{i1} \rangle})(1 - \mathrm{e}^{t\langle \boldsymbol{\lambda}, \mathbf{d}_{i2} \rangle}) \dots (1 - \mathrm{e}^{t\langle \boldsymbol{\lambda}, \mathbf{d}_{in} \rangle})}.$$

Counting the number of lattice points is the same as computing the constant terms of the Laurent series for each summand and adding them up. This can be done using elementary complex residue techniques (see [2]).

Proof of Theorem 1, part (i) and (ii). The proof of part (i) has three steps: *Step 1.* For $i = 1, \dots, k$ let $\bar{v}_i \in \mathbb{Z}$ be an upper bound of polynomial encoding size for the value of f_i over P. Such a bound exists because of the boundedness of P, and it can be computed in polynomial time by linear programming. We will denote the vector of upper bounds by $\bar{\mathbf{v}} \in \mathbb{Z}^k$. We consider the *truncated multi-epigraph* of the objective functions $f_1, \dots, f_k$ over the linear relaxation of the feasible region P,

$$\begin{aligned} P^{\geq}_{f_1,\dots,f_k} = \bigl\{ \, (\mathbf{u}, \mathbf{v}) \in \mathbb{R}^n \times \mathbb{R}^k : \mathbf{u} \in P, \\ \bar{v}_i \geq v_i \geq f_i(\mathbf{u}) \text{ for } i = 1, \dots, k \, \bigr\}, \end{aligned} \tag{7}$$

which is a rational convex polytope in $\mathbb{R}^n \times \mathbb{R}^k$. Let $V^{\geq} \subseteq \mathbb{Z}^k$ denote the integer projection of $P^{\geq}_{f_1,\dots,f_k}$ on the $\mathbf{v}$ variables, i.e., the set

$$V^{\geq} = \bigl\{ \, \mathbf{v} \in \mathbb{Z}^k : \exists \mathbf{u} \in \mathbb{Z}^n \text{ with } (\mathbf{u}, \mathbf{v}) \in P^{\geq}_{f_1,\dots,f_k} \cap (\mathbb{Z}^n \times \mathbb{Z}^k) \, \bigr\}. \tag{8}$$

Clearly, the vectors in $V^{\geq}$ are all integer vectors in the outcome space which are weakly dominated by some outcome vector $\bigl(f_1(\mathbf{u}), f_2(\mathbf{u}), \dots, f_k(\mathbf{u})\bigr)$ for a feasible solution $\mathbf{x}$ in $P \cap \mathbb{Z}^n$; however, we have truncated away all outcome vectors which weakly dominate the computed bound $\bar{\mathbf{v}}$. Let us consider the generating function of $V^{\geq}$, the multivariate polynomial

$$g(V^{\geq}; \mathbf{z}) \;= \sum \bigl\{ \, \mathbf{z}^{\mathbf{v}} : \mathbf{v} \in V^{\geq} \, \bigr\}.$$

In the terminology of polynomial ideals, the monomials in $g(V^{\geq}; \mathbf{z})$ form a truncated ideal generated by the Pareto optima. By the Projection Theorem (our Theorem 6), we can compute $g(V^{\geq}; \mathbf{z})$ in the form of a polynomial-size rational function in polynomial time.

Step 2. Let $V^{\mathrm{Pareto}} \subseteq \mathbb{Z}^k$ denote the set of Pareto optima. Clearly we have

$$V^{\mathrm{Pareto}} = \bigl(V^{\geq} \setminus (\mathbf{e}_1 + V^{\geq})\bigr) \cap \dots \cap \bigl(V^{\geq} \setminus (\mathbf{e}_k + V^{\geq})\bigr)\,,$$

where $\mathbf{e}_i \in \mathbb{Z}^k$ denotes the i-th unit vector and

$$\mathbf{e}_i + V^{\geq} = \{ \, \mathbf{e}_i + \mathbf{v} : \mathbf{v} \in V^{\geq} \, \}.$$

The generating function $g(V^{\mathrm{Pareto}}; \mathbf{z})$ can be computed by the Boolean Operations Lemma (Theorem 5) in polynomial time from $g(V^{\geq}; \mathbf{z})$ as

$$\begin{aligned} g(V^{\mathrm{Pareto}}; \mathbf{z}) = \bigl(g(V^{\geq}; \mathbf{z}) - g(V^{\geq}; \mathbf{z}) * z_1 g(V^{\geq}; \mathbf{z})\bigr) \\ * \dots * \bigl(g(V^{\geq}; \mathbf{z}) - g(V^{\geq}; \mathbf{z}) * z_k g(V^{\geq}; \mathbf{z})\bigr), \end{aligned} \tag{9}$$

where $*$ denotes taking the Hadamard product of the rational functions.

Step 3. To obtain the number of Pareto optima, we compute the specialization $g(V^{\mathrm{Pareto}}; \mathbf{z} = \mathbf{1})$. This is possible in polynomial time using residue techniques as outlined before the beginning of the proof.

