Decomposing a Permutation into a Conjugated Tensor Product

Sebastian Egner * Markus Püschel Thomas Beth
(egner@ira.uka.de) (pueschel@ira.uka.de) (eiss-office@ira.uka.de)

Institut für Algorithmen und Kognitive Systeme
Universität Karlsruhe
FRG-76128 Karlsruhe

Abstract

The problem of decomposing a single permutation into a conjugated tensor product of smaller permutations is solved. In general, the decomposition is not uniquely determined. An algorithm is presented which enumerates all solutions. In particular, it is possible to decide considerably fast if a permutation is tensor-indecomposable.

1 Introduction

The central problem solved in this paper is the decomposition of a single permutation into a conjugated tensor product of permutations of smaller degree. Formally,

Problem 1 Given a permutation $\pi \in S_n$, enumerate all permutations σ_1, σ_2, and τ such that π decomposes as

$$\pi \quad = \quad (\sigma_1 \otimes \sigma_2)^\tau$$

where $\sigma^\tau = \tau^{-1}\sigma\tau$ denotes conjugation and $\sigma_1 \otimes \sigma_2$ denotes the tensor product of permutations (def. in next section). □

Our original motivation to factor a permutation into a tensor product comes from the desire to decompose a permutation representation into representations of smaller degree. Among all types of such decompositions (tensor product, direct sum, induction etc.) the tensor product is the most desirable one, since it is most reducing (multiplicative in the degree!). This has applications in the field of digital signal processing since fast algorithms for many well-known signal transformations (like Fourier Transform and Cosine Transform) can be generated based on the decomposition of associated permutation representations ([7], [10], [5]).

Another field of application is the realization of permutations on a VLSI circuit ([6]) and the routing problem in an interconnection network of a parallel computer architecture. In fact, the algorithms presented in this work discover a bus-like structure of a permutation: Consider a column of input signal pads $x_1, \ldots, x_n$, a column of output signal pads $y_1, \ldots, y_n$, and a fixed permutation π connecting the inputs to the outputs. Any factorization $\pi = (\sigma_1 \otimes \sigma_2)^\tau$ corresponds to a realization of the permutation in the form of a permuted bus of permuted busses, after a simultaneous renumbering of the input and output pads. Although the factorized VLSI realization cannot be smaller than a direct one, it is sometimes important to know which bus-structures a permutation gives rise to. Fig. 1 illustrates this geometric interpretation of factorization. In graph theoretical language the factorization replaces a 1-regular group graph by a 2-regular group graph which is a direct product of two 1-regular group graphs. This decomposition can be used for routing in interconnection networks, as treated in [3], [2], and [4].

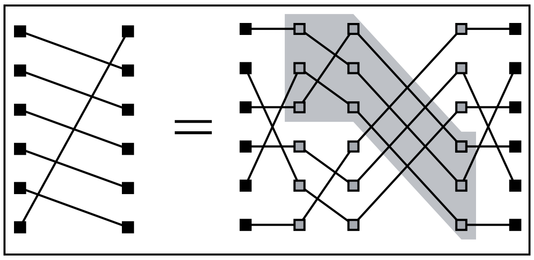

Figure 1: $(1,2,3,4,5,6) = ((1,2) \otimes (1,2,3))^{(2,5)}$

The subject treated in this paper is related to the work [8] of S. P. Glasby (1995) where tensor products of polynomials are considered. In fact, Problem 1 can be restated in terms of the characteristic polynomials of the corresponding permutation matrices. These are products of cyclotomic polynomials. In contrast to [8], we are only interested in factorizations which again yield products of this specific form. Beyond that, we would like to know *all* factorizations, which is rather a combinatorial problem than an algebraic one.

As far as we know, Problem 1 has not been studied before although conjugated tensored permutations are being used in a number of applications as indicated in the preceeding paragraphs. We develop a convenient mathematical description by introducing the ring $\mathcal{P}_*$ (defined in the next section) revealing the algebraic core of Problem 1. The study of this structure, the development of the presented algorithms, and the statements on the computational complexity of the problem are the new results of this paper.

2 Reduction to Cycle Types

The tensor product of permutations is defined via the Kronecker product of the corresponding permutation matrices.

* Grant GRK 209/2-96 "Beherrschbarkeit komplexer Systeme"

Formally, let $\Pi_n = \pi \mapsto [\delta_{i^\pi,j} \mid i,j \in \{1..n\}]$ denote the natural permutation matrix representation of degree n of the symmetric group S_n. Then for $\sigma_1 \in S_{n_1}$ and $\sigma_2 \in S_{n_2}$:

$$\sigma_1 \otimes \sigma_2 = \Pi_{n_1 \cdot n_2}^{-1} \left(\Pi_{n_1}(\sigma_1) \otimes \Pi_{n_2}(\sigma_2) \right) \in S_{n_1 \cdot n_2}.$$

We understand the left operand of the Kronecker product to yield the coarse structure of the result. Hence, $(1,2) \otimes (1,2,3) = (1,5,3,4,2,6)$. In addition to the tensor product, let $\sigma_1 \oplus \sigma_2$ denote the direct sum of permutations, defined by the block diagonal composition of the permutation matrices. For example, $(1,2) \oplus (1,2,3) = (1,2)(3,4,5)$.

Note that we have to distinguish $(1,2) \in S_2$ and $(1,2) \in S_3$. For this end, let $S_* = S_1 \dot\cup S_2 \dot\cup S_3 \dot\cup \cdots$ denote the disjointified union of all symmetric groups and let $\deg(\cdot)$ be the degree operation associated with it. Hence, for all $\pi \in S_*$ the degree $\deg(\pi) = n$ is the unique $n > 0$ such that $\pi \in S_n$. For the purpose of examples we silently drop the degree of permutations and assume absence of trailing fixpoints.

Theorem $(S_*, \oplus)$ is a non-abelian semigroup, $(S_*, \otimes)$ is a non-abelian monoid, and the left and right distributivity law holds.

