GEOMETRY AND COMPLEXITY OF O'HARA'S ALGORITHM

MATJAŽ KONVALINKA AND IGOR PAK

ABSTRACT. In this paper we analyze O'Hara's partition bijection. We present three type of results. First, we show that O'Hara's bijection can be viewed geometrically as a certain scissor congruence type result. Second, we obtain a number of new complexity bounds, proving that O'Hara's bijection is efficient in several special cases and mildly exponential in general. Finally, we prove that for identities with finite support, the map of the O'Hara's bijection can be computed in polynomial time, i.e. much more efficiently than by O'Hara's construction.

1. INTRODUCTION

Ever since the pioneer work by Sylvester and his school, there has been a quest to find bijective proofs of many interesting partition identities. Despite remarkable successes in the last century (see [P3]) and some recent work of both positive and negative nature (see e.g. [P2, P4]), the problem remains ambiguous and largely unresolved. Much of this stems from the lack of clarity as to what exactly constitutes a bijective proof. Depending on whether one accentuates simplicity, ability to generalize, the time complexity, geometric structure, or asymptotic stability, different answers tend to emerge.

In one direction, the subject of partition bijections was revolutionized by Garsia and Milne with their *involution principle* [GM1, GM2]. This is a combinatorial construction which allows to use a few basic bijections and involutions to build more involved combinatorial maps. As a consequence, one can start with a reasonable analytic proof of a partition identity and trace every step to obtain a (possibly extremely complicated) bijective construction. Garsia and Milne used this route to obtain a long sought bijection proving the Rogers-Ramanujan identities, resolving an old problem in this sense [GM2]. Unfortunately, this bijection is too complex to be analyzed and has yet to lead to new Rogers-Ramanujan type partition identities.

After Garsia-Milne paper, there has been a flurry of activity to obtain synthetic bijections for large classes of partition identities. Most of these bijections did not seem to lead anywhere with one notable exception. Remmel and Gordon found (rather involved) bijective

---

proofs of basic partition identities due to Andrews [R, G]. The latter are direct extensions of Euler's distinct/odd theorem and have a similar straightforward analytic proof [A, P3]. Then O'Hara made a surprising discovery that Remmel's and Gordon's bijections can be streamlined to give the same bijective map with a simple construction [O1, O2]. In fact, O'Hara proved that the resulting bijection is a direct generalization of Glaisher's classical bijection proving Euler's theorem. Moreover, in her thesis [O1], O'Hara showed that her bijection is computationally efficient in certain special cases. Until now, the reason why O'Hara's bijection has a number of nice properties distinguishing it from the other "involution principle bijections" remained mysterious.

In this paper we obtain results of both positive and negative type. First, we analyze the complexity of O'Hara's bijection, which we view as a discrete algorithm. We prove a general result (Theorem 3.2), which given an *exact* formula for the number of steps of the algorithm in certain cases. From here we show that O'Hara's bijection is computationally efficient in many special cases. On the other hand, perhaps surprisingly, we prove that the number of steps can be (mildly) exponential in the worst case (Theorem 3.7 part (3)). In fact, even when the natural speed-up is applied, the worst complexity does not improve significantly (see Section 5). This is the first negative result of this kind, proving the analogue of a conjecture that remains open for the Garsia-Milne's "Rogers-Ramanujan bijection" (see Subsection 6.1).

Second, we show that O'Hara's bijection has a rich underlying geometry. In a manner similar to that in [P1, PV], we view this bijection as a map between integer points in polytopes which preserves certain linear functionals. We present an advanced generalization of Andrews's result and of O'Hara's bijection in this geometric setting. In a special case, the working of the map corresponds to the Euclid algorithm and, more generally, to terms in the continuing fractions. Thus one can also think of our generalization as a version of multidimensional continuing fractions.

Finally, by combining the geometric and complexity ideas we show that in the finite dimensional case the map defined by O'Hara's bijection is a solution of an integer linear programming problem. This implies that the map defined by the bijection can be computed in polynomial time, i.e. much more efficiently than by O'Hara's bijection. This suggests that perhaps in the general case O'Hara's bijection can also be sped up to work in polynomial time (see Subsection 3.3).

The paper is structured as follows. We start with definitions and notations in Section 2. In Section 3, we describe the main results on both geometry and complexity. Proofs of most results are postponed until Section 4. We conclude with a quick application in Section 5 and final remarks in Section 6.

2. DEFINITIONS AND BACKGROUND

2.1. **Andrews's theorem.** A *partition* λ is an integer sequence $(\lambda_1, \lambda_2, \ldots, \lambda_\ell)$ such that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_\ell > 0$, where the integers λ_i are called the *parts* of the partition. The sum $n = \sum_{i=1}^{\ell} \lambda_i$ is called the *size* of λ, denoted $|\lambda|$; in this case we say that λ is a partition of n, and write $\lambda \vdash n$. We can also write $\lambda = 1^{m_1} 2^{m_2} \cdots$, where $m_i = m_i(\lambda)$ is the number of

parts of λ equal to i. The *support* of $\lambda = 1^{m_1} 2^{m_2} \cdots$ is the set $\{i : m_i > 0\}$. The set of all positive integers will be denoted by $\mathbb{P}$.

Denote the set of all partitions by $\mathcal{P}$ and the set of all partitions of n by $\mathcal{P}_n$. The number of partitions of n is given by Euler's formula

$$\sum_{\lambda \in \mathcal{P}} t^{|\lambda|} = \sum_{n=0}^{\infty} |\mathcal{P}_n| t^n = \prod_{i=1}^{\infty} \frac{1}{1 - t^i}.$$

For a sequence $\bar{a} = (a_1, a_2, \ldots)$ with $a_i \in \mathbb{P} \cup \{\infty\}$, define $\mathcal{A}$ to be the set of partitions λ with $m_i(\lambda) < a_i$ for all i; write $\mathcal{A}_n = \mathcal{A} \cap \mathcal{P}_n$. Denote by $\mathrm{supp}(\bar{a}) = \{i : a_i < \infty\}$ the *support* of the sequence a.

Let $\bar{a} = (a_1, a_2, \ldots)$ and $\bar{b} = (b_1, b_2, \ldots)$. We say that $\bar{a}$ and $\bar{b}$ are φ-*equivalent*, $\bar{a} \sim_\varphi \bar{b}$, if φ is a bijection $\mathrm{supp}(\bar{a}) \to \mathrm{supp}(\bar{b})$ such that $ia_i = \varphi(i) b_{\varphi(i)}$ for all i. If $\bar{a} \sim_\varphi \bar{b}$ for some φ, we say that $\bar{a}$ and $\bar{b}$ are equivalent, and write $\bar{a} \sim \bar{b}$.

Theorem 2.1 (Andrews) *If $\bar{a} \sim \bar{b}$, then $|\mathcal{A}_n| = |\mathcal{B}_n|$ for all n.*

Proof. We use the notation $t^\infty = 0$. Clearly,

$$\sum_{n=0}^{\infty} |\mathcal{A}_n| t^n = \prod_{i=1}^{\infty} \frac{1 - t^{ia_i}}{1 - t^i} = \prod_{j=1}^{\infty} \frac{1 - t^{jb_j}}{1 - t^j} = \sum_{n=0}^{\infty} |\mathcal{B}_n| t^n,$$

which means that $|\mathcal{A}_n| = |\mathcal{B}_n|$. $\qquad\square$

Consider the classical Euler's theorem on partitions into distinct and odd parts. For $\bar{a} = (2, 2, \ldots)$ and $\bar{b} = (\infty, 1, \infty, 1, \ldots)$, $\mathcal{A}_n$ is the set of all partitions of n into distinct parts, and $\mathcal{B}_n$ is the set of partitions of n into odd parts. The bijection $i \mapsto 2i$ between $\mathrm{supp}(\bar{a}) = \mathbb{P}$ and $\mathrm{supp}(\bar{b}) = 2\mathbb{P}$ satisfies $ia_i = \varphi(i) b_{\varphi(i)}$, so $\bar{a} \sim_\varphi \bar{b}$ and $|\mathcal{A}_n| = |\mathcal{B}_n|$. Throughout the paper, we refer to this example as the *distinct/odd case*.

2.2. **O'Hara's algorithm.** The analytic proof of Andrews's theorem shown above does not give an explicit bijection $\mathcal{A}_n \to \mathcal{B}_n$. Such a bijection is, by Theorem 2.3, given by the following algorithm.

Algorithm 2.2 (O'Hara's algorithm on partitions)

    `Fix:` sequences $\bar{a} \sim_\varphi \bar{b}$
    `Input:` $\lambda \in \mathcal{A}$
    `Set:` $\mu \leftarrow \lambda$
    `While:` μ contains more than b_j copies of j for some j
        `Do:` remove b_j copies of j from μ, add a_i copies of i to μ, where $\varphi(i) = j$
    `Output:` $\psi(\lambda) \leftarrow \mu$

Theorem 2.3 (O'Hara) *Algorithm 2.2 stops after a finite number of steps. The resulting partition $\psi(\lambda) \in \mathcal{B}$ is independent of the order of the parts removed and defines a size-preserving bijection $\mathcal{A} \to \mathcal{B}$.*

Later on (see Subsection 4.2) we deduce O'Hara's theorem from our generalization (Theorem 3.2).

Denote by $L_\varphi(\lambda)$ the number of steps O'Hara's algorithm takes to compute $\psi(\lambda)$, and by $\mathcal{L}_\varphi(n)$ the maximum value of $L_\varphi(\lambda)$ over all $\lambda \vdash n$.

