Meta-Learning-Based Algorithm Selection for Multi-Objective Wind Farm Layout Optimization

1st Gustavo J. N. Silva
Institute of Computing
Federal University of Bahia
Salvador, Brazil
gustavonovaes@ufba.br

2nd João G. L. S. G. Silva
Institute of Computing
Federal University of Bahia
Salvador, Brazil
joao.lofiego@ufba.br

3rd Islame F. C. Fernandes*
Institute of Computing
Federal University of Bahia
Salvador, Brazil
islame.felipe@ufba.br

***Abstract*—Wind farm layout optimization involves multiple conflicting objectives and highly irregular search landscapes, making the selection of suitable multi-objective metaheuristics a challenging task. Although meta-learning has been successfully employed to select metaheuristics for single-objective optimization, its application to real-world multi-objective engineering problems remains unexplored. This paper proposes a meta-learning approach for automatically selecting metaheuristics for the Multi-Objective Wind Farm Layout Optimization Problem (*MoWFLOP*). The proposed approach uses fitness landscape meta-features to predict the performance of multi-objective state-of-the-art metaheuristics under different quality indicators. We evaluate 90 datasets with different sampling configurations from 516 wind farms. The analysis comprised the results of the metaheuristics, the impact of sampling configurations, and the meta-learning performance for metaheuristic selection. Results show that the meta-learning approach surpassed the baseline of selecting the same algorithm for all instances.**

***Index Terms*—Multi-objective Optimization, Meta-learning, Wind Farm Design, Evolutionary Algorithms**

I. Introduction

Context and motivation. The global transition toward renewable energy has intensified the development of wind farm systems, with projections indicating even greater expansion in the coming years. Between 2025 and 2030, global onshore wind installations are expected to grow by 827 GW, averaging 138 GW per year, while offshore wind capacity is projected to increase from 8 GW in 2024 to 34 GW in 2030 [1].

Wind farm design involves several *multi-objective* optimization problems. One of the most relevant problems is determining the optimal placement of turbines to minimize construction costs and maximize energy production, referred to as *Multi-objective Wind Farm Layout Optimization Problem* (*MoWFLOP*) [3]. Turbine placement is critical to ensure the economic and energetic efficiency of wind farms. Studies have indicated that suboptimal turbine placement can reduce energy production by up to 28.9% [1] due to the reduced wind speed region generated by upstream turbines, named *wake effect* [2]. This phenomenon reduces the efficiency and power output of neighboring turbines, leading to significant economic losses and degraded operational performance.

*Corresponding author

Since *MoWFLOP* is *NP-hard* [3], metaheuristics have been widely employed to obtain high-quality solutions [4]–[6]. Nevertheless, these algorithms exhibit different performance profiles across *MoWFLOP* instances, reflecting structural variations in the underlying optimization landscape.

Related work. In recent years, several studies have used *meta-learning* to enhance the performance of metaheuristics in optimization problems [5]. Meta-learning exploits knowledge from past executions to predict algorithm performance on unseen instances [9]. For this purpose, *Fitness Landscape Analysis* (FLA) characterizes the landscape underlying an optimization problem using numerical measures, termed *meta-features* (e.g., ruggedness and dominance relations) [15].

Meta-feature extraction commonly involves sampling and evaluating a representative solution set. The most common sampling method is a *walk* over the landscape [11], which begins from an initial solution and iteratively explores its neighbors until meeting a stopping condition. Parameters, such as sample size, neighborhood structure, and stopping condition, significantly impact the informativeness of meta-features and the accuracy of meta-learning.

Although meta-learning and FLA have been widely studied in single-objective optimization, their application to multi-objective problems remains limited. The study in [8] investigated meta-features based on Pareto dominance and quality indicators, demonstrating strong correlations with the performance of evolutionary algorithms. The authors in [7] introduced decomposition-based meta-features, while those in [11] analyzed the cost and accuracy of sampling strategies and their impact on the algorithm selection. However, these studies focused mainly on theoretical and continuous problems, leaving the use of meta-learning and FLA in real-world engineering problems, such as *MoWFLOP*, unexplored. Moreover, the impact of different sampling strategies on meta-features for *MoWFLOP* remains an open research question.

Proposal. This paper proposes a meta-learning approach for algorithm selection for *MoWFLOP*. The proposed approach uses Pareto-based meta-features to train a Random Forest regression model that estimates the performance of a set of metaheuristics, namely *CoMOLS/D* [12], *MOEA/D* [14], and *NSGA-II* [13]. Given an unseen instance, the framework computes the meta-features and predicts the metaheuristic that performs best. The proposed approach can support engineers in wind farm design by automatically recommending the most effective algorithm, reducing computational effort and improving design efficiency.