Since the cycle type of a permutation is invariant under conjugation and the cycle type of a tensor product does only depend on the cycle types of the tensor factors, the problem can be reduced completely to the problem of 'factoring the cycle type'. This shall now be studied.

Given a permutation $\pi \in S_*$, let $\mathrm{ct}(\pi)$ denote the cycle type of π. The cycle type is the multiset of the cycle lengths of π (written as a product of disjoint cycles), including all cycles of length one. In other words, $\mathrm{ct}(\pi)$ is an unordered partition of the natural number $\deg(\pi)$, the degree of π.

We now define structures related to the unordered partitions of an integer. It turned out that one should consider a slightly generalized notion of partitions which allows negative multiplicities, too. A partition f is denoted by the formal sum

$$f = \sum_{k>0}{}' f_k P(k),$$

with the formal symbol $P(\cdot)$ meaning that the natural number k occurs in f with (generalized) multiplicity $f_k \in \mathbb{Z}$. The prime at the summation sign indicates that the sum runs only over finitely many non-zero summands. This means that the support $\mathrm{sp}(f) = \{k \mid f_k \neq 0\}$ of f is finite. Let $\mathcal{P}_*$ denote the set of all the generalized unordered partitions. We will write $f > 0$ iff $f_k \geq 0$ for all k and at least one $f_k > 0$. Let $\mathcal{P}_*^+$ denote the subset of $\mathcal{P}_*$ which contains the $f > 0$.

The set $\mathcal{P}_*$ can be equipped with componentwise addition and a most interesting multiplication: For $f, g \in \mathcal{P}_*$ define

$$\begin{aligned} f + g &= \sum_k{}' (f_k + g_k) \cdot P(k) \quad \text{and} \\ f \cdot g &= \sum_{i,j}{}' f_i\, g_j \, \gcd(i,j) \cdot P(\mathrm{lcm}(i,j)). \end{aligned}$$

Theorem $(\mathcal{P}_*, +, \cdot)$ is a commutative ring with unity. The ring is non-integral.

Proof Clearly, $(\mathcal{P}_*, +)$ is a $\mathbb{Z}$-module. Direct calculation shows associativity of $\cdot$ and the two distributivity laws. The one-element is $P(1)$. An example for divisors of zero is $P(2) \cdot (P(2) - 2P(1)) = 0$. □

Next, let us investigate the connection of $(S_*, \oplus, \otimes)$ and $(\mathcal{P}_*, +, \cdot)$. This also motivates the definition of the special multiplication in $\mathcal{P}_*$.

Theorem The cycle type operator maps the set S_* surjectivly onto $\mathcal{P}_*^+$. It is a semigroup homomorphism with respect to $\oplus$ and $+$ and a monoid homomorphism with respect to $\otimes$ and $\cdot$.

Proof Clearly, $\mathrm{ct}(\cdot)$ is surjective on $\mathcal{P}_*^+$ and $\mathrm{ct}(\sigma \oplus \tau) = \mathrm{ct}(\sigma) + \mathrm{ct}(\tau)$ for all $\sigma, \tau \in S_*$. Hence, considering the cycle decompositions of σ and τ, using distributivity in S_* and invariance under conjugation it is enough to show that $(1,2,\ldots,b) \otimes (1,2,\ldots,c)$ is a $\gcd(b,c)$-fold $\mathrm{lcm}(b,c)$-cycle. We will show this by considering the characteristic polynomials of the corresponding permutation matrices: Given the characteristic polynomials $f_B = \prod_i (X - \beta_i)$ and $f_C = \prod_j (X - \gamma_j)$ of two matrices B and C it is known that the characteristic polynomial of the Kronecker product $B \otimes C$ is $f_{B\otimes C} = \prod_{i,j}(X - \beta_i \gamma_j)$. The characteristic polynomial of (the permutation matrix of) a single b-cycle is $f_b = X^b - 1 = \prod_{i=0}^{b-1}(X - \zeta_b^i)$ where ζ_b is a primitive b-th root of unity, analogous $f_c = X^c - 1 = \prod_{j=0}^{c-1}(X - \zeta_c^j)$ for a primitive c-th root of unity ζ_c. Hence, the characteristic polynomial of the tensor product can be calculated in terms of the primitive $\mathrm{lcm}(b,c)$-th root of unity $\zeta_{\mathrm{lcm}(b,c)}$ as

$$\begin{aligned} f_{b\otimes c} &= \prod_{i,j}(X - \zeta_b^i \zeta_c^j) \\ &= \left(\prod_{k=0}^{\mathrm{lcm}(b,c)-1} (X - \zeta_{\mathrm{lcm}(b,c)}^k) \right)^{\gcd(b,c)} \\ &= (X^{\mathrm{lcm}(b,c)} - 1)^{\gcd(b,c)}, \end{aligned}$$

which represents a $\gcd(b,c)$-fold $\mathrm{lcm}(b,c)$-cycle. □

With the preceding two statements it is possible to solve the original problem, stated at the beginning of this paper, based on the following reduced problem:

Problem 2 Given a $f \in \mathcal{P}_*^+$, enumerate all $g, h \in \mathcal{P}_*^+$ such that $f = g \cdot h$. □

To see why this problem is the core of the original problem, consider a permutation $\pi \in S_n$. Compute the positive factorizations of $f = \mathrm{ct}(\pi)$ in $\mathcal{P}_*^+$ (the reduced problem). For any factorization $f = g_1 \cdot g_2$ one can easily write down some permutations $\sigma_1, \sigma_2 \in S_*$ with $\mathrm{ct}(\sigma_1) = g_1$ and $\mathrm{ct}(\sigma_2) = g_2$. Hence, $\mathrm{ct}(\sigma_1 \otimes \sigma_2) = \mathrm{ct}(\pi)$. Finally, the conjugating permutations τ which map $\sigma_1 \otimes \sigma_2$ into π have to be computed. They form a coset $\tau_1 C_{S_n}(\pi)$ with respect to the the centralizer $C_{S_n}(\pi)$ with representative τ_1. Their construction is a standard task for permutation groups (which are even symmetric groups in this case). Hence, decomposing a single permutation into a conjugated tensor product has been reduced to factorization in the subset $\mathcal{P}_*^+$ of the commutative ring $\mathcal{P}_*$. However, factorization is not unique.