EXAMPLE 2.4 In the distinct/odd case, O'Hara's algorithm gives the inverse of Glaisher's bijection, which maps $\lambda = 1^{m_1} 3^{m_3} \cdots \in \mathcal{B}$ to the partition $\mu \in \mathcal{A}$ which contains $i2^j$ if and only if m_i has a 1 in the j-th position when written in binary.   □

EXAMPLE 2.5 Let $\bar{a} = (1, 1, 4, 5, 3, 1, 1, \ldots)$, $\bar{b} = (1, 1, 5, 3, 4, 1, 1, \ldots)$ and $\varphi(3) = 4$, $\varphi(4) = 5$, $\varphi(5) = 3$, $\varphi(i) = i$ for $i \neq 3, 4, 5$; observe that $\bar{a} \sim_\varphi \bar{b}$. Then O'Hara's algorithm on $\lambda = 3^3 4^4 5^2$ runs as follows:

$$\mathbf{3^3 4^4 5^2} \quad \rightarrow \quad 3^7 4^1 5^2 \quad \rightarrow \quad 3^2 4^1 5^5 \quad \rightarrow \quad 3^2 4^6 5^1 \quad \rightarrow \quad 3^6 4^3 5^1$$
$$\rightarrow \quad 3^{10} 4^0 5^1 \quad \rightarrow \quad 3^5 4^0 5^4 \quad \rightarrow \quad 3^0 4^0 5^7 \quad \rightarrow \quad 3^0 4^5 5^3 \quad \rightarrow \quad \mathbf{3^4 4^2 5^3}$$

We have $L_\varphi(\lambda) = \mathcal{L}_\varphi(35) = 9$.   □

EXAMPLE 2.6 Take $\bar{a} = (2, 2, 1, 2, 2, 1, \ldots)$ and $\bar{b} = (3, 1, 3, 1, \ldots)$. Here $\mathcal{A}$ is the set of partitions into distinct parts $\equiv \pm 1 \mod 3$, and $\mathcal{B}$ is the set of partitions into odd parts, none appearing more than twice. Define $\varphi: \mathbb{P} \to \mathbb{P}$ as follows:

$$(2.1) \qquad \varphi(i) = \begin{cases} i & \text{if } i \text{ is divisible by } 6 \\ i/3 & \text{if } i \text{ is divisible by } 3, \text{ but not by } 2 \\ 2i & \text{if } i \text{ is not divisible by } 3 \end{cases}.$$

Clearly, $\bar{a} \sim_\varphi \bar{b}$. O'Hara's algorithm on $1^1 2^1 8^1 10^1 14^1 20^1$ runs as follows:

$$\mathbf{1^1 2^1 8^1 10^1 14^1 20^1} \quad \rightarrow \quad 1^1 2^1 8^1 10^3 14^1 \quad \rightarrow \quad 1^1 2^1 7^2 8^1 10^3 \quad \rightarrow \quad 1^1 2^1 5^2 7^2 8^1 10^2$$
$$\rightarrow \quad 1^1 2^1 5^4 7^2 8^1 10^1 \quad \rightarrow \quad 1^1 2^1 5^6 7^2 8^1 \quad \rightarrow \quad 1^1 2^1 4^2 5^6 7^2 \quad \rightarrow \quad 1^1 2^3 4^1 5^6 7^2$$
$$\rightarrow \quad 1^1 2^5 5^6 7^2 \quad \rightarrow \quad 1^3 2^4 5^6 7^2 \quad \rightarrow \quad 1^5 2^3 5^6 7^2 \quad \rightarrow \quad 1^7 2^2 5^6 7^2$$
$$\rightarrow \quad 1^9 2^1 5^6 7^2 \quad \rightarrow \quad 1^{11} 5^6 7^2 \quad \rightarrow \quad 1^{11} 5^3 7^2 15^1 \quad \rightarrow \quad 1^{11} 7^2 15^2$$
$$\rightarrow \quad 1^8 3^1 7^2 15^2 \quad \rightarrow \quad 1^5 3^2 7^2 15^2 \quad \rightarrow \quad 1^2 3^3 7^2 15^2 \quad \rightarrow \quad \mathbf{1^2 7^2 9^1 15^2}$$

The bijection ψ is similar in spirit to Glaisher's bijection: given $\lambda = 1^{m_1} 2^{m_2} 4^{m_4} 5^{m_5} \cdots \in \mathcal{A}$ and $j \in \mathbb{P}$, the number of copies of part $2j - 1$ in $\psi(\lambda)$ is equal to the k-th digit in the ternary expansion of l, where k is the highest power of 3 dividing $2j - 1$, $2j - 1 = 3^k r$, and $l = \sum_i 2^i m_{r2^i}$.   □

2.3. **Equivalent sequences and graphs.** Choose equivalent sequences $\bar{a}$, $\bar{b}$. Define a directed graph G_φ on $\mathrm{supp}(\bar{a}) \cup \mathrm{supp}(\bar{b})$ by drawing an edge from i to j if $\varphi(j) = i$; an arrow from i to j therefore means that O'Hara's algorithm simultaneously removes copies of i and adds copies of j. Each vertex v has $\mathrm{indeg}\, v \leq 1$, $\mathrm{outdeg}\, v \leq 1$ and $\mathrm{indeg}\, v + \mathrm{outdeg}\, v \geq 1$. The graph splits into connected components of the following five types:

(i) cycles of length $m \geq 1$;
(ii) paths of length $m \geq 2$;
(iii) infinite paths with an ending point, but without a starting point;
(iv) infinite paths with a starting point, but without an ending point;

(v) infinite paths without a starting point or an ending point.

EXAMPLE 2.7 Figure 1 shows portions of graphs G_φ for certain φ:

(1) $\bar{a} = (1, 1, 4, 5, 3, 1, 1, \ldots)$, $\bar{b} = (1, 1, 5, 3, 4, 1, 1, \ldots)$, $\varphi(3) = 4$, $\varphi(4) = 5$, $\varphi(5) = 3$, $\varphi(i) = i$ for $i \neq 3, 4, 5$; components of G_φ are of type (i);

(2) $\bar{a} = (\infty, 1, 2, 3, \infty, \infty, \infty, \ldots)$, $\bar{b} = (2, 3, 4, \infty, \infty, \infty, \infty, \ldots)$, $\varphi(2) = 1$, $\varphi(3) = 2$, $\varphi(4) = 3$; G_φ is of type (ii);

(3) the distinct/odd case: $\bar{a} = (2, 2, \ldots)$, $\bar{b} = (\infty, 1, \infty, 1, \ldots)$, $\varphi(i) = 2i$; components of G_φ are of type (iii);

(4) the odd/distinct case: $\bar{a} = (\infty, 1, \infty, 1, \ldots)$, $\bar{b} = (2, 2, \ldots)$, $\varphi(i) = i/2$; components of G_φ are of type (iv);

(5) $\bar{a} = (2, 2, 1, 2, 2, 1, \ldots)$ and $\bar{b} = (3, 1, 3, 1, \ldots)$, φ given by (2.1); components of G_φ are of types (i) and (v). □
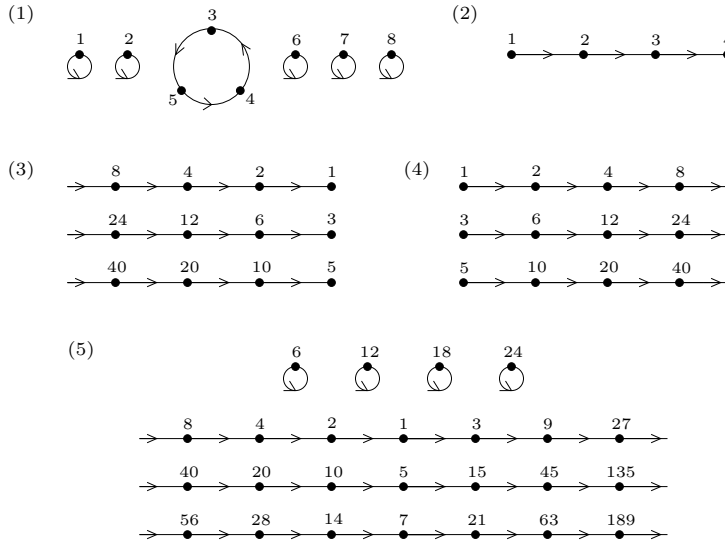

FIGURE 1. Examples of graphs G_φ.

2.4. **Scissor-congruence and Π-congruence.** We say that convex polytopes A, B in $\mathbb{R}^m$ are *congruent*, write $A \simeq B$, if B can be obtained from A by rotation and translation. For convex polytopes $P, Q \subset \mathbb{R}^m$, we say that they are *scissor-congruent* if P can be cut into finitely many polytopes which can be rearranged and assembled into Q, i.e. if P and Q are the disjoint union of congruent polytopes: $P = \cup_{i=1}^n P_i$, $Q = \cup_{i=1}^n Q_i$, $P_i \simeq Q_i$.

Let π be a linear functional on $\mathbb{R}^m$. If Q_i can be obtained from P_i by a translation by a vector in the hyperplane $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^m \colon \pi(\mathbf{x}) = 0\}$, we say that P and Q are *π-congruent*. If P and Q are π-congruent for some linear functional π, we say that they are *Π-congruent*.

If P can be cut into countably many polytopes which can be translated by a vector in the hyperplane $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^m \colon \pi(\mathbf{x}) = 0\}$ and assembled into Q, we say that P and Q are

approximately π-congruent. We say that they are *approximately Π-congruent* if they are approximately π-congruent for some linear functional π. If P and Q are approximately π-congruent, there exist, for every $\varepsilon > 0$, π-congruent polytopes $P_\varepsilon \subseteq P$ and $Q_\varepsilon \subseteq Q$, such that $\mathrm{vol}(P \setminus P_\varepsilon) < \varepsilon$ and $\mathrm{vol}(Q \setminus Q_\varepsilon) < \varepsilon$.

Finally, let $\mathbf{R}(a_1, \ldots, a_m) = [0, a_1) \times \cdots \times [0, a_m)$ be a box in $\mathbb{R}^m$, and let $R(a_1, \ldots, a_m) = \mathbf{R}(a_1, \ldots, a_m) \cap \mathbb{Z}^m$ be the set of its integer points.

EXAMPLE 2.8 Let $d = 2$ and $\pi(x, y) = x + y$. Euclid's algorithm on (a, b) yields a π-congruence between $\mathbf{R}(a, b)$ and $\mathbf{R}(b, a)$: if $b = r_1 a + s_1$ with $0 \le s_1 < a$, divide $[0, a) \times [0, r_1 a)$ into r_1 squares with side a, and translate the square $[0, a) \times [ia, (i + 1)a)$ by the vector $(ia, -ia)$ to $[ia, (i + 1)a) \times [0, a)$. Then write $a = r_2 s_1 + s_2$ with $0 \le s_2 < s_1$, divide $[0, a) \times [r_1 a, b)$ into r_2 squares with side s_1, and translate the square $[is_1, (i+1)s_1) \times [r_1 a, b)$ by the vector $(r_1 a - is_1, is_1 - r_1 a)$ to $[r_1 a, b) \times [is_1, (i+1)s_1)$. Continue until the remainder s_i is equal to 0. The first drawing of Figure 2 gives an example.