Proof of part (ii). Now we recover the Pareto *strategies* that gave rise to the Pareto optima, i.e., we compute a generating function for the set

$$U^{\mathrm{Pareto}} = \left\{ \mathbf{u} \in \mathbb{Z}^n : \mathbf{u} \in P \cap \mathbb{Z}^n \text{ and } \mathbf{f}(\mathbf{u}) \text{ is a Pareto optimum} \right\}.$$

To this end, we first compute the generating function for the set

$$S^{\mathrm{Pareto}} = \left\{ (\mathbf{u}, \mathbf{v}) \in \mathbb{Z}^n \times \mathbb{Z}^k : \mathbf{v} \text{ is a Pareto point with Pareto strategy } \mathbf{u} \right\}.$$

For this purpose, we consider the multi-graph of the objective functions $f_1, \dots, f_k$ over P,

$$\begin{aligned} P^{=}_{f_1,\dots,f_k} = \big\{ (\mathbf{u}, \mathbf{v}) \in \mathbb{R}^n \times \mathbb{R}^k : \mathbf{u} \in P, \\ v_i = f_i(\mathbf{u}) \text{ for } i = 1, \dots, k \big\}. \end{aligned} \tag{10}$$

Using Barvinok's theorem, we can compute in polynomial time the generating function for the integer points in P,

$$g(P; \mathbf{x}) = \sum \left\{ \mathbf{x}^{\mathbf{u}} : \mathbf{u} \in P \cap \mathbb{Z}^n \right\},$$

and also, using the monomial substitution $x_j \to x_j z_1^{f_1(\mathbf{e}_j)} \cdots z_k^{f_k(\mathbf{e}_j)}$ for all j, the generating function is transformed into

$$g(P^{=}_{f_1,\dots,f_k}; \mathbf{x}, \mathbf{z}) = \sum \left\{ \mathbf{x}^{\mathbf{u}} \mathbf{z}^{\mathbf{v}} : (\mathbf{u}, \mathbf{v}) \in P^{=}_{f_1,\dots,f_k} \cap (\mathbb{Z}^n \times \mathbb{Z}^k) \right\},$$

where the variables $\mathbf{x}$ carry on the monomial exponents the information of the $\mathbf{u}$-coordinates of $P^{=}_{f_1,\dots,f_k}$ and the $\mathbf{z}$ variables of the generating function carry the $\mathbf{v}$-coordinates of lattice points in $P^{=}_{f_1,\dots,f_k}$. Now

$$g(S^{\mathrm{Pareto}}; \mathbf{x}, \mathbf{z}) = \big(g(P; \mathbf{x})\, g(V^{\mathrm{Pareto}}; \mathbf{z}) \big) * g(P^{=}_{f_1,\dots,f_k}; \mathbf{x}, \mathbf{z}), \tag{11}$$

which can be computed in polynomial time for fixed dimension by the theorems outlined early on this section. Finally, to obtain the generating function $g(U^{\mathrm{Pareto}}; \mathbf{x})$ of the Pareto strategies, we need to compute the projection of S^{Pareto} into the space of the strategy variables $\mathbf{u}$. Since the projection is one-to-one, it suffices to compute the specialization

$$g(U^{\mathrm{Pareto}}; \mathbf{x}) = g(S^{\mathrm{Pareto}}; \mathbf{x}, \mathbf{z} = \mathbf{1}),$$

which can be done in polynomial time. □

3. Efficiently listing all Pareto optima

The Pareto optimum that corresponds to the "best" compromise between the individual objective functions is often chosen in an *interactive mode*, where a visualization of the Pareto optima is presented to the user, who then chooses a Pareto optimum. Since the outcome space frequently is of a too large dimension for visualization, an important task is to list (explicitly enumerate) the elements of the *projection* of the Pareto set into some lower-dimensional linear space.

It is clear that the set of Pareto optima (and thus also any projection) is of exponential size in general, ruling out the existence of a polynomial-time enumeration algorithm. In order to analyze the running time of an enumeration algorithm, we must turn to *output-sensitive complexity analysis*.

Various notions of output-sensitive efficiency have appeared in the literature; we follow the discussion of [10]. Let $W \subseteq \mathbb{Z}^p$ be a finite set to be enumerated. An enumeration algorithm is said to run in *polynomial total time* if its running time is bounded by a polynomial in the encoding size of the input and the output. A stronger notion is that of *incremental polynomial time*: Such an algorithm receives a list of solutions $\mathbf{w}_1, \dots, \mathbf{w}_N \in W$ as an additional input. In polynomial time, it outputs one solution $\mathbf{w} \in W \setminus \{\mathbf{w}_1, \dots, \mathbf{w}_N\}$ or asserts that there are no more solutions. An even stronger notion is that of a *polynomial-delay* algorithm, which takes only polynomial time (in the encoding size of the input) before the first solution is output, between successive outputs of solutions, and after the last solution is output to the termination of the algorithm. Since the algorithm could take exponential time to output all solutions, it could also build exponential-size data structures in the course of the enumeration. This observation gives rise to an even stronger notion of efficiency, a *polynomial-space polynomial-delay* enumeration algorithm.