3 Factoring in $\mathcal{P}_*^+$

The method we propose to enumerate the positive factorizations of $f \in \mathcal{P}_*^+$ solves the problem in two steps:

1. Enumerate the supports of potential pairs of factors. Technically speaking, compute a set $\mathcal{S}_f$ such that $gh =$

f implies $[\mathrm{sp}(g), \mathrm{sp}(h)] \in \mathcal{S}_f$ for all g, h. In other words, we formally deduce which cycle lengths do not occur in the factors at all.

2. Enumerate the factorizations of given support. That means, consider all pairs $[G, H] \in \mathcal{S}_f$ and enumerate the pairs $[g, h]$ satisfying $\mathrm{sp}(g) = G$, $\mathrm{sp}(h) = H$ and $gh = f$.

The first subproblem is a concise combinatorial problem which is treated in the next section. The second subproblem is the algebraic problem of solving a system of non-linear diophantine equations of a certain type. It is far easier to solve than the original problem of factoring f since the supports of the two factors g and h are known. This dramatically reduces the number of possibilities. The second problem is solved in section 6.

4 Factoring the Support

Let $f \in \mathcal{P}_*^+$ be given and consider a positive factorization $f = gh$. Then the corresponding supports satisfy

$$\mathrm{sp}(f) \quad = \quad \mathrm{Lcm}\left(\mathrm{sp}(g) \times \mathrm{sp}(h)\right).$$

(Here and in the following, $\mathrm{Lcm}(Z) = \{\mathrm{lcm}(x, y) \mid [x, y] \in Z\}$ is a shorthand notation for the set of l.c.m's of a set Z of pairs of integers). The enumeration of exactly all pairs $[\mathrm{sp}(g), \mathrm{sp}(h)]$ with this property is an elementary number theoretical problem which is not easy to solve in closed form. However, in the special case where $\mathrm{sp}(f)$ consists of prime powers only this is possible, as we will show in the next section.

For the general case, observe that both $\mathrm{sp}(g)$ and $\mathrm{sp}(h)$ can only contain divisors of elements from $\mathrm{sp}(f)$. Hence,

$$\mathrm{sp}(g) \times \mathrm{sp}(h) \quad \subseteq \quad D_f \times D_f,$$

where

$$D_f = \{d \mid \exists t \in \mathrm{sp}(f) : d \text{ divides } t\}.$$

Further restrictions for $\mathrm{sp}(g) \times \mathrm{sp}(h)$ are implied by the coefficients of f: The equation for the k-th coefficient of f can be stated as

$$f_k \quad = \sum_{\substack{i,j \\ \mathrm{lcm}(i,j)=k}} \gcd(i,j) g_i h_j, \quad f_k, g_i, h_j \geq 0.$$

This equation implies the following basic fact: If $g_i, h_j > 0$ then $f_{\mathrm{lcm}(i,j)} \geq \gcd(i, j)$, or turned the other way round, if $f_{\mathrm{lcm}(i,j)} < \gcd(i, j)$ then $g_i = 0$ or $h_j = 0$. Hence, $gh = f$ is possible only if

$$\mathrm{sp}(g) \times \mathrm{sp}(h) \subseteq \{[i, j] \mid f_{\mathrm{lcm}(i,j)} \geq \gcd(i, j)\} \subseteq D_f \times D_f.$$

These restrictions naturally lead to the definition of the set

$$Z_f \quad = \quad \{[i, j] \mid f_{\mathrm{lcm}(i,j)} \geq \gcd(i, j)\}$$

as an upper bound on $\mathrm{sp}(g) \times \mathrm{sp}(h)$ but satisfying the necessary condition $\mathrm{Lcm}(Z_f) = \mathrm{sp}(f)$. Thus, the core problem of factoring the support of f, using part of the information in the coefficients, is the following combinatorial problem:

Problem 3 Consider a set Z of pairs of positive integers. Enumerate all pairs $[X, Y]$ of sets X, Y of positive integers such that

$$X \times Y \subseteq Z \quad \text{and} \quad \mathrm{Lcm}(Z) \subseteq \mathrm{Lcm}(X \times Y).$$

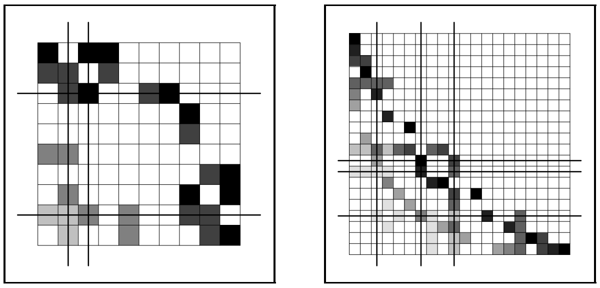

Figure 2: a) Z_{f_3} b) Z_{f_4}

Such a pair $[X, Y]$, or its product $X \times Y$, will be called an *lcm-cover* for Z. □

Let $\mathcal{S}_f$ be the set of lcm-covers of Z_f for a given positive partition f. Then $gh = f$ implies $[\mathrm{sp}(g), \mathrm{sp}(h)] \in \mathcal{S}_f$ for all positive partitions g, h. Hence, the lcm-covers of Z_f contain the possible supports of factorizations.

Example The following examples illustrate that Z_f does not only use $\mathrm{sp}(f)$ as a source of information on the factorizations of f, but also to a certain extent the coefficients f_k.