Methodology. Comprehensive experiments used 516 wind farm instances. The approximation sets generated by the metaheuristics are compared using the Hypervolume (*HV*), Inverted Generational Distance (*IGD*), and ϵ indicators. Two sampling strategies, *random* and *adaptive walks*, extracted meta-features, producing 90 datasets with varying walk lengths and neighborhood rates. Meta-feature values, objective function evaluations, and Spearman correlations are analyzed. The merit indicator evaluated the proposed meta-learning approach in selecting the best-performing algorithm. Standard regression metrics MSE, RMSE, MAE, MAPE, and R^2 are also reported. All instances, datasets, source codes, and results are publicly available in the GitHub repository https://github.com/mowflopcec/wflopcec26.

Contribution. The main contributions of this work are: (i) A meta-learning framework for selecting multi-objective metaheuristics for the *MoWFLOP*, designed to support algorithmic decision-making in wind farm layout design; (ii) The use of Pareto-based fitness landscape meta-features to predict the performance of state-of-the-art multi-objective metaheuristics under multiple quality indicators; (iii) The first large-scale experimental study of landscape-aware algorithm selection for the *MoWFLOP*; (iv) A systematic analysis of the trade-off between sampling cost and meta-learning accuracy, quantifying the impact on meta-feature values and predictive performance.

II. Preliminaries

A. Multi-objective optimization

Multi-objective optimization considers $q > 1$ objective functions, a feasible solution space Γ, and an objective space $\Upsilon \subseteq \mathbb{R}^q$. A function $f : \Gamma \rightarrow \Upsilon$ assigns each $\gamma \in \Gamma$ to a vector $f(\gamma) = (f_1(\gamma), \ldots, f_q(\gamma))$. A generic formulation is given by expression (1).

$$\underset{\gamma \in \Gamma}{\text{optimize}}(f_1(\gamma), \ldots, f_q(\gamma)) \tag{1}$$

The term *optimize* denotes either minimization or maximization under the Pareto dominance relation. Without loss of generality, consider a minimization problem. For $\gamma, \gamma' \in \Gamma$, γ *dominates* γ' (denoted $\gamma \prec \gamma'$) if $f_k(\gamma) \leq f_k(\gamma')$ for all $k = 1, \ldots, q$, and $f_k(\gamma) < f_k(\gamma')$ for at least one k. Solutions γ and γ' are *incomparable* if neither $\gamma \prec \gamma'$ nor $\gamma' \prec \gamma$ holds. An *approximation set* $\Upsilon^{*\prime} \subseteq \Upsilon$ is a set where $\forall \upsilon, \upsilon' \in \Upsilon^{*\prime}$, $\upsilon = f(\gamma)$ and $\upsilon' = f(\gamma')$, $\gamma \nprec \gamma'$ and $\gamma' \nprec \gamma$.

B. Multi-Objective Wind Farm Layout Optimization Problem

MoWFLOP addresses the placement of τ wind turbines across a predefined set of available positions $\mathcal{P}$ in the open sea. For each position $i \in \mathcal{P}$, a binary decision variable z_i indicates whether a turbine is installed ($z_i = 1$) or not ($z_i = 0$). The installation cost at position i is denoted by C_i [4].

Each turbine produces a specific amount of energy from the free wind flow. The Jensen model [2] computes the wake effect among the turbines. Let E be a $|\mathcal{P}| \times |\mathcal{P}|$ matrix, where $E_{ii} \geq 0$ is the power produced by turbine i under free wind conditions, while $E_{ij} \leq 0$ is the power loss at turbine i due to the wake effect caused by turbine j, for $i \neq j$.