We also wish to prescribe an *order*, like the lexicographic order, in which the elements are to be enumerated. We consider term orders $\prec_R$ on monomials $\mathbf{y}^{\mathbf{w}}$ that are defined as in [11] by a non-negative integral $p \times p$-matrix R of full rank. Two monomials satisfy $\mathbf{y}^{\mathbf{w}_1} \prec_R \mathbf{y}^{\mathbf{w}_2}$ if and only if $R\mathbf{w}_1$ is lexicographically smaller than $R\mathbf{w}_2$. In other words, if $\mathbf{r}_1, \dots, \mathbf{r}_n$ denote the rows of R, there is some $j \in \{1, \dots, n\}$ such that $\langle \mathbf{r}_i, \mathbf{w}_1 \rangle = \langle \mathbf{r}_i, \mathbf{w}_2 \rangle$ for $i < j$, and $\langle \mathbf{r}_j, \mathbf{w}_1 \rangle < \langle \mathbf{r}_j, \mathbf{w}_2 \rangle$. For example, the unit matrix $R = I_n$ describes the lexicographic term ordering.

We prove the existence of a polynomial-space polynomial-delay prescribed-order enumeration algorithm in a general setting, where the set W to be enumerated is given as the projection of a set presented by a rational generating function.

Theorem 7. *Let the dimension k and the maximum number ℓ of binomials in the denominator be fixed.*

Let $V \subseteq \mathbb{Z}^k$ be a bounded set of lattice points with $V \subseteq [-M, M]^k$, given only by the bound $M \in \mathbb{Z}_+$ and its rational generating function encoding $g(V; \mathbf{z})$ with at most ℓ binomials in each denominator. Let

$$W = \{ \mathbf{w} \in \mathbb{Z}^p : \exists \mathbf{t} \in \mathbb{Z}^{k-p} \text{ such that } (\mathbf{t}, \mathbf{w}) \in V \}$$

denote the projection of V onto the last p components. Let $\prec_R$ be the term order on monomials in $y_1, \dots, y_p$ induced by a given matrix $R \in \mathbb{N}^{p \times p}$.

There exists a polynomial-space polynomial-delay enumeration algorithm for the points in the projection W, which outputs the points of W in the *order given by $\prec_R$. The algorithm can be implemented without using the Projection Lemma.*

We remark that Theorem 7 is a stronger result than what can be obtained by the repeated application of the monomial-extraction technique of Lemma 7 from [4], which would only give an incremental polynomial time enumeration algorithm.

Proof. We give a simple recursive algorithm that is based on the iterative bisection of intervals.

Input: Lower and upper bound vectors $\mathbf{l}, \mathbf{u} \in \mathbb{Z}^p$.
Output: All vectors $\mathbf{w}$ in W with $\mathbf{l} \leq R\mathbf{w} \leq \mathbf{u}$, sorted in the order $\preceq_R$.

1. If the set $W \cap \{\, \mathbf{w} : \mathbf{l} \leq R\mathbf{w} \leq \mathbf{u} \,\}$ is empty, do nothing.
2. Otherwise, if $\mathbf{l} = \mathbf{u}$, compute the unique point $\mathbf{w} \in \mathbb{Z}^k$ with $R\mathbf{w} = \mathbf{l} = \mathbf{u}$ and output $\mathbf{w}$.
3. Otherwise, let j be the smallest index with $l_j \neq u_j$. We bisect the integer interval $\{l_j, \dots, u_j\}$ evenly into $\{l_j, \dots, m_j\}$ and $\{m_j + 1, \dots, u_j\}$, where $m_j = \left\lfloor \frac{l_j + u_j}{2} \right\rfloor$. We invoke the algorithm recursively on the first part, then on the second part, using the corresponding lower and upper bound vectors.

We first need to compute appropriate lower and upper bound vectors $\mathbf{l}, \mathbf{u}$ to start the algorithm. To this end, let N be the largest number in the matrix R and let $\mathbf{l} = -pMN\mathbf{1}$ and $\mathbf{u} = pMN\mathbf{1}$. Then $\mathbf{l} \leq R\mathbf{w} \leq \mathbf{u}$ holds for all $\mathbf{w} \in W$. Clearly the encoding length of $\mathbf{l}$ and $\mathbf{u}$ is bounded polynomially in the input data.

In step 1 of the algorithm, to determine whether

$$W \cap \{\, \mathbf{w} : \mathbf{l} \leq R\mathbf{w} \leq \mathbf{u} \,\} = \emptyset, \tag{12}$$

we consider the polytope

$$Q_{\mathbf{l},\mathbf{u}} = [-M, M]^{k-p} \times \{\, \mathbf{w} \in \mathbb{R}^p : \mathbf{l} \leq R\mathbf{w} \leq \mathbf{u} \,\} \subseteq \mathbb{R}^k, \tag{13}$$

a parallelelepiped in $\mathbb{R}^k$. Since W is the projection of V and since $V \subseteq [-M, M]^k$, we have (12) if and only if $V \cap Q_{\mathbf{l},\mathbf{u}} = \emptyset$. The rational generating function $g(Q_{\mathbf{l},\mathbf{u}}; \mathbf{z})$ can be computed in polynomial time. By using the Intersection Lemma, we can compute the rational generating function $g(V \cap Q_{\mathbf{l},\mathbf{u}}; \mathbf{z})$ in polynomial time. The specialization $g(V \cap Q_{\mathbf{l},\mathbf{u}}; \mathbf{z} = \mathbf{1})$ can also be computed in polynomial time. It gives the number of lattice points in $V \cap Q_{\mathbf{l},\mathbf{u}}$; in particular, we can decide whether $V \cap Q_{\mathbf{l},\mathbf{u}} = \emptyset$.