- $f_1 = 4P(4)$. $Z_{f_1} = \{[1, 4], [4, 1], [2, 4], [4, 2], [4, 4]\}$. The lcm-covers are $\{1\} \times \{4\}$, $\{1, 2\} \times \{4\}$, $\{1, 4\} \times \{4\}$, $\{2, 4\} \times \{4\}$, and $\{1, 2, 4\} \times \{4\}$ and the five mirrored ones.
- $f_2 = 3P(4)$. $Z_{f_2} = \{[1, 4], [4, 1], [2, 4], [4, 2]\}$. Now there are just four lcm-covers, namely $\{1\} \times \{4\}$ and $\{1, 2\} \times \{4\}$ and two mirrored.
- $f_3 = 4P(2) + 2P(6) + 5P(50) + 5P(75)$. The set Z_{f_3} and the particular lcm-cover $\{2, 3\} \times \{2, 25\}$ are shown graphically in Figure 2a). The rows and columns are indexed by $D_{f_3} = [1, 2, 3, 5, 6, 10, 15, 25, 50, 75]$. Grey fields represent $[i, j] \in Z_{f_3}$ with $\mathrm{lcm}(i, j)$ encoded in brightness. Observe that the grid $\{2, 3\} \times \{2, 25\}$ is contained in grey fields and covers all levels of brightness.
- $f_4 = 4P(12) + 4P(16) + P(42) + P(44) + 4P(48) + 2P(112) + P(132) + P(154)$ The lcm-cover $\{3, 11, 16\} \times \{4, 12, 14\}$ of Z_{f_4} is shown in Figure 2b). □

The problem is combinatorial in nature. It is solved by an enumeration method which recursively enumerates the solutions within a lower and an upper bound. In particular, `enumerate`$(Z; L_X, L_Y, U_X, U_Y)$ outputs all lcm-covers $[X, Y]$ of Z which satisfy $L_X \subseteq X \subseteq U_X$ and $L_Y \subseteq Y \subseteq U_Y$. Starting with $L_X = L_Y = \emptyset$ and $U_X = \{x \mid [x, y] \in Z\}$, $U_Y = \{y \mid [x, y] \in Z\}$, all solutions will be computed by the following procedure represented in pseudo-code.

```
enumerate(Z; L_X, L_Y, U_X, U_Y)
  local-variables x_1, y_1;
  if not (
    L_X ⊆ U_X and L_Y ⊆ U_Y and
    L_X × L_Y ⊆ Z and Lcm(Z) ⊆ Lcm(U_X × U_Y)
  ) then
    return;
  if L_X = U_X and L_Y = U_Y then
    output [L_X, L_Y];
```

```
    return;
  if L_Y = U_Y or
    U_X ≠ L_X and splitX?(Z; L_X, L_Y, U_X, U_Y)
  then
    choose x_1 ∈ U_X \ L_X;
    enumerate(Z; L_X, L_Y, U_X \ {x_1}, U_Y);
    enumerate(Z; L_X ∪ {x_1}, L_Y, U_X, U_Y);
    return;
  else
    choose y_1 ∈ U_Y \ L_Y;
    enumerate(Z; L_X, L_Y, U_X, U_Y \ {y_1});
    enumerate(Z; L_X, L_Y ∪ {y_1}, U_X, U_Y);
    return;
```

Here splitX? is a heuristic function to decide whether to split the X-dimension or the Y-dimensions if there is a choice. The procedure enumerate essentially enumerates all $2^{|U_X|+|U_Y|}$ subsets $X \times Y$ of the initial $U_X \times U_Y$. However, the first if-clause prunes an entire branch of the search-tree if there are obviously no solutions between $L_X \times L_Y$ and $U_X \times U_Y$. Hence, the running time of enumerate is worst-case exponential in $|Z|$, but it may be quite fast. As we will show later, there are instances of Z with an exponential number of lcm-covers $[X, Y]$. This shows that there are no faster algorithms in the sense of problem complexity (which ignores constants and improvements for frequent cases). Some technical remarks on actual implementation:

1. The first if-statement may be removed by assuming its condition to be true by induction. For this to work, the relevant part of the precondition has to be tested before each of the recursive calls. Since a special situation is present at these places the test has a reduced strength – it can be tested assuming the precondition (differentially).
2. In enumerate(Z; $L_X \cup \{x_1\}, L_Y, U_X, U_Y$) we may pass $\{y \mid [x_1, y] \in Z\}$ instead of U_Y, since only these y can extend L_Y later on. The analogous thing applies to U_X in the call enumerate(Z; $L_X, L_Y \cup \{y_1\}, U_X, U_Y$).
3. A heuristic strategy for splitX? is to alternate splitting the X- and the Y-dimension. This quickly applies the restrictions present in Z to U_X and U_Y since Z is assumed sparse.
4. A heuristic strategy to choose x_1 and y_1 is to take $x_1 = \max(U_X \setminus L_X)$ (and analogous for y_1). This strategy makes use of the fact that the pairs in Z mostly occupy the lower-left corner of the diagrams as printed above for Z_{f_3} and Z_{f_4}. The very simple strategy seems to perform better than a more complicated strategy we tried: Choose x_1 minimizing the number of pairs in Z with first component x_1.
5. The most time-critical part of enumerate is the test for $\mathrm{Lcm}(Z) \subseteq \mathrm{Lcm}(U_X \times U_Y)$. A suprisingly fast way to decide this question is to maintain the set $D = \mathrm{Lcm}(Z) \setminus \mathrm{Lcm}(L_X \times L_Y)$ differentially and to check for all $d \in D$ if there is a point $[x, y] \in Z \cap U_X \times U_Y$ with $\mathrm{lcm}(x, y) = d$. The latter can be done fast by precomputing sets $Z_d = \{[x, y] \in Z \mid \mathrm{lcm}(x, y) = d\}$ and checking for $Z_d \cap (U_X \times U_Y) = \emptyset$.

5 Factoring the Support, a Special Case

There is a special case of Problem 3 in which the number of possible supports for the factors can be counted in closed form.