MoWFLOP minimizes construction cost and maximizes total power generation, as expressed in objective functions (2) and (3). Constraint (4) ensures exactly τ turbines are placed. Constraints (5) enforce a minimum separation distance σ between turbines, based on their Euclidean distance d_{ij}, to prevent excessive structural loads. Constraints (6) define the binary domain of decision variables.

$$\min \sum_{i \in \mathcal{P}} C_i z_i \tag{2}$$

$$\max \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{P}} E_{ij} z_i z_j \tag{3}$$

$$\sum_{i \in \mathcal{P}} z_i = \tau \tag{4}$$

$$z_i + z_j \leq 1 \quad \forall i, j \in \mathcal{P},\ d_{ij} \leq \sigma \tag{5}$$

$$z_i \in \{0, 1\} \quad \forall i \in \mathcal{P} \tag{6}$$

III. Meta-learning

Meta-learning is a data-driven approach aimed at predicting the most suitable optimization algorithm for a given problem instance by exploiting knowledge gathered from previously solved instances [9]. In this work, meta-learning is employed to predict the performance of multi-objective metaheuristics applied to *MoWFLOP*, enabling the automatic selection of the best-performing algorithm for new instances.

Figure 1 presents the proposed meta-learning pipeline. The process begins with *meta-data extraction*, which involves assessing the performance of metaheuristics and computing *meta-features* from *MoWFLOP* instances. Performance can be measured using any quality indicator ι. Section IV-B details the quality indicators used in this study. Meta-features are descriptive measures that characterize the search space. The second stage builds a *dataset* for supervised learning by combining meta-features with ι-values for each metaheuristic. Finally, a multi-output *Random Forest* (*RF*) regression model is trained to predict the ι-value for new instances. The metaheuristic with the best predicted ι-value is subsequently selected for that instance. The following sections detail meta-features and metaheuristics used in this study.

A. Meta-features extraction

This paper uses Pareto-based meta-features [11] to capture the structure of *MoWFLOP* search space by analyzing local dominance relations among neighboring solutions. Let $N : \Gamma \rightarrow 2^\Gamma$ be a neighborhood function, where $N(\gamma)$ contains solutions differing from γ by exactly one turbine location. A walk $W = (\gamma_0, \gamma_1, \ldots, \gamma_L)$ is a sequence of L neighboring solutions, where $\gamma_{t+1} \in N(\gamma_t)$ for every $t \in \{0, ..., L-1\}$. Four Pareto-based measures are computed for each γ_t:

1) `inf`: number of neighbors dominated by γ_t;

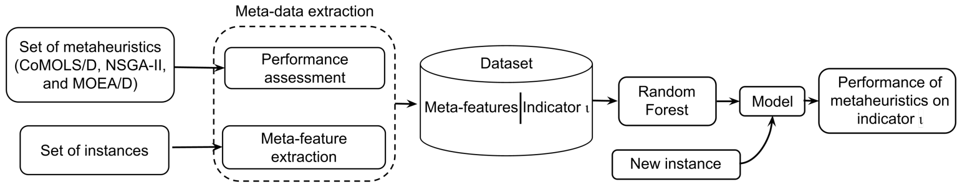


Fig. 1. Meta-learning pipeline for predicting metaheuristic performance on a quality indicator ι

2) `sup`: number of neighbors that dominate γ_t;
3) `inc`: number of neighbors incomparable to γ_t;
4) `ind`: number of locally non-dominated neighbors.

Each measure is aggregated along the walk using five statistics: the mean (`avg`) for the overall tendency of values, the standard deviation (`sd`) for overall variability, the first autocorrelation coefficient (r_1) for local smoothness or ruggedness, the kurtosis (`kr`) for tail heaviness and extreme-value behavior, and the skewness (`sk`) for distribution asymmetry. Combining the four Pareto-based measures with these five statistics yields 20 meta-features. For instance, `inf.avg` is the mean number of dominated neighbors, whereas `sup.sd` is the variability of dominating neighbors.

Meta-features are computed by sampling the solution space using two local search procedures: *random walk* and *adaptive walk* [11]. In the random walk, solution γ_{t+1} is randomly selected with uniform probability from a truncated neighborhood $N_r(\gamma_t) \subseteq N(\gamma_t)$, where $r \in (0, 1]$ is the neighborhood rate. Solutions in $N_r(\gamma_t)$ are sampled uniformly at random from $N(\gamma_t)$. Both r and L are user-defined parameters for random walk. In the adaptive walk, solution $\gamma_{t+1} \in N_r(\gamma_t)$ is one that dominates γ_t, and the walk terminates when no dominating solution exists. Here, only r is user-defined, while L depends on the number of dominating solutions in the walk.

The truncated neighborhood strategy reduces the meta-feature extraction complexity to $\Theta(L \cdot r)$. This paper examines the impact of various combinations of r and L on computational cost, meta-feature values, and meta-learning results.

