This is the author's final version, of the article published in **Evolutionary Computation** 31(3) : 163-199(2023) doi: https://doi.org/10.1162/evco_a_00315.

Characterizing Permutation-Based Combinatorial Optimization Problems in Fourier Space

Anne Elorza[1], Leticia Hernando[2], and Jose A. Lozano[1,3]

[1]Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU, 20018 San Sebastián, Spain
[2]Intelligent Systems Group, Department of Mathematics, University of the Basque Country UPV/EHU, 48940 Leioa, Spain
[3]Basque Center for Applied Mathematics (BCAM), 48009 Bilbao, Spain

Abstract

Comparing combinatorial optimization problems is a difficult task. They are defined using different criteria and terms: weights, flows, distances, etc. In spite of this apparent discrepancy, on many occasions, they tend to produce problem instances with similar properties. One avenue to compare different problems is to project them onto the same space, in order to have homogeneous representations. Expressing the problems in a unified framework could also lead to the discovery of theoretical properties or the design of new algorithms. This paper proposes the use of the Fourier transform over the symmetric group as the tool to project different permutation-based combinatorial optimization problems onto the same space. Based on a previous study (Kondor, 2010), which characterized the Fourier coefficients of the quadratic assignment problem, we describe the Fourier coefficients of three other well-known problems: the symmetric and non-symmetric traveling salesman problem and the linear ordering problem. This transformation allows us to gain a better understanding of the intersection between the problems, as well as to bound their intrinsic dimension.

Keywords

Combinatorial Optimization Problems, Fourier Transform, Permutations, Representation theory, Intrinsic dimension

1 Introduction

The field of permutation-based combinatorial optimization includes a number of problems with very different formulations and meanings. For instance, although the linear ordering problem (LOP) (Martí and Reinelt, 2011), the traveling salesman problem (TSP) (Laporte, 1992), the quadratic assignment problem (QAP) (Loiola et al., 2007) and the permutation flowshop scheduling problem (PFSP) (Framinan et al., 2004) are usually defined over the permutation space via matrices, the meaning of the elements of these matrices is different according to the problem. As an example, a problem can address distances between cities, cope with flows between factories, or concern tasks on machines. In spite of having a completely distinct definition, in practice, two instances of two different problems could present very similar properties, or they could even turn out to be the same instance (if they provide the same objective function values for the solutions). In order to discover these similarities between instances, it could be useful to project all of these problems onto the same space. The Fourier domain seems to be a suitable working framework to achieve this purpose, since any objective function is univocally determined by its Fourier coefficients.

Indeed, Fourier analysis produces a decomposition of a given function into a sum of smaller pieces, from which the original function can be recovered. Whereas its real-line counterpart is universally-known in the area of applied mathematics (Körner, 1989), at the present time, this is not the case of the Fourier transform (FT) over the space of permutations, namely, the symmetric group (Terras, 1985, 1999, 2012). Despite being substantially less renowned than its real analogue, the FT over permutations has recently been gaining attention in the computer science field, giving rise to new proposals of applications and algorithms (Huang et al., 2009; Kondor, 2008), some of which deal with areas such as object tracking or analyzing ranking data. A survey with applications of the generalized FT from the point of view of group representations can be found at Rockmore (2004). Even though its employment has also reached the area of combinatorial optimization (Kondor, 2010; Rockmore et al., 2002) and its boolean twin, the Walsh transform, has also been considered (Christie, 2016; Goldberg, 1989), the use of Fourier analysis still remains limited. However, the FT seems to offer a suitable space for combinatorial optimization problems (COP) to be treated in a homogeneous way, via their Fourier decomposition. The starting point of our research is the observation of (Kondor, 2010; Rockmore et al., 2002), where it is proved that, for an objective function of a QAP, only four of its Fourier coefficients are non-zero. It should be noted that, even though the classic Fourier coefficients of real-valued functions are scalars, in the case of permutation-based functions, the coefficients are stored in matrices. Then, the four Fourier coefficients that can be non-zero in the QAP are four matrices, and not just four real values. The knowledge that only a few matrix coefficients of the QAP can be non-zero can lead to a deeper understanding of the problem. The total number of elements stored in the matrix Fourier coefficients grows factorially with the dimension of the problem, while the number of elements stored in the non-zero matrix Fourier coefficients of the QAP grows polinomially. So, for high dimensions, the family of QAP functions is quite small in comparison with the whole set of permutation-based functions (whose Fourier coefficients can take any value).

The work presented in this paper has a theoretical nature, although it could eventually lead to improvements in the resolution of permutation-based COPs. For instance, Kondor (2010) designed a branch-and-bound algorithm for the QAP that fully operates on Fourier space. Our first contribution is that we characterize the Fourier coefficients of the LOP and both the symmetric and the non-symmetric version of the TSP. That is, given an LOP or a TSP function, we find properties that its Fourier coefficients must necessarily fulfill. This characterization permits us to obtain relevant theoretical information, which is our second contribution. On the one hand, we observe that the LOP, the TSP and the QAP functions can be reparametrized using their Fourier representation. Through this new representation, they can be defined using a lower number of parameters than those given by the input matrices. This implies that their intrinsic dimension is lower than the one suggested by their usual definition. On the other hand, we approach the intersection between problems by studying whether we can find sets of functions that are at the same time instances of two different problems.

The rest of the paper is organized as follows. Section 2 motivates the study presented in this paper, by outlining a variety of related research questions, implications and uses. Sections 3 and 4 introduce the necessary mathematical background regarding the FT and permutation-based COPs, respectively. In Section 5, we firstly summarize the results already present in the literature regarding the Fourier characterization of the QAP. Secondly, we characterize the LOP and the TSP in Fourier space. Sections 6 and 7 are devoted to the study of the intrinsic dimensions of the COPs and the intersections between problems, which can be approached thanks to the Fourier characterizations found in Section 5. Section 8 gathers the conclusions.

2 Motivation

The FT works similarly to elementary landscape decomposition (Whitley et al., 2008; Chicano et al., 2011), in the sense that both of them decompose the fitness function into a sum of orthogonal components. The elementary landscape decomposition of a variety of COPs has been extensively studied and, given a neighborhood, it produces a decomposition of a fitness function into a sum of functions, which have the so-called property of being elementary landscapes with respect to the given neighborhood. Each of these elementary landscapes has a number of properties which explain the interest that this kind of decomposition has generated (Whitley et al., 2008). For example, for each of the elementary landscapes, given the function value of a solution, the average value of the neighbors of the solution can be computed in $O(1)$. Another interesting property is the one that states that the objective function value of a local maximum or minimum must be respectively greater or lower than the mean value of the objective function. These properties, among others, can be exploited in the design of local search or evolutionary algorithms (Benavides et al., 2021; Ceberio et al., 2019). However, both decompositions differ in the fact that the first is associated with a specific neighborhood. We have observed that the Fourier characterizations of the problems studied in this paper have an interesting property. If the lowest coefficient is discarded, which is the one that represents the mean of the function, the number of non-zero Fourier coefficients coincides with the number of components of the elementary landscape decompositions found in the literature. For example, the QAP can be decomposed into three elementary landscapes under the 2-exchange neighborhood (Chicano et al., 2011), the LOP can be decomposed into two under the 2-exchange neighborhood (Ceberio et al., 2019), the symmetric TSP is itself an elementary landscape under the 2-exchange and 2-opt neighborhoods (Stadler, 1996) and the general TSP has two components under the 2-exchange and 2-opt neighborhoods (Stadler, 1996). We conjecture that the Fourier decomposition could be related to elementary landscape decomposition, and that it could be possible to find a neighborhood such that each of the Fourier components is an elementary landscape with respect to the given neighborhood. Regarding this topic, there has also been proposed an optimization strategy for COPs which makes use of elementary landscape decomposition combined with multi-objectivization (Ceberio et al., 2019). This same study could be undertaken considering the Fourier decomposition instead of elementary landscape decomposition, which would lead to a general method for multi-objectivizing COPs.

One of the benefits of characterizing COPs in Fourier space is that all of them are expressed under the same framework. This can facilitate the comparison between different problems as well as the transfer of strategies and knowledge from one problem to another. For instance, one may wonder what the relationship between problem complexity and Fourier coefficients is. For example, in Elorza et al. (2022), taking the LOP as a case study, it is shown that a Fourier coefficient is associated with a P problem, whereas the other is associated with an NP-hard problem. Taking this into account, this work analizes how the behaviour of constructive algorithms degrades as the Fourier coefficients transit from those of a P problem to those of an NP-hard problem. This kind of studies could help in the classification of instance difficulty for common algorithms, as well as in the construction of easy or hard to solve instances.

Another relevant question arises when, instead of fitness functions, one considers rankings of solutions of the search space. Many heuristic algorithms, such as local search or evolutionary algorithms that use tournament or ranking selection operators, do not make use of the exact value of the fitness function of each solution. Instead, they operate using comparisons between the objective function values. The performance of this type of algorithms depends solely on the ranking of the solutions. Therefore, one may wonder what types of rankings are generated by different COPs (which is a research question that has been approached in Hernando et al. (2019)). Thanks to the Fourier characterization of the QAP, it has been proved that, for $n = 4$, the QAP does not generate all possible rankings (Elorza et al., 2019). It could be useful to know what types of rankings are generated by different Fourier coefficients. This information could eventually be used to establish a matching between the Fourier coefficients of problem instances and algorithms, in such a way that a given instance is assigned an algorithm that solves it efficiently.

These are a number of uses that the Fourier characterization of COPs could have. More generally, the FT that we are considering is the permutation version of the Walsh transform, which operates on pseudo-Boolean functions. Therefore, it could be interesting to see whether the studies that have been published for the Walsh transform, such as the use of surrogate functions based on it (Swingler, 2020; Verel et al., 2018), could be extended to permutations as well.

3 Fourier transform on the symmetric group

The following lines offer a brief overview of the FT, which is mainly based on Huang et al. (2009). In addition, the interested reader may refer to Sagan (2013) for a deeper insight into the precise algebraic concepts. Since this is a general introduction to the subject, the most technical details, which are needed to prove part of the theorems of Section 5, can be found in Appendix A.1.

The FT on the symmetric group, Σ_n, comes from a generalization of the well-known transform on the real line to finite groups. Thus, in an initial encounter, it may be useful to understand it in contrast with the real case. Given a function $f : \Sigma_n \longrightarrow \mathbb{R}$, the FT decomposes f into certain coefficients by means of a set of base functions (on the real line, sines and cosines). When working with Σ_n, the base is composed of the *irreducibles* of Σ_n, a set of functions whose image is invertible matrices. These are formally defined in the context of the representation theory.

Definition 1 (Representation)**.** *$\rho : \Sigma_n \longrightarrow GL_m$ is a representation if it preserves the group product, that is, $\rho(\sigma_1\sigma_2) = \rho(\sigma_1) \cdot \rho(\sigma_2)$ for all $\sigma_1, \sigma_2 \in \Sigma_n$. GL_m is the set of invertible complex-valued matrices. The size m of the matrices is also called the dimension of the representation.*

Example 1 (Trivial representation)**.** *The trivial representation, noted by $\rho_{(n)}$, is the constant function $\rho_{(n)}(\sigma) = [1]$, for all $\sigma \in \Sigma_n$.*

Example 2 (Permutation representation)**.** *The first-order permutation representation, noted by $\tau_{(n-1,1)}$, maps σ to its permutation matrix, that is, $[\tau_{(n-1,1)}(\sigma)]_{ij} = \mathbb{1}_{\{\sigma:\sigma(j)=i\}}(\sigma)$. For $n = 3$, the permutation representation $\tau_{(2,1)}$ would be the following:*

$$\tau_{(2,1)}(123) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \tau_{(2,1)}(213) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \tau_{(2,1)}(132) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

$$\tau_{(2,1)}(321) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \tau_{(2,1)}(231) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \tau_{(2,1)}(312) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Two different operations, the equivalence and the direct sum, can be defined, which allow new representations to be built from existing ones.

Definition 2 (Equivalence)**.** *Two representations, ρ_1 and ρ_2, are equivalent if there exists an invertible matrix C such that*

$$\rho_2(\sigma) = C^{-1} \cdot \rho_1(\sigma) \cdot C \qquad \forall \sigma \in \Sigma_n,$$

which is denoted by $\rho_1 \equiv \rho_2$.

Definition 3 (Direct sum)**.** *Given two representations, ρ_1 and ρ_2, their direct sum is the representation $\rho_1 \oplus \rho_2$ that satisfies*

$$\rho_1 \oplus \rho_2(\sigma) = \left[\begin{array}{c|c} \rho_1(\sigma) & 0 \\ \hline 0 & \rho_2(\sigma) \end{array}\right].$$

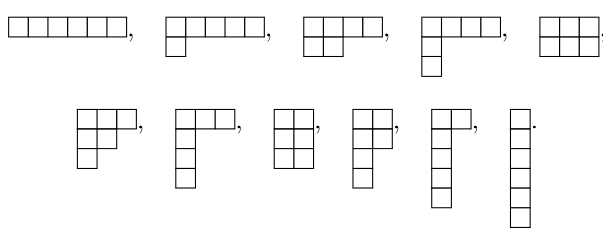
A representation is said to be reducible if it can be "decomposed" into smaller pieces through the direct sum and the equivalence.

Definition 4 (Reducibility)**.** *A representation ρ is reducible if there exist two representations ρ_1 and ρ_2 such that $\rho \equiv \rho_1 \oplus \rho_2$. Otherwise, it is irreducible.*

For a finite group G, the set of irreducible representations (up to equivalence) is finite. This implies that one can select a finite number of representations and build up the rest from them, through the operations of equivalence and direct sum. In addition, in the case of the symmetric group Σ_n, it has been proved that the number of inequivalent irreducible representations is the number of partitions of n.

Definition 5 (Partition of a natural number)**.** *Let $\lambda = (\lambda_1, \cdots, \lambda_k)$ be a tuple where the elements, known as parts, are in decreasing order, that is $\lambda_i \geq \lambda_j$ for $i < j$. Then, λ is a partition of n if the parts sum to n, that is $\sum_{i=1}^{k} \lambda_i = n$. This is denoted by $\lambda \vdash n$.*

Example 3. *The partitions of $n = 6$ are $(6), (5,1), (4,2), (4,1,1), (3,3), (3,2,1), (3,1,1,1),$ $(2,2,2), (2,2,1,1), (2,1,1,1,1)$ and $(1,1,1,1,1,1)$. These partitions can be graphically represented by the so-called Ferrer's diagrams, by placing a λ_i number of squares in the i-th row. The Ferrer's diagrams corresponding to the partitions of $n = 6$ are illustrated as follows:*



As previoulsy mentioned, if one chooses a set of (inequivalent) irreducible representations of Σ_n, also called *system of irreps*, it can be used to reconstruct any representation via the operations of equivalence and direct sum. In addition, the system of irreps can be indexed by the partitions of n, taking the following shape: $\{\rho_{(n)}, \rho_{(n-1,1)}, ..., \rho_{(1,1,\cdots,1)}\}$. The dimension d_λ of each irreducible representation ρ_λ depends on λ. It can be computed using a theorem named the Hook Formula (the reader is addressed to Sagan (2013) for further information). The specific set of irreducible representations that is used in this paper is called Young's Orthogonal Representations (YOR), which is the canonical one (for further information, see Kondor (2010)). The interesting property of these representations is that the matrices $\rho_\lambda(\sigma)$ are real-valued and orthogonal. Once a set of irreducibles $\{\rho_\lambda : \lambda \vdash n\}$ is established, a function f can be projected onto this base via the FT, which produces its Fourier coefficients.

