Fourier Transform-based instance decomposition for k-adic Assignment Problems

Xabier Benavides[1], Leticia Hernando[2], Josu Ceberio[2], and Jose A. Lozano[1,2]

[1] Basque Center for Applied Mathematics (BCAM), Spain
[2] University of the Basque Country UPV/EHU, Spain
xbenavides@bcamath.org, josu.ceberio@ehu.eus,
leticia.hernando@ehu.eus, ja.lozano@ehu.eus

Abstract. The work summarized in this manuscript has been accepted for publication in *IEEE Transactions on Evolutionary Computation* [1].
In combinatorial optimization, the Fourier Transform (FT) on the symmetric group has been used to analyze and optimize permutation-based problems. However, the difficulties that arise when working within Fourier space have limited its potential in practical scenarios. To address this issue, rather than using the FT to decompose the objective function, an alternative possibility is to use the FT as a tool to break down the instance defining the problem at hand. This strategy captures the information from each Fourier coefficient as a separate instance of the same problem, resulting in a potentially more usable setting in the context of optimization. So far, the FT-based instance decomposition has been limited to a small sample of problems characterized by low-order Fourier coefficients. In this work, we generalize this notion to a much broader class of problems, known as k-adic Assignment Problems (k-APs). We present a general framework for the decomposition of k-AP instances and discuss its practical applications. As a possible use case, we propose an instance generation method that exploits the FT-based instance decomposition to create artificial instances with tunable difficulty levels.

Keywords: Combinatorial optimization · Fourier transform · Assignment problems · Permutations.

1 Introduction

The Fourier Transform (FT) is a mathematical tool that decomposes functions into orthogonal components. Originally developed for the real domain, it was extended to finite groups, including permutations, enabling the decomposition of objective functions of combinatorial optimization problems. This provides a common framework for analyzing different problems and has been used to compare problem characteristics, study fitness landscapes, build surrogate models, and support the design of metaheuristic optimization methods.

For permutation-based problems, however, the Fourier space is often difficult to interpret. To address this issue, we develop this work focusing on the broad class of the k-adic assignment problems (k-APs). Given $D = [d_{i_1,\ldots,i_k}]$

and $H = [h_{i_1,\ldots,i_k}]$ such that $i_1, \ldots, i_k \in \{1, \ldots, n\}$, denoted as *distance* and *flow* tensors, respectively, the k-AP consists of finding the permutation $\sigma \in \Sigma_n$ that minimizes the following objective function:

$$f(\sigma) = \sum_{i_1=1}^{n} \cdots \sum_{i_k=1}^{n} d_{i_1,\ldots,i_k} h_{\sigma(i_1),\ldots,\sigma(i_k)}.$$

We propose decomposing problem instances obtaining a sum of problem instances associated with different Fourier coefficients. We also introduce an instance generation method based on this decomposition, allowing the creation of artificial instances through linear combinations of Fourier components. Experimental results show that this framework can be used to generate instances with controllable difficulty, at least for local search-based algorithms, potentially supporting the development of FT-based optimization strategies.

2 Proposed k-AP instance decomposition strategy

Any instance of the k-AP can be broken down into the sum of a set of instances of the same problem using the FT of the original objective function[3]. Each instance in this decomposition encodes information about a single Fourier coefficient. Thus, the *FT-based instance decomposition* provides an alternative way of leveraging the information provided by the classical FT while operating within the instance space of the problem. The proposed strategy does not compute the Fourier coefficients explicitly.

Considering a fixed k such that $k << n$, which is the most common situation in real-world scenarios, the introduced decomposition algorithm runs in polynomial time on the instance size n.

3 Use case: Instance generation

In general, controlling the properties and difficulty of the generated instances is not a trivial task, partly because the concept of difficulty may vary depending on the algorithm. We present an instance generation method for the k-AP built upon the FT-based instance decomposition. It consists of linearly combining the sub-instances obtained from the FT-based instance decomposition, by providing a weight β_i to each component, to create artificial instances with tunable difficulty levels.

4 Experimental analysis

We empirically validate that adjusting the weights β_i of the components obtained from the FT-based instance decomposition allows us to create artificial

[3] The code for the FT-based instance decomposition is available in `https://github.com/Av-Repos/FT-Instance-Decomposition`.

instances with tunable difficulty levels based on the fitness landscape concept. The neighborhood function used is the 2-*interchange* neighborhood. k-AP instances of size $n = 10$ with $k \in \{2, 3, 4\}$ are generated, considering different configurations of the weight parameters β_i, with 100 instances per problem type and parameter configuration. For simplicity, just the results for $k = 4$ are shown here. The measures used in the analysis are: autocorrelation coefficient, number of local optima, relative error with respect to the global optima, rank position of the output solution and distance to the global optima. The local search algorithm is executed 10000 times to obtain average results for the measures.