B. Metaheuristics

Three state-of-the-art metaheuristics were considered due to their different techniques and well-established performance:

1) **CoMOLS/D** [12]: a decomposition-based coevolutionary local search with two populations: one drives convergence toward the Pareto front, while the other preserves diversity. Both populations evolve through a local search procedure and cooperate by sharing solutions. The neighborhood for *MoWFLOP* is defined as solutions differing by the position of a single turbine.
2) **NSGA-II** [13]: an elitist and dominance-based evolutionary algorithm. Each generation produces offspring through genetic operators, followed by non-dominated sorting and crowding distance to select individuals for the next population.
3) **MOEA/D** [14]: a decomposition-based evolutionary algorithm that optimizes multiple scalar subproblems in parallel. Neighboring subproblems exchange information to balance convergence and diversity.

The metaheuristics use the same solution encoding and genetic operators. Solutions are $|\mathcal{P}|$-dimensional vectors indicating turbine positions. The initial population is generated uniformly at random. Crossover uses a one-point strategy: the offspring inherits the left segment from one parent and fills the remaining positions with non-redundant turbines from the other parent. Mutation randomly selects a turbine and relocates it to a randomly chosen empty position.

IV. Experimental Methodology

A. Instance set

This study used 516 wind farm instances generated with the method proposed by [4]. Each instance features 1–3 zones with 0–3 obstacles and 462–47,902 available positions for placing 5–648 turbines of 15 MW. Additionally, 4–75 pre-fixed 10 MW turbines, representing nearby farms, are included, which are not optimized but may generate wake effects. Foundation costs for turbine placement are computed using the model in [2], considering water depths between 10 and 40 meters. The power, thrust curves, and wind scenarios from [4] are adopted in this study.

B. Multi-objective test and assessment

Results concern 20 executions of each metaheuristic per instance, with 10^6 objective function evaluations as the stopping criterion. The *irace* tool [16] tuned the parameters. All algorithms used a population size of 100. *NSGA-II* adopted mutation and crossover rates of 0.5 and 0.6, while *MOEA/D* used rates of 0.5 and 0.5. *CoMOLS/D* employed $\theta = 2$ to balance diversity and convergence.

HV, *IGD*, and additive ϵ quality indicators evaluated the approximation set Υ' obtained by the metaheuristics [10]. All indicators relied on a reference set constructed from the non-dominated union of solutions generated by *CoMOLS/D*, *NSGA-II*, and *MOEA/D*. All objective functions were normalized to the interval $[1, 2]$. *HV* measures the volume dominated by Υ', bounded by $(2.1, 0.9)$. *IGD* computes the average distance from each reference point to its nearest neighbor in Υ', whereas ϵ denotes the minimum additive factor required for Υ' to dominate the reference set. For all indicators, lower values indicate better approximation quality.

The Kruskal-Wallis test assessed statistical significance. This test first checks for any significant difference and then proceeds with pairwise post-hoc comparisons. All experimental data are available on GitHub.

C. Dataset construction

Datasets associate the meta-features with average ι-values, $\iota \in \{HV, IGD, \epsilon\}$. Each row represents a single *MoWFLOP* instance, while the columns store the 20 meta-features and the average performance of each metaheuristic with respect to ι. In total, 90 datasets were generated, 30 for each indicator ι. For each ι, the random walk produced 25 sampling configurations by combining parameters $L \in \{5, 10, 25, 50, 100\}$ and $r \in \{0.05, 0.1, 0.25, 0.5, 1.0\}$, while the adaptive walk generated five configurations with $r \in \{0.05, 0.1, 0.25, 0.5, 1.0\}$. The goal is to assign the meta-learning pipeline for different sampling configurations and quality indicators.

D. Meta-feature analysis

We evaluate the Spearman correlation between meta-features and the number of turbines τ for each (L, r) configuration [11]. We also report the variation of each meta-feature, referred to as *GAP*, defined in expression (7). For each meta-feature ξ, the *GAP* is the difference between the value of ξ under a given (L, r) configuration, denoted by $V_{L,r}(\xi)$, and the value of ξ under the most computationally expensive configuration $V_{\max}(\xi)$. For random walks, $V_{\max}(\xi)$ corresponds to $L = 100$ and $r = 1.0$; for adaptive walks, $V_{\max}(\xi)$ uses $r = 1.0$.

$$GAP = \frac{|V_{\max}(\xi) - V_{L,r}(\xi)|}{V_{\max}(\xi)} \tag{7}$$

E. Meta-learning test and assessment

The methodology in [11] evaluated the meta-learning performance. The instance set was divided into a training set $\mathcal{T}$ and a validation set $\mathcal{V}$. Set $\mathcal{T}$ was used to train the multi-output *RF* regression model. Given the meta-features of an unseen instance in $\mathcal{V}$, the model predicts the ι-value for each metaheuristic and outputs the best one. Random Search tuned the RF hyperparameters for each indicator: for HV, *n_estimators* = 292, *depth* = 20, *min_split* = 7, and *min_leaf* = 4; for IGD, *n_estimators* = 215, *depth* = 25, *min_split* = 3, and *min_leaf* = 1; and for ϵ, *n_estimators* = 100, *depth* = 30, *min_split* = 2, and *min_leaf* = 1.