The second drawing shows that boxes $\mathbf{R}(12, 8)$ and $\mathbf{R}(32, 3)$ are π-congruent for $\pi(x, y) = x + 4y$. Finally, in Figure 3 we give a π-congruence between $\mathbf{R}(4, 5, 3)$ and $\mathbf{R}(5, 3, 4)$ for $\pi(x, y, z) = 3x + 4y + 5z$.   □
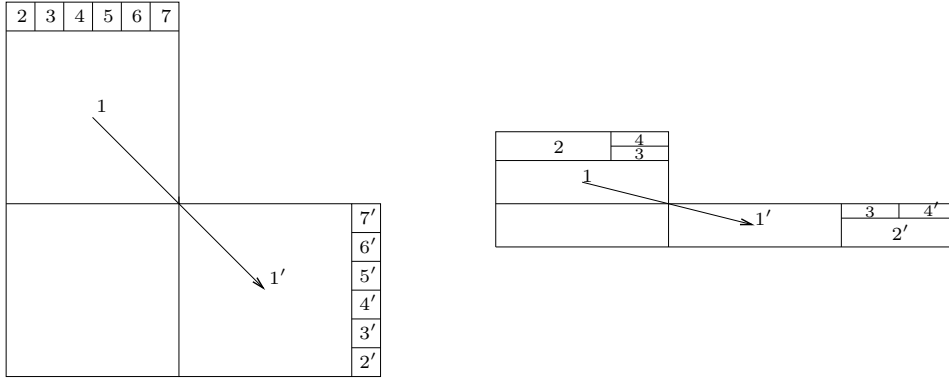

FIGURE 2. Two Π-congruences.

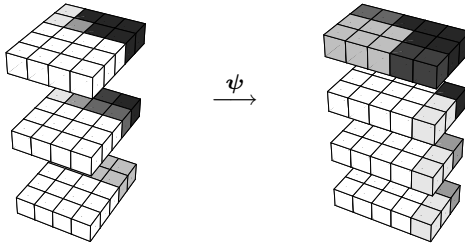

FIGURE 3. π-congruence between $\mathbf{R}(4, 5, 3)$ and $\mathbf{R}(5, 3, 4)$.

3. Main results

3.1. **Continuous O'Hara's algorithm and Π-congruences.** Take the case when G_φ is a cycle $i_1 \to i_m \to i_{m-1} \to \ldots \to i_1$. In this case, $\varphi(i_1) = i_2$, $\varphi(i_2) = i_3$, etc. Throughout this section, identify a partition $i_1^{t_1} \cdots i_m^{t_m}$ with the vector $\mathbf{t} = (t_1, \ldots, t_m)$. By Theorem 2.3, O'Hara's algorithm defines a bijection $\psi \colon R(a_1, \ldots, a_m) \to R(b_1, \ldots, b_m)$, where $i_j a_j = i_{j+1} b_{j+1}$ for all j, where the indices are taken cyclically. The following algorithm (see also Theorem 3.2) generalizes ψ to the continuous setting. It gives a bijection $\boldsymbol{\psi} \colon \mathbf{R}(a_1, \ldots, a_m) \to \mathbf{R}(b_1, \ldots, b_m)$, which is defined also for non-integer a_j, b_j. When a_j, b_j are integers, it is an extension of $\psi \colon R(a_1, \ldots, a_m) \to R(b_1, \ldots, b_m)$. As an immediate corollary, we prove that two boxes with rational coordinates and with equal volume are Π-congruent. We show in Subsection 4.2 how we can use Theorem 3.2 to give an alternative proof of Theorem 2.3.

Algorithm 3.1 (continuous O'Hara's algorithm)

```
Fix: i = (i₁, ..., iₘ) ∈ ℝᵐ₊
     a = (a₁, ..., aₘ) ∈ ℝᵐ₊, b = (b₁, ..., bₘ) ∈ ℝᵐ₊ with iⱼaⱼ = iⱼ₊₁bⱼ₊₁
Input: t ∈ R(a₁, ..., aₘ)
Set: s ← t
While: s contains a coordinate sⱼ ≥ bⱼ
     Do: sⱼ ← sⱼ − bⱼ, sⱼ₋₁ ← sⱼ₋₁ + aⱼ₋₁
Output: ψ(t) ← s
```

It is clear that the algorithm starts with an element of $P = \mathbf{R}(a_1, \ldots, a_m)$ and, if the while loop terminates, outputs an element of $Q = \mathbf{R}(b_1, \ldots, b_m)$. It is not obvious, however, that the loop terminates in every case, or that the output $\boldsymbol{\psi}(\mathbf{t})$ and the number of steps $\mathbf{L}_\varphi(\mathbf{t})$ depend only on $\mathbf{t}$, not on the choices made in the while loop.

Theorem 3.2 *Algorithm 3.1 has the following properties.*

(1) *The algorithm stops after a finite number of steps, and the resulting vector $\boldsymbol{\psi}(\mathbf{t})$ and the number of steps $\mathbf{L}_\varphi(\mathbf{t})$ are independent of the choices made during the execution of the algorithm.*

(2) *The algorithm defines a bijection $\boldsymbol{\psi} \colon P \to Q$ which satisfies $\boldsymbol{\psi}(\mathbf{t}) - \mathbf{t} \in \mathcal{H}$, where $\mathcal{H}$ is the hyperplane defined by $i_1 x_1 + \ldots + i_m x_m = 0$.*

(3) *We have*

$$\mathbf{L}_\varphi(\mathbf{t} + \mathbf{t}') \geq \mathbf{L}_\varphi(\mathbf{t}) + \mathbf{L}_\varphi(\mathbf{t}') \text{ for every } \mathbf{t}, \mathbf{t}', \mathbf{t} + \mathbf{t}' \in P.$$

In particular, $\mathbf{L}_\varphi(\mathbf{t}') \leq \mathbf{L}_\varphi(\mathbf{t})$ if $\mathbf{t}' \leq \mathbf{t}$.

(4) *Let $\mathbf{t}, \mathbf{t}' \in P$, $\mathbf{s} = \boldsymbol{\psi}(\mathbf{t})$, with $t_j \leq t'_j < t_j + \varepsilon_j$, where $\varepsilon_j = b_j - s_j$. Then*

$$\boldsymbol{\psi}(\mathbf{t}') - \mathbf{t}' = \boldsymbol{\psi}(\mathbf{t}) - \mathbf{t} \quad and \quad \mathbf{L}_\varphi(\mathbf{t}') = \mathbf{L}_\varphi(\mathbf{t}).$$

(5) *For all $\mathbf{a}, \mathbf{b} \in \mathbb{Z}^m_+$, we have*

$$\max_{\mathbf{t} \in P} \mathbf{L}_\varphi(\mathbf{t}) = \operatorname{lcm}(c_1, \ldots, c_m) \cdot \left(\frac{1}{c_1} + \ldots + \frac{1}{c_m} \right) - m,$$

where $c_j = a_1 \cdots a_{j-1} b_j \cdots b_{m-1}$.

The proof of the theorem is given in Subsection 4.1.

We call boxes $P = \mathbf{R}(a_1, \ldots, a_m)$, $Q = \mathbf{R}(b_1, \ldots, b_m)$ *relatively rational* if there exists λ, $\lambda \neq 0$, such that $\lambda a_j \in \mathbb{Z}, \lambda b_j \in \mathbb{Z}$. Clearly, two boxes P and Q with rational side-lengths are relatively rational.

Corollary 3.3 *Boxes* $P = \mathbf{R}(a_1, \ldots, a_m)$, $Q = \mathbf{R}(b_1, \ldots, b_m)$ *with equal volume are approximately* Π-*congruent. Moreover, when* P *and* Q *are relatively rational and have equal volume, they are* Π-*congruent.*

Proof. For $j = 1, \ldots, m$, take $i_j = a_1 \cdots a_{j-1} b_{j+1} \cdots b_m$. Clearly $i_j a_j = i_{j+1} b_{j+1}$ for $j = 1, \ldots, m-1$, and $a_1 \cdots a_m = b_1 \cdots b_m$ implies $i_m a_m = i_1 b_1$. Therefore, the numbers i_j, a_j, b_j satisfy the conditions of Algorithm 3.1. By Theorem 3.2 part (2), the algorithm defines a bijection $\psi \colon P \to Q$. Parts (4) and (2) of Theorem 3.2 imply that we can cut P into (countably many) smaller boxes, each of which is translated by a vector in the plane $i_1 x_1 + \ldots + i_m x_m = 0$.

If P and Q are relatively rational, we can assume without loss of generality that all a_j, b_j are integers. For any integer vector $\mathbf{t}$, we have $\psi(\mathbf{t}') - \mathbf{t}' = \psi(\mathbf{t}) - \mathbf{t}$ and $\mathbf{L}_\varphi(\mathbf{t}') = \mathbf{L}_\varphi(\mathbf{t})$ whenever $t_j \leq t_j' < t_j + 1$, so P and Q are divided into a finite number (at most $a_1 \cdots a_m$) of boxes. $\qquad \square$

EXAMPLE 3.4 Even in the 3-dimensional case the Π-congruence defined by the algorithm can be quite complex, as the next figure suggests. Here the same shading is used for parallel translations by the same vector. $\qquad \square$
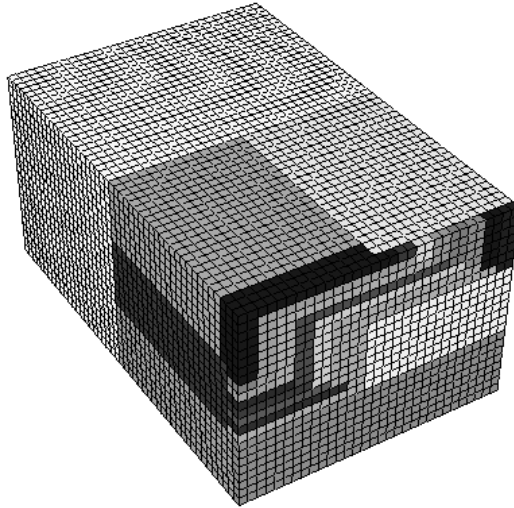

FIGURE 4. The decomposition of the box $\mathbf{R}(31, 47, 23)$ given by O'Hara's algorithm (only the top, right, and back sides are shown) .