Definition 6 (FT at a representation)**.** *The FT of a function $f : \Sigma_n \longrightarrow \mathbb{R}$ at a representation ρ, also called Fourier coefficient at ρ, is defined as*

$$\hat{f}_\rho = \sum_{\sigma \in \Sigma_n} f(\sigma)\rho(\sigma). \tag{1}$$

As can be seen from the definition, $\hat{f}_\rho$ is a matrix whose dimension is the same as the dimension of $\rho(\sigma)$ (for any σ).

Definition 7 (FT of a function)**.** *Given a system of irreps, $\{\rho_\lambda : \lambda \vdash n\}$, the FT of a function f is defined as the collection of the Fourier coefficients at each of the irreducibles of the system:*

$$\{\hat{f}_{\rho_\lambda} : \lambda \vdash n\}.$$

Whenever we allude to Fourier coefficients without specifying the representations, it can be assumed that it refers to irreducible representations. The FT of a function f at irreducible ρ_λ is denoted by $\hat{f}_{\rho_\lambda}$. Even so, we may simplify this notation to $\hat{f}_\lambda$ for readability purposes and whenever this does not lead to ambiguity. Note again that, unlike the real case, the coefficients are not simply numerical values, they are stored in matrices, instead. Even so, several of the properties on the real line, such as invertibility (Theorem 1), linearity and the convolution theorem still hold for the symmetric group.

Theorem 1 (Inverse FT)**.** *A function f can be computed in terms of its Fourier coefficients according to the following formula:*

$$f(\sigma) = \frac{1}{|\Sigma_n|} \sum_{\lambda \vdash n} d_\lambda \operatorname{Tr}\left[\hat{f}^T_{\rho_\lambda} \cdot \rho_\lambda(\sigma)\right],$$

where d_λ is the dimension of the representation ρ_λ and Tr *denotes the trace of a matrix.*

Let us recall that the Fourier coefficients and the image of the irreducible representations are matrices, so, in this formula, $\hat{f}_{\rho_\lambda}$ and $\rho_\lambda(\sigma)$ are square matrices of dimension d_λ. The inversion theorem is important because it implies that there is a one-to-one correspondence between functions and Fourier coefficients. A function f has certain Fourier coefficients $\{\hat{f}_{\rho_\lambda} : \lambda \vdash n\}$, which can be computed via the FT. At the same time, f can be recovered, via the inversion formula, from the set $\{\hat{f}_{\rho_\lambda} : \lambda \vdash n\}$. In other words, the Fourier space offers an alternative way of representing functions defined over the symmetric group.

4 Permutation-based combinatorial optimization problems

In the field of permutation-based combinatorial optimization, the aim is to optimize an *objective function* $f : \Sigma_n \longrightarrow \mathbb{R}$, that is, to find

$$\operatorname*{argmin}_{\sigma \in \Sigma_n} f(\sigma),$$

if we are minimizing.

Analogously, maximization problems could be defined, but they are not considered in this paper. Since we aim at comparing problems, all of the studied problems must be expressed in the same terms. In this sense, we focus on the minimization version of three different problems: the QAP, LOP and TSP.

4.1 QAP

In the QAP, a set of n facilities has to be assigned to n locations. The aim is to reduce the cost of the flow between the facilities, which depends on the distances between their locations. If $A = [a_{ij}]$ is the distance matrix, which measures the distance between locations, and $A' = [a'_{ij}]$ is the flow matrix, which measures the flow between facilities, then the QAP consists of minimizing the following function:

$$f(\sigma) = \sum_{i \neq j} a_{\sigma(i)\sigma(j)} \cdot a'_{ij}, \tag{2}$$

where $\sigma(k)$ represents the location to which facility k is assigned. We assume in the rest of the paper that the diagonal elements of A and A' are zero.

As we will immediately see, the other two problems studied in this paper can be reformulated as particular cases of the QAP, if the values of matrix A' are properly fixed.

4.2 LOP

The LOP is given by a square matrix $A = [a_{ij}]$ of size n, and consists of finding the joint permutations of rows and columns that maximize the sum of the upper-diagonal elements, or equivalently, minimize the sum of the lower-diagonal elements. Even though the maximization version is the one usually found in literature, we consider the minimization one, in which the problem consists of minimizing the following function:

$$f(\sigma) = \sum_{j=1}^{n-1} \sum_{i=j+1}^{n} a_{\sigma(i)\sigma(j)}, \tag{3}$$

where $\sigma(k)$ denotes the number of the row/column of the original matrix which is located in the k-th position.

By setting

$$a'_{ij} = \begin{cases} 1 & \text{if } i > j \\ 0 & \text{otherwise,} \end{cases}$$

the objective function of the QAP becomes the objective function of the LOP.

4.3 TSP

The TSP consists of finding the shortest route that a traveling salesperson should take to visit a number of cities once and then come back to the starting point. The distances between the cities are given by a square matrix $A = [a_{ij}]$ of size n, where n is the number of cities. The problem consists of minimizing the following function:

$$f(\sigma) = a_{\sigma(n)\sigma(1)} + \sum_{i=1}^{n-1} a_{\sigma(i)\sigma(i+1)}, \tag{4}$$

where $\sigma(i)$ denotes the number of the city visited in the i-th place. If A is a symmetric matrix, the problem is called symmetric traveling salesman problem (STSP).

By setting

$$a'_{ij} = \begin{cases} 1 & \text{if } j = i+1 \text{ or } (i = n \text{ and } j = 1) \\ 0 & \text{otherwise,} \end{cases}$$

the objective function of the QAP becomes the objective function of the TSP.

5 Characterizing problems in Fourier domain

Finding an accurate characterization of the Fourier coefficients of COPs is one of the fundamental goals of our research. This means that we are interested in describing the properties that all the objective functions associated with a certain problem, such as the LOP, the TSP or the QAP, have in common. Characterizing the problems in Fourier space could be interesting since we could compare different problems, with disparate definitions, by looking at the similarities and differences between their Fourier coefficients.

In this vein, and having as a starting point the observations that have already been made on the Fourier coefficients of the QAP (Kondor, 2010; Rockmore et al., 2002), we study the Fourier representation of the LOP and the TSP. We find definite conditions that the LOP and both the symmetric and non-symmetric TSP functions must satisfy (Theorems 3, 4 and 5).

5.1 Characterization of the QAP (Kondor, 2010; Rockmore et al., 2002)

The starting point of our research is the work of Kondor (2010); Rockmore et al. (2002) on the Fourier representation of the QAP. Thanks to these studies, the following theorem can be stated:

Theorem 2 (FT of the QAP)**.** *If $f : \Sigma_n \longrightarrow \mathbb{R}$ is the objective function of a QAP instance, that is, f is expressed as in (2), then its FT has the following properties:*

1. $\hat{f}_\lambda = 0$*, if* $\lambda \neq (n), (n-1,1), (n-2,2), (n-2,1,1)$.
2. $\hat{f}_\lambda$ *has at most rank one for* $\lambda = (n-2,2), (n-2,1,1)$.
3. $\hat{f}_\lambda$ *has at most rank two for* $\lambda = (n-1,1)$.

Let us introduce an example of how the Fourier coefficients of the QAP can be obtained for a reduced dimension.

Example 4. *Consider the QAP instance of dimension $n = 3$ whose input matrices are the following:*

$$A = \begin{bmatrix} 0 & 4 & -2 \\ 5 & 0 & 1 \\ 8 & 3 & 0 \end{bmatrix} \quad \text{and} \quad A' = \begin{bmatrix} 0 & -3 & 1 \\ 4 & 0 & 2 \\ -5 & 3 & 0 \end{bmatrix}.$$

Then, the objective function f of this instance is calculated (according to Equation (2)) and shown in Table 1.

The Fourier coefficients of f can be computed through Equation (1). A function defined over Σ_3 has three matrix Fourier coefficients: $\hat{f}_{(3)}$, $\hat{f}_{(2,1)}$ and $\hat{f}_{(1,1,1)}$. In order to calculate them, the values of the irreducible representations $\rho_{(3)}$, $\rho_{(2,1)}$ and $\rho_{(1,1,1)}$ are needed. The values of these representations are shown in Table 2. $\rho_{(3)}$ is the trivial representation and $\rho_{(1,1,1)}$ is the signature function.

Table 1: *Values of the objective function f of the QAP instance of Example 4.*

σ	$f(\sigma)$
[1 2 3]	-12
[1 3 2]	13
[2 1 3]	13
[2 3 1]	-5
[3 1 2]	-29
[3 2 1]	26

Table 2: Values of the irreducible representations of Σ_3, as shown in Huang et al. (2009).

σ	$\rho_{(3)}(\sigma)$	$\rho_{(2,1)}(\sigma)$	$\rho_{(1,1,1)}(\sigma)$
[1 2 3]	[1]	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	[1]
[1 3 2]	[1]	$\begin{bmatrix} 1/2 & \sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix}$	[-1]
[2 1 3]	[1]	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	[-1]
[2 3 1]	[1]	$\begin{bmatrix} -1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix}$	[1]
[3 1 2]	[1]	$\begin{bmatrix} -1/2 & \sqrt{3}/2 \\ -\sqrt{3}/2 & -1/2 \end{bmatrix}$	[1]
[3 2 1]	[1]	$\begin{bmatrix} 1/2 & -\sqrt{3}/2 \\ -\sqrt{3}/2 & -1/2 \end{bmatrix}$	[-1]

Taking into account the values shown in Table 2 and the formula of the FT,

$$\hat{f}_{(3)} = \sum_{\sigma \in \Sigma_3} f(\sigma)\rho_{(3)}(\sigma) = \sum_{\sigma \in \Sigma_3} f(\sigma) \cdot [1] =$$
$$[-12] + [13] + [13] + [-5] + [-29] + [26] = [6].$$
$$\hat{f}_{(2,1)} = \sum_{\sigma \in \Sigma_3} f(\sigma)\rho_{(2,1)}(\sigma) = -12 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + 13 \cdot \begin{bmatrix} 1/2 & \sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix} + 13 \cdot \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$
$$- 5 \cdot \begin{bmatrix} -1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix} - 29 \cdot \begin{bmatrix} -1/2 & \sqrt{3}/2 \\ -\sqrt{3}/2 & -1/2 \end{bmatrix} + 26 \cdot \begin{bmatrix} 1/2 & -\sqrt{3}/2 \\ -\sqrt{3}/2 & -1/2 \end{bmatrix} =$$
$$\begin{bmatrix} \frac{23}{2} & -\frac{37}{2}\sqrt{3} \\ \frac{11}{2}\sqrt{3} & -\frac{3}{2} \end{bmatrix}.$$
$$\hat{f}_{(1,1,1)} = \sum_{\sigma \in \Sigma_3} f(\sigma)\rho_{(1,1,1)}(\sigma) = \sum_{\sigma \in \Sigma_3} f(\sigma) \cdot [\mathrm{sgn}(\sigma)] =$$
$$- 12 \cdot [1] + 13 \cdot [-1] + 13 \cdot [-1] - 5 \cdot [1] - 29 \cdot [1] + 26 \cdot [-1] = [-98].$$

This example does not have any coefficient equal to zero because, for a dimension of $n = 3$, the Fourier coefficients of f ($f_{(3)}, f_{(2,1)}$ and $f_{(1,1,1)}$) are the ones which are not necessarily zero according to Theorem 2. However, if higher dimensions are considered, zero-valued matrices appear among the coefficients. As an example, if a QAP objective function f with a space dimension of $n = 4$ is considered, there are 5 possible Fourier coefficients: coefficients (4), $(3,1)$, $(2,2)$, $(2,1,1)$ and $(1,1,1,1)$. Among these coefficients, coefficient $(1,1,1,1)$ must satisfy $\hat{f}_{(1,1,1,1)} = [0]$.

Taking these results already found in the literature as a basis, the next sections provide a step forward in detailing the Fourier coefficients of the LOP and the symmetric and non-symmetric TSP.

5.2 Characterization of the LOP

The aim of this section is to enumerate the properties that the Fourier coefficients of any LOP function must fulfill. These are given in Theorem 3. The proof requires a number of intermediate steps, which are stated in the form of propositions.

The first step consists of reducing the problem. Kondor (2010) proved (by means of the convolution theorem) that the Fourier coefficients of a QAP can be expressed in terms of the distance data and the flow data. The statement of this result requires the definition of a new concept: graph functions. The *graph function* associated with a matrix A is defined as

$$f_A(\sigma) = a_{\sigma(n)\sigma(n-1)}.$$

Kondor's result is the following, which is stated without proof because it is already given in Kondor (2010).

Proposition 1. *(Kondor, 2010) If $f : \Sigma_n \longrightarrow \mathbb{R}$ is the objective function of a QAP instance, that is, f is expressed as in (2), then its FT has the following properties:*

1. *$\hat{f}_\lambda = 0$, if $\lambda \neq (n), (n-1,1), (n-2,2), (n-2,1,1)$.*
2. *The values of coefficients $\lambda = (n-1,1), (n-2,2)$ and $(n-2,1,1)$ can be factored in terms of the coefficients of the graph functions of A and A':*

$$\hat{f}_\lambda = \frac{1}{(n-2)!} \hat{f}_{A\lambda} \cdot \hat{f}_{A'\lambda}^{T}.$$

Since the LOP is a particular case of the QAP, this factorization can be applied to the LOP as well.

Proposition 2. *If $f : \Sigma_n \longrightarrow \mathbb{R}$ is the objective function of an LOP instance, that is, f is expressed as in (3), then its FT has the following properties:*

1. *$\hat{f}_\lambda = 0$, if $\lambda \neq (n), (n-1,1), (n-2,2), (n-2,1,1)$.*
2. *The values of coefficients $\lambda = (n-1,1), (n-2,2)$ and $(n-2,1,1)$ can be factored as the following product:*

$$\hat{f}_\lambda = \frac{1}{(n-2)!} \hat{f}_{A\lambda} \cdot \hat{f}_{A'\lambda}^{T}, \tag{5}$$

where f_A is the graph function of A and

$$f_{A'} = \mathbb{1}_{\{\sigma:\, \sigma(n-1) < \sigma(n)\}}.$$

Proof. The LOP is a particular case of the QAP. By setting

$$a'_{ij} = \begin{cases} 1 & \text{if } j < i \\ 0 & \text{otherwise,} \end{cases}$$

Equation (2) becomes the objective function of the LOP. Then,

$$f_{A'}(\sigma) = a'_{\sigma(n)\sigma(n-1)} = \begin{cases} 1 & \text{if } \sigma(n-1) < \sigma(n) \\ 0 & \text{otherwise,} \end{cases}$$

which is the indicator function over the set $\{\sigma : \; \sigma(n-1) < \sigma(n)\}$. At this point, the result immediately follows from Propostion 1.