The merit m^* [11] evaluated the distance between the predicted algorithm and two baselines: (i) the *Single Best Solver* (*SBS*), i.e., the algorithm with the best average performance over $\mathcal{T}$, and (ii) the *Virtual Best Solver* (*VBS*), which selects the best algorithm per instance. Let $\mathcal{A}$ be the set of algorithms and $\overline{r\iota}(\alpha, \mathcal{J})$ the average ι-value obtained by $\alpha \in \mathcal{A}$ over a subset of instances $\mathcal{J}$. The *SBS* and m^* are defined in (8) and (9), respectively, where α_s is the algorithm predicted by RF. Values $m^* = 0$ indicate a perfect oracle, and $m^* < 1$ ($m^* > 1$) indicate that RF performs better (worse) than *SBS*.

$$SBS = \arg\min_{\alpha \in \mathcal{A}} \overline{r\iota}(\alpha, \mathcal{T}) \tag{8}$$

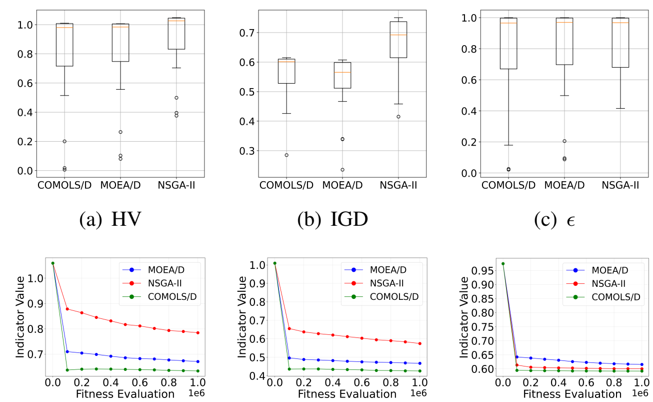
$$m^* = \frac{\overline{r\iota}(\alpha_s, \mathcal{V}) - \overline{r\iota}(VBS, \mathcal{V})}{\overline{r\iota}(SBS, \mathcal{V}) - \overline{r\iota}(VBS, \mathcal{V})} \tag{9}$$

Standard regression metrics MSE, MAE, RMSE, MAPE, and R^2 are also reported. A repeated random hold-out strategy with a 90/10% split between $\mathcal{T}$ and $\mathcal{V}$ was employed. Reported values correspond to the mean across 100 independent folds.

V. Results and Discussion

A. Metaheuristics results

CoMOLS/D achieved the best average *HV*, *IGD*, and ϵ in 98.45%, 96.51%, and 98.65% of the instances, respectively, indicating the strong effectiveness of the decomposition-based co-evolutionary search for *MoWFLOP*. Moreover, *MOEA/D* consistently outperformed *NSGA-II*, further reinforcing the superiority of decomposition-based strategies over non-dominated sorting in this problem.



(a) HV (b) IGD (c) ϵ

(d) HV (e) IGD (f) ϵ

Fig. 2. Boxplots and average evolution of HV, IGD, and ϵ across objective function evaluations

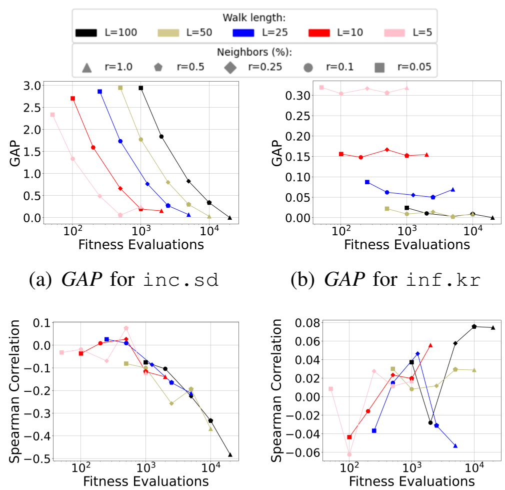
The Kruskal–Wallis test found no significant differences among the algorithms in 91.47% of the instances for HV, 66.47% for IGD, and 100% for ϵ. In the instances where statistical differences were detected, the pairwise tests indicated that *CoMOLS/D* outperformed *MOEA/D*, which in turn outperformed *NSGA-II*. Figures 2(a)–(c) illustrate the distributions of HV, IGD, and ϵ, respectively. While the metaheuristics yield different IGD values, the distributions for HV and ϵ are similar.