It is clear that the algorithm outputs the elements of W in the order given by $\prec_R$. We next show that the algorithm is a polynomial-space polynomial-delay enumeration algorithm. The subproblem in step 1 only depends on the input data as stated in the theorem and on the vectors $\mathbf{l}$ and $\mathbf{u}$, whose encoding length only decreases in recursive invocations. Therefore each of the subproblems can be solved in polynomial time (thus also in polynomial space).

The recursion of the algorithm corresponds to a binary tree whose nodes are labeled by the bound vectors $\mathbf{l}$ and $\mathbf{u}$. There are two types of leaves in the tree, one corresponding to the "empty-box" situation (12) in step 1, and one corresponding to the "solution-output" situation in step 2. Inner nodes of the tree correspond to the recursive invocation of the algorithm in step 3. It is clear that the depth of the recursion is $\mathrm{O}(p\log(pMN))$, because the integer intervals are bisected evenly. Thus the stack space of the algorithm is polynomially bounded. Since the algorithm does not maintain any global data structures, the whole algorithm uses polynomial space only.

Let $\mathbf{w}_i \in W$ be an arbitrary solution and let $\mathbf{w}_{i+1}$ be its direct successor in the order $\prec_R$. We shall show that the algorithm only spends polynomial time between the output of $\mathbf{w}_i$ and the output of $\mathbf{w}_{i+1}$. The key property of the recursion tree of the algorithm is the following:

> Every inner node is the root of a subtree that contains at least one solution-output leaf. (14)

The reason for that property is the test for situation (12) in step 1 of the algorithm. Therefore, the algorithm can visit only $\mathrm{O}(p\log(pMN))$ inner nodes and empty-box leaves between the solution-output leaves for $\mathbf{w}_i$ and $\mathbf{w}_{i+1}$. For the same reason, also the time before the first solution is output and the time after the last solution is output are polynomially bounded. □

The following corollary, which is a stronger formulation of Theorem 1 (iii), is immediate.

Corollary 8. *Let n and k be fixed integers. There exist polynomial-space polynomial-delay enumeration algorithms to enumerate the set of Pareto optima of the multicriterion integer linear program* (1)*, the set of Pareto strategies, or arbitrary projections thereof in lexicographic order (or an arbitrary term order).*

Remark 9. We remark that Theorem 7 is of general interest. For instance, it also implies the existence of a polynomial-space polynomial-delay prescribed-order enumeration algorithm for Hilbert bases of rational polyhedral cones in fixed dimension.

Indeed, fix the dimension d and let $C = \operatorname{cone}\{\mathbf{b}_1,\dots,\mathbf{b}_n\} \subseteq \mathbb{R}^d$ be a pointed rational polyhedral cone. The *Hilbert basis* of C is defined as the inclusion-minimal set $H \subseteq C \cap \mathbb{Z}^d$ which generates $C \cap \mathbb{Z}^d$ as a monoid. For *simplicial* cones C (where $\mathbf{b}_1,\dots,\mathbf{b}_n$ are linearly independent), Barvinok and Woods [3] proved that one can compute the rational generating function $g(H;\mathbf{z})$ (having a constant number of binomials in the denominators) of the Hilbert basis of $C \cap \mathbb{Z}^d$ using the Projection Theorem. The same technique works for non-simplicial pointed cones. Now Theorem 7 gives a polynomial-space polynomial-delay prescribed-order enumeration algorithm.

4. Selecting a Pareto optimum using global criteria

Now that we know that all Pareto optima of a multicriteria integer linear programs can be encoded in a rational generating function, and that they can be listed efficiently on the output size, we can aim to apply selection criteria stated by a user. The advantage of our setup is that when we optimize a global objective function it guarantees to return a Pareto optimum, because we evaluate the global criterion only on the Pareto optima. Let us start with the simplest global criterion which generalizes the use of the ℓ_1 norm distance function:

Theorem 10. *Let the dimension k and the maximum number ℓ of binomials in the denominator be fixed.*

Let $V \subseteq \mathbb{Z}^k$ be a bounded set of lattice points with $V \subseteq [-M, M]^{n+k}$, given only by the bound $M \in \mathbb{Z}_+$ and its rational generating function encoding $g(V; \mathbf{z})$ with at most ℓ binomials in the denominators.

Let $Q \subseteq \mathbb{R}^k$ be a rational convex central-symmetric polytope with $\mathbf{0} \in \operatorname{int} Q$, given by its vertex or inequality description. Let the polyhedral norm $\|\cdot\|_Q$ be defined using the Minkowski functional

$$\|\mathbf{y}\|_Q = \inf\{\, \lambda \geq 0 : \mathbf{y} \in \lambda Q \,\}. \tag{15}$$

Finally, let a prescribed point $\hat{\mathbf{v}} \in \mathbb{Z}^k$ be given.