□

According to Proposition 2, if f is an LOP function, its FT is proportional to (the proportionality constant being $1/(n-2)!$) the product of matrix $\hat{f}_{A\lambda}$, whose values depend on the instance, and matrix $\hat{f}_{A'\lambda}^{\,T}$, which remains the same for any instance of size n. The next step is to see the specific properties that $\hat{f}_{A'\lambda}$, the constant factor that the Fourier coefficients of any LOP share, fulfills. This is in order to deduce how it affects the whole set of LOP functions.

Mania et al. (2018) proved certain properties that the FT of the so-called Kendall kernel satisfies. The following proposition is an immediate result of the exposition of Mania et al. (2018), so the proof has not been included in this paper. To see the details, the reader is refered to the aforementioned paper.

Proposition 3. *The FT of* $f_{A'}(\sigma) = \mathbb{1}_{\{\sigma:\, \sigma(n-1)<\sigma(n)\}}$ *satisfies*

1. $\hat{f}_{A'\lambda} = 0$ *if* $\lambda \neq (n), (n-1,1), (n-2,1,1)$.

2. $\hat{f}_{A'\lambda}$ *has rank one for* $\lambda = (n-1,1), (n-2,1,1)$.

Having stated Propositions 2 and 3, all the necessary pieces to prove Theorem 3 have been presented.

Theorem 3 (FT of the LOP)**.** *If* $f : \Sigma_n \longrightarrow \mathbb{R}$ *is the objective function of an LOP instance, that is,* f *is expressed as in (3), then its FT has the following properties:*

1. $\hat{f}_\lambda = 0$*, if* $\lambda \neq (n), (n-1,1), (n-2,1,1)$.

2. $\hat{f}_\lambda$ *has at most rank one for* $\lambda = (n-1,1), (n-2,1,1)$. *Having rank one is equivalent to the fact that the matrix columns are proportional.*

3. *For* $\lambda = (n-1,1), (n-2,1,1)$ *and a fixed dimension* n*, the proportions among the columns of* $\hat{f}_\lambda$ *are the same for all the instances.*

Proof. Taking into account Equation (5), it is clear that the Fourier coefficients of an LOP function are the product of two matrices multiplied by $1/(n-2)!$. One of them, $\hat{f}_{A\lambda}$, depends on the values of the input matrix A. According to Proposition 3, the other one, $\hat{f}_{A'\lambda}^{\,T}$, has rank one for $\lambda = (n-1,1), (n-2,1,1)$ and is 0 for any other partition except for $\lambda = (n)$. A basic result of linear algebra states that, given two matrices A and B,

$$\text{rank}(AB) \leq \min\{\text{rank}(A), \text{rank}(B)\}.$$

This means that the rank of the product of two matrices is at most the lowest of the ranks of both matrices. Hence, when multiplying matrices $\hat{f}_{A\lambda}$ and $\hat{f}_{A'\lambda}^{T}$, the resulting rank must be less than or equal to 1.

Secondly, to prove that the proportions among columns are fixed for a given dimension n, observe the following: If $\lambda = (n-1,1)$ or $\lambda = (n-2,1,1)$, then $\hat{f}_{A'\lambda}$ has rank 1, and so does its transpose $\hat{f}_{A'\lambda}^{T}$. This means that all the rows and columns are proportional. Thus, we can write all the columns of $\hat{f}_{A'\lambda}^{T}$ proportionally to the first column, which we denote by $\mathbf{v}$, while we denote the proportions by α_i for $i = 2, \ldots, d_\lambda$. Therefore,

$$\hat{f}_{A'\lambda}^{T} = \begin{bmatrix} \mathbf{v} & \alpha_2 \mathbf{v} & \cdots & \alpha_{d_\lambda} \mathbf{v} \end{bmatrix}.$$

If $\hat{f}_{A\lambda}$ is expressed as follows:

$$\hat{f}_{A\lambda} = \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \cdots \\ \mathbf{w}_{d_\lambda}^T \end{bmatrix},$$

where $\mathbf{w}_1, \cdots, \mathbf{w}_{d_\lambda}$ are arbitrary column vectors of length d_λ, then

$$\hat{f}_\lambda = \frac{1}{(n-2)!} \cdot \hat{f}_{A\lambda} \cdot \hat{f}_{A'\lambda}^{T} = \frac{1}{(n-2)!} \cdot \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \cdots \\ \mathbf{w}_{d_\lambda}^T \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v} & \alpha_2 \mathbf{v} & \cdots & \alpha_{d_\lambda} \mathbf{v} \end{bmatrix}$$

$$= \frac{1}{(n-2)!} \cdot \begin{bmatrix} \mathbf{w}_1^T \cdot \mathbf{v} & \mathbf{w}_1^T \cdot \alpha_2 \mathbf{v} & \cdots & \mathbf{w}_1^T \cdot \alpha_{d_\lambda} \mathbf{v} \\ \mathbf{w}_2^T \cdot \mathbf{v} & \mathbf{w}_2^T \cdot \alpha_2 \mathbf{v} & \cdots & \mathbf{w}_2^T \cdot \alpha_{d_\lambda} \mathbf{v} \\ \vdots & \vdots & & \vdots \\ \mathbf{w}_{d_\lambda}^T \cdot \mathbf{v} & \mathbf{w}_{d_\lambda}^T \cdot \alpha_2 \mathbf{v} & \cdots & \mathbf{w}_{d_\lambda}^T \cdot \alpha_{d_\lambda} \mathbf{v} \end{bmatrix}.$$

For $\lambda = (n-1,1), (n-2,1,1)$, all the columns of $\hat{f}_\lambda$ are proportional to the first one and its proportions are equal to those in $\hat{f}_{A'\lambda}^{T}$. Let us recall that matrix $\hat{f}_{A'\lambda}^{T}$ does not depend on the instance and is fixed for a given dimension n. This is the reason why the proportions among columns are the same for any instance.

□

Note that there exist LOP functions with coefficients $(n-1,1)$ or $(n-2,1,1)$ equal to zero. For example, the constant function $f(\sigma) = 1$ is an LOP function with $\hat{f}_{(n-1,1)} = 0$ and $\hat{f}_{(n-2,1,1)} = 0$. In this case, the proportions among columns could be any number, since 0 is proportional to 0 with any proportion factor. The constant proportions that are mentioned in Theorem 3 are those of $\hat{f}_{A'(n-1,1)}$ and $\hat{f}_{A'(n-2,1,1)}$, which appear in any LOP function f whose Fourier coefficients $\hat{f}_{(n-1,1)}$ and $\hat{f}_{(n-2,1,1)}$ are non-zero.

Having found such a precise condition, one may wonder whether the reciprocal is true: if one defines a function f as the inverse of certain Fourier coefficients that satisfy the conditions described in the theorem, will f be an LOP? To clarify this question, we designed the following experiment for dimensions $n = 3, 4, 5, 6$:

1. Generate a random LOP function g. Repeat this process until g has non-null $(n-1,1)$ and $(n-2,1,1)$ coefficients. Then, compute $\hat{g}_\lambda$ for $\lambda = (n-1,1)$, $(n-2,1,1)$ and extract the column proportions.

2. Generate random Fourier coefficients $\hat{f}_\lambda$ following the patterns shown in Theorem 3.

   - For this purpose, firstly set $\hat{f}_\lambda = 0$, for $\lambda \neq (n), (n-1,1), (n-2,1,1)$.
   - Generate a uniformly random number in the interval $(-1,1)$ for $\hat{f}_{(n)}$.
   - For $\lambda = (n-1,1), (n-2,1,1)$, build coefficient $\hat{f}_\lambda$ in the following way: firstly, generate a random column of size d_λ (using a uniform distribution in $(-1,1)$). This is the first column of the matrix. The other columns are proportional to the first, with the same proportions as in $\hat{g}_\lambda$, obtained in step 1.

3. Define function f as the inverse of the coefficient family $\{\hat{f}_\lambda : \lambda \vdash n\}$ via the inverse FT.

4. Check whether there exists an input matrix for the LOP such that the objective function associated with that matrix is f.

The details for implementing the last step of the experiment are explained in Appendix B, where the problem is reduced to analyzing the solvability of a certain linear system. We run 1000 repetitions of the experiment for each of the dimensions $n = 3, 4, 5, 6$. The result was, invariably, that the inverse function was an LOP.

This experiment indicates that, probably, for dimensions $n = 3, 4, 5, 6$, the implication stated in Theorem 3 is actually an equivalence. We hypothesize that this could be generalized for any dimension n or, in other words, that Theorem 3 gathers the properties that exactly characterize the LOP in Fourier domain, as we express in the shape of the following conjecture.

Conjecture 1. *If f is an LOP function with non-null $(n-1,1)$ and $(n-2,1,1)$ Fourier coefficients, and a function g satisfies the conditions mentioned in Theorem 3, that is,*

1. $\hat{g}_\lambda = 0$, *for* $\lambda \neq (n), (n-1,1), (n-2,1,1)$.
2. $\hat{g}_{(n-1,1)}$ *is 0 or rank-one with the same column proportions as* $\hat{f}_{(n-1,1)}$.
3. $\hat{g}_{(n-2,1,1)}$ *is 0 or rank-one with the same column proportions as* $\hat{f}_{(n-2,1,1)}$.

Then, g is the objective function of an LOP instance.

5.3 Characterization of the TSP

As for the previous case, we find conditions that the Fourier coefficients of non-symmetric and symmetric TSP functions have to meet, which are stated in Theorems 4 and 5. The proofs require more steps than in the case of the LOP, so, for readability purposes, they are developed in Appendix A.2.

Theorem 4 (FT of the TSP). *If $f : \Sigma_n \longrightarrow \mathbb{R}$ is the objective function of a TSP instance, that is, f is expressed as in (4), then its FT has the following properties:*

1. $\hat{f}_\lambda = 0$, *if* $\lambda \neq (n), (n-2,2), (n-2,1,1)$.

2. $\hat{f}_\lambda$ *has at most rank one for* $\lambda = (n-2,2),(n-2,1,1)$*. Having rank one is equivalent to the fact that the matrix columns are proportional.*

3. *For* $\lambda = (n-2,2),(n-2,1,1)$ *and a fixed dimension* n*, the proportions among the columns of* $\hat{f}_\lambda$ *are the same for all the instances.*

Theorem 5 (FT of the STSP)**.** *If* $f : \Sigma_n \longrightarrow \mathbb{R}$ *is the objective function of an STSP instance, that is,* f *is expressed as in (4) and its input matrix A is symmetric, then its FT has the following properties:*

1. $\hat{f}_\lambda = 0$*, if* $\lambda \neq (n),(n-2,2)$*.*

2. $\hat{f}_\lambda$ *has at most rank one for* $\lambda = (n-2,2)$*. Having rank one is equivalent to the fact that the matrix columns are proportional.*

3. *For* $\lambda = (n-2,2)$ *and a fixed dimension* n*, the proportions among the columns of* $\hat{f}_\lambda$ *are the same for all the instances.*

As for the reciprocal implication, we have proceeded in a similar fashion to the case of the LOP. We generated random Fourier coefficients satisfying the properties of Theorems 4 and 5, applied the inverse FT to them and checked whether the resulting function was a TSP or an STSP, respectively. We repeated this process, both for the TSP and for the STSP, 1000 times and for dimensions $n = 3,4,5,6$. The results were always positive: the function obtained in the experiments was the corresponding TSP or STSP. So we hypothesize, again, that Theorems 4 and 5 hold the exact characterization of the TSP and the STSP. Therefore, as for Conjecture 1 in the case of the LOP, equivalent conjectures can be stated for the TSP and the STSP:

Conjecture 2. *If* f *is a TSP function with non-null* $(n-2,2)$ *and* $(n-2,1,1)$ *Fourier coefficients, and a function* g *satisfies the conditions mentioned in Theorem 4, that is,*

1. $\hat{g}_\lambda = 0$*, for* $\lambda \neq (n),(n-2,2),(n-2,1,1)$*.*

2. $\hat{g}_{(n-2,2)}$ *is 0 or rank-one with the same column proportions as* $\hat{f}_{(n-2,2)}$*.*

3. $\hat{g}_{(n-2,1,1)}$ *is 0 or rank-one with the same column proportions as* $\hat{f}_{(n-2,1,1)}$*.*

Then, g *is the objective function of a TSP instance.*

Conjecture 3. *If* f *is an STSP function with a non-null* $(n-2,2)$ *Fourier coefficient, and a function* g *satisfies the conditions mentioned in Theorem 5, that is,*

1. $\hat{g}_\lambda = 0$*, for* $\lambda \neq (n),(n-2,2)$*.*

2. $\hat{g}_{(n-2,2)}$ *is 0 or rank-one with the same column proportions as* $\hat{f}_{(n-2,2)}$*.*

Then, g *is the objective function of an STSP instance.*

6 Intrinsic dimensions of COPs

This section concerns the intrinsic (or parametric) dimension of COPs, which is the minimum number of parameters needed to define an instance of a problem. Note that n, the length of the permutations used to codify the solutions, is a different type of dimension.

Thanks to the Fourier inversion formula (see Theorem 1), we know that a function can be recovered from its Fourier coefficients, which means that we have an alternative representation of the function in Fourier domain. In general, the number of parameters needed to define a permutation-based function in Fourier domain would be $n!$, which is the same number needed to define the function by choosing the value that it takes at each permutation. However, as has been observed, the Fourier coefficients of the problems analyzed have some special properties or, in other words, they have to meet certain restrictions. This means that (given a solution dimension n), if one wants to define an LOP, a TSP or a QAP instance in Fourier space, many of the parameters that were tunable for general functions now remain fixed. This idea can be captured by the concept of intrinsic dimension.

Although the term of intrinsic dimension has been used in other optimization scenarios, we have not found it specifically applied to combinatorial optimization. For instance, in the field of Bayesian optimization, Wang et al. (2016) observe that there are problems in which most dimensions do not significantly change the value of the objective function, and this can be exploited in the optimization strategy. The authors define the effective dimensionality of a function as the number of dimensions that contribute to the value of a function $f : \mathbb{R}^p \longrightarrow \mathbb{R}$. This notion is not exactly the one that we are considering, as it is straightforwardly related with the dimension of the solutions. In our case, we refer to COPs, and to the intrinsic dimension of these problems, which we define as follows.