Figures 2(d)–(f) illustrate the average evolution of HV, IGD, and ϵ as a function of the number of objective function evaluations. Although all algorithms start from the same initial population, their trajectories diverge markedly. *CoMOLS/D* and *MOEA/D* achieve high-quality HV and IGD values within the first 2×10^5 evaluations, exhibiting slower improvement thereafter. In contrast, *NSGA-II* progresses more slowly throughout and fails to obtain better HV and IGD values up to 1×10^6 evaluations. For ϵ, all three algorithms converge rapidly to similar values by 2×10^5 evaluations, with gradual refinement up to 1×10^6.

B. Meta-features analysis

In random walks, meta-features exhibit distinct sensitivities to L and r. Features in the `sd` and `sk` groups are primarily influenced by r, due to its effect on neighborhood diversity. Because kurtosis measures the frequency of extreme values and tail heaviness, kr-based features depend mainly on L. The r1 group depends on both L and r, because the combination of these parameters impacts the local smoothness or ruggedness of the landscape. The avg group presents irregular and non-monotonic trends, indicating that mean dominance levels are less structurally tied to either parameter.

Figures 3(a) and 3(b) illustrate two representative meta-features, inc.sd and inf.kr, plotting their *GAP* values (y-axis) against the number of objective function evaluations (x-axis) for all (L, r) configurations. Each line color corresponds to an L value, with geometric markers denoting r values. The number of evaluations grows with both L and r, since higher values expand the solution sample. For inc.sd, increasing r reduces the *GAP*, with L exerting minimal influence. Conversely, inf.kr exhibits the opposite pattern: larger L reduces the *GAP*, whereas r has a limited impact.



(a) *GAP* for inc.sd (b) *GAP* for inf.kr

(c) Correlation for inc.sd (d) Correlation for inf.kr

Fig. 3. *GAP* and Spearman correlation for random walk

Further, Figure 3(c) shows that the Spearman correlation between inc.sd and τ decreases as both r and L increase, indicating a clear trend of anti-correlation between the number of turbines and inc.sd. In contrast, Figure 3(d) shows that this pattern does not extend to inf.kr, where Spearman correlations are irregular and exhibit no monotonic relationship with either parameter.

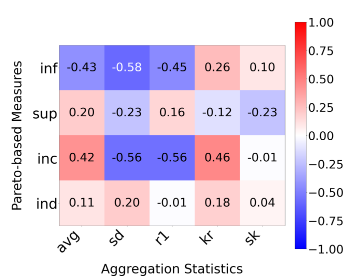


Fig. 4. Average Spearman correlations for adaptive walk

Figure 4 reports the average Spearman correlation between τ and the meta-features computed by adaptive walks. The color indicates the Spearman correlation for the meta-feature identified by the Pareto-based measure (rows) and aggregation statistics (columns). Feature inf.avg is negatively correlated with τ, whereas sup.avg, inc.avg, and ind.avg are positively correlated. This indicates that increasing τ reduces the average number of dominated neighbors while increasing the proportion of dominating (sup) and non-dominated ones (inc and ind). For inf and inc, larger values of τ also lead to lower r1 and higher kr, suggesting that instance size increases local ruggedness and amplifies tail heaviness and extreme-value behavior in the sampled distributions. However, the negative correlation between τ and $\star$.sd, $\star \in \{\texttt{inf}, \texttt{sup}, \texttt{inc}\}$ suggests that the adaptive walk struggles to capture such variability.

C. *Meta-learning results*

Table I presents the merit m^* across all sampling configurations and quality indicators, with gray cells indicating $m^* < 1$ and bold entries denoting the best m^* for random and adaptive walks. Overall, 47.78% of configurations achieved $m^* < 1$, demonstrating that nearly half of the sampling settings enable the meta-learner to outperform the *SBS*. Across indicators, 56.66% of the HV configurations yielded $m^* < 1$, compared to 40% for IGD and 46.66% for ϵ. These differences reflect how each indicator measures solution quality. HV assesses both convergence and diversity, which aligns well with the landscape structural information captured by Pareto-based meta-features. Conversely, IGD focuses on diversity, and ϵ on additive dominance relations.