3.2. **Complexity of O'Hara's algorithm.** The complexity of O'Hara's algorithm has been an open problem, with the exception of the elementary distinct/odd case (see [O1] and Example 3.8).

It turns out that the complexity depends heavily on the type of the graph G_φ defined in Subsection 2.3. Part (5) of Theorem 3.2 gives the maximum number of steps that O'Hara's algorithm takes when G_φ is a cycle. The following lemma gives an estimate for $\mathcal{L}_\varphi(n)$ when G_φ is a path.

Lemma 3.5 *Let G_φ be a finite or infinite path on $\mathcal{I} \subseteq \mathbb{P}$. Then $\mathcal{L}_\varphi(n) \leq n(\log n + 1)$. Moreover, if*

$$D = \sum_{i \in \mathcal{I}} \frac{1}{i a_i} = \sum_{j \in \mathcal{I}} \frac{1}{j b_j} < \infty,$$

then $\mathcal{L}_\varphi(n) \leq Dn$.

This lemma and the other results in this subsection are proved in Section 4. Here and throughout the paper, by $\log n$ we mean the natural logarithm of n.

We combine these estimates to prove Theorem 3.6, the second main result of this paper, which gives upper bounds for the complexity of the algorithm in the general case.

Theorem 3.6 *Let $\overline{a}, \overline{b}$ be φ-equivalent sequences.*

    (1) *If G_φ has only a finite number of cycles of length > 2, then $\mathcal{L}_\varphi(n) = O(n \log n)$, and the constants implied by the O-notation are universal.*
    (2) *If G_φ has only a finite number of cycles of length $> m$ for some $m > 2$, then $\mathcal{L}_\varphi(n) = O(n^{m-1})$, and the constants implied by the O-notation depend only on m.*

The following theorem gives the corresponding lower bound on the worst case complexity. It shows that the estimates of Theorem 3.6 are close to being sharp.

Theorem 3.7 *There exist φ-equivalent sequences $\overline{a}$ and $\overline{b}$, such that:*

    (1) *G_φ is a path and $\mathcal{L}_\varphi(n) = \Omega(n \log \log n)$;*
    (2) *G_φ contains only cycles of length $\leq m$ and $\mathcal{L}_\varphi(n) = \Omega(n^{m-1-\varepsilon})$ for every $\varepsilon > 0$;*
    (3) *$\mathcal{L}_\varphi(n) = \exp \Omega(\sqrt[3]{n})$.*

In other words, depending on the type of the graph, we have nearly matching upper and lower bounds on $\mathcal{L}_\varphi(n)$. For example, for an m-cycle, Theorem 3.6 shows that $\mathcal{L}_\varphi(n)$ is $O(n^{m-1})$, while Theorem 3.7 shows that it is $\Omega(n^{m-1-\varepsilon})$ for every $\varepsilon > 0$. Similarly, part (3) shows that O'Hara's algorithm can be very slow in general since the total number of partitions of n is asymptotically $\exp \Theta(\sqrt{n})$.

EXAMPLE 3.8 In the distinct/odd case, the graph G_φ is composed of infinite paths

$$\ldots \to 8j \to 4j \to 2j \to j \quad \text{for each odd } j.$$

A partition $\lambda \vdash n$ can be broken up into partitions $\lambda^{(j)} \vdash n_j$ such that the support of $\lambda^{(j)}$ is contained in $\{(2j-1)2^k \colon k \in \mathbb{N}\}$. We have

$$\sum_{k=0}^{\infty} \frac{1}{2^{k+1}(2j-1)} = \frac{1}{2j-1},$$

and Lemma 3.5 implies that O'Hara's algorithm takes at most $n_j/(2j-1)$ steps to compute $\psi(\lambda^{(j)})$. This implies that

$$L_{\varphi}(\lambda) \leq \sum_{j=1}^{\infty} \frac{n_j}{2j-1} \leq \sum_{j=1}^{n} n_j = n.$$

In other words, O'Hara's algorithm takes at most n steps to compute $\psi(\lambda)$. This bound is (almost) sharp since the algorithm takes $2^k - 1$ steps to compute $\psi(2^k) = 1^{2^k}$. $\qquad \square$

3.3. O'Hara's algorithm as an integer linear programming problem. Let us now give a new description of O'Hara's algorithm.

Proposition 3.9 *Let* $\mathbf{i}, \mathbf{a}, \mathbf{b} \in$ *be as above such that* $i_j a_j = i_{j+1} b_{j+1}$ *for* $j = 1, \ldots, m$. *Fix a vector* $\mathbf{t} \in \mathbf{R}(a_1, \ldots, a_m)$. *Then* $\mathbf{s} = \boldsymbol{\psi}(\mathbf{t})$ *satisfies the following:*

$$\mathbf{s} = \mathbf{t} + A\mathbf{k},$$

where

$$A = \begin{pmatrix} -b_1 & a_1 & 0 & \cdots & 0 \\ 0 & -b_2 & a_2 & \cdots & 0 \\ 0 & 0 & -b_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_m & 0 & 0 & \cdots & -b_m \end{pmatrix}$$

and $\mathbf{k} = (k_1, \ldots, k_m)$ *is the unique vector minimizing*

$$k_1 + \ldots + k_m$$

with constraints

$$\mathbf{k} \in \mathbb{Z}^m, \qquad \mathbf{k} \geq \mathbf{0}, \qquad A\mathbf{k} \geq -\mathbf{t}, \qquad A\mathbf{k} \leq \mathbf{b} - \mathbf{1} - \mathbf{t}.$$

The proposition is proved in Subsection 4.6. The advantage of this approach is that one can use standard integer linear programming results to speed up the computation of $\boldsymbol{\psi}(\mathbf{t})$.

Theorem 3.10 *For every* $m \geq 1$, *there exists a deterministic algorithm which computes the continuous O'Hara's bijection* $\boldsymbol{\psi}$ *in polynomial time, for all integer vectors* $\mathbf{i}, \mathbf{a}, \mathbf{b}$ *as above.*

Proof. It is well known that for a bounded dimension m, there exists an algorithm for solving integer linear programming problem $A\mathbf{x} \leq \mathbf{b}$, for which the number of steps is bounded by a polynomial in the logarithm of the largest entry of A, $\mathbf{b}$ for integer A and $\mathbf{b}$ (see e.g. [S, Corollary 18.7b]). By Proposition 3.9, this implies the result. $\qquad \square$

Theorem 3.10 can be used to obtain a significant speed-up of (the usual) O'Hara's algorithm, in the case when G_φ contains only cycles of bounded length. Namely, we obtain the following result.

Theorem 3.11 *Let $\bar{a} \sim_\varphi \bar{b}$. If the lengths of cycles of G_φ are bounded, there exists a deterministic algorithm which computes $\psi(\lambda)$ in $O(n \log n)$ steps for $\lambda \in \mathcal{A}_n$.*

Proof. Without loss of generality, the support of $\lambda \in \mathcal{A}_n$ is contained in one of the connected components of G_φ. If this connected component is a path, O'Hara's algorithm takes $O(n \log n)$ steps by Lemma 3.5. If it is a cycle of length m, the algorithm described in the previous theorem takes $O(\log^c n)$ steps for some c, and obviously the $O(n \log n)$ term dominates. $\square$

REMARK 3.12 Let us note that the inner workings of the algorithms in Theorem 3.10 and Theorem 3.11 have a geometric rather than combinatorial nature, and are very different from those of O'Hara's algorithm. However, both kinds of algorithms, when applied to the same input, have the same output, which means that they produce the same partition bijection.

4. Proofs and examples

4.1. **Proof of Theorem 3.2.** Throughout the section, indices are taken modulo m.

Lemma 4.1 *Take a vector $\mathbf{t} \in P$, choose $\varepsilon_j < a_j - t_j$, and do the algorithm on $\mathbf{t}$: denote the vectors we get by $\mathbf{s}^0 = \mathbf{t}$, $\mathbf{s}^1$, $\mathbf{s}^2$, etc. Then each box of size $\varepsilon_1 \times \cdots \times \varepsilon_m$ contains at most one of $\mathbf{s}^i$.*

Proof. Assume that we have $|s_j^{k'} - s_j^k| \leq \varepsilon_j$ for all j for some $0 \leq k < k'$, i.e. that we hit an $\varepsilon_1 \times \cdots \times \varepsilon_m$ box twice. Say that in the course of getting from $\mathbf{t}$ to $\mathbf{s}^k$ (respectively $\mathbf{s}^{k'}$, respectively $\mathbf{s}^{k'-k}$), we subtracted b_j from the j-th coordinate and added a_j to the $(j-1)$-th coordinate k_j times (respectively k_j' times, respectively k_j'' times). Clearly we have $\sum k_j = k$, $\sum k_j' = k'$ and $\sum k_j'' = k' - k$. Furthermore, the equations $s_j^k = t_j - b_j k_j + a_j k_{j+1}$, $s_j^{k'} = t_j - b_j k_j' + a_j k_{j+1}'$ and $s_j^{k'-k} = t_j - b_j k_j'' + a_j k_{j+1}''$, which hold for all j, can be written as

$$\begin{aligned} \mathbf{s}^k &= \mathbf{t} + A\mathbf{k}, \\ \mathbf{s}^{k'} &= \mathbf{t} + A\mathbf{k}', \\ \mathbf{s}^{k'-k} &= \mathbf{t} + A\mathbf{k}'', \end{aligned}$$

where

$$A = \begin{pmatrix} -b_1 & a_1 & 0 & \cdots & 0 \\ 0 & -b_2 & a_2 & \cdots & 0 \\ 0 & 0 & -b_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_m & 0 & 0 & \cdots & -b_m \end{pmatrix}.$$

There are two cases to consider: $\mathbf{k}'' = \mathbf{k}' - \mathbf{k}$ and $\mathbf{k}'' \neq \mathbf{k}' - \mathbf{k}$. We obtain a contradiction in each case. Assume first that $\mathbf{k}'' = \mathbf{k}' - \mathbf{k}$. Then we have

$$\mathbf{s}^{k'-k} - \mathbf{t} = \mathbf{t} + A(\mathbf{k}' - \mathbf{k}) - \mathbf{t} = \mathbf{s}^{k'} - \mathbf{s}^k.$$

In particular, we have $|s_j^{k'-k} - t_j| \leq \varepsilon_j$ and $s_j^{k'-k} < a_j$ for all j. On the other hand, $\mathbf{s}^{k'-k}$ was obtained from $\mathbf{s}^{k'-k-1}$ by choosing j with $s_j^{k'-k-1} \geq b_j$ and then taking $s_j^{k'-k} = s_j^{k'-k-1} - b_j$ and $s_{j-1}^{k'-k} = s_{j-1}^{k'-k-1} + a_{j-1}$. Therefore $s_{j-1}^{k'-k} \geq a_{j-1}$, a contradiction.