Theorem Consider a positive partition f of the form

$$f \;=\; f_{p^{e_1}} P(p^{e_1}) + \cdots + f_{p^{e_r}} P(p^{e_r})$$

where p is some fixed prime number, $0 \le e_1 < \cdots < e_r$ are the exponents, and $f_{p^{e_k}} \ge p^{e_k}$ for all k. (The last means that the coefficients do not impose restrictions on Z_f.) Then the number of lcm-covers $X \times Y$ of Z_f of size $|X| = s$ and $|Y| = t$ is the coefficient of $x^s y^t$ in the polynomial

$$\begin{aligned} G_f(x, y) \;=\; & (x + y + xy)^{r-1} \cdot \\ & \big(y(x+1)^{e_1+1} + x(y+1)^{e_1+1} - (x + y + xy)\big). \end{aligned}$$

In particular, the number of lcm-covers of Z_f is

$$G_f(1, 1) = 3^{r-1}(2^{e_1+1} - 3).$$

Proof Before we give the actual proof let us state some important observations on the result and on the problem.

1. $G_f(x, y)$ does only depend on r and e_1. (corollary).
2. The total number of lcm-covers is exponential in r and in e_1 (corollary).
3. $G_f(x, y)$ is symmetric in x and y since $Y \times X$ is an lcm-cover iff $X \times Y$ is (trivial).
4. The only lcm-cover $X \times Y$ where $X = Y$ is $Z \times Z$. Note that $\mathrm{lcm}(X \times X) = X$ for any set X which consists of p-powers only (trivial, too).

Induction on r. For this end, we write $Z_r = \{p^{e_1}, \ldots, p^{e_r}\}$ for the support of f. G_r denotes the corresponding generating function G_f.

$r = 1$:
Let $[X, Y]$ be an lcm-cover of $Z_1 = \{p^{e_1}\}$. Then $X = \{p^{e_1}\}$ or $Y = \{p^{e_1}\}$. In the case $X = \{p^{e_1}\}$ any non-vacuous subset of $\{1, p, \ldots, p^{e_1}\}$ is a suitable Y. Hence, this case contributes

$$x \sum_{t=1}^{e_1+1} \binom{e_1 + 1}{t} y^t = x\left((y+1)^{e_1+1} - 1\right)$$

to the generating function G_1. Analogous, the case $Y = \{p^{e_1}\}$ contributes

$$y \sum_{s=1}^{e_1+1} \binom{e_1 + 1}{s} x^s = y\left((x+1)^{e_1+1} - 1\right).$$

The common case $X = Y = \{p^{e_1}\}$ has been counted twice this way. This yields

$$G_1 = y(x+1)^{e_1+1} + x(y+1)^{e_1+1} - (x + y + xy).$$

$r > 1$:
Let $[X, Y]$ be an lcm-cover of Z_{r-1}. Then

$$[X \cup \{p^{e_r}\}, Y],\ [X, Y \cup \{p^{e_r}\}],\text{ and } [X \cup \{p^{e_r}\}, Y \cup \{p^{e_r}\}]$$

are different lcm-covers of Z_r. Conversely, if $[X', Y']$ is an lcm-cover of Z_r then $p^{e_r} \in X'$ or $p^{e_r} \in Y'$. Thus, $[X' \setminus \{p^{e_r}\}, Y' \setminus \{p^{e_r}\}]$ is an lcm-cover of Z_{r-1} which implies the recurrence

$$G_r = (x + y + xy) \cdot G_{r-1}.$$

Use of the induction hypothesis concludes the proof. □

The preceding theorem shows that the number of possible factorizations of the support (i.e. the number of solutions to Problem 3) may be exponential in the number of different occuring cycle types. However, a factorization of the support does not necessarily lead to one or more factorizations of the corresponding cycle type. In light of Problem 2 the number of actual factorizations of given degree is the quantity of interest. It is treated in section 7 below.

6 Factoring in $\mathcal{P}_*^+$ with Known Support

Consider a factorization $f = gh$ of the form

$$\begin{aligned} \sum_k f_k P(k) &= \Big(\sum_i g_i P(i)\Big)\Big(\sum_j h_j P(j)\Big) \\ &= \sum_{i,j} \gcd(i,j)\, g_i\, h_j\, P(\mathrm{lcm}(i,j)). \end{aligned}$$

where all $f_k, g_i, h_j \geq 0$. Then the equations for the coefficients may be rearranged into equations for g_k, h_k in terms of f_k and the g_i, h_j for $i, j < k$. The equation for the coefficient of $P(k)$ is

$$\begin{aligned} g_k h_k \cdot \Big\{k\Big\} + g_k \cdot \Big\{\sum_{\substack{j|k \\ j<k}} j h_j\Big\} + h_k \cdot \Big\{\sum_{\substack{i|k \\ i<k}} i g_i\Big\} = \\ = \Big\{f_k - \sum_{\substack{\mathrm{lcm}(i,j)=k \\ i,j<k}} \gcd(i,j) g_i h_j\Big\} \end{aligned}$$

which is a polynomial in g_k and h_k where the expressions in curly braces are non-negative constants. This triangular system of equations can be solved inductively by solving the equations for $k = 1, 2, 3, \ldots$ in turn. Since g_k, h_k are trivially bounded by $\max_i f_i$ the number of solutions is finite.

Solving the system of diophantine equations becomes practical by knowing the support of g and h, as computed in section 4. In fact, only the equations for $k \in \mathrm{sp}(g) \cup \mathrm{sp}(h)$ need to be solved since $g_k = h_k = 0$ for all other k. In addition we know which summands are present in the equation for k. The problem solved here can be stated as

Problem 4 Consider a set $f \in \mathcal{P}_*^+$ and two sets of positive integers G and H. Enumerate all $g, h \in \mathcal{P}_*^+$ such that $\mathrm{sp}(g) = G$, $\mathrm{sp}(h) = H$ and $g \cdot h = f$.

We solve the system of non-linear diophantine equations with positive solutions by considering one equation at a time, enumerating the solutions, and solving the specialized (hence reduced) system recursively. The general form of a single equation to be solved for integers $g_i, h_j > 0$ is

$$a = \sum_i b_i g_i + \sum_j c_j h_j + \sum_{i,j} d_{ij} g_i h_j, \quad a > 0,\, b_i, c_j, d_{ij} \geq 0.$$

For the enumeration of the solutions of a single equation we distinguish various cases which allow nice solutions.