Most combinatorial optimization problems are defined by means of a subset of parameters of size p_n when the solution dimension is n. A value for each of the p_n parameters produces a different instance and, therefore, a different function f. Given a COP with a set of solutions of size n, X, it is possible to define a function F that assigns to each set of parameters the generated objective function in the COP.

$$F : \mathbb{R}^{p_n} \longrightarrow \{f \mid f : X \longrightarrow \mathbb{R} \text{ is a function, i.e., an instance of the COP}\}.$$

For example, in the case of the LOP, F would map the elements of the entry matrix to the function that assigns to each permutation the sum of the lower-diagonal elements of the matrix when the rows and columns are jointly reordered. Then, the COP given by F can be reparametrized to

$$G : \mathbb{R}^{k_n} \longrightarrow \{f \mid f : X \longrightarrow \mathbb{R} \text{ is a function, i.e., an instance of the COP}\}$$

if there exists a function $\phi : \mathbb{R}^{p_n} \longrightarrow \mathbb{R}^{k_n}$ such that $F(\mathbf{a}) = G(\phi(\mathbf{a}))$, for any vector of parameters $\mathbf{a} \in \mathbb{R}^{p_n}$. $\phi(\mathbf{a})$ would be the new vector of parameters, which has k_n elements. If such a ϕ exists, then the COP given by F, which depends on p_n parameters, can be expressed by k_n parameters through G, which means that the COP has an alternative representation which depends on a different number of parameters (if $k_n \neq p_n$). Then, the *intrinsic dimension* of a COP would be the lowest dimension k_n for which there exists a reparametrization of F.

The number of parameters needed to define any instance using the standard representation (for example, via input matrices, as in the case of the LOP, TSP or QAP) is, by default, an upper bound of the instrinsic dimension of the problem. However, if we reparametrize the problems using the Fourier domain, the number of parameters needed changes, and so does the upper bound of the intrinsic dimension, as we will immediately see. Table 3 lists the dimensions of the irreducibles $\rho_{(n)}, \rho_{(n-1,1)}, \rho_{(n-2,2)}$ and $\rho_{(n-2,1,1)}$, which are the ones needed to compute the FT of the COPs studied in this paper.

Intrinsic dimension of the LOP. The only non-zero coefficients of the LOP are those indexed by $\lambda = (n)$, $(n-2,1,1)$ and $(n-1,1)$. Taking into account Table 3 and the

Table 3: Dimensions of the irreps d_λ according to the partitions of n (Kondor, 2010).

λ	d_λ
(n)	1
$(n-1,1)$	$n-1$
$(n-2,2)$	$\frac{n(n-3)}{2}$
$(n-2,1,1)$	$\frac{(n-1)(n-2)}{2}$

fact that the ranks of coefficients $\lambda = (n-2,1,1)$ and $(n-1,1)$ are 1, that is, all the columns are proportional to the first one in both coefficient matrices, the number of possible tunable values is just the number of rows. Thus, the intrinsic dimension of the LOP is, at most:

$$1 + \frac{(n-1)(n-2)}{2} + (n-1) = \frac{n(n-1)}{2} + 1.$$

Intrinsic dimension of the TSP. The only non-zero coefficients of the TSP are those indexed by $\lambda = (n)$, $(n-2,1,1)$ and $(n-2,2)$. Again, taking into account Table 3 and that the ranks of coefficients $\lambda = (n-2,1,1)$ and $(n-2,2)$ are 1, the number of possible tunable values is just the number of rows of the coefficient matrices. Thus, the intrinsic dimension of the TSP is, at most:

$$1 + \frac{(n-1)(n-2)}{2} + \frac{n(n-3)}{2} = (n-1)(n-2).$$

Intrinsic dimension of the STSP. For the STSP, coefficient $\lambda = (n-2,1,1)$ is also zero. Therefore, the intrinsic dimension of the STSP is, at most:

$$1 + \frac{n(n-3)}{2} = \frac{(n-1)(n-2)}{2}.$$

Intrinsic dimension of the QAP. The only non-zero coefficients of the QAP are those indexed by $\lambda = (n)$, $(n-2,1,1)$, $(n-1,1)$ and $(n-2,2)$. Once again, taking into account Table 3 and knowing that the rank of coefficient $\lambda = (n-1,1)$ is at most 2, and the rank of coefficients $\lambda = (n-2,1,1)$ and $(n-2,2)$ is at most 1, the intrinsic dimension of the QAP would be upper bounded by the following expression (note that in this case, the proportions among columns are not fixed):

$$1 + 2(n-1) + 2(n-1-2) + \frac{n(n-3)}{2} + \frac{n(n-3)}{2} - 1 + \frac{(n-1)(n-2)}{2} + \frac{(n-1)(n-2)}{2} - 1 = 2n^2 - 2n - 7.$$

To sum up, Table 4 gathers the number of parameters needed to define the different COPs considering both the usual representation (via input matrices) and the Fourier representation. In the case of the LOP, the number of parameters when the problem is reparametrized using the Fourier space is approximately half of the number of parameters needed with the usual representation. This is significantly lower. In the case of the ATSP and the STSP, the difference is not so remarkable, but it is still noticeable. In the case of the QAP, the difference is fixed, only of 7 parameters. As can be seen, we have obtained lower bounds for the intrinsic dimensions of the problems than those suggested by their usual parametrization. In this sense, it has been proven that any instance of the COPs studied in this paper can be defined by a lower number of parameters than the number of elements that appear in the input matrices.

Table 4: Number of parameters needed to define the different COPs considering both the usual representation (via input matrices) and the Fourier representation. The COPs taken into account are the LOP, TSP, STSP and QAP.

COP	Usual representation	Fourier representation
LOP	$n^2 - n$	$\frac{n^2-n}{2} + 1$
TSP	$n(n-1)$	$(n-1)(n-2)$
STSP	$\frac{n(n-1)}{2}$	$\frac{(n-1)(n-2)}{2}$
QAP	$2(n^2 - n)$	$2(n^2 - n) - 7$

The following three examples show how certain instances that are codified differently in the usual representation give rise to the same objective function. In addition, the second matrix of each of the examples has the same number of non-zero values as the number of parameters of the Fourier representation.

Example 5. *Let A and B be the following input matrices of two LOP instances:*

$$A = \begin{bmatrix} 0 & 4 & 7 & 10 \\ 1 & 0 & 8 & 11 \\ 2 & 5 & 0 & 12 \\ 3 & 6 & 9 & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 & 3 & 5 & 7 \\ 0 & 0 & 3 & 5 \\ 0 & 0 & 0 & 29 \\ 0 & 0 & 26 & 0 \end{bmatrix}.$$

Then, both of the instances have the same objective function.

Example 6. *Let C and D be the following input matrices of two TSP instances:*

$$C = \begin{bmatrix} 0 & 2 & 5 & 6 \\ 3 & 0 & 4 & 8 \\ 7 & 3 & 0 & 5 \\ 11 & 4 & 2 & 0 \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} 0 & 0 & -2 & 2 \\ 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 \\ 22 & 19 & 12 & 0 \end{bmatrix}.$$

Then, both of the instances have the same objective function.

Example 7. *Let E and F be the following input matrices of two STSP instances:*

$$E = \begin{bmatrix} 0 & 3 & 1 & 5 \\ 3 & 0 & 4 & 7 \\ 1 & 4 & 0 & 2 \\ 5 & 7 & 2 & 0 \end{bmatrix} \quad \text{and} \quad F = \begin{bmatrix} 0 & 0 & 8 & 9 \\ 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 5 \\ 9 & 0 & 5 & 0 \end{bmatrix}.$$

Then, both of the instances have the same objective function.

7 Intersection between COPs

Characterizing COPs in Fourier space opens up the possibility of comparing problems that originally have completely distinct representations, by looking at their Fourier coefficients. In this section, we specifically study the intersections between the LOP, TSP and QAP.

Once a COP is understood as a set of functions, we can view the intersection between two COPs as the subset composed of the functions that belong to both sets. It is understood that a function f belongs to two different COPs if there exist instances of both problems that provide the same objective function values as f. As previously mentioned, the LOP and the TSP are particular cases of the QAP. Thus, the intersection between the LOP and the QAP is the LOP, while the intersection between the TSP and the QAP is the TSP.

In order to analyze the intersection between the LOP and the TSP, let us consider the characterization of the Fourier coefficients of these problems. On the one hand, if a function is an LOP, it must have all the Fourier coefficients equal to zero, except for the ones indexed by $\lambda = (n)$, $(n-2,1,1)$ and $(n-1,1)$. On the other hand, if the function is at the same time a TSP, it must have all the Fourier coefficients equal to zero, except for the ones indexed by $\lambda = (n)$, $(n-2,1,1)$ and $(n-2,2)$. Then, in this case, the only non-zero coefficients that both problems share are those indexed by $\lambda = (n)$ and $(n-2,1,1)$. Thus, the intrinsic dimension of this intersection is, at most,

$$1 + \frac{(n-1)(n-2)}{2}.$$

However, it is necessary, for this intersection not to be trivial, that the proportions between the columns of coefficient $(n-2,1,1)$ of the LOP and the TSP (see Theorems 3 and 4) are the same. We have experimentally checked that this happens for dimension $n = 4$, but the same does not hold for $n = 5, 6, 7, 8, 9, 10$. Then, for dimensions $n = 5, 6, 7, 8, 9, 10$, we can assure that, for a function that belongs both to the LOP and the TSP, coefficient $(n-2,1,1)$ must be zero. This means that, for these dimensions, the functions that belong to the intersection of the LOP and the TSP are only composed of coefficient (n), which implies that they must be constant functions. Based on this observation, we conjecture that, for $n > 4$, the intersection between the LOP and the TSP is trivial, in the sense that it is composed of constant functions.

In the case of the intersection between the LOP and the STSP, the only Fourier coefficient that both problems have in common is the one corresponding to $\lambda = (n)$. Therefore, we can conclude that the only functions that belong at the same time to both problems are the constant functions. The intrinsic dimension in this case is 1.

8 Conclusions

Based on Kondor (2010), we have proposed a framework from which COPs can be studied: the Fourier domain. We have characterized the LOP and the TSP in this little-known space, and we have also included the characterization of the QAP of Kondor (2010) in order to further analyse and compare the three COPs. In the case of the STSP, the FT has two non-zero Fourier coefficients, while, in the case of the LOP and the TSP, there are three non-zero Fourier coefficients and, finally, in the case of the QAP, there are four. As the dimension, n, grows, so does the number of Fourier coefficients of a function, and, for the COPs that we have studied, the number of functions that can be generated with those two, three or four non-zero coefficients is minimal in comparison with the whole set of permutation-based functions.

One of the consequences of the Fourier characterizations of the COPs considered in this paper is that the intrinsic dimensions of the problems are lower than the dimensions suggested by their usual parametrization, which means that they can be defined by a lower number of parameters. In addition, the FT allows us to express the different problems using the same representation, instead of having disparate definitions depending on how the input matrices are interpreted. This can have a number of advantages. Firstly, it is easier to compare instances of different problems, and we have made an example of this by studying the intersection between problems. We have proved that the intersection between the LOP and the STSP is trivial, and have bounded the intersection between the LOP and the TSP, which we conjecture, based on empirical evidence, that it is also trivial for $n > 4$. To the best of our knowledge, this topic has not been addressed before. Nevertheless, it is true that this issue has already been approached from a different point of view. Many heuristic algorithms do not take into account the exact objective function value of the solutions, but only a comparison between them. This is the reason why the intersection between problems has been studied in terms of rankings of the solutions of the search space (Hernando et al., 2019). As future work, we would like to study the connection between rankings and Fourier coefficients, and use this information to analyze the intersection of rankings between the LOP and the TSP. We believe that this can be relevant because, if two problems A and B share many rankings, then algorithms that are efficient for problem A, and which do not take into account the exact objective function value, would also perform well on B.

Future work may also envision developing the characterization theorems presented here (Theorems 2, 3, 4 and 5), and prove Conjectures 1, 2 and 3. That is, we would like to prove that a function whose Fourier coefficients have the shapes described in the theorems must necessarily be a QAP, an LOP or a TSP. It could be interesting to add new problems to our research, such as the PFSP. Another possible research line would be the one that relates Fourier coefficients to problem complexity. Regarding this topic, many relevant research questions arise. For instance, one may try to determine the minimal Fourier structure of a COP so that it results in an NP-hard problem.

Acknowledgments This work is supported by the Basque Government (BERC 2022-2025 and IT1504-22) and by the Spanish Ministry of Economy and Competitiveness MINECO (projects PID2019-104966GB-I00 and PID2019-106453GA-I00/AEI/10.13039/501100011033). Jose A. Lozano acknowledges support by the Spanish Ministry of Science, Innovation and Universities through BCAM Severo Ochoa accreditation (SEV-2017-0718). Anne Elorza holds a predoctoral grant (ref. PIF17/293) from the University of the Basque Country.

Appendix

This appendix has two sections. Section A contains the proofs of the theorems stated in Section 5.3. Section B explains how to code the functions needed to carry out the experiments mentioned in Section 5.

A Proofs of Section 5.3

This appendix contains the proofs of Theorems 4 and 5, which characterize the Fourier coefficients of the TSP. Part of the mathematical background about the FT and the COPs has already been introduced in Sections 3 and 4, respectively. However, more technical background is needed for the mathematical proof of the theorems.

A.1 Further mathematical notation and details on the FT

Regarding the FT, let us recall that the Fourier coefficients of a function are indexed by the partitions of n. In addition, for a given partition $\lambda \vdash n$, the rows and columns of $\hat{f}_\lambda$ are indexed by *standard tableaus* of shape λ (a standard tableau is a Young tableau in which numbers $1, 2, \cdots, n$ are placed exactly once and in increasing order both from left to right and from top to bottom). For example, the standard tableaus of shape $\lambda = (2, 1)$ are just the following two:

$$\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 & \multicolumn{1}{c}{} \\ \cline{1-1} \end{array} \qquad \begin{array}{|c|c|}\hline 1 & 3 \\ \hline 2 & \multicolumn{1}{c}{} \\ \cline{1-1} \end{array}$$

Apart from Young tableaus, there also exist *Young tabloids*, which, intuitively, are Young tableaus in which the elements in the same row are not sorted. Young tabloids have a slightly different representation:

$$\begin{array}{cc}\hline 1 & 2 \\ \hline 3 & \\ \cline{1-1} \end{array} \qquad \begin{array}{cc}\hline 1 & 3 \\ \hline 2 & \\ \cline{1-1} \end{array} \qquad \begin{array}{cc}\hline 2 & 3 \\ \hline 1 & \\ \cline{1-1} \end{array}$$

We assume that, for a given partition $\lambda \vdash n$, its Young tabloids are sorted in a certain way, $t_1, t_2, \cdots$ (to see the particular ordering we refer the reader to the appendix of Huang et al. (2009)).