Suppose now that $\mathbf{k}'' \neq \mathbf{k}' - \mathbf{k}$. Since $\sum k_j'' = \sum (k_j' - k_j)$, there is a j so that $k_j'' \geq k_j' - k_j$ and $k_{j+1}'' < k_{j+1}' - k_{j+1}$. We have

$$\mathbf{s}^{k'-k} = \mathbf{t} + A\mathbf{k}'' = \mathbf{t} + A(\mathbf{k}'' - \mathbf{k}' + \mathbf{k}) + \mathbf{s}^{k'} - \mathbf{s}^k.$$

This implies

$$0 \leq s_j^{k'-k} \leq t_j - b_j \underbrace{(k_j'' - k_j' + k_j)}_{\geq 0} + a_j \underbrace{(k_{j+1}'' - k_{j+1}' + k_{j+1})}_{\leq -1} + \varepsilon_j \leq t_j - a_j + \varepsilon_j < 0,$$

a contradiction. Therefore we can never hit an $\varepsilon_1 \times \cdots \times \varepsilon_m$ box twice, which completes the proof. $\qquad\square$

Now we are ready to prove all parts of Theorem 3.2.

(1) Clearly we have $i_1 s_1^k + \ldots + i_m s_m^k = i_1 t_1 + \ldots + i_m t_m$ and $s_j^k \geq 0$ for all j and k. Since the set $\{\mathbf{x} \in \mathbb{R}^m : i_1 x_1 + \ldots + i_m x_m = C, x_j \geq 0\}$ can be covered by a finite number of boxes of size $\varepsilon_1 \times \cdots \times \varepsilon_m$, the number of steps is finite.

Let us prove that the output vector $\mathbf{s} \in Q$ does not depend on the choices we make in the course of O'Hara's algorithm. Assume that when we run the algorithm twice, we obtain vectors $\mathbf{s}$ and $\mathbf{s}'$. Denote by $k = |\mathbf{k}| = k_1 + \ldots + k_m$ the number of steps to obtain $\mathbf{s}$, and by $k' = |\mathbf{k}'|$ the number of steps to obtain $\mathbf{s}'$, and assume that $k \leq k'$. Denote by $\mathbf{s}''$ the vector we obtain after k steps in the second run of the algorithm. Think of $\mathbf{s}''$ as being on the path from $\mathbf{t}$ to $\mathbf{s}'$. Write $\mathbf{s} = \mathbf{t} + A\mathbf{k}$, $\mathbf{s}'' = \mathbf{t} + A\mathbf{k}''$, where $|\mathbf{k}| = |\mathbf{k}''| = k$. Then $\mathbf{s}'' = \mathbf{s} + A(\mathbf{k}'' - \mathbf{k})$, and $k_j'' > k_j$, $k_{j+1}'' \leq k_{j+1}$ would imply

$$0 \leq s_j'' = s_j - b_j \underbrace{(k_j'' - k_j)}_{\geq 1} + a_j \underbrace{(k_{j+1}'' - k_{j+1})}_{\leq 0} < b_j - b_j = 0,$$

which is a contradiction. Therefore $\mathbf{k}'' = \mathbf{k}$ and $\mathbf{s}'' = \mathbf{s}$, therefore also $k' = k$ and $\mathbf{s}' = \mathbf{s}$.

(2) Let us construct an explicit inverse map $\boldsymbol{\psi}^{-1}$. Denote by τ the flip $\tau \colon (x_1, \ldots, x_m) \mapsto (x_m, \ldots, x_1)$. Consider a map $\boldsymbol{\psi}' \colon \mathbf{R}(b_m, \ldots, b_1) \to \mathbf{R}(a_m, \ldots, a_1)$ obtained by replacing $(i_1, \ldots, i_m)$, $(a_1, \ldots, a_m)$ and $(b_1, \ldots, b_m)$ with $(i_m, \ldots, i_1)$, $(b_m, \ldots, b_1)$ and $(a_m, \ldots, a_1)$, respectively. Observe that

$$\boldsymbol{\psi}^{-1} := \tau \circ \boldsymbol{\psi}' \circ \tau : Q \to P$$

is the inverse map of $\boldsymbol{\psi}$. Therefore, $\boldsymbol{\psi}$ is one-to-one.

It remains to check that $\boldsymbol{\psi}$ satisfies $\boldsymbol{\psi}(\mathbf{t}) - \mathbf{t} \in \mathcal{H}$, where $\mathcal{H}$ is the hyperplane defined by $i_1 x_1 + \ldots + i_m x_m = 0$. Note that the columns of A lie in $\mathcal{H}$. Because $\boldsymbol{\psi}(\mathbf{t}) - \mathbf{t} = A\mathbf{k}$ for some $\mathbf{k}$, $\boldsymbol{\psi}(\mathbf{t}) - \mathbf{t}$ lies in $\mathcal{H}$.

(3) Let $\mathbf{s}^0 = \mathbf{t}, \mathbf{s}^1, \mathbf{s}^2, \ldots, \mathbf{s}^k = \boldsymbol{\psi}(\mathbf{t})$, $k = \mathbf{L}_\varphi(\mathbf{t})$, be the intermediate steps of the algorithm which computes $\boldsymbol{\psi}(\mathbf{t})$. Similarly, let $\mathbf{r}^0 = \mathbf{t}', \mathbf{r}^1, \mathbf{r}^2, \ldots, \mathbf{r}^{k'} = \boldsymbol{\psi}(\mathbf{t}')$, $k' = \mathbf{L}_\varphi(\mathbf{t}')$, be the intermediate steps of the algorithm which computes $\boldsymbol{\psi}(\mathbf{t})$. Every vector in the sequence $\mathbf{t} + \mathbf{t}', \mathbf{s}^1 + \mathbf{t}', \ldots, \mathbf{s}^k + \mathbf{t}', \mathbf{s}^k + \mathbf{r}^1, \ldots, \mathbf{s}^k + \mathbf{r}^{k'-1}$ has at least one of the coordinates $\geq b_j$, so O'Hara's algorithm takes at least $k + k' = \mathbf{L}_\varphi(\mathbf{t}) + \mathbf{L}_\varphi(\mathbf{t}')$ steps to compute $\boldsymbol{\psi}(\mathbf{t} + \mathbf{t}')$.

(4) Let $\mathbf{t} \in P$, $\mathbf{s} = \boldsymbol{\psi}(\mathbf{t})$ and $\varepsilon_j = b_j - s_j$. For every $\mathbf{t}'$ with $t_j \leq t'_j < t_j + \varepsilon_j$, make the steps of the algorithm which inputs $\mathbf{t}'$ the same as the one which inputs $\mathbf{t}$. Let $\mathbf{s}^i$ and $\mathbf{r}^i$ be as in part (3). Then the vector $\mathbf{r}^i$ satisfies $\mathbf{r}^i = \mathbf{s}^i + (\mathbf{t}' - \mathbf{t})$. For $k = \mathbf{L}_\varphi(\mathbf{t})$, we have $\mathbf{s}^k = \mathbf{s}$ and $\mathbf{r}^k = \mathbf{s} + (\mathbf{t}' - \mathbf{t}) \in Q$. Therefore, $\mathbf{r}^k = \boldsymbol{\psi}(\mathbf{t}')$, $\psi(\mathbf{t}') = \psi(\mathbf{t}) + \mathbf{t}' - \mathbf{t}$ and $\mathbf{L}_\varphi(\mathbf{t}') = \mathbf{L}_\varphi(\mathbf{t})$, as desired.

(5) Assume that a_j and b_j are integers for all $j = 1, \ldots, m$. Let $\mathbf{t} \in P$ and assume that O'Hara's algorithm with input $\mathbf{t}$ subtracts b_j from the j-th coordinate and adds a_{j-1} to the $(j - 1)$-st coordinate exactly k_j times. This implies that $\mathbf{s} = \psi(\mathbf{t}) = \mathbf{t} + A\mathbf{k}$.

Observe that the matrix A has rank $m - 1$, and its kernel is spanned by the vector

$$\left(\frac{1}{b_1 b_2 \cdots b_{m-2} b_{m-1}}, \frac{1}{a_1 b_2 \cdots b_{m-2} b_{m-1}}, \ldots, \frac{1}{a_1 a_2 \cdots a_{m-2} a_{m-1}} \right) = \left(\frac{1}{c_1}, \frac{1}{c_2}, \ldots, \frac{1}{c_m} \right).$$

The integer vectors in $\ker A$ are integer multiples of the vector

$$\mathrm{lcm}(c_1, c_2, \ldots, c_m) \cdot \left(\frac{1}{c_1}, \frac{1}{c_2}, \ldots, \frac{1}{c_m} \right).$$

Suppose an integer vector $\mathbf{n} = (n_1, \ldots, n_m)$ with at least one negative and at least one non-negative coordinate is a solution of

(4.1) $$\mathbf{s} = \mathbf{t} + A\mathbf{n}.$$

Suppose $n_j < 0$ and $n_{j+1} \geq 0$. Then we have $-b_j n_j + a_j n_{j+1} \geq b_j$ and

$$b_j > s_j = t_j - b_j n_j + a_j n_{j+1} \geq b_j,$$

a contradiction. This implies that the coordinates of every solution to (4.1) are either all negative or all non-negative. In particular, there is exactly one solution $\mathbf{n}$ with

$$0 \leq n_j \leq \frac{\mathrm{lcm}(c_1, \ldots, c_m)}{c_j} - 1$$

for each j.

