Balanced Bipartition into Homogeneous Groups with Positional Correspondence

Margarita Ruiz-Olazar[1], Diego Ihara[1], Benjamín Barán[1], and Miguel Méndez-Garabetti[2,3,4,5]*

[1] Universidad Comunera, Asunción, Paraguay
margarita.ruiz@ucom.edu.py, diego.ihara@ucom.edu.py, bbaran@cba.com.py
[2] Universidad Nacional de Cuyo, Mendoza, Argentina
miguel.mendez@uncuyo.edu.ar
[3] Universidad Nacional de San Juan, San Juan, Argentina
[4] Universidad de Mendoza, Mendoza, Argentina
[5] Universidad CAECE, Mar del Plata, Argentina

Abstract. In many applications requiring fair comparisons (e.g., control vs. treatment or equitable resource allocation), forming "internally compact" groups is not sufficient: it is necessary to partition a dataset into two balanced subsets that are as similar as possible to each other, and additionally, with an explicit one-to-one correspondence between elements. This work studies the case of two groups ($c = 2$) as a combinatorial optimization problem, minimizing a cost function based on the sum of squared distances between pairs matched at equivalent positions (positional matching). As the main contribution, a hybrid heuristic is proposed that constructs an initial solution via nearest neighbors supported by spatial indexing (KD-Tree) and refines it with local search through swaps accepted by pure descent, exploiting the additive nature of the cost to evaluate improvements efficiently. The experimental evaluation on synthetic and real datasets shows that the method attains consistently low objective values, practical execution times, and stable behavior across runs. Finally, future directions are discussed to extend the approach to the general case $c > 2$ and study scalability strategies (e.g., parallelization and ANN structures), maintaining positional correspondence as the central constraint.

Keywords: anticlustering · balanced partitioning · positional matching · KD-Tree · local search · combinatorial optimization

* This work was carried out during a research stay funded by the Consejo Nacional de Ciencia y Tecnología (CONACYT), Paraguay, in support of researcher Miguel Méndez-Garabetti, under Resolution No. [459/2025].

Bipartición balanceada en grupos homogéneos con correspondencia posicional

Resumen. En numerosas aplicaciones donde se requieren comparaciones justas (por ejemplo, control vs. tratamiento o asignación equitativa de recursos), no basta con formar grupos "internamente compactos": se necesita particionar un conjunto de datos en dos subconjuntos balanceados que sean lo más similares posible entre sí y, además, con correspondencia explícita uno a uno entre elementos. Este trabajo estudia el caso de dos grupos ($c = 2$) como problema de optimización combinatoria, minimizando una función de costo basada en la suma de distancias cuadráticas entre pares emparejados en posiciones equivalentes (matching posicional). Como contribución principal, se propone una heurística híbrida que construye una solución inicial mediante vecinos cercanos apoyados en indexación espacial (KD-Tree) y la refina con búsqueda local por intercambios (swaps) aceptados por descenso puro, explotando la naturaleza aditiva del costo para evaluar mejoras de forma eficiente. La evaluación experimental sobre datasets sintéticos y reales muestra que el método obtiene soluciones de alta calidad con buen desempeño temporal y estabilidad entre ejecuciones. Finalmente, se discuten líneas futuras para extender el enfoque al caso general $c > 2$ y estudiar estrategias de escalabilidad (p. ej., paralelización y estructuras ANN), manteniendo la correspondencia posicional como restricción central.

Palabras clave: anticlustering · particionamiento balanceado · emparejamiento posicional · KD-Tree · búsqueda local · optimización combinatoria

1 Introduction

Partitioning a dataset into balanced subsets that are as similar as possible is essential in applications requiring fair comparisons, such as matched experimental designs or equitable resource allocation. Anticlustering methods [8] promote distributional equivalence between groups but do not enforce that each element has a structurally comparable counterpart at the same position in the other group—a constraint we term *positional one-to-one correspondence*. Matching methods [9] impose such correspondence but do not primarily optimize group-level similarity under an anticlustering-style objective. In previous work [10], we addressed the general multi-group case via a Genetic Algorithm; that formulation raises computational challenges for large datasets. The present paper focuses on the bipartition case ($c = 2$), where the positional objective decomposes into independent row-wise terms, enabling efficient local evaluation.

We propose a hybrid heuristic combining KD-Tree-based initialization with stochastic swap-based refinement. The main contributions are: (i) derivation of the bipartition form of W_{pos}; (ii) a four-phase heuristic (preprocessing, spatial

indexing, greedy pairing, local refinement); and (iii) an empirical evaluation of solution quality, stability, and execution time on datasets of varying size and dimensionality.