TABLE I
AVERAGE MERIT m^* FOR DIFFERENT SAMPLING CONFIGURATIONS

r	Random Walk L = 100	L = 50	L = 25	L = 10	L = 5	Adaptive Walk
(a) HV						
1.00	0.85	0.74	0.99	0.94	0.97	0.99
0.50	0.98	1.02	1.06	0.96	1.08	**0.78**
0.25	**0.58**	0.99	1.03	1.13	1.10	0.94
0.10	0.87	0.98	0.99	1.07	1.11	0.86
0.05	0.72	1.01	1.02	1.09	1.10	1.08
(b) IGD						
1.00	0.94	0.93	1.01	0.97	1.07	0.96
0.50	1.03	1.09	0.97	1.03	1.10	**0.91**
0.25	**0.86**	1.05	1.00	0.99	1.05	0.99
0.10	0.94	1.07	1.06	1.03	1.06	0.96
0.05	0.98	1.06	1.05	1.07	1.08	1.05
(c) ϵ						
1.00	0.86	0.86	0.91	0.97	1.05	0.85
0.50	1.13	0.90	0.99	0.99	1.06	**0.80**
0.25	**0.78**	1.11	0.98	1.06	1.08	1.06
0.10	0.89	1.05	1.09	1.08	1.03	0.88
0.05	0.89	1.04	1.04	1.08	1.05	1.07

For all three indicators, the best m^* was achieved with random walks using $L = 100$ and $r = 0.25$. Merit improves with increasing L, as longer walks enhance landscape characterization through diversified sampling. Notably, even low neighborhood rates ($r = 0.05$ and $r = 0.10$) yield competitive m^* values when combined with $L = 100$. These results demonstrate that balancing diversification (via L) and intensification (via r) without exhaustive neighborhood exploitation leads to high-quality m^* values through random walks. Regression metrics confirm high prediction accuracy at $L = 100$. MSE achieved 3.5×10^{-6}, RMSE 1.9×10^{-3}, MAE 3.9×10^{-4}, MAPE 4.6%, and R^2 ranged from 0.84 to 0.99 across HV, IGD, and ϵ indicators.

Adaptive walks exhibited the best m^* for all indicators at $r = 0.50$, revealing that exploring 50% of the neighborhood is sufficient for accurate metaheuristic selection. Smaller r values yielded poorer m^* results due to limited access to dominating neighbors, which restricts the intensification capability of adaptive walks. The accuracy at $r = 0.50$ was: MSE $\leq 1.0 \times 10^{-5}$, RMSE $\leq 3.2 \times 10^{-3}$, MAE $\leq 5.8 \times 10^{-4}$, MAPE $\leq 8.5\%$, and R^2 between 0.92 and 0.99.

Figure 5 illustrates the normalized meta-feature importance for predicting HV under the best random ($L = 100$, $r = 0.25$) and adaptive ($r = 0.50$) walk configurations. The meta-learner utilizes all four feature groups (`inf`, `sup`, `inc`, `ind`) from random walks, while focusing primarily on `inf` and `inc` groups from adaptive walks. These results reflect the difference between unbiased exploratory sampling in random walks and the dominance-driven behavior of adaptive walks. The latter emphasizes meta-features associated with local improvements.

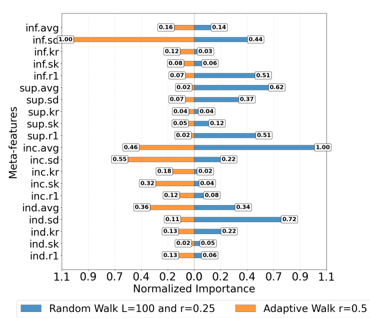


Fig. 5. Meta-feature importance for predicting the HV indicator

VI. Conclusion

This paper proposed a meta-learning approach for algorithm selection in *MoWFLOP*, a problem with applications in wind farm design. By integrating fitness landscape meta-features with Random Forest regression, the framework predicts the performance of multi-objective metaheuristics and selects the most suitable one for unseen instances. Using 516 instances and 90 sampling configurations, the study provided a comprehensive assessment of sampling strategies, landscape characterization, and predictive performance.

