P-NP instance decomposition based on the Fourier transform for solving the Linear Ordering Problem

Xabier Benavides
Basque Center for Applied Mathematics
Bilbao, Spain
xbenavides@bcamath.org

Leticia Hernando
University of the Basque Country UPV/EHU
Leioa, Spain
leticia.hernando@ehu.eus

Josu Ceberio
University of the Basque Country UPV/EHU
Donostia-San Sebastian, Spain
josu.ceberio@ehu.eus

Jose A. Lozano
Basque Center for Applied Mathematics
University of the Basque Country UPV/EHU
Bilbao, Spain
jlozano@bcamath.org

Abstract

Despite recent developments on the application of the Fourier transform in combinatorial optimization, few meta-heuristic algorithms have been proposed in the literature that exploit the information provided by this technique. In this work, we address this research gap by considering the case of the Linear Ordering Problem (LOP). Based on the Fourier transform of the problem's objective function, we propose an instance decomposition strategy that divides any LOP instance into the sum of two LOP instances associated with a P and an NP-Hard optimization problem. We take advantage of this decomposition to design a meta-heuristic algorithm called P-Descent Search (PDS). The proposed method intelligently adjusts the proportion of the P and NP-Hard components in the decomposition to define a sequence of surrogate instances suitable for optimization. By iteratively solving those instances, PDS is able to find better solutions than classical algorithms operating on the original problem.

The following document is a brief summary of the paper by Benavides et al. (2025). For further information, we refer the interested reader to the original paper.

1 Introduction

Given a matrix $M = [m_{i,j}]_{n\times n}$, the Linear Ordering Problem (LOP) is a Combinatorial Optimization Problem (COP) that consists in finding the joint permutation of rows and columns that maximizes the sum of the entries above the main diagonal. Thus, any given permutation $\sigma \in \Sigma_n$ is evaluated as

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

From a Fourier transform perspective, the LOP consists of two non-constant components that capture the different order information of the objective function (Elorza et al., 2022). Thus, Eq 1 can be seen as the sum of two orthogonal functions, plus a constant that comes from the mean objective value:

$$f(\sigma) = f_{\rho_{(n-1,1)}}(\sigma) + f_{\rho_{(n-2,1,1)}}(\sigma) + \overline{f} \ .$$

Note that $f_{\rho_{(n-1,1)}}$ and $f_{\rho_{(n-2,1,1)}}$ come from the inverse Fourier transform, where $f_{\rho_{(n-1,1)}}$ captures the first-order information of the problem and $f_{\rho_{(n-2,1,1)}}$ measures its second-order information.

XVI XVI Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (maeb 2025).

Elorza et al. (2022) proved that the optimization problem defined by the first-order component $f_{\rho_{(n-1,1)}}$ can be exactly solved in polynomial time. In the same work, they also showed that the problem defined by $f_{\rho_{(n-2,1,1)}}$ remains NP-Hard, which implies that the complexity of the LOP stems from the second-order component. Thus, we hypothesize that modifying the proportion of the P and NP-Hard components in the target instance could allow us to control the properties of the objective function in a way that is beneficial for optimization. This hypothesis is validated through the proposal of a meta-heuristic algorithm based on this foundation, called P-Descent Search (PDS).

The main drawback of the previous strategy is that working with the decomposed functions is very costly (Maslen, 1998). Directly evaluating $f_{\rho_{(n-1,1)}}$ and $f_{\rho_{(n-2,1,1)}}$ requires performing matrix operations that introduce a considerable computational burden, limiting the practical application of the Fourier transform to small problem sizes. To avoid this issue, we propose an alternative approach that focuses on decomposing the problem instance instead of the objective function. This instance decomposition, which does not require computing the Fourier coefficients, allows us to access the information provided by the Fourier transform much more efficiently. We denote this method as P-NP instance decomposition. The PDS algorithm is built upon this new, more efficient framework.

2 P-NP instance decomposition

Given an LOP instance $M = [m_{i,j}]_{n\times n}$, the P-NP instance decomposition consists in finding two LOP instances defined by a pair of matrices $P = [p_{i,j}]_{n\times n}$ and $H = [h_{i,j}]_{n\times n}$ such that $M = P + H$ whose objective functions are equal to the $f_{\rho_{(n-1,1)}}$ and $f_{\rho_{(n-2,1,1)}}$ functions of the original instance, respectively (plus a constant). This is equivalent to solving a system of linear equations of the form

$$\begin{cases} p_{i,j} + h_{i,j} = m_{i,j} & \forall\, i,j = 1,...,n \\ p_{i,j} - p_{j,i} + p_{j,k} - p_{k,j} + p_{k,i} - p_{i,k} = 0 & \forall\, i,j,k = 1,...,n \\ \sum_{j=1}^{n}(h_{i,j} - h_{j,i}) = 0 & \forall\, i = 1,...,n \end{cases},$$

which is always solvable regardless of the input instance M. The previous system can be efficiently constructed and solved in $O(n^4)$ by considering that the coefficient matrix is always the same for a certain instance size n. Once the instance is decomposed, the value of $f_{\rho_{(n-1,1)}}$ and $f_{\rho_{(n-2,1,1)}}$ can be efficiently recovered from P and H in just $O(n^2)$.

3 P-Descent Search

Given the P-NP instance decomposition of an LOP, it is possible to generate surrogate instances by modifying the proportion of the P and NP-Hard components. According to experimental results, moderately increasing the proportion of the P component results in a less rugged fitness landscape with a similar ranking of solutions. This allows us to create surrogate instances that are more suitable for optimization while maintaining as much information about the original problem as possible.