The remainder of this paper is organized as follows. Section 2 reviews related work on anticlustering, the Maximally Diverse Grouping Problem, and matching methods. Section 3 formalizes the problem and derives the bipartition form of the positional objective. Section 4 presents the proposed four-phase heuristic. Section 5 reports the experimental evaluation, including an ablation study, a sensitivity analysis of the refinement parameter, and scalability results on five datasets. Section 6 discusses the findings and concludes with directions for future work.

2 Related Work

The problem considered herein lies at the intersection of anticlustering, the Maximally Diverse Grouping Problem (MDGP), and multi-group matching. While involving complex combinatorial assignments, none of these lines simultaneously enforce equal group size, global similarity, and explicit positional one-to-one correspondence between elements.

Anticlustering aims to construct groups with maximally similar distributions, originating with iterative reassignments by Späth [12] and formalized via a k-means objective inversion by Papenberg and Klau [8]. This framework has been extended to balance higher-order moments [7], combine diversity and dispersion [3], optimize MDGP objectives [15], and scale to massive datasets [1]. Although these methods ensure global distributional equivalence, they do not enforce explicit local positional correspondence between elements across groups.

Similarly, the NP-hard [4] MDGP [14] assigns elements to maximize intra-group diversity. Various metaheuristics, such as tabu search [5] and iterated maxima strategies [6], have been proposed, alongside balanced block-constrained formulations [11]. While handling equal-size groups and complex assignments, MDGP variants neither seek positional correspondence nor minimize pairwise dissimilarity between aligned elements.

Conversely, matching methods, such as propensity-score matching [9, 13], explicitly enforce element-to-element correspondence by pairing similar units. However, while strong in local alignment, they do not primarily optimize the global distributional similarity of the resulting groups compared to anticlustering objectives.

This highlights a clear methodological gap: anticlustering provides global equivalence without local correspondence, whereas matching provides local correspondence without global equivalence. Our previous work [10] addressed this combined requirement for multiple groups via an evolutionary approach. In the present paper, we focus on the bipartition case ($c = 2$). This specialization enables the objective to be evaluated locally at the paired-row level, facilitating a highly efficient heuristic treatment via swap-based refinement.

3 Problem Formulation

Let $X = \{x_1, x_2, \ldots, x_N\}$ be a set of N elements, where each element is represented by a vector of D numerical attributes:

$$x_i = (x_i^1, x_i^2, \ldots, x_i^D) \in \mathbb{R}^D.$$

The goal is to build two disjoint groups, denoted by S_1 and S_2, each of size M, while preserving an explicit one-to-one correspondence between their elements. This means that the solution is not only a partition into two subsets, but also an ordered configuration in which the element placed at position i in S_1 must be compared with the element placed at the same position i in S_2.

To quantify the quality of such a configuration, we use the *Positional Accumulated Variance* criterion, denoted by W_{pos}. In its general form for c groups, the objective is:

$$W_{pos} = \sum_{i=1}^{M} \sum_{j=1}^{c} \sum_{k=j+1}^{c} \|S_j[i] - S_k[i]\|^2 , \tag{1}$$

where $S_j[i]$ denotes the element assigned to position i in group S_j, and $\|\cdot\|^2$ is the squared Euclidean norm.

In the particular case considered here, namely $c = 2$, the expression reduces to:

$$W_{pos} = \sum_{i=1}^{M} \|S_1[i] - S_2[i]\|^2 . \tag{2}$$

This formulation has a direct interpretation: each row i defines a matched pair, and the global objective is simply the sum of squared distances over all such pairs. Lower values of W_{pos} indicate that the two groups are composed of more similar counterparts at corresponding positions. Fig. 1 illustrates the matrix structure induced by this formulation. In the bipartition case, the solution can be seen as an $M \times 2$ matrix, where each row corresponds to one matched pair. This row-wise structure is important because it also determines the locality of the refinement moves introduced later.