The results showed that the parameters have a strong influence on meta-feature values, computational cost, and meta-learning accuracy. The walk length and the neighborhood size promoted an effective balance between diversification and intensification. The proposed pipeline consistently outperformed the *SBS* and achieved high regression accuracy, confirming its robustness and practical relevance for supporting wind farm design.

Future work includes exploring additional meta-feature types (e.g., decomposition-based or indicator-based) and extending the framework to other renewable energy design problems. Further research may also incorporate the proposed meta-learning pipeline into multi-agent metaheuristics or hyper-heuristic architectures, enabling adaptive control of search components and online algorithm selection.

Acknowledgements

This work was supported by FAPESB through the PIBIC, MOVE, and INCITE programs; CAPES; CNPq (grant 402842/2023-5); and Rede BAH.IAna/UFBA/Kunumi. Experiments were supported by CMCAD/UFBA. The authors used LLM to improve the clarity and readability of all sections. Content was created and reviewed by the authors.

References

[1] Global Wind Energy Council (GWEC), *Global Wind Report 2025*. GWEC, Brussels, 2025.

[2] M. S. Adaramola and P.-Å. Krogstad, “Experimental investigation of wake effects on wind turbine performance,” *Renewable Energy*, vol. 36, no. 8, pp. 2078–2086, Aug. 2011.

[3] S. Rodrigues, P. Bauer, and P. A. N. Bosman, “Multi-objective optimization of wind farm layouts–Complexity, constraint handling and scalability,” *Renewable and Sustainable Energy Reviews*, vol. 65, pp. 587–609, 2016.

[4] D. Cazzaro and D. Pisinger, “Variable neighborhood search for large offshore wind farm layout optimization,” *Computers & Operations Research*, vol. 138, p. 105588, 2022.

[5] G. J. N. Silva, J. G. L. S. G. Silva, and I. F. C. Fernandes, “On the Use of Pareto-Based Features in Meta-learning for Multi-objective Offshore Wind Farm Layout Optimization,” in *Brazilian Conference on Intelligent Systems*, Cham, Switzerland: Springer Nature Switzerland, 2025.

[6] Y. Zheng, et al., “MORSA: Multi-objective reptile search algorithm based on elite non-dominated sorting and grid indexing mechanism for wind farm layout optimization problem,” *Energy*, p. 130771, 2024.

[7] R. Cosson, B. Derbel, A. Liefooghe, H. Aguirre, K. Tanaka, and Q. Zhang, “Decomposition-based multi-objective landscape features and automated algorithm selection,” in *Proceedings of the 21st Evolutionary Computation in Combinatorial Optimization (EvoCOP 2021)*, Virtual Event, Apr. 7–9, 2021, Springer, 2021, pp. 34–50.

[8] A. Liefooghe, F. Daolio, S. Verel, B. Derbel, H. Aguirre, and K. Tanaka, “Landscape-aware performance prediction for evolutionary multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 6, pp. 1063–1077, 2019.

[9] P. Brazdil, J. N. V. Rijn, C. Soares, and J. Vanschoren, “Metalearning: Applications to Automated Machine Learning and Data Mining”. *Springer Nature*, 2022.

[10] J. D. Knowles, L. Thiele, and E. Zitzler, “A tutorial on the performance assessment of stochastic multiobjective optimizers”. *TIK Report*, vol. 214, 2006.

[11] R. Cosson, B. Derbel, A. Liefooghe, S. Verel, H. Aguirre, Q. Zhang, and K. Tanaka, “Cost-vs-accuracy of sampling in multi-objective combinatorial exploratory landscape analysis,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2022, pp. 493–501.

[12] X. Cai, M. Hu, D. Gong, Y. Guo, Y. Zhang, Z. Fan, and Y. Huang, “A decomposition-based coevolutionary multiobjective local search for combinatorial multiobjective optimization,” *Swarm and Evolutionary Computation*, vol. 49, pp. 178–193, 2019.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[14] Q. Zhang and H. Li, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[15] F. Zou, D. Chen, H. Liu, S. Cao, X. Ji, and Y. Zhang, “A survey of fitness landscape analysis for optimization,” *Neurocomputing*, vol. 503, pp. 129–139, 2022.

[16] M. López-Ibáñez, J. Dubois-Lacoste, L. P. Cáceres, M. Birattari, and T. Stützle, “The irace package: Iterated racing for automatic algorithm configuration,” *Operations Research Perspectives*, vol. 3, pp. 43–58, 2016.