First, let us show that this solution is equal to $\mathbf{k}$. Denote by n'_j the number of times we subtract b_j from the j-th coordinate and add a_{j-1} to the $(j-1)$-th coordinate in the first $|\mathbf{n}| = n_1 + \ldots + n_m$ steps of the algorithm with input $\mathbf{t}$. Let $\mathbf{s}' = \mathbf{t} + A\mathbf{n}'$. Then $\mathbf{s}' = \mathbf{s} + A(\mathbf{n}' - \mathbf{n})$, and the same argument as above shows that we cannot have $n'_j - n_j < 0$, $n'_{j+1} - n_{j+1} \geq 0$. Clearly the sum of the coordinates of $\mathbf{n}' - \mathbf{n}$ is 0, so we must have $\mathbf{n}' = \mathbf{n}$ and $\mathbf{s}' = \mathbf{s}$. This implies that $\mathbf{n} = \mathbf{k}$. Therefore, the maximum number of steps $\mathbf{L}_\varphi(\mathbf{t})$ is at most

(4.2) $$\mathrm{lcm}(c_1, \ldots, c_m) \cdot \left(\frac{1}{c_1} + \ldots + \frac{1}{c_m} \right) - m.$$

Consider the vector $(a_1 - 1, \ldots, a_m - 1)$ and observe that

$$i_1(a_1 - 1) + \ldots + i_m(a_m - 1) = i_1(b_1 - 1) + \ldots + i_m(b_m - 1).$$

Since $\boldsymbol{\psi}(\mathbf{t})$ is an integer vector, we have $\mathbf{s} = \boldsymbol{\psi}(\mathbf{t}) = (b_1 - 1, \ldots, b_m - 1)$. Furthermore, the vector $(-1, -1, \ldots, -1)$ is a solution of the system

$$b_j - 1 = a_j - 1 - b_j n_j + a_j n_{j+1}, \quad 1 \leq j \leq m.$$

Therefore,

$$k_j = \frac{\mathrm{lcm}(c_1, \ldots, c_m)}{c_j} - 1$$

and the maximum number of steps $\mathbf{L}_\varphi(\mathbf{t})$ over all $\mathbf{t}$, is given by the equation (4.2). This finishes the proof of Theorem 3.2. $\square$

4.2. **Proof of Theorem 2.3.** In this subsection, we present an alternative proof of O'Hara's theorem by deducing it from Theorem 3.2. Take a partition $\lambda \in \mathcal{A}_n$. Without loss of generality we may assume that its support is contained in one of the connected components of G_φ. If this connected component is a cycle, the fact that O'Hara's algorithm stops and that the final result is independent of the choices made in the algorithm follows from Theorem 3.2 part (1). If the connected component is a path, we can assume that $\lambda = i_1^{t_1} \cdots i_m^{t_m}$, where $i_1 \to i_2 \to \ldots \to i_m \to i_{m+1} \to \ldots$ is a part of G_φ. Assume that $i_j b_j = i_{j+1} a_{j+1}$. Remove copies of i_1 until their number is smaller than b_1. Remove copies of i_2 until their number is smaller than b_2, etc. Since i_j are all different, we have $i_k > n$ for some k. The algorithm cannot increase the number of copies of i_k, so it stops before that. Furthermore, it is clear that if k_j is the number of copies of i_j removed, then $k_1 = \lfloor t_1/b_1 \rfloor$, $k_2 = \lfloor (t_2 + k_1 a_2)/b_2 \rfloor$, etc. Therefore the final result is independent of the order in which we remove and add parts. $\square$

4.3. **Proof of Lemma 3.5.** Assume that the path is $\ldots \to i_{-1} \to i_0 \to i_1 \to i_2 \to \ldots$ (it can be finite, or infinite in one or both direction). A partition λ has only a finite number of parts, without loss of generality we can assume that $\lambda = i_1^{t_1} i_2^{t_2} \cdots i_m^{t_m}$ and $\psi(\lambda) = i_1^{s_1} i_2^{s_2} \cdots i_m^{s_m}$. O'Hara's algorithm is straightforward in this case: first remove $b_1 k_1$ copies of i_1, where k_1 is the smallest integer with $t_1 - b_1 k_1 < b_1$, and add $a_2 k_1$ copies of i_2; then remove $b_2 k_2$ copies of i_2, where k_2 is the smallest integer with $t_2 + a_2 k_1 - b_2 k_2 < b_2$, and add $a_3 k_2$ copies of i_3, etc. Clearly,

$$k_1 \leq \frac{t_1}{b_1} = \frac{i_1 t_1}{i_1 b_1},$$

$$k_2 \leq \frac{t_2 + a_2 k_1}{b_2} = \frac{i_2 t_2 + i_1 b_1 k_1}{i_2 b_2} \leq \frac{i_1 t_1 + i_2 t_2}{i_2 b_2},$$

$$k_3 \leq \frac{t_3 + a_3 k_2}{b_3} = \frac{i_3 t_3 + i_2 b_2 k_2}{i_3 b_3} \leq \frac{i_1 t_1 + i_2 t_2 + i_3 t_3}{i_3 b_3},$$

etc. Therefore, the total number of steps is at most

$$(i_1 t_1 + \ldots + i_m t_m) \left(\frac{1}{i_1 b_1} + \ldots + \frac{1}{i_m b_m} \right).$$

Since $i_1 t_1 + \ldots + i_m t_m$ is the size n of λ, the number of steps is at most Dn when $D < \infty$. In general, since $i_1, \ldots, i_m$ are distinct integers, there is a j with $i_j \geq m$. On the other hand, when $i_j > n$, O'Hara's algorithm stops at i_j. This implies that $m \leq n$, and

$$\frac{1}{i_1 b_1} + \ldots + \frac{1}{i_m b_m} \leq \frac{1}{i_1} + \ldots + \frac{1}{i_m} \leq 1 + \frac{1}{2} + \ldots + \frac{1}{m} \leq \log m + 1 \leq \log n + 1.$$

This completes the proof. $\qquad\square$

4.4. **Proof of Theorem 3.6.** (1) For a 1-cycle $\{i\}$ and a partition λ with only parts i, we have $\psi(\lambda) = \lambda$ and O'Hara's algorithm takes 0 steps. For a 2-cycle $i \to j \to i$, by Theorem 3.2 part (5), the largest number of steps of O'Hara's algorithm is equal to

$$\operatorname{lcm}(a_i, b_i) \cdot \left(\frac{1}{a_i} + \frac{1}{b_i} \right) - 2 \leq a_i b_i \cdot \left(\frac{1}{a_i} + \frac{1}{b_i} \right) - 2 = a_i + b_i - 2.$$

Denote by $\lambda_{\max} = i^{a_i-1} j^{a_j-1}$ the largest partition in $\mathcal{A}$. Then O'Hara's algorithm takes at most

$$a_i + b_i - 2 \leq i(a_i - 1) + i(b_i - 1) = |\lambda_{\max}| \qquad \text{steps.}$$

For a partition $\lambda \vdash n \geq |\lambda_{\max}|/2$, we need, by Theorem 3.2 part (3), at most $|\lambda_{\max}| \leq 2n$ steps. Similarly, for a partition $\lambda = i^t j^s \vdash n < |\lambda_{\max}|/2$, we have $L_\varphi(\lambda) \leq \max\{t, s\} \leq n$, since $it + js$ is less than $i(b_i - 1)$ or $j(b_j - 1)$.

Consider a cycle $C : i_1 \to i_m \to i_{m-1} \to \ldots \to i_1$ in G_φ of length $m > 2$. Denote by M_C the maximum number of steps O'Hara's algorithm can take on partitions with support in $\{i_1, \ldots, i_m\}$. Recall that M_C is given by Theorem 3.2 part (5). Finally, denote by M the sum of M_C over all cycles C of length > 2 in G_φ.

Note that every partition $\lambda \in \mathcal{A}$ is decomposed into partitions $\lambda^{(1)}, \ldots, \lambda^{(r)}$ of sizes $n_1, \ldots, n_r$ with support in one of the components of G_φ. From above and by Lemma 3.5, the number of steps of O'Hara's algorithm is at most

$$n_1(\log n_1 + 1) + \ldots + n_r(\log n_r + 1) + 2n + M \leq n(\log n + 3) + M \leq (1 + \varepsilon)n \log n,$$

for every fixed $\varepsilon > 0$ and n large enough.

(2) Take a cycle $i_1 \to i_m \to i_{m-1} \to \ldots \to i_1$ of length m. In the notation of Theorem 3.2 part (5), we have

$$\begin{aligned} L_\varphi(\lambda) &\leq \operatorname{lcm}(c_1, \ldots, c_m) \cdot \left(\frac{1}{c_1} + \ldots + \frac{1}{c_m} \right) - m \\ &\leq a_1 a_2 \cdots a_{m-1} b_1 b_2 \cdots b_{m-1} \cdot \left(\frac{1}{c_1} + \ldots + \frac{1}{c_m} \right) - m \\ &\leq \sum_{j=1}^m b_1 b_2 \cdots b_{j-1} a_j \cdots a_{m-2} a_{m-1} - m. \end{aligned}$$

Denote by N is the largest of $\{a_1, \ldots, a_m, b_1, \ldots, b_m\}$. Then the size of the largest partition $\lambda_{\max} \in \mathcal{A}$ is at least $N - 1$, and by the calculation above the number of steps is at most $m(N^{m-1} - 1)$. Therefore $L_\varphi(\lambda_{\max}) \leq m|\lambda_{\max}|^{m-1}$. For $\lambda \vdash n \geq |\lambda_{\max}|/m$, we have $L_\varphi(\lambda) \leq m|\lambda_{\max}|^{m-1} \leq m^m n^{m-1}$. On the other hand, when $n < |\lambda_{\max}|/m$, we have

$n \leq i_j(b_j - 1)$ for some j, and O'Hara's algorithm does not remove any copies of i_j. By Lemma 3.5, this implies that $L_\varphi(\lambda) \leq n(1 + 1/2 + \ldots + 1/m)$.

In the notation of part (1), write M' for the sum of M_C over all cycles of length $> m$. As before, every partition $\lambda \in \mathcal{A}_n$ is decomposed into partitions $\lambda^{(1)}, \ldots, \lambda^{(r)}$ of sizes $n_1, \ldots, n_r$ with support in one of the components of G_φ. From above and by Lemma 3.5, the number of steps of O'Hara's algorithm is at most

$$m^m n_1^{m-1} + \ldots + m^m n_r^{m-1} + M' \leq m^m n^{m-1} + M' = O(n^{m-1}). \qquad \square$$

4.5. **Proof of Theorem 3.7.** We begin with the following key example.