A feasible solution must satisfy three conditions. First, the groups must be disjoint:

$$S_1 \cap S_2 = \varnothing. \tag{3}$$

Second, both groups must contain exactly M elements:

$$|S_1| = |S_2| = M. \tag{4}$$

Third, when $N > 2M$, not all elements need to be included in the paired configuration. The remaining elements are placed in a reserve pool B, so that:

$$X = S_1 \cup S_2 \cup B, \qquad S_1 \cap S_2 = S_1 \cap B = S_2 \cap B = \varnothing. \tag{5}$$

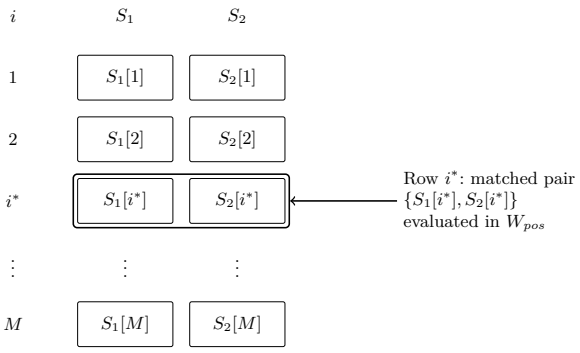

Fig. 1: Matrix representation $M \times 2$ of the bipartition. Each column corresponds to one group S_j and each row i defines a matched pair $\{S_1[i], S_2[i]\}$, the unit of comparison in W_{pos}.

Thus, the problem is not necessarily an exhaustive partition of all elements into the two matched groups. Instead, it consists of selecting two balanced subsets of size M together with their explicit positional arrangement, while optionally leaving a residual set outside the matched structure.

In this work, the reserve pool B plays no operational role during optimization: when $N > 2M$, the residual elements are simply excluded from the matched configuration once it is built. For the bipartition case studied here, $M = \lfloor N/2 \rfloor$, so $|B| = N - 2M \in \{0, 1\}$: B is therefore only a by-product of the cardinality constraint $|S_1| = |S_2| = M$, with no further role in the heuristic once Phase 3 (Algorithm 1) has completed.

4 Proposed Method

To address the bipartition problem defined above, we propose a hybrid heuristic composed of four sequential phases: preprocessing, spatial indexing, greedy initialization, and local refinement. The rationale behind this design is to combine a geometry-aware construction phase, capable of producing a low-cost initial configuration, with a subsequent local search stage that improves the solution through small, efficiently evaluated perturbations. The method is specifically tailored to the case $c = 2$. In this setting, the objective function decomposes into independent row-wise terms, which makes it possible to evaluate many candidate moves by recomputing only the affected pairs rather than the full solution.

Before optimization, the data are optionally normalized in order to avoid distortions caused by heterogeneous feature scales. The method supports standard normalization alternatives such as Z-score or Min-Max scaling, as well as the use of raw data when preserving the original physical interpretation of the variables is considered more important than scale invariance. This preprocessing affects only the distance computations used by the heuristic. The identities of the original instances remain unchanged throughout the algorithm, so that

the final paired configuration can always be interpreted in terms of the original dataset.

Once the data have been normalized, a KD-Tree [2] is built to accelerate nearest-neighbor queries. This structure is used during the constructive phase to identify candidate elements that are geometrically close to a selected seed.

The use of KD-Tree is especially appealing in low- and moderate-dimensional settings, where nearest-neighbor retrieval can be performed efficiently. Although the performance of exact spatial indexing may deteriorate as dimensionality increases, it still provides a simple and effective initialization mechanism for the problem sizes considered here.

The initial solution is built greedily by forming M pairs. At each step, one available element is selected as a seed, and its nearest available neighbor is retrieved using the KD-Tree. The two elements are then assigned to the same row of the bipartition matrix and removed from the available pool. This process is repeated until M pairs have been formed. The remaining elements, if any, are assigned to the reserve pool B. As a consequence, the initial solution already satisfies the balance constraint and starts from a configuration with relatively low positional cost, since each row is constructed from geometrically close elements. This initialization strategy is intentionally simple: it does not attempt to solve the global optimization problem during construction, but rather to position the search in a promising region of the solution space from which local refinement can proceed effectively.

Starting from the greedy configuration, the solution is refined through stochastic local search based on swap-like moves. A move is accepted only if it strictly reduces the objective value, that is, the search follows a pure descent criterion. The efficiency of this phase relies on the additive structure of Eq. (2). Since the objective is the sum of row-wise pair costs, a local modification affects only a small number of rows. Therefore, instead of recomputing the full W_{pos} after each candidate move, it is sufficient to evaluate only the rows involved.