The *permutation representation* at partition $\lambda \vdash n$ is denoted as τ_λ, that is,

$$\tau_\lambda(\sigma)_{ij} = \begin{cases} 1 & \text{if } \sigma(t_j) = t_i \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

For a detailed definition of τ_λ, see Huang et al. (2009).

The formulas below show the decompositions of the permutation representations for certain partitions in terms of the irreducible representations:

$$\begin{aligned} \tau_{(n)} &\equiv \rho_{(n)} \\ \tau_{(n-1,1)} &\equiv \rho_{(n)} \oplus \rho_{(n-1,1)} \\ \tau_{(n-2,2)} &\equiv \rho_{(n)} \oplus \rho_{(n-1,1)} \oplus \rho_{(n-2,2)} \\ \tau_{(n-2,1,1)} &\equiv \rho_{(n)} \oplus \rho_{(n-1,1)} \oplus \rho_{(n-1,1)} \oplus \rho_{(n-2,2)} \oplus \rho_{(n-2,1,1)} \end{aligned} \tag{7}$$

That is, the left-sided representations are equal to the direct sums of the right side when a change of basis is applied. In matrix terminology, this is the same as saying that there exists an invertible matrix C_λ, such that

$$\tau_\lambda(\sigma) = C_\lambda \left[\bigoplus_{\mu \geq \lambda} \bigoplus_{i=1}^{k_{\lambda\mu}} \rho_\mu(\sigma) \right] C_\lambda^{-1}.$$

This decomposition implies that the FT of a function f at τ_λ can be computed in terms of the FT at ρ_μ, with $\mu \geq \lambda$:

$$\hat{f}_{\tau_\lambda} = C_\lambda \left[\bigoplus_{\mu \geq \lambda} \bigoplus_{i=1}^{k_{\lambda\mu}} \hat{f}_{\rho_\lambda} \right] C_\lambda^{-1}.$$

A.2 Characterization of the TSP: Proof

The aim of this section is to prove Theorems 4 and 5 (previously stated in section 5.3), following the line of thought of the proof for the LOP. This means that the steps for proving the theorem remain the same. Firstly, we observe that the FT of a TSP function is proportional to the product of two matrices, one depending on the instance and the other one ($\hat{f}_{A'}{}_{\lambda}^{T}$) being constant. Secondly, we analyse the constant factor $\hat{f}_{A'}{}_{\lambda}$ and see how its particular shape affects the Fourier coefficients of the TSP.

Even though this process is very similar to the one followed with the LOP, this time the proof is not as straightforward, because, even though in the case of the LOP the constant factor has already been analyzed in the literature, we have not found the analogue for the TSP. So, unlike in the previous section, we have to carry out the entire analysis of the constant factor.

Proposition 4. *If $f : \Sigma_n \longrightarrow \mathbb{R}$ is the objective function of a TSP instance, that is, f is expressed as in (4), then its FT has the following properties:*

1. *$\hat{f}_{\lambda} = 0$, if $\lambda \neq (n), (n-1,1), (n-2,2), (n-2,1,1)$.*
2. *The values of coefficients $\lambda = (n-1,1), (n-2,2)$ and $(n-2,1,1)$ can be factored as the following product:*
$$\hat{f}_{\lambda} = \frac{1}{(n-2)!} \hat{f}_{A\lambda} \cdot \hat{f}_{A'}{}_{\lambda}^{T},$$
where f_A is the graph function of A and

$$f_{A'}(\sigma) = \begin{cases} 1 & \textit{if } \sigma(n-1) = \sigma(n)+1 \\ & \qquad \textit{or} \\ & (\sigma(n-1) = 1 \textit{ and } \sigma(n) = n) \\ 0 & \textit{otherwise.} \end{cases} \tag{8}$$

Proof. The TSP is a particular case of the QAP. By setting

$$a'_{ij} = \begin{cases} 1 & \text{if } j = i+1 \text{ or } (i = n \text{ and } j = 1) \\ 0 & \text{otherwise,} \end{cases}$$

Equation (2) becomes the objective function of the TSP. Then,

$$f_{A'}(\sigma) = a'_{\sigma(n)\sigma(n-1)} = \begin{cases} 1 & \text{if } \sigma(n-1) = \sigma(n)+1 \\ & \qquad \text{or} \\ & (\sigma(n-1) = 1 \text{ and } \sigma(n) = n) \\ 0 & \text{otherwise.} \end{cases}$$

At this point, the result immediately follows from Propostion 1.

□

Let us proceed with the analysis of $\hat{f}_{A'}{}_{\lambda}$ as defined in Proposition 4. This analysis is not too complex, but it requires many calculations. For the sake of clarity, the analysis of $\hat{f}_{A'}{}_{\lambda}$ has been divided into a number of propositions (Propositions 6, 7, 8, 9 and 10) and corollaries (Corollaries 1 and 2).

The first step in the computation of $\hat{f}_{A'}{}_{\lambda}$ consists of computing $\hat{f}_{A'}{}_{\rho_{(n)}}$.

Proposition 5. *Given $f_{A'}$ defined as in Proposition 4, the FT of $f_{A'}$ at irreducible $\rho_{(n)}$ is*

$$\hat{f}_{A'}{}_{\rho_{(n)}} = [n \cdot (n-2)!] \,. \tag{9}$$

Proof. $\rho_{(n)} = 1$, therefore,

$$\hat{f}_{A'}{}_{\rho_{(n)}} = \sum_{\sigma} f_{A'}(\sigma) \cdot \rho_{(n)}(\sigma) = \sum_{\sigma} f_{A'}(\sigma).$$

According to the definition of $f_{A'}$ in Equation (8), $\sum_\sigma f_{A'}(\sigma)$ is the number of permutations $\sigma \in \Sigma_n$ such that $\sigma(n-1) = \sigma(n) + 1$ or $(\sigma(n) = n$ and $\sigma(n-1) = 1)$. The condition $\sigma(n-1) = \sigma(n) + 1$ or $(\sigma(n-1) = 1$ and $\sigma(n) = n)$ can be understood as mapping two elements ($n-1$ and n) to fixed values, while the rest are mapped to arbitray values. The pairs of values to which $n-1$ and n can be mapped are listed:

$$\begin{aligned} \sigma(n) = 1 \ \text{ and } \ \sigma(n-1) = 2, \\ \sigma(n) = 2 \ \text{ and } \ \sigma(n-1) = 3, \\ \vdots \\ \sigma(n) = n-1 \ \text{ and } \ \sigma(n-1) = n, \\ \sigma(n) = n \ \text{ and } \ \sigma(n-1) = 1. \end{aligned} \tag{10}$$

There are n possible mappings and, in each of them, the remaining $(n-2)$ elements can have any possible order. This means that the number of permutations that satisfy system (10) is

$$\hat{f}_{A'}{}_{\rho_{(n)}} = [n \cdot (n-2)!] \,.$$

□

At this point, and before proceeding with the exposition, let us note that the FT is linear and that, for any constant function c, $\hat{c}_{\rho_\lambda} = 0$ for each $\lambda \neq (n)$. So, when a function f is translated (that is, f is transformed to a function $h = f + c$, being h the new function), its FT remains the same for any partition except for $\lambda = (n)$ (that is, $\hat{f}_{\rho_\lambda} = \hat{h}_{\rho_\lambda}$ for $\lambda \neq (n)$). So we could translate $f_{A'}$ and its non-trivial Fourier coefficients would remain the same.

Instead of working with $f_{A'}$, in what follows, the analysis is carried out with a translation of $f_{A'}$, $h = f_{A'} + c$. We adjust our constant c, such that $\hat{h}_{\rho_{(n)}} = 0$. Even though this choice may initially seem unclear, it is elucidated later in Proposition 10. For the time being, probably it suffices to observe that this choice eventually makes the mathematical reasoning easier. Taking into account Equation (9), it is immediate to see that the function h with $\hat{h}_{\rho_{(n)}} = 0$ is the following:

$$h(\sigma) = f_{A'}(\sigma) - \frac{1}{n-1}. \tag{11}$$

Directly studying $\hat{h}_{\rho_\lambda}$ is not an easy task. Instead, we compute $\hat{h}_{\tau_\lambda}$ and deduce the properties of $\hat{h}_{\rho_\lambda}$ by means of the decompositions of Equation (7). The computation of $\hat{h}_{\tau_\lambda}$ is quite straightforward, but requires a high number steps based on elementary combinatorics.

FT at $\tau_{(n)}$ From the fact that $\tau_{(n)} = \rho_{(n)} = 1$, it obviously follows Proposition 6.

Proposition 6. *The FT of h at irreducible $\tau_{(n)}$ is zero.*

FT at $\tau_{(n-1,1)}$ Before computing $\hat{h}_{\tau_{(n-1,1)}}$ in Proposition 7, it is convenient to have a look at representation $\tau_{(n-1,1)}$. Its definition is based on the standard tabloids of shape $(n-1,1)$, which are the following

$$\begin{array}{l} \hline 2 \quad 3 \quad \cdots \qquad n \\ \hline 1 \\ \hline \end{array} \qquad \begin{array}{l} \hline 1 \quad 3 \quad \cdots \qquad n \\ \hline 2 \\ \hline \end{array} \qquad \begin{array}{l} \hline 1 \quad 2 \quad \cdots \qquad n \\ \hline 3 \\ \hline \end{array} \qquad \cdots$$

$$\cdots \qquad \begin{array}{l} \hline 1 \quad 2 \quad \cdots \quad n-1 \\ \hline n \\ \hline \end{array}$$

$\tau_{(n-1,1)}$ is given by

$$\tau_{(n-1,1)}(\sigma)_{ij} = \begin{cases} 1 & \text{if } \sigma(\tau_j) = \tau_i \\ 0 & \text{otherwise.} \end{cases}$$

Note that there is a one-to-one correspondence between each of the tabloids of shape $(n-1,1)$ and the element in the second row. Therefore, the tabloids can be indexed by their second-row element. With this ordering, t_1 and t_2 would be the following:

$$t_1 = \begin{array}{l} \hline 2 \quad 3 \quad \cdots \qquad n \\ \hline 1 \\ \hline \end{array} \quad \text{and} \quad t_2 = \begin{array}{l} \hline 1 \quad 3 \quad \cdots \qquad n \\ \hline 2 \\ \hline \end{array}$$

The condition $\sigma(t_1) = t_2$ means that we are checking whether

$$\begin{array}{l} \hline \sigma(2)\ \sigma(3)\ \ \cdots\ \sigma(n) \\ \hline \sigma(1) \\ \hline \end{array} = \begin{array}{l} \hline 1 \quad 3 \quad \cdots \quad n \\ \hline 2 \\ \hline \end{array}$$

This is equivalent to checking if $\sigma(1) = 2$. So $\tau_{(n-1,1)}$ can also be formulated as

$$\tau_{(n-1,1)}(\sigma)_{ij} = \begin{cases} 1 & \text{if } \sigma(j) = i \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Proposition 7. *The FT of h at irreducible $\tau_{(n-1,1)}$ is zero.*

Proof. The FT of h at representation $\tau_{(n-1,1)}$ is

$$\hat{h}_{\tau_{(n-1,1)}} = \sum_{\sigma} h(\sigma) \cdot \tau_{(n-1,1)}(\sigma).$$

Remember that a representation maps permutations to matrices, and $\tau_{(n-1,1)}(\sigma)$ is an $(n-1) \times (n-1)$ matrix, so $\hat{h}_{\tau_{(n-1,1)}}$ is an $(n-1) \times (n-1)$ matrix too. Each element of matrix $\hat{h}_{\tau_{(n-1,1)}}$ is

$$[\hat{h}_{\tau_{(n-1,1)}}]_{ij} = \sum_{\sigma} h(\sigma) \cdot [\tau_{(n-1,1)}(\sigma)]_{ij}.$$

Using the definition of h (Equation (11)),

$$\begin{aligned} [\hat{h}_{\tau_\lambda}]_{ij} &= \sum_{\sigma} \left(f_{A'}(\sigma) - \frac{1}{n-1} \right) \cdot [\tau_\lambda(\sigma)]_{ij} \\ &= \sum_{\sigma} f_{A'}(\sigma) \cdot [\tau_\lambda(\sigma)]_{ij} \; - \; \frac{1}{n-1} \sum_{\sigma} [\tau_\lambda(\sigma)]_{ij}. \end{aligned} \tag{13}$$

Considering the definition of $\tau_{(n-1,1)}$ of Equation (12), $\sum_\sigma [\tau_{(n-1,1)}(\sigma)]_{ij}$ is the number of permutations such that $\sigma(j) = i$, that is, $(n-1)!$. Therefore, the second term of the subtraction is

$$\frac{1}{n-1} \sum_{\sigma} [\tau_{(n-1,1)}(\sigma)]_{ij} = \frac{(n-1)!}{n-1} = (n-2)!. \tag{14}$$

To compute $\sum_\sigma f_{A'}(\sigma) \cdot [\tau_{(n-1,1)}(\sigma)]_{ij}$, notice that for each permutation σ,

$$f_{A'}(\sigma) \cdot [\tau_{(n-1,1)}(\sigma)]_{ij}$$

can take either one of two values. If $f_{A'}(\sigma) = 1$ and $[\tau_{(n-1,1)}(\sigma)]_{ij} = 1$,

$$f_{A'}(\sigma) \cdot [\tau_{(n-1,1)}(\sigma)]_{ij} = 1,$$

otherwise it is 0. So

$$\sum_{\sigma} f_{A'}(\sigma) \cdot [\tau_{(n-1,1)}(\sigma)]_{ij}$$

is the number of permutations σ such that $f_{A'}(\sigma) = 1$ and $[\tau_{(n-1,1)}(\sigma)]_{ij} = 1$. A permutation σ satisfies $f_{A'}(\sigma) = 1$ and $[\tau_{(n-1,1)}(\sigma)]_{ij} = 1$ if and only if (see equations (8) and (12), which define $f_{A'}$ and $\tau_{(n-1,1)}$) it satisfies the following system of equations:

$$\begin{cases} \sigma(n-1) = \sigma(n) + 1 \quad \text{or} \quad (\sigma(n-1) = 1 \text{ and } \sigma(n) = n), \\ \sigma(j) = i. \end{cases} \tag{15}$$

The number of permutations that satisfy these conditions depends on the values of indices i and j.