(i) If $a < \sum_i b_i + \sum_j c_j + \sum_{i,j} d_{ij}$ or $\gcd_{i,j}(b_i, c_j, d_{ij}) \nmid a$ then there are no integer solutions $g_i, h_j > 0$.

(ii) If $a = \sum_i b_i + \sum_j c_j + \sum_{i,j} d_{ij} > 0$ then there is the single solution $g_i = h_j = 1$ for all i, j.

(iii) The solutions of $a = g_i h_j$ can be enumerated by running through the divisors of a. After factoring a, which we expect not to exceed a few million, the divisors can be enumerated very fast.

(iv) Equations of the form $a = c_j h_j + \sum_i d_{ij} g_i h_j$ may considered as of the form $a = (c_j + \sum_i d_{ij} g_i) h_j$. Hence, the solutions can be enumerated by running through the divisors h_j of a.

(v) In general, the equation may be considered as of the form $a = BX + CY$ where X, Y are sums of terms in the equation. The constants $B, C > 0$ are to be as large as possible. The new equation is a linear diophantine equation. It may be reduced by dividing off $\gcd(B, C)$, since there is no solution if $\gcd(B, C) \nmid a$. Now assume $\gcd(B, C) = 1$. Then there are integers U, V such that $1 = UB + VC$. The set of integer solutions of the equation is $a[U, V] + \mathbb{Z} \cdot [C, -B]$. Hence, the positive integer solutions may be enumerated by running through $a[U, V] + k[C, -B]$ for $\lceil (1 - aU)/C \rceil \leq k \leq \lfloor (aV - 1)/B \rfloor$.

To make the intention clear: The triangular system of equations stated at the beginning of this section *can* be solved recursively — even without knowing the support of the factors. However, it becomes *fast* if we know the supports in advance (no summand possibly vanishes) and we consider all equations at once to solve the nice ones first. The reason for this is the combinatorial nature of the problem: Solving a single equation generally means enumerating many solutions, one by one. Iterating this recursively results in an exponential growth of possible solutions, which are finally discarded. Solving the nice equations first defeats this growth. (Although we have not been able to prove hard facts on this subject.)

To illustrate the method consider the example f_3 from section 4 above.

Example Factor $f = 4P(2) + 2P(6) + 5P(50) + 5P(75)$ using the particular lcm-cover $\{2, 3\} \times \{2, 25\}$. With $g = g_2 P(2) + g_3 P(3)$ and $h = h_2 P(2) + h_{25} P(25)$ the equation $f = gh$ yields

$$4 = 2g_2 h_2,\ 2 = g_3 h_2,\ 5 = g_2 h_{25},\ 5 = g_3 h_{25}.$$

The last equation is of type (iii) and implies $h_{25} \in \{1, 5\}$. Specializing the equations for $h_{25} = 1$ results in $g_3 = 5$ and $2 = g_3 h_2$, which is unsolvable. Hence, $h_{25} = 5$ and by recursive specialization finally $g_2 = g_2 = 1$ and $h_2 = 2$. □

7 Complexity

In this section the complexity of the factorization problem (Problem 2) is investigated. Throughout this section let $N(f)$ denote the number of positive factorizations of a cycle type $f \in \mathcal{P}_*^+$. It turns out that most f do only posses the two trivial factorizations $f = 1 \cdot f = f \cdot 1$, so $N(f) = 2$ for nearly all f. On the other hand, there are elements f of degree d which give rise to more factorizations than any polynomial in d, as d goes to infinity. In fact, the following lemmas deal with a class of cycle types where $N(f)$ grows like $d^{\theta(\log(d))}$. (We write $g(x) = \Theta(h(x))$ iff there are constants $c_0, c_1 > 0$ and $x_0 > 0$ such that $0 \leq c_0 h(x) \leq g(x) \leq c_1 h(x)$ for all $x > x_0$.) Note that the logarithm is unbounded, so the function eventually grows faster than any polynomial in d. For this end, consider the following special case where the number of factorizations can be counted in closed form. The notation we use for asymptotical behavior follows [9].

Lemma Let $N(f)$ denote the number of positive factorizations of an element $f \in \mathcal{P}_*^+$. Then, for any prime p and

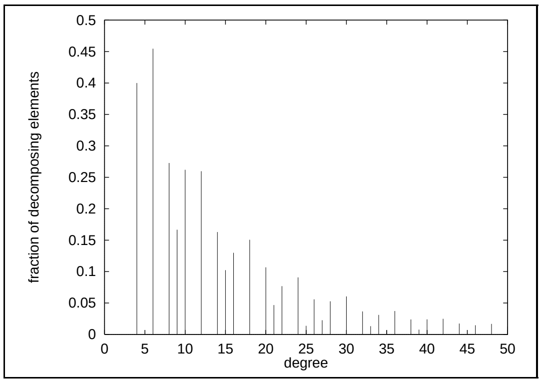

Figure 3: Fraction of decomposing cycle types of degree n.

exponent $r \geq 0$

$$N\left(p^r P(p^r)\right) \quad = \quad 2\sum_{k=0}^{r} m_p(p^k) - 1$$

where $m_p(R)$ denotes the number of $\{1, p, p^2, \ldots\}$-restricted partitions of the number $R > 0$. That is, $m_p(R)$ is the number of sequences $(a_0, a_1, \ldots)$ of integers $a_i \geq 0$ such that $\sum_i a_i p^i = R$.