(i) *There exists a polynomial-time algorithm to find a point $\mathbf{v} \in V$ that minimizes the distance $d_Q(\mathbf{v}, \hat{\mathbf{v}}) = \|\mathbf{v} - \hat{\mathbf{v}}\|_Q$ from the prescribed point.*

(ii) *There exists a polynomial-space polynomial-delay enumeration algorithm for enumerating the points of V in the order of increasing distances d_Q from the prescribed point $\hat{\mathbf{v}}$, refined by an arbitrary term order $\prec_R$ given by a matrix $R \in \mathbb{N}^{k\times k}$.*

Theorem 2, as stated in the introduction, is an immediate corollary of this theorem.

Proof. Since the dimension k is fixed, we can compute an inequality description

$$Q = \{\, \mathbf{y} \in \mathbb{R}^k : A\mathbf{y} \leq \mathbf{b} \,\}$$

of Q with $A \in \mathbb{Z}^{m\times k}$ and $\mathbf{b} \in \mathbb{Z}^k$ in polynomial time, if Q is not already given by an inequality description. Let $\mathbf{v} \in V$ be arbitrary; then

$$\begin{aligned} d_Q(\hat{\mathbf{v}}, \mathbf{v}) &= \|\mathbf{v} - \hat{\mathbf{v}}\|_Q \\ &= \inf\big\{\, \lambda \geq 0 : \mathbf{v} - \hat{\mathbf{v}} \in \lambda Q \,\big\} \\ &= \min\big\{\, \lambda \geq 0 : \lambda \mathbf{b} \geq A(\mathbf{v} - \hat{\mathbf{v}}) \,\big\}. \end{aligned}$$

Thus there exists an index $i \in \{1, \dots, m\}$ such that

$$d_Q(\hat{\mathbf{v}}, \mathbf{v}) = \frac{(A\mathbf{v})_i - (A\hat{\mathbf{v}})_i}{b_i};$$

so $d_Q(\hat{\mathbf{v}}, \mathbf{v})$ is an integer multiple of $1/b_i$. Hence for every $\mathbf{v} \in V$, we have that

$$d_Q(\hat{\mathbf{v}}, \mathbf{v}) \in \frac{1}{\operatorname{lcm}(b_1, \dots, b_m)} \mathbb{Z}_+, \tag{16}$$

where $\operatorname{lcm}(b_1, \dots, b_m)$ clearly is a number of polynomial encoding size. On the other hand, every $\mathbf{v} \in V$ certainly satisfies

$$d_Q(\hat{\mathbf{v}}, \mathbf{v}) \le ka\bigl(M + \max\{|\hat{v}_1|, \dots, |\hat{v}_d|\}\bigr) \tag{17}$$

where a is the largest number in A, which is also a bound of polynomial encoding size.

Using Barvinok's algorithm, we can compute the rational generating function $g(\hat{\mathbf{v}} + \lambda Q; \mathbf{z})$ for any rational λ of polynomial enoding size in polynomial time. We can also compute the rational generating function $g(V \cap (\hat{\mathbf{v}} + \lambda Q); \mathbf{z})$ using the Intersection Lemma. By computing the specialization $g(V \cap (\hat{\mathbf{v}} + \lambda Q); \mathbf{z} = \mathbf{1})$, we can compute the number of points in $V \cap (\hat{\mathbf{v}} + \lambda Q)$, thus we can decide whether this set is empty or not.

Hence we can employ binary search for the smallest $\lambda \ge 0$ such that $V \cap (\hat{\mathbf{v}} + \lambda Q)$ is nonempty. Because of (16) and (17), it runs in polynomial time. By using the recursive bisection algorithm of Theorem 7, it is then possible to construct one Pareto optimum in $V \cap (\hat{\mathbf{v}} + \lambda Q)$ for part (i), or to construct a sequence of Pareto optima in the desired order for part (ii). □

Now we consider a global criterion using a distance function corresponding to a non-polyhedral norm like the Euclidean norm $\|\cdot\|_2$ (or any other ℓ_p-norm for $1 < p < \infty$). We are able to prove a very strong type of approximation result, a so-called fully polynomial-time approximation scheme (FPTAS), in a somewhat more general setting.

Definition 11 (FPTAS)**.** Consider the optimization problems

$$\max\{\, f(\mathbf{v}) : \mathbf{v} \in V \,\}, \tag{18a}$$
$$\min\{\, f(\mathbf{v}) : \mathbf{v} \in V \,\}. \tag{18b}$$

A *fully polynomial-time approximation scheme (FPTAS)* for the maximization problem (18a) or the minimization problem (18b), respectively, is a family $\{\, \mathcal{A}_\epsilon : \epsilon \in \mathbb{Q},\ \epsilon > 0 \,\}$ of approximation algorithms $\mathcal{A}_\epsilon$, each of which returns an *ϵ-approximation*, i.e., a solution $\mathbf{v}_\epsilon \in V$ with

$$f(\mathbf{v}_\epsilon) \ge (1 - \epsilon) f^* \quad \text{where} \quad f^* = \max_{\mathbf{v} \in V} f(\mathbf{v}), \tag{19a}$$

or, respectively,

$$f(\mathbf{v}_\epsilon) \le (1 + \epsilon) f^* \quad \text{where} \quad f^* = \min_{\mathbf{v} \in V} f(\mathbf{v}), \tag{19b}$$

such that the algorithms $\mathcal{A}_\epsilon$ run in time polynomial in the input size and $\frac{1}{\epsilon}$.