- If $j \neq n-1, n$,
  the possible values of σ that satisfy the first condition of system (15) have already been listed in (10). In addition, system (15) imposes $\sigma(j) = i$, so this additional condition makes us discard two of the possibilities listed in (10), because $\sigma(n), \sigma(n-1) \neq i$. Consequently, there are $(n-2)$ possible pairs of values that $\sigma(n)$ and $\sigma(n-1)$ can take. For each of these possibilities, we are fixing three elements: $\sigma(n), \sigma(n-1)$ and $\sigma(i)$. Hence, the number of permutations that satisfy system (15) is
  $$\sum_{\sigma} f_{A'}(\sigma) \cdot [\tau_{(n-1,1)}(\sigma)]_{ij} = (n-2) \cdot (n-3)! = (n-2)!.$$

- If $j = n$,
  system (15) is simplified:

$$\begin{cases} \sigma(n-1) = \sigma(n) + 1 \quad \text{or} \quad (\sigma(n-1) = 1 \text{ and } \sigma(n) = n), \\ \sigma(n) = i. \end{cases} \tag{16}$$

  $\sigma(n) = i$ is fixed and so is $\sigma(n-1)$. Then, the number of permutations that satisfy system (16) is the number of permutations that fix two elements:

$$\sum_{\sigma} f_{A'}(\sigma) \cdot [\tau_{(n-1,1)}(\sigma)]_{ij} = (n-2)!.$$

- If $j = n - 1$,
  this case is analogous to the previous case, where $j = n$; then,

$$\sum_{\sigma} f_{A'}(\sigma) \cdot [\tau_{(n-1,1)}(\sigma)]_{ij} = (n-2)!.$$

  We have just seen that, regardless of the values of i and j,

$$\sum_{\sigma} f_{A'}(\sigma) \cdot [\tau_{(n-1,1)}(\sigma)]_{ij} = (n-2)!.$$

  Taking (14) into account,

  $[\hat{h}_{\tau_{(n-1,1)}}]_{ij} = \sum_\sigma f_{A'}(\sigma) \cdot [\tau_{(n-1,1)}(\sigma)]_{ij} - \dfrac{1}{n-1} \sum_\sigma [\tau_{(n-1,1)}(\sigma)]_{ij} =$

  $(n-2)! - (n-2)! = 0.$

$\square$

FT at $\boldsymbol{\tau_{(n-2,2)}}$ A Young tabloid of shape $\lambda = (n-2, 2)$ can be exactly determined by the elements in the second and third row. For example, the following tabloid

$$\begin{array}{llll} \hline 1 & 2 & \cdots & n-2 \\ \hline n-1 & n & & \\ \cline{1-2} \end{array}$$

can be identified by the unordered tuple $\{n-1, n\}$. The permutation representation can be expressed as

$$\tau_{(n-2,2)}(\sigma)_{ij} = \begin{cases} 1 & \text{if } \sigma(\{j_1, j_2\}) = \{i_1, i_2\} \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

where the indices $i = \{i_1, i_2\}$ and $j = \{j_1, j_2\}$ are unordered tuples.

Proposition 8. *The FT of h at irreducible $\tau_{(n-2,2)}$ can be indexed by unordered tuples and takes the following values depending on the row $i = \{i_1, i_2\}$ and the column $j = \{j_1, j_2\}$:*

- *If $\{j_1, j_2\} \cap \{n-1, n\} = \emptyset$,*

$$[\hat{h}_{\tau_{(n-2,2)}}]_{ij} = \begin{cases} \dfrac{2(n-3)!}{n-1} & \textit{if } |i_1 - i_2| \equiv 1 \textit{ (mod } n\textit{)} \\ -\dfrac{4(n-4)!}{n-1} & \textit{if } |i_1 - i_2| \not\equiv 1 \textit{ (mod } n\textit{).} \end{cases}$$

- *If* $\{j_1, j_2\} \cap \{n-1, n\} = \{n-1\}$ *or* $\{n\}$,

$$[\hat{h}_{\tau_{(n-2,2)}}]_{ij} = \begin{cases} \dfrac{(3-n)(n-3)!}{n-1} & \text{if } |i_1 - i_2| \equiv 1 \text{ (mod } n) \\ \dfrac{2(n-3)!}{n-1} & \text{if } |i_1 - i_2| \not\equiv 1 \text{ (mod } n). \end{cases}$$

- *If* $\{j_1, j_2\} \cap \{n-1, n\} = \{n-1, n\}$,

$$[\hat{h}_{\tau_{(n-2,2)}}]_{ij} = \begin{cases} \dfrac{(n-3)(n-2)!}{n-1} & \text{if } |i_1 - i_2| \equiv 1 \text{ (mod } n) \\ -\dfrac{2(n-2)!}{n-1} & \text{if } |i_1 - i_2| \not\equiv 1 \text{ (mod } n). \end{cases}$$

Proof. To compute $\hat{h}_{\tau_{(n-2,2)}}$ as specified by Equation (13), we compute again the two terms of the subtraction separately.

$\sum_\sigma [\tau_{(n-2,2)}(\sigma)]_{ij}$ is the number of permutations such that $\sigma(j_1) = i_1$ and $\sigma(j_2) = i_2$, or, $\sigma(j_1) = i_2$ and $\sigma(j_2) = i_1$. There are $2 \cdot (n-2)!$ permutations satisfying this condition, then

$$\frac{1}{n-1} \sum_\sigma [\tau_{(n-1,1)}(\sigma)]_{ij} = \frac{2 \cdot (n-2)!}{n-1}. \tag{18}$$

To compute $\sum_\sigma f_{A'}(\sigma)[\tau_{(n-2,2)}(\sigma)]_{ij}$, we have to count the number of permutations for which $f_{A'}(\sigma) = 1$ and $[\tau_{(n-2,2)}(\sigma)]_{ij} = 1$, that is, how many permutations satisfy the following system of equations:

$$\begin{cases} \sigma(n-1) \equiv \sigma(n) + 1 \pmod{n}, \\ \sigma(\{j_1, j_2\}) = \{i_1, i_2\}. \end{cases} \tag{19}$$

The number of permutations that satisfy these equations depends on the values of indices i and j. We list the values of $\sum_\sigma f_{A'}(\sigma) \cdot [\tau_{(n-2,2)}(\sigma)]_{ij}$, which depend on certain conditions over the indices. Even though there are 6 distinct cases, we only prove the first and second, since the rest of them are computed similarly.

- If $\{j_1, j_2\} \cap \{n, n-1\} = \emptyset$,
  - If $|i_1 - i_2| \equiv 1 \pmod{n}$,

    the condition for a permutation σ to satisfy $\sigma(n-1) \equiv \sigma(n)+1 \pmod{n}$ restricts the pairs of values that $\sigma(n-1)$ and $\sigma(n)$ can hold. All the n possible pairs of values that $\sigma(n-1)$ and $\sigma(n)$ can take are listed in Equation (10). However, the additional condition of system (19), that is $\sigma(\{j_1, j_2\}) = \{i_1, i_2\}$, discards some of these pairs, since we have the restrictions $\sigma(n-1) \neq i_1, i_2$ and $\sigma(n) \neq i_1, i_2$. Assume, without loss of generality, that $i_2 \equiv i_1 + 1 \pmod{n}$. Then, the discarded pairs are

$$\sigma(n) \equiv i_1 - 1 \pmod{n} \text{ and } \sigma(n-1) = i_1,$$

$$\sigma(n) = i_1 \text{ and } \sigma(n-1) \equiv i_2 \pmod{n},$$

$$\sigma(n) = i_2 \text{ and } \sigma(n-1) \equiv i_2 + 1 \pmod{n}.$$

So we are left with $(n-3)$ different possible values of $\sigma(n-1)$ and $\sigma(n)$. In addition, $\sigma(\{j_1, j_2\}) = \{i_1, i_2\}$ has also two possibilities, that is $\sigma(j_1) = i_1$ and $\sigma(j_2) = i_2$, or $\sigma(j_1) = i_2$ and $\sigma(j_2) = i_1$. So we have $2 \cdot (n-3)$ possible combinations of values for $\sigma(n-1), \sigma(n), \sigma(j_1)$ and $\sigma(j_2)$. For each combination, the rest of the $(n-4)$ elements can be reordered arbitrarily. This implies that the number of permutations that satisfy system (19) is

$$\sum_{\sigma} f_{A'}(\sigma)[\tau_{(n-2,2)}(\sigma)]_{ij} = 2 \cdot (n-3) \cdot (n-4)!.$$

– If $|i_1 - i_2| \not\equiv 1 \pmod{n}$,
This case is very similar to the previous one, but, since $|i_1 - i_2| \not\equiv 1 \pmod{n}$, the pairs of values of $\sigma(n)$ and $\sigma(n-1)$ discarded due to the condition $\sigma(\{j_1, j_2\}) = \{i_1, i_2\}$ are different:

$$\begin{aligned}
&\sigma(n) \equiv i_1 - 1 \pmod{n} \text{ and } \sigma(n-1) = i_1, \\
&\sigma(n) = i_1 \text{ and } \sigma(n-1) \equiv i_1 + 1 \pmod{n}, \\
&\sigma(n) \equiv i_2 - 1 \pmod{n} \text{ and } \sigma(n-1) = i_2, \\
&\sigma(n) = i_2 \text{ and } \sigma(n-1) \equiv i_2 + 1 \pmod{n}.
\end{aligned}$$

Therefore, the number of permutations satisfying system (19) is

$$\sum_{\sigma} f_{A'}(\sigma) \cdot [\tau_{(n-2,2)}(\sigma)]_{ij} = 2 \cdot (n-4) \cdot (n-4)!.$$

- If $\{j_1, j_2\} \cap \{n-1, n\} = \{n-1\}$ or $\{j_1, j_2\} \cap \{n-1, n\} = \{n\}$,
  – If $|i_1 - i_2| \equiv 1 \pmod{n}$,

$$\sum_{\sigma} f_{A'}(\sigma)[\tau_{(n-2,2)}(\sigma)]_{ij} = (n-3)!.$$

  – If $|i_1 - i_2| \not\equiv 1 \pmod{n}$,

$$\sum_{\sigma} f_{A'}(\sigma)[\tau_{(n-2,2)}(\sigma)]_{ij} = 2\,(n-3)!.$$

- If $\{j_1, j_2\} = \{n-1, n\}$,
  – If $|i_1 - i_2| \equiv 1 \pmod{n}$,

$$\sum_{\sigma} f_{A'}(\sigma)[\tau_{(n-2,2)}(\sigma)]_{ij} = (n-2)!.$$

  – If $|i_1 - i_2| \not\equiv 1 \pmod{n}$,

$$\sum_{\sigma} f_{A'}(\sigma)[\tau_{(n-2,2)}(\sigma)]_{ij} = 0.$$

$\hat{h}_{\tau_{(n-2,2)}}$ is computed by subtracting the two terms in Equation (13). The first term is $\sum_{\sigma} f_{A'}(\sigma) \cdot [\tau_{(n-2,2)}(\sigma)]_{ij}$ and the second has been calculated in (18). The subtraction immediately leads to the statement of our proposition.

□

Corollary 1. *$\hat{h}_{\tau_{(n-2,2)}}$ is a rank-one matrix.*

Proof. We have already computed the value of $\hat{h}_{\tau_{(n-2,2)}}$ in Proposition 8. Rows and columns are indexed by unordered tuples $i = \{i_1, i_2\}$ and $j = \{j_1, j_2\}$, respectively. Notice that there are only two different rows in $\hat{h}_{\tau_{(n-2,2)}}$, depending on whether $|i_1 - i_2| \equiv 1 \pmod{n}$ or $|i_1 - i_2| \not\equiv 1 \pmod{n}$. It is easy to check that a row given by $|i_1 - i_2| \equiv 1 \pmod{n}$ is $2/(3-n)$ times a row given by $|i_1 - i_2| \not\equiv 1 \pmod{n}$.

Let us see it by differentiating column cases. Assume that $i = \{i_1, i_2\}$ is an index such that $|i_1 - i_2| \equiv 1 \pmod{n}$ and $i' = \{i'_1, i'_2\}$ is such that $|i'_1 - i'_2| \not\equiv 1 \pmod{n}$.

- If $\{j_1, j_2\} \cap \{n-1, n\} = \emptyset$,

$$[\hat{h}_{\tau_{(n-2,2)}}]_{i'j} = -\frac{4(n-4)!}{n-1} = \frac{2}{3-n} \cdot \frac{2(n-3)!}{n-1} = \frac{2}{3-n} \cdot [\hat{h}_{\tau_{(n-2,2)}}]_{ij}.$$

- If $\{j_1, j_2\} \cap \{n-1, n\} = \{n-1\}$ or $\{j_1, j_2\} \cap \{n-1, n\} = \{n\}$,

$$[\hat{h}_{\tau_{(n-2,2)}}]_{i'j} = \frac{2(n-3)!}{n-1} = \frac{2}{3-n} \cdot \frac{(3-n)(n-3)!}{n-1} = \frac{2}{3-n} \cdot [\hat{h}_{\tau_{(n-2,2)}}]_{ij}.$$

- If $\{j_1, j_2\} = \{n-1, n\}$,

$$[\hat{h}_{\tau_{(n-2,2)}}]_{i'j} = -\frac{2(n-2)!}{n-1} = \frac{2}{3-n} \cdot \frac{(n-3)(n-2)!}{n-1} = \frac{2}{3-n} \cdot [\hat{h}_{\tau_{(n-2,2)}}]_{ij}.$$

This implies that all the rows in $\hat{h}_{\tau_{(n-2,2)}}$ are proportional and, in consequence, $\hat{h}_{\tau_{(n-2,2)}}$ is rank-one.

□

FT at $\tau_{(n-2,1,1)}$ A Young tabloid of shape $(n-2, 1, 1)$ has the following representation:

$$\begin{array}{cccc} \hline 1 & 2 & \cdots & n-2 \\ \hline n-1 & & & \\ \cline{1-1} n & & & \\ \cline{1-1} \end{array}$$

It can be exactly identified by the elements in the second and third row, which can be represented, for instance, with the tuple $(n-1, n)$. The permutation representation for partition $\lambda = (n-2, 1, 1)$ can then be expressed as

$$\tau_{(n-2,1,1)}(\sigma)_{ij} = \begin{cases} 1 & \text{if } \sigma(j_1) = i_1 \text{ and } \sigma(j_2) = i_2 \\ 0 & \text{otherwise,} \end{cases} \tag{20}$$

where the indices are ordered tuples $i = (i_1, i_2)$ and $j = (j_1, j_2)$.