Let rows p and q be affected by a move. The relevant local cost before the move is:

$$W_{\text{local}}^{\text{old}} = \|S_1[p] - S_2[p]\|^2 + \|S_1[q] - S_2[q]\|^2, \tag{6}$$

and the corresponding cost after the move is:

$$W_{\text{local}}^{\text{new}} = \|S_1'[p] - S_2'[p]\|^2 + \|S_1'[q] - S_2'[q]\|^2. \tag{7}$$

If $W_{\text{local}}^{\text{new}} < W_{\text{local}}^{\text{old}}$, the move is accepted. Because all unaffected rows keep the same cost, this criterion guarantees a strict improvement in the global objective.

A single swap operator is applied at each iteration.

Same-group case ($\pi_1 = \pi_2$). Two rows $p \neq q$ are drawn uniformly at random, together with two positions $\pi_1, \pi_2 \in \{1, 2\}$ drawn independently and uniformly, one for each row. The elements $S_{\pi_1}[p]$ and $S_{\pi_2}[q]$ are exchanged. When $\pi_1 = \pi_2$, the exchange takes place within the same group between rows p and q, preserving the balance structure while exploring alternative pairings among already selected elements.

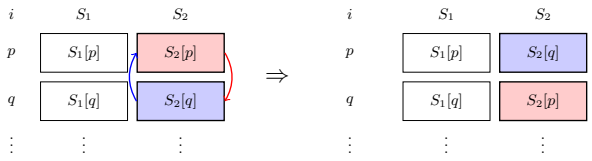

Fig. 2: Same-group case ($\pi_1 = \pi_2$) of the swap operator. Elements $S_2[p]$ (red) and $S_2[q]$ (blue) exchange positions between rows p and q within group S_2; the symmetric case within S_1 follows the same procedure. This case occurs with probability $1/2$ at each iteration.

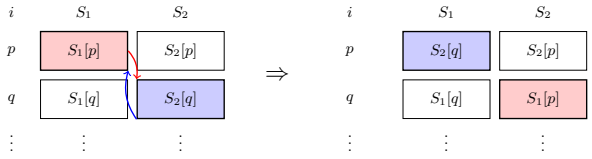

Fig. 3: Cross-group case ($\pi_1 \neq \pi_2$) of the swap operator. Element $S_1[p]$ (red) from row p and element $S_2[q]$ (blue) from row q exchange positions across groups S_1 and S_2. This case occurs with probability $1/2$ at each iteration.

Cross-group case ($\pi_1 \neq \pi_2$). When $\pi_1 \neq \pi_2$, the same exchange crosses groups S_1 and S_2 between rows p and q. Since π_1 and π_2 are independent and uniform over $\{1, 2\}$, the same-group and cross-group cases each occur with probability $1/2$ at every iteration; no element from the reserve pool B is involved in this operator. Fig. 2 and Fig. 3 illustrate the two cases.

Algorithm 1 summarizes the complete hybrid heuristic. The procedure begins with optional normalization and the construction of a KD-Tree. It then forms an initial set of M pairs through greedy nearest-neighbor assignments. Finally, it applies a fixed number S of local-search iterations, each proposing one random swap between two rows and accepting it only if it strictly reduces the objective.

5 Experimental Evaluation

The empirical evaluation pursues four complementary objectives: (i) to assess the quality of the solutions produced by the proposed heuristic; (ii) to quantify the contribution of its main components; (iii) to analyze the sensitivity of the method to the refinement parameter S; and (iv) to examine whether execution times remain practical as dataset size and dimensionality increase.

Algorithm Parameters. All experiments were conducted with $c = 2$, which implies $M = \lfloor N/2 \rfloor$ matched pairs. Unless otherwise stated, the number of refinement iterations was fixed to $S = 10{,}000$, a choice justified in Section 5.2. The algorithm is stochastic in two respects: (i) the constructive phase depends on

Algorithm 1: Hybrid Heuristic for Balanced Bipartition with Positional Correspondence

Input: Dataset X, number of pairs M, number of local-search iterations S
Output: Pair configuration $\mathcal{P}^*$, reserve pool B

`// Phase 1: Preprocess`
$X_{norm} \leftarrow \textsc{Normalize}(X)$;

`// Phase 2: Index`
Build KD-Tree $\mathcal{T}$ on X_{norm};