We visualize the marginal impact of the β_i parameters in the instance generation method by displaying the distribution of the difficulty measures in instances where interactions of each specific order are *predominant* (see Figure 1). We denote i-order interactions as predominant when their weight in the generated instances is equal to or higher than the weight of any other order interactions. The number of local optima seems related to the weight of the different-order interactions in the generated instances. In particular, instances with a higher predominant order seem to contain more local optima than those dominated by lower-order interactions. According to the average relative error and distance to the global optimum, the quality of the solutions produced by the algorithm deteriorates as the order of the predominant interactions increases. Additionally, the average rank of the solutions produced by the algorithm generally increases with the order of the predominant interactions.

5 Conclusions

The article [1] presents the first general framework for Fourier-based instance decomposition in the broad family of k-adic Assignment Problems. The proposed method transforms Fourier information into a representation based on optimization instances, making spectral analysis significantly more interpretable and easier to use in practice. It also provides a computationally attractive alternative for many applications.

A particularly valuable contribution is the instance generation methodology, which enables the creation of benchmark problems with tunable difficulty levels for, at least, local search-based algorithms. Experimental results show that manipulating the spectral composition of instances has a clear impact on landscape characteristics and algorithmic performance. So, it permits to create diverse benchmarks for the k-AP.

Acknowledgments. This work has been funded by the Basque Government (IT2109-26, BERC 2022-2025 program, projects KK-2024/00030 and KK-2025/00098 through Elkartek Program), the Spanish Ministry of Science, Innovation and Universities (projects PID2022-137442NB-I00 and PID2023-149195NB-I00, BCAM Severo Ochoa accreditation CEX2021-001142-S, FPU21/00607 grant) and the Universidad Empresa call from the University of the Basque Country (under project US24/22).

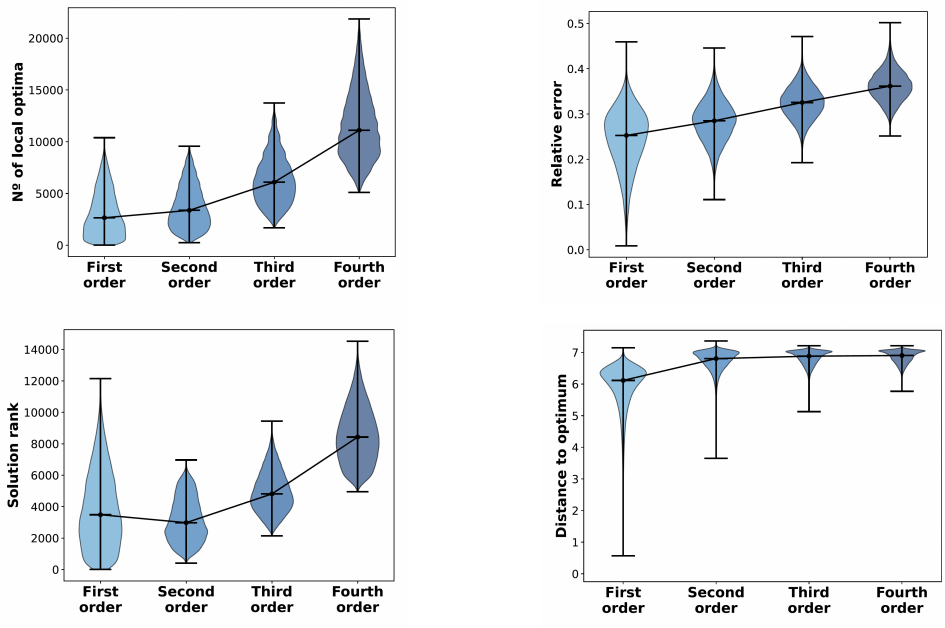


Fig. 1: Distribution of the measures according to the predominant order in the 4-AP instances.

References

1. Benavides, X., Hernando, L., Ceberio, J., Lozano, J.A.: Fourier Transform-based instance decomposition for k-adic Assignment Problems. IEEE Transactions on Evolutionary Computation (2026). https://doi.org/10.1109/TEVC.2026.3674527