Proposition 9. *The FT of h at irreducible $\tau_{(n-2,1,1)}$ can be indexed by ordered tuples and takes the following values depending on the row $i = (i_1, i_2)$ and the column $j = (j_1, j_2)$:*

- *If* $\{j_1, j_2\} \cap \{n-1, n\} = \emptyset$,

$$[\hat{h}_{\tau_{(n-2,2)}}]_{ij} = \begin{cases} \dfrac{(n-3)!}{n-1} & \text{if } |i_1 - i_2| \equiv 1 \text{ (mod } n) \\ -\dfrac{2(n-4)!}{n-1} & \text{if } |i_1 - i_2| \not\equiv 1 \text{ (mod } n). \end{cases}$$

- *If* $j_1 = n$ *and* $j_2 \neq n-1$, *or* $j_1 \neq n$ *and* $j_2 = n-1$, *or* $j_1 = n$ *and* $j_2 = n-1$,

$$[\hat{h}_{\tau_{(n-2,2)}}]_{ij} = \begin{cases} -\dfrac{(n-2)!}{n-1} & \text{if } j_1 = n, j_2 \neq n-1 \text{ and } i_2 \equiv i_1 + 1 \text{ (mod } n) \\ \dfrac{(n-3)!}{n-1} & \text{if } j_1 = n, j_2 \neq n-1 \text{ and } i_2 \not\equiv i_1 + 1 \text{ (mod } n) \\ & \\ -\dfrac{(n-2)!}{n-1} & \text{if } j_1 \neq n, j_2 = n-1 \text{ and } i_2 \equiv i_1 + 1 \text{ (mod } n) \\ \dfrac{(n-3)!}{n-1} & \text{if } j_1 \neq n, j_2 = n-1 \text{ and } i_2 \not\equiv i_1 + 1 \text{ (mod } n) \\ & \\ \dfrac{(n-2)(n-2)!}{n-1} & \text{if } j_1 = n, j_2 = n-1 \text{ and } i_2 \equiv i_1 + 1 \text{ (mod } n) \\ -\dfrac{(n-2)!}{n-1} & \text{if } j_1 = n, j_2 = n-1 \text{ and } i_2 \not\equiv i_1 + 1 \text{ (mod } n). \end{cases}$$

- *If* $j_1 = n-1$ *and* $j_2 \neq n$, *or* $j_1 \neq n-1$ *and* $j_2 = n$, *or* $j_1 = n-1$ *and* $j_2 = n$,

$$[\hat{h}_{\tau_{(n-2,2)}}]_{ij} = \begin{cases} -\dfrac{(n-2)!}{n-1} & \text{if } j_1 \neq n-1, j_2 = n \text{ and } i_1 \equiv i_2 + 1 \text{ (mod } n) \\ \dfrac{(n-3)!}{n-1} & \text{if } j_1 \neq n-1, j_2 = n \text{ and } i_1 \not\equiv i_2 + 1 \text{ (mod } n) \\ & \\ -\dfrac{(n-2)!}{n-1} & \text{if } j_1 = n-1, j_2 \neq n \text{ and } i_1 \equiv i_2 + 1 \text{ (mod } n) \\ \dfrac{(n-3)!}{n-1} & \text{if } j_1 = n-1, j_2 \neq n \text{ and } i_1 \not\equiv i_2 + 1 \text{ (mod } n) \\ & \\ \dfrac{(n-2)(n-2)!}{n-1} & \text{if } j_1 = n-1, j_2 = n \text{ and } i_1 \equiv i_2 + 1 \text{ (mod } n) \\ -\dfrac{(n-2)!}{n-1} & \text{if } j_1 = n-1, j_2 = n \text{ and } i_1 \not\equiv i_2 + 1 \text{ (mod } n). \end{cases}$$

Proof. We proceed as in the proofs of Propositions 7 and 8, by computing $\hat{h}_{\tau_{(n-2,1,1)}}$ with Equation (13). In the subtraction, $\sum_\sigma [\tau_{(n-2,1,1)}(\sigma)]_{ij}$ is the number of permutations such that $\sigma(j_1) = i_1$ and $\sigma(j_2) = i_2$. There are $(n-2)!$ permutations satisfying this condition, then

$$\frac{1}{n-1}\sum_{\sigma}[\tau_{(n-2,1,1)}(\sigma)]_{ij} = \frac{(n-2)!}{n-1}. \tag{21}$$

Regarding $\sum_\sigma f_{A'}(\sigma)\cdot[\tau_{(n-2,1,1)}(\sigma)]_{ij}$, note that $f_{A'}(\sigma)\cdot[\tau_{(n-2,1,1)}(\sigma)]_{ij} = 1$ if and only if σ satisfies the following system of equations:

$$\begin{cases} \sigma(n-1) \equiv \sigma(n)+1 \pmod{n}, \\ \sigma(j_1) = i_1, \\ \sigma(j_2) = i_2. \end{cases} \tag{22}$$

Similarly to the proofs of Propositions 7 and 8, there are different cases. Since all the cases can be calculated using basic combinatorics, we only develop the third and forth as an example (we skip the first and second ones because they are very similar to those explained in the proof of Proposition 8).

- If $\{j_1, j_2\} \cap \{n-1, n\} = \emptyset$,
  - If $|i_1 - i_2| \equiv 1$ (mod n),
  $$\sum_\sigma f_{A'}(\sigma)\cdot[\tau_{(n-2,1,1)}(\sigma)]_{ij} = \frac{(n-3)!}{n-1}.$$
  - If $|i_1 - i_2| \not\equiv 1$ (mod n),
  $$\sum_\sigma f_{A'}(\sigma)\cdot[\tau_{(n-2,1,1)}(\sigma)]_{ij} = -\frac{2(n-4)!}{n-1}.$$
- If $j_1 = n$ and $j_2 \neq n-1$, system (22) becomes
$$\begin{cases} \sigma(n)+1 \equiv \sigma(n-1) \pmod{n}, \\ \sigma(n) = i_1, \\ \sigma(j_2) = i_2. \end{cases} \tag{23}$$
  - If $i_2 \equiv i_1 + 1$ (mod n),
  System (23) is incompatible because
  $$\sigma(j_2) = i_2 = i_1 + 1 = \sigma(n) + 1 \equiv \sigma(n-1) \text{ (mod } n).$$
  Then, $j_2 = n-1$, which is a contradiction. Therefore,
  $$\sum_\sigma f_{A'}(\sigma)\cdot[\tau_{(n-2,1,1)}(\sigma)]_{ij} = 0.$$
  - Otherwise,
  $\sigma(n)$, $\sigma(n-1)$ and $\sigma(j_2)$ are fixed and the rest of the elements can be arbitrarily reordered. So the number of permutations that satisfy system (23) is the number of permutations that fix 3 elements, that is,
  $$\sum_\sigma f_{A'}(\sigma)\cdot[\tau_{(n-2,1,1)}(\sigma)]_{ij} = (n-3)!.$$
- If $j_1 \neq n-1$ and $j_2 = n$,

- If $i_1 \equiv i_2 + 1 \pmod{n}$,

$$\sum_\sigma f_{A'}(\sigma) \cdot [\tau_{(n-2,1,1)}(\sigma)]_{ij} = 0.$$

- Otherwise,

$$\sum_\sigma f_{A'}(\sigma) \cdot [\tau_{(n-2,1,1)}(\sigma)]_{ij} = (n-3)!.$$

- If $j_1 = n-1$ and $j_2 \neq n-1$,
  - If $i_1 \equiv i_2 + 1 \pmod{n}$,

$$\sum_\sigma f_{A'}(\sigma) \cdot [\tau_{(n-2,1,1)}(\sigma)]_{ij} = 0.$$

  - Otherwise,

$$\sum_\sigma f_{A'}(\sigma) \cdot [\tau_{(n-2,1,1)}(\sigma)]_{ij} = (n-3)!.$$

- If $j_1 \neq n$ and $j_2 = n-1$,
  - If $i_2 \equiv i_1 + 1 \pmod{n}$,

$$\sum_\sigma f_{A'}(\sigma) \cdot [\tau_{(n-2,1,1)}(\sigma)]_{ij} = 0.$$

  - Otherwise,

$$\sum_\sigma f_{A'}(\sigma) \cdot [\tau_{(n-2,1,1)}(\sigma)]_{ij} = (n-3)!.$$

- If $j_1 = n$ and $j_2 = n-1$,
  - If $i_2 \equiv i_1 + 1 \pmod{n}$,

$$\sum_\sigma f_{A'}(\sigma) \cdot [\tau_{(n-2,1,1)}(\sigma)]_{ij} = 0.$$

  - Otherwise,

$$\sum_\sigma f_{A'}(\sigma) \cdot [\tau_{(n-2,1,1)}(\sigma)]_{ij} = (n-3)!.$$

- If $j_1 = n-1$ and $j_2 = n$,
  - If $i_1 \equiv i_2 + 1 \pmod{n}$,

$$\sum_\sigma f_{A'}(\sigma) \cdot [\tau_{(n-2,1,1)}(\sigma)]_{ij} = (n-2).$$

  - Otherwise,

$$\sum_\sigma f_{A'}(\sigma) \cdot [\tau_{(n-2,1,1)}(\sigma)]_{ij} = 0.$$

The result stated in the proposition follows from subtracting the value calculated in (21) to $\sum_\sigma f_{A'}(\sigma) \cdot [\tau_{(n-2,1,1)}(\sigma)]_{ij}$, as indicated by Equation (13).

□

Corollary 2. *The rank of $\hat{h}_{\tau_{(n-2,1,1)}}$ is 2.*

Proof. A way of proving that a matrix has rank 2 is by showing that the linear space spanned by its columns has dimension 2. We take this path by defining two column vectors $\mathbf{v}$ and $\mathbf{w}$ which can generate any of the columns of $\hat{h}_{\tau_{(n-2,1,1)}}$. Before defining $\mathbf{v}$ and $\mathbf{w}$, note that, independently of the size of $\hat{h}_{\tau_{(n-2,1,1)}}$, there are only 7 different columns, determined by the following conditions on $j = (j_1, j_2)$:

1. $\{j_1, j_2\} = \emptyset$.
2. $j_1 = n-1,\ j_2 \neq n$.
3. $j_1 \neq n-1,\ j_2 = n$.
4. $j_1 = n-1,\ j_2 = n$.
5. $j_1 = n,\ j_2 \neq n-1$.
6. $j_1 \neq n,\ j_2 = n-1$.
7. $j_1 = n,\ j_2 = n-1$.

However, it is easy to see that, actually, there are only 3 types of columns, if we consider proportional columns to be equivalent.

1. Columns satisfying $\{j_1, j_2\} = \emptyset$ are proportional to $\mathbf{z}$ defined as

$$\mathbf{z}_i = \begin{cases} 1 & \text{if } |i_1 - i_2| \equiv 1 \pmod{n} \\ -\dfrac{2}{n-3} & \text{if } |i_1 - i_2| \not\equiv 1 \pmod{n}. \end{cases}$$

2. Columns satisfying $j_1 = n-1$ and $j_2 \neq n$, or $j_1 \neq n-1$ and $j_2 = n$, or $j_1 = n-1$ and $j_2 = n$ are proportional to $\mathbf{v}$ defined as

$$\mathbf{v}_i = \begin{cases} 1 & \text{if } i_1 \equiv i_2 + 1 \pmod{n} \\ -\dfrac{1}{n-2} & \text{if } i_1 \not\equiv i_2 + 1 \pmod{n}. \end{cases}$$

3. Columns satisfying $j_1 = n$ and $j_2 \neq n-1$, or $j_1 \neq n$ and $j_2 = n-1$, or $j_1 = n$ and $j_2 = n-1$ are propotional to $\mathbf{w}$ defined as

$$\mathbf{w}_i = \begin{cases} 1 & \text{if } i_2 \equiv i_1 + 1 \pmod{n} \\ -\dfrac{1}{n-2} & \text{if } i_2 \not\equiv i_1 + 1 \pmod{n}. \end{cases}$$

It could seem, then, that the columns of $\hat{h}_{\tau_{(n-2,1,1)}}$ are spanned by three vectors, namely $\mathbf{v}$, $\mathbf{w}$ and $\mathbf{z}$; but $\mathbf{z}$ is a linear combination of $\mathbf{v}$ and $\mathbf{w}$. Indeed,

$$\mathbf{z} = (\mathbf{v} + \mathbf{w}) \cdot \frac{n-2}{n-3}.$$

It follows that $\hat{h}_{\tau_{(n-2,1,1)}}$ is rank-2, since its columns can be expressed as linear combinations of 2 vectors: $\mathbf{v}$ and $\mathbf{w}$.

$\square$

Proposition 10 (FT of the indicator $f_{A'}$). *The FT of $f_{A'}$ as defined by Equation (8) satisfies*

1. $\hat{f}_{A'}{}_{\rho_{(n)}} \neq 0$,
2. $\hat{f}_{A'}{}_{\rho_{(n-1,1)}} = 0$,
3. $\hat{f}_{A'}{}_{\rho_\lambda}$ *has rank one for* $\lambda = (n-2,2), (n-2,1,1)$.

Proof. 1. We have seen this in Proposition 6.

2. We know that $\hat{h}_{\tau_\lambda} = 0$ for $\lambda = (n)$ (Proposition 6) and for $\lambda = (n-1,1)$ (Proposition 7). We can see, thanks to the decompositions of the permutation representations of Equation (7), that $\hat{h}_{\tau_{(n-1,1)}}$ is equivalent to the direct sum of $\hat{h}_{\rho_{(n)}}$ and $\hat{h}_{\rho_{(n-1,1)}}$. Since $\hat{h}_{\tau_{(n-1,1)}} = 0$ and $\hat{h}_{\rho_{(n)}} = 0$, we conclude that $\hat{h}_{\rho_{(n-1,1)}} = 0$.

3. In the previous step, we have seen that $\hat{h}_{\rho_{(n)}} = 0$ and $\hat{h}_{\rho_{(n-1,1)}} = 0$, then the decomposition of $\hat{h}_{\tau_{(n-2,2)}}$ is reduced to:

$$\hat{h}_{\tau_{(n-2,2)}} \equiv \hat{h}_{\rho_{(n)}} \oplus \hat{h}_{\rho_{(n-1,1)}} \oplus \hat{h}_{\rho_{(n-2,2)}} \iff \hat{h}_{\tau_{(n-2,2)}} \equiv \hat{h}_{\rho_{(n-2,2)}} \oplus 0 \oplus 0.$$

   We have proved in Corollary 1 that $\hat{h}_{\tau_{(n-2,2)}}$ is rank-one, then, since the only non-zero component of $\hat{h}_{\tau_{(n-2,2)}}$ is $\hat{h}_{\rho_{(n-2,2)}}$, it has to be rank-one too.

   The only non-zero components in the decomposition of $\hat{h}_{\tau_{(n-2,1,1)}}$ in terms of the irreducible representations are $\hat{h}_{\rho_{(n-2,2)}}$ and $\hat{h}_{\rho_{(n-2,1,1)}}$, that is

$$\hat{h}_{\tau_{(n-2,1,1)}} \equiv 0 \oplus 0 \oplus 0 \oplus \hat{h}_{\rho_{(n-2,2)}} \oplus \hat{h}_{\rho_{(n-2,1,1)}}.$$

   $\hat{h}_{\rho_{(n-2,2)}}$ has rank 1 and $\hat{h}_{\tau_{(n-2,1,1)}}$ rank 2 (see Corollary 2). Then, $\hat{h}_{\rho_{(n-2,1,1)}}$ must have rank one too.