Proof Let $gh = p^r P(p^r)$ be a positive factorization. Then $\operatorname{sp}(g) = \{p^r\}$ or $\operatorname{sp}(h) = \{p^r\}$. The special case $\operatorname{sp}(g) = \operatorname{sp}(h) = \{p^r\}$ implies $g = h = P(p^r)$. Hence, by commutativity $N(p^r P(p^r)) = 2N_1 - 1$ where N_1 denotes the number of factorizations of the form $p^r P(p^r) = g \cdot h_{p^r} P(p^r)$. Clearly, the coefficient h_{p^r} is a divisor of p^r. Hence, $N_1 = \sum_{k=0}^{r} N_2(k)$ where $N_2(k)$ denotes the number of factorizations of the form $p^r P(p^r) = g \cdot p^k P(p^r)$. Now consider $h_{p^r} = p^k$ and $g = \sum_{i=0}^{r} g_{p^i} P(p^i)$. Then the factorization equations read

$$p^r P(p^r) \quad = \quad g \cdot p^k P(p^r) = \left(\sum_{i=0}^{r} g_{p^i} p^{i+k}\right) P(p^r).$$

Hence, g represents a $\{1, p, \ldots, p^{r-k}\}$-restricted partition of p^{r-k} and $N_2(k) = m_p(p^{r-k})$, which results in the expression for N stated in the lemma. $\square$

The next lemma investigates the asymptotic behavior of $m_p(p^k)$ for fixed $p > 1$ as $k \geq 0$ approaches infinity. For this purpose, let us introduce $M_p(n, k)$ as the number of $\{1, p, \ldots, p^k\}$-restricted partitions of $np^k > 0$. Formally,

$$M_p(n,k) \quad = \quad \#\left\{(a_0, a_1, \ldots, a_k) \,\middle|\, \sum_{i=0}^{k} a_i p^i = np^k,\ a_i \geq 0\right\}.$$

Lemma For integers $p > 1$ and $n, k \geq 0$ the following holds

(i) $m_p(p^k) = M_p(1, k)$.

(ii) $M_p(n, 0) = 1$.

(iii) $M_p(n,k) = \sum_{i=0}^{n} M_p(ip, k-1), \quad$ for $k \geq 1$.

(iv) $\frac{1}{k!} n^k p^{k(k-1)/2} \leq M_p(n,k) \leq n^k p^{(k+1)k/2}$.

Proof (i) by definition. (ii) trivial. (iii) The $(k+1)$-tuple $(a_0, \ldots, a_k)$ is a p-power-restricted partition of np^k if and only if $a_k \in \{0, \ldots, n\}$ and the k-tuple $(a_0, \ldots, a_{k-1})$ is a p-power-restricted partition of $(n - a_k)p \cdot p^{k-1}$. This implies the recursion formula.
(iv) Lower bound: by induction on k. For $k = 0$ the inequality trivially holds, for $k > 0$:

$$\begin{aligned}
M_p(n,k) &= \sum_{i=0}^{n} M_p(ip, k-1) \quad \text{by (iii)} \\
&\geq \sum_{i=0}^{n} \frac{1}{(k-1)!} (ip)^{k-1} p^{(k-1)(k-2)/2} \quad \text{(ind. hyp.)} \\
&= \frac{1}{(k-1)!} p^{k(k-1)/2} \sum_{i=0}^{n} i^{k-1} \\
&\geq \frac{1}{(k-1)!} p^{k(k-1)/2} \cdot \frac{n^k}{k},
\end{aligned}$$

yielding the desired result. For the upper bound let $n > 0$:

$$\begin{aligned}
M_p(n,k) &= \sum_{i=0}^{n} M_p(ip, k-1) \quad \text{by (iii)} \\
&\leq (n+1) M_p(np, k-1) \quad \text{by monotony of } M_p \\
&\leq (n+1)(np+1) M_p(np^2, k-2) \\
&\vdots \\
&\leq (n+1)(np+1)\cdots(np^{k-1}+1) M_p(np^{k-1}, 0) \\
&\leq 2^k n^k p^{k(k-1)/2} \quad \text{using } 1 \leq np^i \text{ for all } i \\
&\leq n^k p^{(k+1)k/2} \quad \text{using } 2 \leq p. \qquad \square
\end{aligned}$$

Now the main statement concerning a lower bound on the complexity of factorization in $\mathcal{P}_*^+$ can be formulated and proved.

Theorem The problem of enumerating all positive factorizations of a given cycle type (Problem 2) is not polynomial time in the degree of the cycle type. Precisely, for a fixed prime p and increasing $r > 0$ the cycle type

$$f \quad = \quad p^r P(p^r)$$

of degree $d = p^{2r}$ gives rise to

$$N(f) \quad = \quad d^{\Theta(\log(d))}$$

many positive factorizations.

Proof By the preceding two lemmas the number $N(f)$ of factorizations of f is bounded by

$$2\sum_{k=0}^{r} \frac{1}{(k-1)!} p^{k(k-1)/2} - 1 \leq N(f) \leq 2\sum_{k=0}^{r} p^{(k+1)k/2} - 1.$$

Weakening the lower bound, we only keep the largest summand, omitting the coefficient 2 and the -1, using $r! \leq r^r$. For the upper bound, all summands are bounded by the largest term. Then omitting -1 and trading $2r$ for $r+1$ yields

$$r^{-r} p^{r(r-1)/2} \leq N(f) \leq 4rp^{(r+1)r/2}.$$

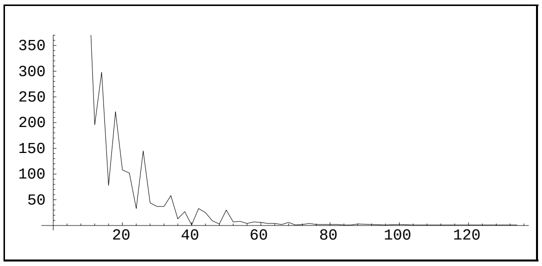

Figure 4: Distribution of $N(f)$ for $\deg(f) = 72$.

Restating these bounds as powers of p gives

$$\frac{1}{2}r^2 - r\log_p(r) - \frac{1}{2}r \le \log_p(N(f)) \le \frac{1}{2}r^2 + \frac{1}{2}r + \log_p(4r).$$

Hence, $\log_p(N(f)) = \Theta(r^2)$ and for $\log_d(x) = \log_p(x)/(2r)$

$$\log_d(N(f)) = \Theta(r) = \Theta(\log(d)). \qquad \square$$

The preceding theorem deals with the worst case of factorization. However, most cycle types do not factor at all. This is illustrated by looking at the fraction of elements which give rise to a non-trivial factorization in relation to the number of all elements of the same degree. Fig. 3 displays this fraction for $1 \le \deg(f) \le 40$, which clearly is zero for prime degree. The next lemma states that the fraction, even exponentially, goes to zero.