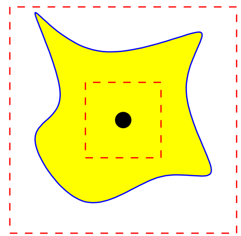

FIGURE 1. A set defining a pseudo-norm with the inscribed and circumscribed cubes αB_∞ and βB_∞ (dashed).

Remark 12. An FPTAS is based on the notion of ϵ-approximation (19), which gives an approximation guarantee relative to the value f^* of an optimal solution. It is clear that this notion is most useful for objective functions f that are non-negative on the feasible region V. Since the approximation quality of a solution changes when the objective function is changed by an additive constant, it is non-trivial to convert an FPTAS for a maximization problem to an FPTAS for a minimization problem.

We shall present an FPTAS for the problem of minimizing the distance of a Pareto optimum from a prescribed outcome vector $\hat{\mathbf{v}} \in \mathbb{Z}^k$. We consider distances $d(\hat{\mathbf{v}}, \cdot)$ induced by a pseudo-norm $\|\cdot\|_Q$ via

$$d(\hat{\mathbf{v}}, \mathbf{v}) = \|\mathbf{v} - \hat{\mathbf{v}}\|_Q \tag{20a}$$

To this end, let $Q \subseteq \mathbb{R}^k$ be a compact basic semialgebraic set with $\mathbf{0} \in \operatorname{int} Q$, which is described by one polynomial inequality,

$$Q = \{\, \mathbf{y} \in \mathbb{R}^k : q(\mathbf{y}) \leq 1 \,\}, \tag{20b}$$

where $q \in \mathbb{Q}[y_1, \dots, y_k]$ is a homogeneous polynomial of (even) degree D. The pseudo-norm $\|\cdot\|_Q$ is now defined using the Minkowski functional

$$\|\mathbf{y}\|_Q = \inf\{\, \lambda \geq 0 : \mathbf{y} \in \lambda Q \,\} \tag{20c}$$

Note that we do not make any assumptions of convexity of Q, which would make $\|\cdot\|_Q$ a norm. Since Q is compact and $\mathbf{0} \in \operatorname{int} Q$, there exist positive rational numbers (norm equivalence constants) α, β with

$$\alpha B_\infty \subseteq Q \subseteq \beta B_\infty \quad \text{where} \quad B_\infty = \{\, \mathbf{y} \in \mathbb{R}^k : \|\mathbf{y}\|_\infty \leq 1 \,\}; \tag{21}$$

see Figure 1.

Now we can formulate our main theorem, which has Theorem 3, which we stated in the introduction, as an immediate corollary.

Theorem 13. *Let the dimension n and the number k of objective functions be fixed. Moreover, let a degree D and two rational numbers $0 < \alpha \leq \beta$ be fixed. Then there exists a fully polynomial-time approximation scheme for the problem of minimizing the distance $d_Q(\hat{\mathbf{v}}, \mathbf{v})$, defined via (20) by a homogeneous polynomial $q \in \mathbb{Q}[y_1, \dots, y_k]$ of degree D satisfying (21), whose* *coefficients are encoded in binary and whose exponent vectors are encoded in unary, of a Pareto optimum of* (1) *from a prescribed outcome vector* $\hat{\mathbf{v}} \in \mathbb{Z}^k$.

The proof is based on the following result, which is a more general formulation of Theorem 1.1 from [5].

Theorem 14 (FPTAS for maximizing non-negative polynomials over finite lattice point sets)**.** *For all fixed integers k (dimension) and s (maximum number of binomials in the denominator), there exists an algorithm with running time polynomial in the encoding size of the problem and $\frac{1}{\epsilon}$ for the following problem.*

Input: *Let $V \subseteq \mathbb{Z}^k$ be a finite set, given by a rational generating function in the form*

$$g(V; \mathbf{x}) = \sum_{i \in I} \gamma_i \frac{\mathbf{x}^{\mathbf{c}_i}}{(1 - \mathbf{x}^{\mathbf{d}_{i1}}) \dots (1 - \mathbf{x}^{\mathbf{d}_{is_i}})}$$

where the the numbers s_i of binomials in the denominators are at most s. Furthermore, let two vectors $\mathbf{v}_{\mathrm{L}}$, $\mathbf{v}_{\mathrm{U}} \in \mathbb{Z}^k$ be given such that V is contained in the box $\{\, \mathbf{v} : \mathbf{v}_{\mathrm{L}} \leq \mathbf{v} \leq \mathbf{v}_{\mathrm{U}} \,\}$.

Let $f \in \mathbb{Q}[v_1, \dots, v_k]$ be a polynomial with rational coefficients that is non-negative on V, given by a list of its monomials, whose coefficients are encoded in binary and whose exponents are encoded in unary.

Finally, let $\epsilon \in \mathbb{Q}$.

Output: *Compute a point $\mathbf{v}_\epsilon \in V$ that satisfies*

$$f(\mathbf{v}_\epsilon) \geq (1 - \epsilon) f^* \quad \textit{where} \quad f^* = \max_{\mathbf{v} \in V} f(\mathbf{v}).$$

In [5] the above result was stated and proved only for sets V that consist of the lattice points of a rational polytope; however, the same proof yields the result above.