`// Phase 3: Initialize (greedy pairs)`
$\mathcal{P} \leftarrow \emptyset$; $X_{avail} \leftarrow X_{norm}$; **while** $|\mathcal{P}| < M$ **do**
  $s \leftarrow \textsc{SelectRandomSeed}(X_{avail})$;
    $k \leftarrow \min\big(\max(c \cdot 10, 50), |X_{avail}| + 1\big)$`//` $c = 2$ `here`
  $C \leftarrow$ the k nearest neighbors of s in $\mathcal{T}$; **if** $C \cap (X_{avail} \setminus \{s\}) \neq \varnothing$ **then**
    $nn \leftarrow$ closest element of $C \cap (X_{avail} \setminus \{s\})$ to s;
  else
    $nn \leftarrow$ uniform random draw from $X_{avail} \setminus \{s\}$ `// fallback`
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{(s, nn)\}$; $X_{avail} \leftarrow X_{avail} \setminus \{s, nn\}$;
$B \leftarrow X_{avail}$ `// residual elements,` $|B| = N - 2M \in \{0, 1\}$`, unused hereafter`

`// Phase 4: Refine (local search)`
for $t = 1$ **to** S **do**
  Select rows $P_a, P_b \in \mathcal{P}$ $(a \neq b)$ and positions $\pi_1, \pi_2 \in \{1, 2\}$, all uniformly and independently at random
  if $W_{loc}^{new}\big(\textsc{Swap}(P_a[\pi_1], P_b[\pi_2])\big) < W_{loc}(P_a, P_b)$ **then**
    Accept swap

return $\mathcal{P}^*$, B

the internal nearest-neighbor retrieval process of the KD-Tree, and (ii) the rows and element positions involved in each local move are selected at random.

Quality Metric. Since all experiments fix $c = 2$, the objective function reduces to the sum of squared Euclidean distances between the two elements of each matched pair:

$$W_{pos} = \sum_{i=1}^{M} \|x_1^{(i)} - x_2^{(i)}\|^2, \tag{8}$$

where $x_1^{(i)}$ and $x_2^{(i)}$ denote the two elements of the i-th pair. Lower values of W_{pos} indicate greater positional homogeneity. Prior to distance computation, all attributes were normalized to the range $[0, 1]$ using column-wise min-max scaling.

Datasets. Five datasets covering a broad range of sizes and dimensionalities were selected (Table 1). This collection includes both low-dimensional benchmark datasets and large-scale high-dimensional instances.

Table 1: Datasets used in the experimental evaluation.

Dataset	N	D	**Source**	**Access**
Synthetic 2D	1 000	2	`make_blobs` (scikit-learn)	scikit-learn
Iris	150	4	UCI ML Repository	scikit-learn
HCV	589	12	UCI ML Repository	UCI
SIFT	10 000	128	TEXMEX (INRIA)	Public
MNIST	10 000	784	OpenML / NIST	scikit-learn

Synthetic 2D was generated with `make_blobs` (10 centers, `random_state=42`). **HCV** contains clinical attributes from hepatitis C patients; records with missing values and non-numeric columns were removed. **SIFT** corresponds to descriptors from the `siftsmall_base` subset of the TEXMEX benchmark, stored in `.fvecs` format. **MNIST** comprises the first 10,000 images of the standard handwritten-digit dataset represented as 784-dimensional vectors.

Execution Protocol. The proposed algorithm was executed 30 independent times on each dataset using different random seeds ($\texttt{seed} = 42 + i$, $i = 0, \ldots, 29$), following standard practice in heuristic evaluation. For the ablation study, the number of repetitions was reduced to 10 for SIFT and 5 for MNIST due to their higher computational cost. Therefore, those ablation results should be interpreted as indicative estimates rather than as statistical estimates directly comparable in strength to those of the smaller datasets. By contrast, the scalability results reported in Table 3 are based on 30 full executions for all five datasets. Statistical significance was assessed using the two-sided Mann–Whitney U test, which is appropriate when no normality assumption is made on the distribution of the objective values.

Computational Environment. All experiments were conducted on a machine equipped with an AMD Ryzen 7 PRO 3700U processor with Radeon Vega Mobile Graphics (4 cores, 8 threads), 29 GiB of available system memory, and Ubuntu 22.04.5 LTS (Jammy Jellyfish) running Linux kernel 6.8.0-107-generic. The implementation was developed in Python 3.10, using NumPy for vectorized operations and `scipy.spatial.KDTree` for the constructive phase. No explicit parallelism was employed; all executions were performed sequentially on a single core.