EXAMPLE 4.2 Let $p_1, p_2, \ldots, p_m$ be distinct primes. Take $\varphi(p_j) = p_{j+1}$ (where $p_{m+1} = p_1$), $a_j = p_{j+1}$, $b_j = p_{j-1}$ (where $p_0 = p_m$). Then

$$a_1 a_2 \cdots a_{j-1} b_j \cdots b_{m-2} b_{m-1} = p_2 p_3 \cdots p_{j-1} p_j p_{j-1} p_j \cdots p_{m-2}$$

for $j = 1, \ldots, m$, and the lowest common multiple of these numbers is

$$p_1 p_2^2 p_3^2 \cdots p_{m-2}^2 p_{m-1} p_m.$$

By Theorem 3.2 part (5), the maximum number of steps of O'Hara's algorithm is equal to

$$p_1 p_2 \cdots p_m \cdot \left(\frac{1}{p_1 p_2} + \frac{1}{p_2 p_3} + \ldots + \frac{1}{p_{m-1} p_m} + \frac{1}{p_m p_1} \right) - m.$$

When $p_1, \ldots, p_m$ are approximately equal, i.e. $|p_j/N - 1| < \varepsilon$ for some N, we obtain partitions of relatively small size (around mN^2), for which O'Hara's algorithm takes a large number of steps (around mN^{m-2}). $\qquad \square$

Now we are ready to prove Theorem 3.7.

(1) Take

i_j	1	$\to$	2	$\to$	3	$\to$	6	$\to$	5	$\to$	10	$\to$	7	$\to$	14	$\to$	9	$\to$	$\ldots$
a_j	∞		1		2		1		6		1		10		1		14		$\ldots$
b_j	2		3		2		5		2		7		2		9		2		$\ldots$

i.e. $i_{2j-1} = 2j-1$, $i_{2j} = 4j-2$, $\varphi(i_j) = i_{j-1}$, $a_1 = \infty$, $a_{2j-1} = 4j-6$ for $j \geq 2$, $a_{4j-2} = 1$, $b_{2j-1} = 2$, $b_{4j-2} = 2j+1$. O'Hara's algorithm on the partition $\lambda^{(k)} = 1^{2^k \cdot (2k-1)!!}$ runs as follows:

$$
\begin{array}{lll}
\mathbf{1^{2^k \cdot 3 \cdot 5 \cdots (2k-1)}} & \longrightarrow & 2^{2^{k-1} \cdot 3 \cdot 5 \cdots (2k-1)} \qquad \text{in } 2^{k-1} \cdot 3 \cdot 5 \cdot 7 \cdot 9 \cdots (2k-1) \text{ steps} \\
2^{2^{k-1} \cdot 3 \cdot 5 \cdots (2k-1)} & \longrightarrow & 3^{2^k \cdot 1 \cdot 5 \cdot 7 \cdots (2k-1)} \\
3^{2^k \cdot 1 \cdot 5 \cdot 7 \cdots (2k-1)} & \longrightarrow & 6^{2^{k-1} \cdot 1 \cdot 5 \cdot 7 \cdots (2k-1)} \qquad \text{in } 2^{k-1} \cdot 1 \cdot 5 \cdot 7 \cdot 9 \cdots (2k-1) \text{ steps} \\
6^{2^{k-1} \cdot 1 \cdot 5 \cdot 7 \cdots (2k-1)} & \longrightarrow & 5^{2^k \cdot 1 \cdot 3 \cdot 7 \cdots (2k-1)} \\
5^{2^k \cdot 1 \cdot 3 \cdot 7 \cdots (2k-1)} & \longrightarrow & 10^{2^{k-1} \cdot 1 \cdot 3 \cdot 7 \cdots (2k-1)} \qquad \text{in } 2^{k-1} \cdot 1 \cdot 3 \cdot 7 \cdot 9 \cdots (2k-1) \text{ steps, etc.}
\end{array}
$$

Note that $n := |\lambda^{(k)}| = 2^k (2k-1)!! = k^{\Theta(k)}$. We conclude that

$$L_\varphi(\lambda^{(k)}) \geq 2^k \cdot (2k-1)!! \left(\frac{1}{2} + \frac{1}{6} + \frac{1}{10} + \ldots + \frac{1}{2(2k-1)} \right) = \Omega(n \log \log n).$$

(2) Take D_k to be a superpolynomial (for example exponential) integer function of k. For each k large enough, we can choose m distinct primes $p_1^k, \ldots, p_m^k$ between D_k and $2D_k$. Choose $i_j^k = km + j$ for $j = 1, \ldots, m$, $a_j^k = i_{j+1}^k p_j^k$, $b_j^k = i_{j-1}^k p_{j-1}^k$ (where indices are written cyclically). Therefore, $i_1^k \to i_m^k \to \ldots \to i_1^k$ is an m-cycle in G_φ. The largest partition $\lambda^{(k)} \in \mathcal{A}$ with support in $\{i_1^k, \ldots, i_m^k\}$ has size at most $2(k+1)m^2 D_k$. Then

$$
\begin{aligned}
\mathrm{lcm}(c_1, \ldots, c_m) &= \mathrm{lcm}(p_m p_1 p_2 \cdots p_{m-2}, p_1^2 p_2 \cdots p_{m-2}, \ldots, p_1 p_2 \cdots p_{m-1}) \\
&= p_1^2 p_2^2 \cdots p_{m-2}^2 p_{m-1} p_m
\end{aligned}
$$

where we omitted the upper index k of p_j^k for simplicity. Therefore,

$$
L_\varphi(\lambda^{(k)}) \geq p_1 p_2 \cdots p_{m-1} + p_2 p_3 \cdots p_m + \ldots + p_m p_1 \cdots p_{m-2} - m \geq m(D_k^{m-1} - 1).
$$

On the other hand, for every C and k large enough, we have

$$
m(D_k^{m-1} - 1) \geq C(2(k+1)m^2 D_k)^{m-1-\varepsilon},
$$

since otherwise D_k grows polynomially, contradicting our assumptions. This implies that the corresponding $\bar{a}, \bar{b}$ and φ satisfy the conditions of (2).

(3) Let $p_1, p_2, \ldots$ denote the sequence of all primes. Set $\varphi(p_j) = p_{j+1}$ for $j \neq k^k$, $\varphi(p_1) = p_1$, $\varphi(p_{k^k}) = p_{(k-1)^{k-1}+1}$ for $k \geq 2$, $a_j = p_{j+1}$ for $j \neq k^k$, $a_1 = p_1$, $a_{k^k} = p_{(k-1)^{k-1}+1}$ for $k \geq 2$, $b_j = p_{j-1}$ for $j \neq 1$, $j \neq k^k + 1$, $b_1 = p_1$, $b_{k^k+1} = p_{(k+1)^{k+1}}$ for $k \geq 1$. The following table summarizes these values.

i_j	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	...	101	103	...
a_j	2	5	7	3	13	17	19	23	29	31	37	41	43	47	53	...	103	11	...
b_j	2	7	3	5	103	11	13	17	19	23	29	31	37	41	43	...	97	101	...

Clearly, G_φ consists of cycles of length 1 and $k^k - (k-1)^{k-1}$ for $k \geq 2$. For $k \geq 2$, write $m_1 = (k-1)^{k-1}$, $m_2 = k^k$, and define $\lambda^{(k)} \in \mathcal{A}$ to be the partition

$$
(p_{m_1+1})^{(p_{m_1+2})-1}(p_{m_1+2})^{(p_{m_1+3})-1} \cdots (p_{m_2-1})^{(p_{m_2})-1}(p_{m_2})^{(p_{m_1+1})-1}
$$

of size

$$
n_k = p_{m_1+1} p_{m_1+2} + p_{m_1+2} p_{m_1+3} + \ldots + p_{m_2-1} p_{m_2} + p_{m_2} p_{m_1+1} - p_{m_1+1} - p_{m_1+2} - \ldots - p_{m_2}.
$$

By the calculation in Example 4.2, O'Hara's algorithm takes exactly

$$
p_{m_1+1} p_{m_1+2} \cdots p_{m_2-2} + p_{m_1+2} p_{m_1+3} \cdots p_{m_2-1} + \ldots + p_{m_2} p_{m_1+1} \cdots p_{m_2-3} - (m_2 - m_1)
$$

steps to compute $\psi(\lambda^{(k)})$. By the distribution law of prime numbers, we have $p_n = n \log n(1 + o(1))$. Therefore,

$$
n_k \leq (m_2 - m_1)p_{m_2}^2 = (k^k - (k-1)^{k-1})p_{k^k}^2 \sim k^k(k^k \log(k^k))^2 = k^{3k+2}(\log k)^2
$$

and

$$
\log L_\varphi(\lambda^{(k)}) \geq \log\left((m_2 - m_1)(p_{m_1+1}^{m_2-m_1-2} - 1)\right) \sim k^k \log\left((k-1)^{k-1}\right) \sim k^{k+1} \log k.
$$

Thus, $L_\varphi(\lambda^{(k)}) > \exp(\sqrt[3]{n_k})$ for k large enough, as desired. $\qquad\square$

4.6. **Proof of Proposition 3.9.** By construction of the algorithm, $\mathbf{k} \in \mathbb{Z}^m$ and it satisfies the inequalities in Proposition 3.9. Therefore, it suffices to prove that $|\mathbf{k}| < |\mathbf{n}|$ for every non-negative integer vector $\mathbf{n} \neq \mathbf{k}$ satisfying $\mathbf{0} \leq \mathbf{t} + A\mathbf{n} \leq \mathbf{b} - \mathbf{1}$.