Lemma Let $F_n = \{f \in \mathcal{P}_n^+ \mid N(f) > 2\}$ be the set of cycle types of degree $n > 0$ which decompose non-trivially. Then the fraction of these elements goes to zero as

$$|F_n|/|\mathcal{P}_n^+| \;=\; O\left(ne^{-c\sqrt{n}}\right)$$

for an arbitrary constant $0 < c < \pi\sqrt{2/3}$.

Proof Recall the asymptotic behavior of the number of unrestricted partitions of the integer $n > 0$ ([1], 24.2.1.III.):

$$\frac{1}{8\sqrt{3}n}e^{\pi\sqrt{2n/3}} \le |\mathcal{P}_n^+| \le \frac{1}{4\sqrt{3}n}e^{\pi\sqrt{2n/3}}.$$

Every non-trivial factorization $f = gh$ where $g, h \ne 1$ implies that the degrees of g and h are divisors of $n = \deg(f)$ in the range $1 < \deg(g), \deg(h) < n$. Hence,

$$\begin{aligned} |F_n| &\le \sum_{\substack{d \mid n \\ 1<d<n}} |\mathcal{P}_d^+| \cdot |\mathcal{P}_{n/d}^+| \le 2 \sum_{\substack{d \mid n \\ 1<d\le n/2}} |\mathcal{P}_d^+| \cdot |\mathcal{P}_{n/d}^+| \\ &\le (\tau(n) - 2) \cdot |\mathcal{P}_{\lfloor\sqrt{n}\rfloor}^+|^2 \le n \cdot |\mathcal{P}_{\lfloor\sqrt{n}\rfloor}^+|^2 \end{aligned}$$

where $\tau(n)$ denotes the number of divisors of n. Using the bounds on $|\mathcal{P}_n^+|$ and $\sqrt{n}/2 \le \lfloor\sqrt{n}\rfloor \le \sqrt{n}$ yields

$$|F_n|/|\mathcal{P}_n^+| \le \frac{2}{\sqrt{3}}ne^{\pi\sqrt{2/3}\cdot(2n^{1/4} - n^{1/2})} = O\left(ne^{-c\sqrt{n}}\right)$$

for an arbitrary constant $0 < c < \pi\sqrt{2/3}$. $\square$

Finally, have a look at the distribution of the number $N(f)$ of factorizations for given $\deg(f)$. Fig. 4 displays the distribution for $\deg(f) = 72$. However, please bear in mind that the distribution for other $\deg(f)$ is largely influenced by the number theoretic nature of the factorization problem. For example, if $\deg(f)$ is prime then $N(f) = 2$.

8 Conclusion and Acknowledgements

The problem of decomposing a single permutation into a conjugated tensor product of two permutations of lower degree is treated (Problem 1). It can be reduced to factoring the cycle type as a positive element in the ring $\mathcal{P}_*^+$ of generalized partitions, equipped with componentwise addition and a special multiplication which corresponds to the tensor product of cycle types (Problem 2). Factorization is not unique since the ring $\mathcal{P}_*^+$ is not integral. In particular, the special case where the support consists of powers of a fixed prime gives rise to a large number of factorizations, as stated by a counting theorem above. An example is given, where the number of factorizations is not polynomial in the degree.

The problem of enumerating the positive factorizations of a positive element in $\mathcal{P}_*^+$ is solved in two steps. First, the possible supports for the two factors have to be enumerated (Problem 3). This is accomplished by a recursive tree-transversal algorithm. Second, all factorizations of the given element are enumerated such that the factors have a specified support (Problem 4). This problem is solved by recursively solving one equation at a time and solving the specialized system recursively. The approach enumerates all factorizations without any need to store the resulting list of factorizations. All algorithms presented have been implemented in the programming language GAP, [11].

The authors would like to thank Martin Rötteler and Michael Schmid for reading the proofs and discussing the subject. In addition, we would like to thank the referees for their constructive comments.

References

[1] Abramowitz, M., and Stegun, I. A. *Handbook of mathematical functions*, 9 ed. Dover Publications, New York, 1972.

[2] Akers, S. B., and Krishnamurthy, B. A group-theoretic model for symmetric interconnection networks. *IEEE Transactions on Computers 38*, 4 (1989), 555–566.

[3] Annexstein, F., and Baumslag, M. A unified approach to off-line permutation routing on parallel networks. In *2nd Annual ACM Symposium on Parallel Algorithms and Architectures, Island of Crete, Greece* (1990).

[4] Annexstein, F., Baumslag, M., and Rosenberg, A. L. Group action graphs and parallel architectures. *SIAM Journal on Computing 19*, 3 (1990), 544–569.

[5] Beth, T. *Methoden der Schnellen Fouriertransformation.* Teubner Verlag, 1984.

[6] Brady, M. L., Brown, D. J., and McGuinness, P. J. Disjoint paths through a 3-dimensional grid. In *2nd Annual ACM Symposium on Parallel Algorithms and Architectures, Island of Crete, Greece* (1990).

[7] Clausen, M., and Baum, U. *Fast Fourier Transforms.* BI-Wiss.-Verl., 1993.

[8] Glasby, S. P. Tensor products of polynomials. Tech. rep., School of Mathematics and Statistics, University of Sydney, 1995.

[9] Graham, R. L., Knuth, D. E., and Patashnik, O. *Concrete Mathematics.* Addison-Wesley, 1992.

[10] MINKWITZ, T. *Algorithmensynthese für lineare Systeme mit Symmetrie.* PhD thesis, Universität Karlsruhe, 1993.

[11] SCHÖNERT, M. ET AL. GAP — groups, algorithms and programming. Tech. rep., Lehrstuhl D für Mathematik, Rheinisch Westfälische Technische Hochschule, Aachen, Germany., fifth ed., 1995.