5.1 Ablation Study

To evaluate the contribution of each component of the hybrid heuristic, we conducted an ablation study with four configurations. **Random** produces a balanced pairing with no structural guidance. **Random+Swaps** adds local refinement to that random initialization, making it possible to assess the isolated effect of the swap phase. **Greedy** corresponds to the KD-Tree-based constructive phase alone, without any subsequent refinement. **Proposed** denotes the

complete method, combining the greedy initialization with $S = 10{,}000$ refinement iterations. Table 2 summarizes the mean and standard deviation of W_{pos} obtained by each configuration.

Table 2: Ablation study: $\bar{W}_{pos} \pm \sigma$ for each algorithmic variant ($c = 2$, $S = 10{,}000$).

Dataset	Random	Rand.+Swaps	Greedy	Proposed
Synthetic 2D	136.3 ± 3.8	11.0 ± 1.3	10.5 ± 0.0	$\mathbf{3.5 \pm 0.5}$
Iris	40.9 ± 4.3	1.2 ± 0.2	1.4 ± 0.0	$\mathbf{0.9 \pm 0.0}$
HCV	7089.0 ± 123.4	3176.2 ± 226.0	2429.8 ± 0.0	$\mathbf{2083.9 \pm 91.4}$
SIFT	63461.3 ± 292.6	47994.7 ± 188.9	15643.8 ± 0.0	$\mathbf{15424.6 \pm 33.3}$
MNIST	527214.9 ± 1212.3	446888.8 ± 581.9	178121.4 ± 0.0	$\mathbf{176671.9 \pm 239.0}$

The comparison between **Greedy** and **Proposed** was evaluated with the Mann–Whitney U test, yielding $p < 0.05$ on all five datasets. This indicates that the local refinement stage provides a statistically significant improvement over the constructive phase alone in every case considered.

Two observations are particularly relevant. First, the KD-Tree-based initialization makes a strong contribution to final solution quality. The **Random+Swaps** variant, despite using the same refinement budget as the full method, remains consistently far from the quality achieved by the proposed algorithm. For instance, on SIFT it yields $\bar{W}_{pos} = 47{,}994$, whereas the proposed method achieves $\bar{W}_{pos} = 15{,}424$, a relative difference of 69.0%. This suggests that local refinement alone is not sufficient to fully compensate for a poor initial configuration under the selected iteration budget.

Second, the local refinement phase contributes consistent additional improvements on top of the greedy initialization. The reduction in W_{pos} between **Greedy** and **Proposed** ranges from 0.9% on SIFT to 33.4% on Synthetic 2D. The relative impact of refinement decreases as dimensionality increases, which is consistent with the fact that the constructive phase already produces effective initial pairings in high-dimensional spaces.

5.2 Sensitivity to the Refinement Parameter S

To justify the choice of $S = 10{,}000$, the proposed method was evaluated with $S \in \{0, 100, 500, 1{,}000, 5{,}000, 10{,}000, 50{,}000\}$ on three representative small- and medium-scale datasets (Synthetic 2D, Iris, and HCV), using 10 repetitions per configuration. This analysis was not extended to SIFT and MNIST because of their higher computational cost per run; moreover, since the marginal contribution of refinement is smaller in very high-dimensional settings, the selected datasets provide a conservative scenario for studying convergence behavior. Fig. 4 shows the relative objective value as a function of the refinement budget, normalized with respect to the Greedy initialization ($S = 0$, set to

100%). The curves reveal a rapid improvement during the first 10^3 swap iterations, followed by a clear stabilization from $S = 5{,}000$ onward. This behavior is especially pronounced in Synthetic 2D, where the constructive phase still leaves substantial room for improvement, whereas Iris and HCV exhibit a smoother decrease, indicating that the greedy initialization already provides relatively strong starting solutions.

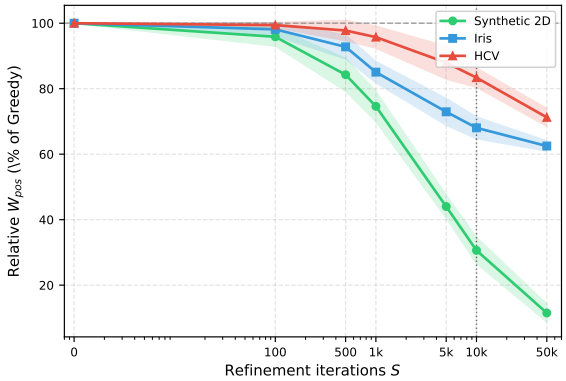

Fig. 4: Relative objective value as a function of the refinement budget S, normalized with respect to the Greedy initialization ($S = 0$, set to 100%). Curves show the mean over 10 independent runs and shaded bands indicate ± 1 standard deviation. Most of the improvement is achieved within the first 10^3 iterations, while gains become progressively smaller beyond $S = 5{,}000$. The vertical dashed line marks the value $S = 10{,}000$ adopted in the experimental study.