   $\hat{f}_{A'}{}_{\rho_\lambda} = \hat{h}_{\rho_\lambda}$, except for $\lambda = (n)$, which means that their rank properties are the same, for any coefficient $\lambda \neq (n)$. This concludes our proof.

$\square$

Having proved Propositions 4 and 10, the proof of Theorem 4 in section 5.3 follows the same pattern as the proof of Theorem 3 presented in section 5.2. This is why we have omitted this proof. On the other hand, to prove Theorem 5, it suffices to take into account two pieces of information. The first element to take into account is the characterization of the Fourier coefficients of the TSP given in Theorem 4. Secondly, Rockmore et al. (2002) proved that if the distance matrix of the QAP is symmetric, the objective function is built from (n), $(n-1,1)$, and $(n-2,2)$ components. This implies that in the case of an STSP function f, $\hat{f}_\lambda = 0$ if $\lambda = (n-2,1,1)$.

B `isLOP` and `isTSP` functions

In this section, the details of the code of the experiments mentioned in Section 5 are explained.

B.1 `isLOP` function

The aim of `isLOP` is to check whether a function $f : \Sigma_n \rightarrow \mathbb{R}$ corresponds to an LOP or not. Given a set of objective function values, it specifically checks whether there exists a matrix A such that f is the objective function of an LOP. Assume that there exists a certain ordering among the permutations of size n, which means that they are indexed from 1 to $n!$. Then, we can write $\Sigma_n = \{\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{n!}\}$. Our question can specifically be phrased as follows: given certain objective function values $v_1, v_2, \ldots, v_{n!}$, does a matrix $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ exist such that the LOP function f obtained with input matrix A satisfies $f(\sigma_l) = v_l$, for $l = 1, \ldots, n!$? This is the same as wondering whether there exists A such that

$$v_l = f(\sigma_l) = \sum_{j=1}^{n-1} \sum_{i=j+1}^{n} a_{\sigma_l(i)\sigma_l(j)}, \qquad l = 1, 2, \ldots, n! \tag{24}$$

Equation (24) can be expressed as a linear system, by performing a few operations:

$$v_l = \sum_{j=1}^{n-1} \sum_{i=j+1}^{n} a_{\sigma_l(i)\sigma_l(j)} = \sum_{s=1}^{n} \sum_{t=1}^{n} m_{st}(\sigma_l) a_{st}, \tag{25}$$

with $m_{st}(\sigma_l) = \begin{cases} 1 & \text{if } \sigma_l^{-1}(s) > \sigma_l^{-1}(t) \\ 0 & \text{otherwise.} \end{cases}$

Equation (25) can be further transformed by mapping the double indices st to a single index r, by using the following relation:

$$r = t + (s - 1) \cdot n.$$

So, by setting $\tilde{a}_r = a_{st}$ and $\tilde{m}_{lr} = m_{st}(\sigma_l)$, Equation (25) can be rewritten as a linear system:

$$v_l = \sum_{r=1}^{n^2} \tilde{m}_{lr} \tilde{a}_r, \tag{26}$$

with

$$\tilde{m}_{lr} = \begin{cases} 1 & \text{if } \sigma_l^{-1}(s) > \sigma_l^{-1}(t) \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

Since $\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_{n^2}$ are the unknown variables, by defining $\tilde{M} = [\tilde{m}_{lr}]$ and $\mathbf{v} = [v_1 v_2 \cdots v_{n!}]^T$, one can know if the system defined by Equation (24) has a solution (that is, if the given values $v_1, v_2, \ldots, v_{n!}$ are the objective function values of an LOP) by knowing if the following linear system is solvable:

$$\tilde{M}\mathbf{x} = \mathbf{v}.$$

Note that $\tilde{M} \in \mathbb{R}^{n!\times n^2}$ is not square. A way of tackling this problem is by finding the least-squares solution. Thus, we solve the problem

$$\min_{\mathbf{x}\in\mathbb{R}^{n^2}} ||\tilde{M}\mathbf{x} - \mathbf{v}||.$$

If there exists $\mathbf{x}$ such that the norm is 0, then we have found the coefficients of the input matrix of the LOP, and the answer is positive. This works theoretically, but, since the problem is solved computationally, one has to establish a threshold to check whether the norm is approximately 0. If the norm definitely is non-zero, $v_1, v_2, \ldots, v_{n!}$ cannot be the objective function values of an LOP. Algorithm 1 summarizes the procedure. A given ordering among permutations is assumed (we used the one given by SnFFT Julia package (Plumb et al., 2015)).

Algorithm 1 Pseudocode of `isLOP`

Input: $v_1, v_2, \ldots, v_{n!}$
Output: $isLOP$
Build matrix $\tilde{M} = [\tilde{m}_{lr}]$, as described by Eq. (27)
Solve $r = \min_{\mathbf{x} \in \mathbb{R}^{n^2}} ||\tilde{M}\mathbf{x} - \mathbf{v}||$
$isLOP = (r == 0)$
return $isLOP$

B.2 `isTSP` function

`isTSP` is the twin function of `isLOP`, and it checks whether a function $f : \Sigma_n \to \mathbb{R}$ corresponds to a TSP or not. Assume again that there exists a certain ordering among the permutations of size n, so $\Sigma_n = \{\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{n!}\}$. Then, our question can specifically be phrased as follows: given certain values $v_1, v_2, \ldots, v_{n!}$, does a matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ exist such that the TSP function f obtained with input matrix A satisfies $f(\sigma_l) = v_l$, for $l = 1, \ldots, n!$? This is the same as wondering whether there exists A such that

$$v_l = f(\sigma_l) = a_{\sigma_l(n)\sigma_l(1)} + \sum_{i=1}^{n-1} a_{\sigma_l(i)\sigma_l(i+1)}, \qquad l = 1, 2, \ldots, n! \tag{28}$$

Similarly to the case of the LOP, Equation (28) can be expressed as a linear system, by performing a few operations:

$$v_l = a_{\sigma_l(n)\sigma_l(1)} + \sum_{i=1}^{n-1} a_{\sigma_l(i)\sigma_l(i+1)} = \sum_{s=1}^{n} \sum_{t=1}^{n} m_{st}(\sigma_l) a_{st}, \tag{29}$$

with $m_{st}(\sigma_l) = \begin{cases} 1 & \text{if } \sigma^{-1}(s) + 1 \equiv \sigma^{-1}(t) \quad (mod\, n) \\ 0 & \text{otherwise.} \end{cases}$

Equation (29) can be further transformed by mapping the double indices st to a single index r, by using the same relation as for `isLOP`:

$$r = t + (s - 1) \cdot n.$$

So, by setting $\tilde{a}_r = a_{st}$ and $\tilde{m}_{lr} = m_{st}(\sigma_l)$, Equation (29) can be rewritten as a linear system:

$$v_l = \sum_{r=1}^{n^2} \tilde{m}_{lr} \tilde{a}_r, \tag{30}$$

with

$$\tilde{m}_{lr} = \begin{cases} 1 & \text{if } \sigma^{-1}(s) + 1 \equiv \sigma^{-1}(t) \quad (mod\, n) \\ 0 & \text{otherwise.} \end{cases} \tag{31}$$

Since $\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_{n^2}$ are the unknown variables, by defining $\tilde{M} = [\tilde{m}_{lr}]$ and $\mathbf{v} = [v_1 v_2 \cdots v_{n!}]^T$, one can know if system (28) has a solution (that is, if the given values $v_1, v_2, \ldots, v_{n!}$ are the objective function values of a TSP) by knowing if the following linear system is solvable:

$$\tilde{M}\mathbf{x} = \mathbf{v}.$$

Note that $\tilde{M} \in \mathbb{R}^{n! \times n^2}$ is not square. A way of tackling this problem is by finding the least-squares solution. Thus, we would like to solve the problem

$$\min_{\mathbf{x} \in \mathbb{R}^{n^2}} ||\tilde{M}\mathbf{x} - \mathbf{v}||.$$

However, this cannot be directly solved, unlike the case of the LOP (see section B.1), because matrix $\tilde{M}$ is never full-rank. This happens because there are many permutations that represent the same solution, e.g, for $n = 4$, (1234) and (3412). So, there are many rows of $\tilde{M}$ that are repeated. In order to solve the least-squares problem, the repeated rows have to be removed. Before removing them, however, it is necessary to check if permutations representing the same solution share the same objective function value. If this does not happen, then we can assure, without solving the least squares, that $v_1, v_2, \ldots, v_{n!}$ cannot be generated by a TSP. In this intermediate step, one has to take into account whether we are considering the symmetric or the non-symmetric TSP, because the number of equivalent permutations depends on the case. After having removed the redundant rows, if the objective function values are consistent (that is, if equivalent solutions share the same objective function values), then the least squares is solved on the reduced matrix.

Algorithms 2 and 3 summarize the procedure. A given ordering among permutations is assumed again (the one given by SnFFT Julia package (Plumb et al., 2015)), and $permutation\_to\_index(\tau)$ is the function that returns the index of τ according to the given ordering. In Algorithm 3, $representative(\sigma)$ is a function that computes a representative of the equivalence class of σ. That is, among all the permutations that encode the same solution as σ, a single one is chosen to represent the whole group. Note that this function varies depending on whether we are working with the symmetric or the non-symmetric version of the TSP and, what is more, this is the only part of `isTSP` that differs between the symmetric and the non-symmetric cases.

References

Benavides, X., Ceberio, J., and Hernando, L. (2021). On the symmetry of the quadratic assignment problem through elementary landscape decomposition. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1414–1422.

Ceberio, J., Calvo, B., Mendiburu, A., and Lozano, J. A. (2019). Multi-objectivising combinatorial optimisation problems by means of elementary landscape decompositions. *Evolutionary computation*, 27(2):291–311.

Chicano, F., Whitley, L. D., and Alba, E. (2011). A methodology to find the elementary landscape decomposition of combinatorial optimization problems. *Evolutionary Computation*, 19(4):597–637.

Algorithm 2 Pseudocode of `isCONSISTENT`

```
Input: v_1, v_2, ..., v_{n!}
Output: isConsistent
isConsistent = true
for i = 1, ..., n! do
    τ = representative(σ_i)
    j = permutation_to_index(τ)
    if v_i != v_j then
        isConsistent = false
        break
    end if
end for
return isConsistent
```

Algorithm 3 Pseudocode of `isTSP`

```
Input: v_1, v_2, ..., v_{n!}
Output: isTSP
isTSP = false
isConsistent = isCONSISTENT(v_1, v_2, ..., v_{n!})
if isConsistent then
    Build matrix M̃ = [m̃_{lr}], as described by Eq. (31)
    for i = n!, ..., 1 do
        if σ_i != representative(σ_i) then
            Delete element i from v_i and row i from M̃
        end if
    end for
    Solve r = min_{x ∈ ℝ^{n^2}} ||M̃x − v||
    if r = 0 then
        isTSP = true
    end if
end if
return isTSP
```

Christie, L. A. (2016). The role of walsh structure and ordinal linkage in the optimisation of pseudo-boolean functions under monotonicity invariance.

Elorza, A., Hernando, L., and Lozano, J. A. (2019). Taxonomization of combinatorial optimization problems in fourier space. *arXiv preprint arXiv:1905.10852*.

Elorza, A., Hernando, L., and Lozano, J. A. (2022). Transitions from p to np-hardness: the case of the linear ordering problem. *arXiv preprint arXiv:2205.12943*.

Framinan, J. M., Gupta, J. N., and Leisten, R. (2004). A review and classification of heuristics for permutation flow-shop scheduling with makespan objective. *Journal of the Operational Research Society*, 55(12):1243–1255.

Goldberg, D. E. (1989). Genetic algorithms and walsh functions: Part i, a gentle introduction. *Complex systems*, 3(2):129–152.

Hernando, L., Mendiburu, A., and Lozano, J. A. (2019). Characterising the rankings produced by combinatorial optimisation problems and finding their intersections. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 266–273.

Huang, J., Guestrin, C., and Guibas, L. (2009). Fourier theoretic probabilistic inference over permutations. *Journal of machine learning research*, 10(May):997–1070.

Kondor, I. R. (2008). *Group theoretical methods in machine learning*. Columbia University.

Kondor, R. (2010). A fourier space algorithm for solving quadratic assignment problems. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1017–1028. SIAM.

Körner, T. W. (1989). *Fourier analysis*. Cambridge university press.

Laporte, G. (1992). The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2):231–247.

Loiola, E. M., de Abreu, N. M. M., Boaventura-Netto, P. O., Hahn, P., and Querido, T. (2007). A survey for the quadratic assignment problem. *European journal of operational research*, 176(2):657–690.

Mania, H., Ramdas, A., Wainwright, M. J., Jordan, M. I., Recht, B., et al. (2018). On kernel methods for covariates that are rankings. *Electronic Journal of Statistics*, 12(2):2537–2577.

Martí, R. and Reinelt, G. (2011). *The linear ordering problem: exact and heuristic methods in combinatorial optimization*, volume 175. Springer Science & Business Media.

Plumb, G., Pachauri, D., Kondor, R., and Singh, V. (2015). Snfft: A julia toolkit for fourier analysis of functions over permutations. *The Journal of Machine Learning Research*, 16(1):3469–3473.

Rockmore, D., Kostelec, P., Hordijk, W., and Stadler, P. F. (2002). Fast fourier transform for fitness landscapes. *Applied and Computational Harmonic Analysis*, 12(1):57–76.

Rockmore, D. N. (2004). Recent progress and applications in group ffts. In *Computational noncommutative algebra and applications*, pages 227–254. Springer.

Sagan, B. E. (2013). *The symmetric group: representations, combinatorial algorithms, and symmetric functions*, volume 203. Springer Science & Business Media.

Stadler, P. F. (1996). Landscapes and their correlation functions. *Journal of Mathematical chemistry*, 20(1):1–45.

Swingler, K. (2020). Learning and searching pseudo-boolean surrogate functions from small samples. *Evolutionary computation*, 28(2):317–338.

Terras, A. (1985). *Harmonic analysis on symmetric spaces and applications*. Springer.

Terras, A. (1999). *Fourier analysis on finite groups and applications*, volume 43. Cambridge University Press.

Terras, A. (2012). *Harmonic analysis on symmetric spaces and applications II*. Springer Science & Business Media.

Verel, S., Derbel, B., Liefooghe, A., Aguirre, H., and Tanaka, K. (2018). A surrogate model based on walsh decomposition for pseudo-boolean functions. In *International Conference on Parallel Problem Solving from Nature*, pages 181–193. Springer.

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and de Feitas, N. (2016). Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387.

Whitley, D., Sutton, A. M., and Howe, A. E. (2008). Understanding elementary landscapes. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 585–592.