Proof of Theorem 13. Using Theorem 1, we first compute the rational generating function $g(V^{\mathrm{Pareto}}; \mathbf{z})$ of the Pareto optima. With binary search using the Intersection Lemma with generating functions of cubes as in section 3, we can find the smallest non-negative integer γ such that

$$(\hat{\mathbf{v}} + \gamma B_\infty) \cap V^{\mathrm{Pareto}} \neq \emptyset. \tag{22}$$

If $\gamma = 0$, then the prescribed outcome vector $\hat{\mathbf{v}}$ itself is a Pareto optimum, so it is the optimal solution to the problem.

Otherwise, let $\mathbf{v}_0$ be an arbitrary outcome vector in $(\hat{\mathbf{v}} + \gamma B_\infty) \cap V^{\mathrm{Pareto}}$. Then

$$\begin{aligned} \gamma \geq \|\mathbf{v}_0 - \hat{\mathbf{v}}\|_\infty &= \inf\{\, \lambda : \mathbf{v}_0 - \hat{\mathbf{v}} \in \lambda B_\infty \,\} \\ &\geq \inf\{\, \lambda : \mathbf{v}_0 - \hat{\mathbf{v}} \in \lambda \tfrac{1}{\alpha} Q \,\} = \alpha \|\mathbf{v}_0 - \hat{\mathbf{v}}\|_Q , \end{aligned}$$

thus $\|\mathbf{v}_0 - \hat{\mathbf{v}}\|_Q \leq \gamma/\alpha$. Let $\delta = \beta\gamma/\alpha$. Then, for every $\mathbf{v}_1 \in \mathbb{R}^k$ with $\|\mathbf{v}_1 - \hat{\mathbf{v}}\|_\infty \geq \delta$ we have

$$\begin{aligned}\delta \leq \|\mathbf{v}_1 - \hat{\mathbf{v}}\|_\infty &= \inf\{\, \lambda : \mathbf{v}_1 - \hat{\mathbf{v}} \in \lambda B_\infty \,\} \\ &\leq \inf\{\, \lambda : \mathbf{v}_1 - \hat{\mathbf{v}} \in \lambda \tfrac{1}{\beta} Q \,\} = \beta \|\mathbf{v}_1 - \hat{\mathbf{v}}\|_Q \,,\end{aligned}$$

thus

$$\|\mathbf{v}_1 - \hat{\mathbf{v}}\|_Q \geq \delta/\beta = \gamma/\alpha \geq \|\mathbf{v}_0 - \hat{\mathbf{v}}\|_Q \,.$$

Therefore, a Pareto optimum $\mathbf{v}^* \in V^{\text{Pareto}}$ minimizing the distance d_Q from the prescribed outcome vector $\hat{\mathbf{v}}$ is contained in the cube $\hat{\mathbf{v}}+\delta B_\infty$. Moreover, for all points $\mathbf{v} \in \hat{\mathbf{v}} + \delta B_\infty$ we have

$$\|\mathbf{v}_0 - \hat{\mathbf{v}}\|_Q \leq \delta/\alpha = \beta\gamma/\alpha^2.$$

We define a function f by

$$f(\mathbf{v}) = \left(\beta\gamma/\alpha^2\right)^D - \|\mathbf{v} - \hat{\mathbf{v}}\|_Q^D \,, \tag{23}$$

which is non-negative over the cube $\hat{\mathbf{v}} + \delta B_\infty$. Since q is a homogeneous polynomial of degree D, we obtain

$$f(\mathbf{v}) = \left(\beta\gamma/\alpha^2\right)^D - q(\mathbf{v} - \hat{\mathbf{v}}) \tag{24}$$

so f is a polynomial.

We next compute the rational generating function

$$g(V^{\text{Pareto}} \cap (\hat{\mathbf{v}} + \delta B_\infty); \mathbf{z})$$

from $g(V^{\text{Pareto}}; \mathbf{z})$ using the Intersection Lemma. Let $\epsilon' > 0$ be a rational number, which we will determine later. By Theorem 14, we compute a solution $\mathbf{v}_{\epsilon'} \in V^{\text{Pareto}}$ with

$$f(\mathbf{v}_{\epsilon'}) \geq (1 - \epsilon') f(\mathbf{v}^*),$$

or, equivalently,

$$f(\mathbf{v}^*) - f(\mathbf{v}_{\epsilon'}) \leq \epsilon' \, f(\mathbf{v}^*).$$

Thus,

$$\begin{aligned}[d_Q(\hat{\mathbf{v}}, \mathbf{v}_{\epsilon'})]^D - [d_Q(\hat{\mathbf{v}}, \mathbf{v}^*)]^D &= \|\mathbf{v}_{\epsilon'} - \hat{\mathbf{v}}\|_Q^D - \|\mathbf{v}^* - \hat{\mathbf{v}}\|_Q^D \\ &= f(\mathbf{v}^*) - f(\mathbf{v}_{\epsilon'}) \\ &\leq \epsilon' \, f(\mathbf{v}^*) \\ &= \epsilon' \, \left(\left(\beta\gamma/\alpha^2\right)^D - \|\mathbf{v}^* - \hat{\mathbf{v}}\|_Q^D \right).\end{aligned}$$