Increasing the refinement budget from 10,000 to 50,000 iterations yields gains below 1% in Iris and HCV, while Synthetic 2D still improves, although with clearly diminishing returns. Since runtime grows approximately linearly with S, the results support $S = 10{,}000$ as a reasonable compromise between solution quality and computational cost for the present study.

5.3 Scalability Results

Table 3 summarizes the descriptive statistics of the proposed algorithm over 30 independent executions on each dataset.

Execution time increases with both dataset size and dimensionality, but remains within practical limits for the instances considered: below one second on Iris and under one minute on MNIST ($N = 10{,}000$, $D = 784$). In addition, the relative standard deviation $\sigma/\bar{W}_{pos}$ ranges from 0.17% on SIFT to 15.4% on Synthetic 2D, indicating that the method produces solutions of comparable quality across independent runs. The larger relative variation observed in

Table 3: Results of the proposed algorithm ($c = 2$, $S = 10{,}000$, 30 independent runs).

Dataset	M	n	$\bar{W}_{pos}$	σ	W_{pos}^{min}	W_{pos}^{max}	$\bar{t}$ (s)
Synthetic 2D	500	30	3.534	0.543	2.521	4.520	0.82
Iris	75	30	0.918	0.036	0.843	0.983	0.70
HCV	294	30	2 083.886	93.005	1 884.077	2 263.661	0.76
SIFT	5 000	30	15 427.707	33.590	15 371.676	15 492.184	17.29
MNIST	5 000	30	176 559.824	295.413	176 074.447	177 145.002	54.70

low-dimensional datasets is consistent with the greater room for refinement improvements in those instances.

5.4 Analysis and Limitations

Absolute values of W_{pos} increase with D because squared Euclidean distance accumulates more terms as dimensionality grows. Therefore, direct comparison of raw W_{pos} values across datasets with different dimensionalities is not meaningful without additional normalization. The experimental analysis does not include exact optimal solutions. For instances of practical size, the underlying problem is computationally intractable to solve exactly within reasonable time. For this reason, solution quality should be interpreted primarily in relative terms: through internal comparisons among algorithmic variants, as well as through the consistency and stability of the results across runs. On MNIST ($D = 784$), KD-Tree efficiency degrades substantially, which helps explain why runtime is approximately three times higher than on SIFT despite both datasets having the same number of instances. In very high-dimensional settings, approximate nearest-neighbor structures such as HNSW or LSH may provide a more favorable trade-off between initialization cost and final solution quality.

Finally, the experimental study does not include a direct comparison with related approaches. The methods surveyed in Section 2 address the three requirements of the present formulation—equal group size, global group similarity, and explicit positional correspondence—only partially, so that adapting them to serve as valid baselines requires enforcing the positional constraint as an additional step. Designing such a comparison protocol constitutes a limitation of the present study and is addressed as future work.

6 Discussion and Conclusions

The experiments support three main conclusions. First, the ablation study shows that the constructive and refinement phases play complementary roles: KD-Tree initialization places the search in a low-cost region, while swap-based refinement yields statistically significant improvements. The weak performance of **Random+Swaps** indicates that local search alone cannot recover from a poor initialization under the adopted refinement budget. Second, the convergence analysis shows that most gains are obtained in the early refinement steps, after which

improvements become marginal. This supports the use of a moderate refinement budget rather than a substantially larger one. Third, the method remains practical across datasets of very different sizes and dimensionalities. Although runtime increases on high-dimensional datasets such as MNIST, the algorithm still produces stable solutions within times compatible with offline optimization.

At the same time, the reduced impact of refinement in high-dimensional spaces suggests that the constructive phase already captures much of the accessible local structure. This points to a trade-off: while KD-Tree initialization is effective at moderate dimensionality, approximate neighbor-search structures may be preferable when D is very large.

This work presented a hybrid heuristic for balanced bipartition with explicit positional correspondence, targeting the $c = 2$ case of a combinatorial optimization problem with applications in fair comparison design, matched experimental settings, and equitable resource allocation. The proposed method combines KD-Tree-based initialization with stochastic local refinement, exploiting the additive structure of the positional objective in the bipartition setting.