Assume that $|\mathbf{n}| \leq |\mathbf{k}|$. Denote by n'_j the number of times we remove b_j copies of i_j in the first $|\mathbf{n}|$ steps of O'Hara's algorithm. Define $\mathbf{s}' = \mathbf{t} + A\mathbf{n}'$. From above, $\mathbf{s}' = \mathbf{s} + A(\mathbf{n}' - \mathbf{n})$. If there exists j such that $n'_j - n_j > 0$ and $n'_{j+1} - n_{j+1} \leq 0$, then

$$0 \leq s'_j = s_j - b_j(n'_j - n_j) + a_j(n'_{j+1} - n_{j+1}) < b_j + (-b_j) = 0,$$

which is impossible. Therefore, all the coordinates of $\mathbf{n}' - \mathbf{n}$ have the same sign. On the other hand, the sum of the coordinates of $\mathbf{n}' - \mathbf{n}$ is 0, which implies $\mathbf{n}' = \mathbf{n} = \mathbf{k}$. Therefore, for all $\mathbf{n} \neq \mathbf{k}$ as above, we have $|\mathbf{n}| > |\mathbf{k}|$, as desired. $\qquad\square$

5. Application: the speedy O'Hara's algorithm

In [P3, §8.2], a simple speed-up of O'Hara's algorithm was given: at each step, if the number of parts i in $\lambda \vdash n$ is $rb_i + s$ for $0 \leq s < b_i$, remove rb_i copies of part i and add ra_j copies of part j, where $\varphi(j) = i$. This replaces r steps of O'Hara's algorithm with one step. This algorithm produces the same output as the original algorithm, but the number of steps may depend on the choices we make in the execution. It is called the *speedy O'Hara's algorithm*. Observe now that it cannot be much faster than the usual O'Hara's algorithm: since $r \leq i(rb_i + s) \leq n$, the number of steps can be reduced by at most the order of n, where $\lambda \vdash n$. Thus the reasoning from the proof of Theorem 3.7 part (3) still gives superpolynomial lower bounds for the number of steps. The following examples show that the speed-up is not substantial (for example, of logarithmic complexity) even in the case described in Theorem 3.6 part (1), and that it gives the same bound as Theorem 3.6 part (2).

EXAMPLE 5.1 Take

i_j	$\ldots$	$\to$	18	$\to$	7	$\to$	14	$\to$	5	$\to$	10	$\to$	3	$\to$	6	$\to$	1
a_j	$\ldots$		1		18		1		14		1		10		1		6
b_j	$\ldots$		7		2		5		2		3		2		1		∞

i.e. $i_{2j-1} = 2j - 1$, $i_{2j} = 4j + 2$, $\varphi(i_j) = i_{j+1}$, $a_{2j-1} = 4j + 2$, $a_{4j+2} = 1$, $b_1 = \infty$, $b_{2j-1} = 2$ for $j \geq 2$, $b_{4j+2} = 2j - 1$. This is similar to the example in the proof of Theorem 3.7 part (1). The speedy O'Hara's algorithm on $\lambda^{(k)} = (2k - 1)^{4k-2}$ for $k \geq 5$ runs as follows:

$$(\mathbf{2k - 1})^{\mathbf{4k-2}} \to (4k-2)^{2k-1} \to (4k-2)^2 (2k-3)^{4k-2} \to (4k-2)^2 (4k-6)^{2k-1} \to (4k-2)^2 (4k-6)^4 (2k-5)^{4k-6}$$

$$\to (4k-2)^2 (4k-6)^4 (4k-10)^{2k-3} \to (4k-2)^2 (4k-6)^4 (4k-10)^4 (2k-7)^{4k-10} \to \ldots \to (4k-2)^2 (4k-6)^4 \cdots 14^4 10^7$$

$$\to (4k-2)^2 (4k-6)^4 \cdots 14^4 10^1 3^{20} \to (4k-2)^2 (4k-6)^4 \cdots 14^4 10^1 6^{10} \to (\mathbf{4k - 2})^{\mathbf{2}} (\mathbf{4k - 6})^{\mathbf{4}} \cdots \mathbf{14^4 10^1 1^{60}}.$$

In particular, the speedy O'Hara's algorithm takes $2k - 2 = \Theta(\sqrt{n})$ steps for a partition of size $n = (2k - 1)(4k - 2) = 2(2k - 1)^2$. $\qquad\square$

Note that the proof that $m \leq n$ in the proof of Lemma 3.5 together with the proof of Theorem 3.6 part (1) shows that the speedy O'Hara algorithm takes $O(n)$ steps when G_φ has a finite number of cycles of length greater than 2. This is smaller than $\Omega(n \log \log n)$ obtained earlier. In other words, the speedy O'Hara's algorithm can be asymptotically faster.

EXAMPLE 5.2 Take $\bar{a}$, $\bar{b}$ and φ constructed in the proof of part (2) of Theorem 3.7, see Subsection 4.5. For each k, we have $i_m^k/i_1^k < (k+1)/k \le 2$, so at each step the number of parts i_j is at most

$$\frac{1}{i_j^k}\left(i_1^k(i_2^kp_1^k-1)+\ldots+i_m^k(i_1^kp_m^k-1)\right) \le 8mi_1^kp_1^k.$$

Therefore, in the speedy algorithm we replace $b_j^k = i_{j-1}^kp_{j-1}^k$ copies of i_j by a_j^k copies of i_{j-1} at most $8m$ times. This means a speed-up by only a constant factor, and the number of steps of the speedy O'Hara's algorithm is $\Omega(n^{m-1-\varepsilon})$ for every $\varepsilon > 0$. $\square$

6. Final remarks

6.1. The polynomial time algorithm in the proof of Theorem 3.10 is given implicitly, by using the general results in integer linear programming. It is saying that the function $\psi : \mathcal{A}_n \to \mathcal{B}_n$ can be computed much faster, by circumventing the elegant construction of O'Hara's algorithm. It would be interesting to give an explicit construction of such an algorithm.

In a different direction, it might prove useful to restate other involution principle bijections in the language of linear programming, such as the Rogers-Ramanujan bijection in [GM2] or in [BP]. If this works, this might lead to a new type of a bijection between these two classes of partitions. Alternatively, this might resolve the conjecture by the second author on the mildly exponential complexity of Garsia-Milne's Rogers-Ramanujan bijection, see [P3, Conjecture 8.5].

6.2. Note the gap between the number $\exp\Theta(\sqrt{n})$ of partitions of n and the lower bound $\mathcal{L}_\varphi(n) = \exp\Omega(\sqrt[3]{n})$ in Theorem 3.7. It would be interesting to decide which of the two worst complexity bounds on the number of steps of O'Hara's algorithm is closer to the truth.

Note that we applied our linear programming approach only in the bounded cycle case. We do not know if there is a way to apply the same technique to the general case. However, we believe that there are number theoretic obstacles preventing that and in fact, computing O'Hara's bijection as a function on partitions may be hard in the formal complexity sense.

6.3. It would be interesting to find φ such that the graph G_φ is a path and $\mathcal{L}_\varphi(n) = \Theta(n\log n)$. From the proof of Lemma 3.5 it follows that the number of steps of (the usual) O'Hara's algorithm is at most $n(\log k + 1)$, where k is the number of steps of the speedy O'Hara's algorithm. In Subsection 5 we constructed an example with $k \sim \sqrt{n/2}$, so it certainly seems possible that such examples exist.

6.4. Most recently, variations on the O'Hara's bijection and applications of rewrite systems were found in [SSM] and [K1, K2]. It would be interesting to see connections between our analysis and this work.

6.5. In the finite dimensional case, the structure of the map ψ establishing approximate
Π-congruence remains largely unexplored. For example, it would be nice to obtain some
convergence result in the irrational case using the rational approximations which follow
from part (3) of Theorem 3.2 and upper bounds in part (2) of Theorem 3.6.

Recall also that the 2-dimensional case can be viewed as the Euclid algorithm which in turn
corresponds to the usual continued fractions (see Example 2.8). Thus the geometry of ψ
can be viewed as a delicate multidimensional extension of continued fractions. Given the
wide variety of (different) multidimensional continued fractions available in the literature,
it would be interesting to see if there is a connection to at least one of these notions.

References

[A]   G. E. Andrews, *The theory of partitions* (Second ed.), Cambridge U. Press, Cambridge, 1998.
[BP]   C. Boulet and I. Pak, A combinatorial proof of the Rogers-Ramanujan identities, *J. Combin. Theory
        Ser. A* **113** (2006), 1019–1030.
[GM1]   A. M. Garsia and S. C. Milne, Method for constructing bijections for classical partition identities,
        Proc. Nat. Acad. Sci. U.S.A. **78** (1981), no. 4, 2026–2028.
[GM2]   A. M. Garsia and S. C. Milne, A Rogers-Ramanujan bijection *J. Combin. Theory Ser. A* **31** (1981),
        289–339.
[G]   B. Gordon, Sieve-equivalence and explicit bijections, *J. Combin. Theory Ser. A* **34** (1983), 90–93.
[K1]   M. Kanovich, Finding direct partition bijections by two-directional rewriting techniques, *Discrete
        Math.* **285** (2004), 151–166.
[K2]   M. Kanovich, The two-way rewriting in action: removing the mystery of Euler-Glaisher's map,
        Discrete Math. **307** (2007), 1909–1935.
[O1]   K. M. O'Hara, *Structure and Complexity of the Involution Principle for Partitions*, Ph.D. thesis,
        UC Berkeley, California, 1984, 135 pp.
[O2]   K. M. O'Hara, Bijections for partition identities, *J. Combin. Theory Ser. A* **49** (1988), 13–25.
[P1]   I. Pak, Partition identities and geometric bijections, *Proc. A.M.S.* **132** (2004), 3457–3462.
[P2]   I. Pak, The nature of partition bijections I. Involutions, *Adv. Applied Math.* **33** (2004), 263–289.
[P3]   I. Pak, Partition bijections, a survey, *Ramanujan J.* 12 (2006), 5–75.
[P4]   I. Pak, The nature of partition bijections II. Asymptotic stability, preprint, 32 pp., available at
        `http://www-math.mit.edu/~pak/`
[PV]   I. Pak and E. Vallejo, Combinatorics and geometry of Littlewood-Richardson cones, *Europ. J.
        Combin.* **26** (2005), 995–1008.
[R]   J. B. Remmel, Bijective proofs of some classical partition identities. *J. Combin. Theory Ser. A*, **33**
        (1982), 273–286.
[S]   A. Schrijver, *Theory of linear and integer programming*, John Wiley, Chichester, 1986.
[SSM]   J. A. Sellers, A. V. Sills and G. L. Mullen, Bijections and congruences for generalizations of partition
        identities of Euler and Guy, *Electronic J. Combin.* **11** (2004), no. 1, RP 43, 19 pp.

DEPARTMENT OF MATHEMATICS, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MA 02139
konvalinka@math.mit.edu
http://www-math.mit.edu/~konvalinka/

SCHOOL OF MATHEMATICS, UNIVERSITY OF MINNESOTA, MINNEAPOLIS, MN 55455   and
DEPARTMENT OF MATHEMATICS, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MA 02139
pak@math.mit.edu
http://www-math.mit.edu/~pak/