Since γ is the smallest integer with (22) and also $\|\mathbf{v}^* - \hat{\mathbf{v}}\|_\infty$ is an integer, we have

$$\gamma \leq \|\mathbf{v}^* - \hat{\mathbf{v}}\|_\infty \leq \beta \|\mathbf{v}^* - \hat{\mathbf{v}}\|_Q \,.$$

Thus,

$$[d_Q(\hat{\mathbf{v}}, \mathbf{v}_{\epsilon'})]^D - [d_Q(\hat{\mathbf{v}}, \mathbf{v}^*)]^D \leq \epsilon' \left[\left(\frac{\beta}{\alpha}\right)^{2D} - 1 \right] \|\mathbf{v}^* - \hat{\mathbf{v}}\|_Q^D \,.$$

An elementary calculation yields

$$d_Q(\hat{\mathbf{v}}, \mathbf{v}_{\epsilon'}) - d_Q(\hat{\mathbf{v}}, \mathbf{v}^*) \le \frac{\epsilon'}{D} \left[\left(\frac{\beta}{\alpha} \right)^{2D} - 1 \right] d_Q(\hat{\mathbf{v}}, \mathbf{v}^*).$$

Thus we can choose

$$\epsilon' = \epsilon\, D \left[\left(\frac{\beta}{\alpha} \right)^{2D} - 1 \right]^{-1} \tag{25}$$

to get the desired estimate. Since α, β and D are fixed constants, we have $\epsilon' = \Theta(\epsilon)$. Thus the computation of $\mathbf{v}_{\epsilon'} \in V^{\mathrm{Pareto}}$ by Theorem 14 runs in time polynomial in the input encoding size and $\frac{1}{\epsilon}$. □

Remark 15. It is straightforward to extend this result to also include the ℓ_p norms for *odd* integers p, by solving the approximation problem separately for all of the $2^k = \mathrm{O}(1)$ shifted orthants $\hat{\mathbf{v}} + O_{\boldsymbol{\sigma}} = \{\, \mathbf{v} : \sigma_i(v_i - \hat{v}_i) \ge 0 \,\}$, where $\boldsymbol{\sigma} \in \{\pm 1\}^k$. On each of the orthants, the ℓ_p-norm has a representation by a polynomial as required by Theorem 13.

References

[1] A. I. Barvinok. Polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Mathematics of Operations Research*, 19:769–779, 1994.

[2] A. I. Barvinok and J. E. Pommersheim. An algorithmic theory of lattice points in polyhedra. In *New Perspectives in Algebraic Combinatorics*, volume 38 of *Math. Sci. Res. Inst. Publ.*, pages 91–147. Cambridge Univ. Press, Cambridge, 1999.

[3] A. I. Barvinok and K. Woods. Short rational generating functions for lattice point problems. *Journal of the AMS*, 16(4):957–979, 2003.

[4] J. A. De Loera, D. Haws, R. Hemmecke, P. Huggins, B. Sturmfels, and R. Yoshida. Short rational functions for toric algebra and applications. *Journal of Symbolic Computation*, 38(2):959–973, 2004.

[5] J. A. De Loera, R. Hemmecke, M. Köppe, and R. Weismantel. Integer polynomial optimization in fixed dimension. *Mathematics of Operations Research*, 31(1):147–153, 2006.

[6] M. Ehrgott and X. Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spektrum*, 22:425–460, 2000.

[7] V. A. Emelichev and V. A. Perepelitsa. On the cardinality of the set of alternatives in discrete many-criterion problems. *Discrete Mathematics and Applications*, 2:461–471, 1992.

[8] J. Figueira, S. Greco, and M. Ehrgott, editors. *Multiple Criteria Decision Analysis. State of the Art Surveys*. Springer, 2005.

[9] H. Isermann. Proper efficiency and the linear vector maximum problem. *Operations Research*, 22:189–191, 1974.

[10] D. S. Johnson, M. Yannakakis, and Ch. H. Papadimitriou. On generating all maximal independent sets. *Information Processing Lettes*, 27:119–123, 1988.

[11] T. Mora and L. Robbiano. The Gröbner fan of an ideal. *Journal of Symbolic Computation*, 6(2–3):183–208, 1988.

[12] Y. Sawaragi, H. Nakayama, and T. Tanino, editors. *Theory of Multiobjective Optimization*. Academic Press, 1985.

[13] I. V. Sergienko and V. A. Perepelitsa. Finding the set of alternatives in discrete multi-criterion problems. *Cybernetics*, 3:673–683, 1991.

J. A. De Loera: University of California, Davis, Department of Mathematics, One Shields Avenue, Davis CA 95616, USA

E-mail address: `deloera@math.ucdavis.edu`

R. Hemmecke: Otto-von-Guericke-Universität Magdeburg, Department of Mathematics, Institute for Mathematical Optimization (IMO), Universitätsplatz 2, 39106 Magdeburg, Germany

E-mail address: `hemmecke@imo.math.uni-magdeburg.de`

M. Köppe: Otto-von-Guericke-Universität Magdeburg, Department of Mathematics, Institute for Mathematical Optimization (IMO), Universitätsplatz 2, 39106 Magdeburg, Germany

E-mail address: `mkoeppe@imo.math.uni-magdeburg.de`