Based on this idea, we propose a meta-heuristic algorithm that uses the P-NP instance decomposition to create a sequence of less rugged surrogate instances by iteratively adjusting the proportion of the P and NP-Hard components. We call this method P-Descent Search (PDS). Given an LOP instance $M = P + H$, PDS starts by creating an initial surrogate instance as

$$M_{\alpha_0} = (1+\alpha_0)\cdot P + (1-\alpha_0)\cdot H \quad \text{s.t.} \quad \alpha_0 \in [0,1]\ ,$$

where the P component has an equal or higher weight than the NP-Hard component in the linear combination. Once the initial instance is defined, PDS performs a greedy local search using the insert neighborhood on M_{α_0} starting from a random solution. The output of this process is considered as a starting point for another greedy local search on a new instance M_{α_1} with $0 \leq \alpha_1 < \alpha_0$. This chained procedure is carried out iteratively, feeding the output of a local search on M_{α_i} as the input for a local search on $M_{\alpha_{i+1}}$ such that $0 \leq \alpha_{i+1} < \alpha_i$. The value of α_{i+1} is calculated in an adaptive way to avoid redundant local searches that do not lead to changes in the solution. When the original instance is reached ($\alpha_i = 0$), the loop stops and the best solution found during the search is returned.

3.1 Experimentation

In this section, we test a multi-start version of PDS, denoted as MS-PDS, consisting of a collection of independent PDS runs with different initial solutions. Note that, if the starting point of the algorithm

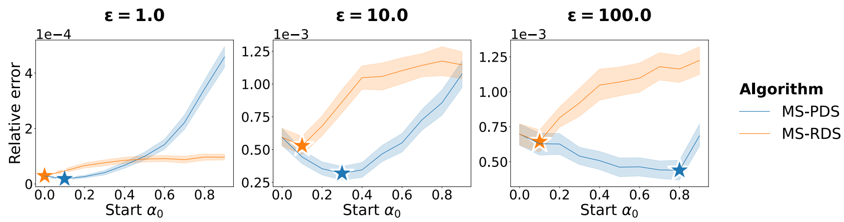

Figure 1: Average relative error of MS-PDS and MS-RDS with respect to the best solutions found according to α_0. The best-case configurations are marked by a star symbol.

is set to $\alpha_0 = 0$, the MS-PDS is equivalent to a classical Multi-Start Local Search (MS-LS) on the original problem. Thus, if the optimal performance of MS-PDS is achieved with $\alpha_0 > 0$, then the information derived from the P-NP instance decomposition could be useful for optimization.

The experimentation is conducted on a set of artificial LOP instances whose entries are generated based on a $\mathcal{U}(0, 1)$ distribution ($n = 100$). We consider instances in which the P and NP-Hard components have different degrees of influence on the objective function. This feature is controlled using an ϵ parameter, where the value of ϵ positively correlates with the weight of the NP-Hard component in the problem. The algorithm is evaluated on three sets of 100 generated instances with $\epsilon = \{1, 10, 100\}$. In each test instance, MS-PDS is executed once with a time limit of n seconds for each $\alpha_0 = \{0.0, 0.1, ..., 0.9\}$. In order to check if the observed performance is due to the P-NP instance decomposition, we carry out additional experiments by repeating the algorithm executions with a random instance decomposition. We call this baseline strategy MS-RDS. The average relative error obtained for each instance set, algorithm and parameter configuration is shown in Figure 1.

In all cases, the best average performance of MS-PDS is achieved with $\alpha_0 > 0$, thus supporting that MS-PDS obtains better results than a classical MS-LS that only operates on the original problem. Moreover, as the value of ϵ increases, the algorithm requires starting closer to the P component ($\alpha_0 \to 1$) to exploit the benefits of the Fourier transform-based strategy. This suggests that there exists a relationship between the weight of the P-NP components in the target instance and the behaviour of MS-PDS. These particularities are not observed in the random decomposition-based MS-RDS, proving that the performance of MS-PDS is caused by the P-NP instance decomposition.

4 Conclusions and future work

In this work, we have presented a P-NP instance decomposition strategy for the LOP that allows us to efficiently access the information provided by the Fourier transform. Based on this new framework, we have proposed a meta-heuristic algorithm called P-Descent Search (PDS) that increases the proportion of the P component to obtain a sequence of surrogate instances that are more suitable for optimization. Experimental results show that the multi-start version of PDS exhibits a promising performance that seems related to specific characteristics of the problem.

Although this work has focused on the LOP, we plan to extend the proposed methodology to other problems. Given an arbitrary COP, could we design a similar instance decomposition based on the Fourier transform? If so, would increasing the proportion of lower-order components create surrogate instances suitable for optimization? Addressing these questions would allow us to further study the potential of the information derived from the Fourier transform in guiding meta-heuristic algorithms.

References

Benavides, X., Hernando, L., Ceberio, J., and Lozano, J. A. (2025). P-np instance decomposition based on the fourier transform for solving the linear ordering problem. *Evolutionary Computation Journal*. Accepted.

Elorza, A., Hernando, L., and Lozano, J. A. (2022). Transitions from p to np-hardness: the case of the linear ordering problem. *2022 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8.

Maslen, D. (1998). The efficient computation of fourier transforms on the symmetric group. *Mathematics of Computation*, 67(223):1121–1147.