Three main conclusions emerge from the study. First, the constructive phase based on nearest-neighbor search makes a strong contribution to the quality of the final solution. Second, the local refinement phase consistently improves the greedy initialization and does so with statistically significant gains across all evaluated datasets. Third, the method remains computationally practical across a broad range of dataset sizes and dimensionalities, while maintaining stable performance across independent runs.

The analysis of the refinement parameter further shows that most of the benefit of local search is achieved within the first few thousand iterations, which supports the selection of $S = 10{,}000$ as a reasonable compromise between quality and runtime in the present configuration.

Future work will focus on extending the present formulation beyond the bipartition case. In particular, the next step is to generalize the proposed strategy to the case of $c > 2$ groups, where the construction of matched tuples and the design of efficient local operators become substantially more complex. Additional directions include exploring parallel implementations for the refinement phase and replacing exact KD-Tree search with approximate nearest-neighbor structures in very high-dimensional settings. A further direction is the design of a systematic comparison protocol against related approaches—including anticlustering-based methods and propensity-score matching—adapted to enforce the positional correspondence constraint, in order to situate the proposed heuristic within the broader landscape of balanced partitioning methods.

References

1. Baumann, P., Goldschmidt, O., Hochbaum, D.S., Yang, J.: A fast and effective method for Euclidean anticlustering: The Assignment-Based-Anticlustering algorithm (2026). https://doi.org/10.48550/arXiv.2601.06351

2. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Communications of the ACM **18**(9), 509–517 (1975). https://doi.org/10.1145/361002.361007
3. Brusco, M.J., Cradit, J.D., Steinley, D.: Combining diversity and dispersion criteria for anticlustering: A bicriterion approach. British Journal of Mathematical and Statistical Psychology **73**(3), 375–396 (2020). https://doi.org/10.1111/bmsp.12186
4. Feo, T.A., Goldschmidt, O., Khellaf, M.: One-half approximation algorithms for the k-partition problem. Operations Research **40**(S1), S170–S173 (1992). https://doi.org/10.1287/opre.40.1.S170
5. Gallego, M., Laguna, M., Martí, R., Duarte, A.: Tabu search with strategic oscillation for the maximally diverse grouping problem. Journal of the Operational Research Society **64**(5), 724–734 (2013). https://doi.org/10.1057/jors.2012.66
6. Lai, X., Hao, J.K.: Iterated maxima search for the maximally diverse grouping problem. European Journal of Operational Research **254**(3), 780–800 (2016). https://doi.org/10.1016/j.ejor.2016.04.011
7. Papenberg, M.: K-plus anticlustering: An improved k-means criterion for maximizing between-group similarity. British Journal of Mathematical and Statistical Psychology **77**(1), 80–102 (2024). https://doi.org/10.1111/bmsp.12315
8. Papenberg, M., Klau, G.W.: Using anticlustering to partition data sets into equivalent parts. Psychological Methods **26**(2), 161–174 (2021). https://doi.org/10.1037/met0000301
9. Rosenbaum, P.R., Rubin, D.B.: The central role of the propensity score in observational studies for causal effects. Biometrika **70**(1), 41–55 (1983). https://doi.org/10.1093/biomet/70.1.41
10. Ruiz-Olazar, M., Ihara, D., Barán, B.: Data partitioning into homogeneous groups using a genetic algorithm. In: 2025 IEEE CHILEAN Conference on Electrical, Electronics Engineering and Information and Communication Technologies (CHILECON). IEEE (2025)
11. Schulz, A.: The balanced maximally diverse grouping problem with block constraints. European Journal of Operational Research **294**(1), 42–53 (2021). https://doi.org/10.1016/j.ejor.2021.01.029
12. Späth, H.: Anticlustering: Maximizing the variance criterion. Control and Cybernetics **15**(2), 213–218 (1986)
13. Stuart, E.A.: Matching methods for causal inference: A review and a look forward. Statistical Science **25**(1), 1–21 (2010). https://doi.org/10.1214/09-STS313
14. Weitz, R.R., Lakshminarayanan, S.: An empirical comparison of heuristic and graph theoretic methods for creating maximally diverse groups. Omega **25**(4), 473–482 (1997). https://doi.org/10.1016/S0305-0483(97)00007-5
15. Yang, X., Cai, Z., Jin, T., Tang, Z., Gao, S.: A three-phase search approach with dynamic population size for solving the maximally diverse grouping problem. European Journal of Operational Research **302**(3), 925–953 (2022). https://doi.org/10.1016/j.ejor.2022.02.003