A Fitness Landscape Analysis of Multi-Objective Neural Architecture Search

1st Cosijopii Garcia-Garcia
IDSSI Department
Universidad del Istmo, Campus Ixtepec
Ciudad Ixtepec, 70110, Oaxaca, Mexico
cosijopii@unistmo.edu.mx

2nd Bilel Derbel
Univ. Lille, CNRS, Inria, Centrale Lille,
UMR 9189 CRIStAL,
F-59000 Lille, France
bilel.derbel@univ-lille.fr

***Abstract*—The performance of local search in Multi-Objective Neural Architecture Search (MONAS) is strongly influenced by the landscape topology, which is itself directly induced by the choice of a neighborhood operator. In this paper, we investigate how different neighborhood definitions shape the MONAS landscape and, in turn, affect the empirical performance of Pareto Local Search (PLS). Specifically, we conduct a systematic multi-objective landscape analysis of the NATS-BENCH topology search space. Using the Pareto Local Optima Networks (PLOS-nets) framework, we analyze the landscapes induced by six neighborhood operators. Our results show that "disruptive" operators generate highly fragmented, poorly searchable landscapes, whereas incremental, "small-step" operators produce well-connected landscapes that effectively link Pareto local optima to the global Pareto front. Building on these insights, we examine the performance of two PLS variants and analyze their relative effectiveness as a function of both the underlying neighborhood structure and the exploration strategy employed.**

***Index Terms*—Multi-objective Optimization, NAS, Fitness Landscape Analysis, Pareto Local Optimal Networks, Neighborhood Operator.**

I. INTRODUCTION

Neural Architecture Search (NAS) has emerged as a fundamental technique for automating the design of high-performance deep learning models. A critical sub-field is Multi-objective NAS , which seeks to optimize the inherent trade-offs between several competing objectives, such as maximizing classification accuracy while simultaneously minimizing computational cost (e.g., FLOPs) [1]–[3]. The discrete nature, high evaluation cost, and exponential size of NAS search spaces make the problem computationally challenging.

To navigate these vast spaces, metaheuristics such as evolutionary algorithms and local search are commonly employed. The success of any such algorithm is fundamentally dependent on the underlying "searchability" of the underlying fitness landscape. In the multi-objective context, this landscape can be analyzed through the relationships among Pareto Local Optima (PLO). However, a critical and often-overlooked component is the neighborhood operator, which directly induce the topology of this landscape. In particular, there is a limited understanding of how different genotypic neighborhood definitions systematically systematically shape the NAS landscape and, in turn, the empirical performance of search algorithms.

Indeed, while some studies have explored the fitness landscape of NAS, a fundamental understanding of what makes the problem difficult remains incomplete. Foundational work in this area has begun to characterize these discrete landscapes using Local Optima Networks (LONs) for single-objective problems, analyzing both the Topology Space [4], [5] and the Size Space (channel configuration) [6]. Furthermore, some work decided to move to continuous representations or real-valued relaxations of the search space [3], [7], [8]. In contrast, the topological properties of the underlying discrete genotypic spaces, particularly in the multi-objective context which this paper investigates, are far less understood. A primary barrier to this understanding is the prohibitive computational cost of evaluating architectures. Analyzing these landscapes in depth has only become feasible with the introduction of tabular benchmarks, such as the NATS-BENCH, which provide pre-computed data for the entire search space. This paper aims to bridge this gap through a systematic analysis of the multi-objective landscape. We utilize the NATS-BENCH topology search space [9], focusing on the bi-objective trade-off between classification error and model complexity. We define six distinct neighborhood operators (N1-N6). Using the PLOS-nets framework [10], we construct the complete landscape graph for each neighborhood. We then analyze the structural properties of these graphs, including their connectivity, fragmentation, path lengths, and assortativity. Our findings demonstrate that the neighborhood definition is one of the most critical factor determining landscape searchability. We show that "disruptive" operators create highly fragmented, problematic landscapes that are effectively unsearchable. Conversely, incremental "small-step" operators generate well-connected graphs that link the vast majority of local optima to the global Pareto front. Finally, we correlate these structural findings with the empirical performance of the well-established Pareto Local Search (PLS) algorithm [11]. The results show that the landscape topology is highly predictive: the most efficient landscape (N1) requires the fewest function evaluations, while the most robust landscapes (N2, N3) are more computationally expensive but converge with the highest confidence. In the remainder, Section II provides some background. Section III details the multi-objective landscape formulation. Section IV reports our findings. Finally, Section V concludes the paper.

II. Background

A. Multi-Objective Optimization

A Multiobjective optimization Problem (MOP) involves the simultaneous optimization (minimization or maximization) of multiple, often conflicting, objective functions: $f_m(\mathbf{x}), m = 1, 2, \ldots, M$, where $\mathbf{x} \in \mathbb{R}^m$ is a n decision variables vector which must satisfy all constraints and variable bounds.

Consequently, there is typically no single, superior solution; instead, the goal is to identify a set of optimal trade-offs, known as Pareto-optimal solutions. The definition of this set relies on the concept of dominance [12]. For a minimization problem with m objectives, a solution vector $\mathbf{u} = (u_1, u_2, \ldots, u_k)$ is said to dominate another vector $\mathbf{v} = (v_1, v_2, \ldots, v_k)$ (denoted $\mathbf{u} \prec \mathbf{v}$) if and only if $\mathbf{u}$ is no worse than $\mathbf{v}$ in all objectives and is strictly better in at least one objective: $\forall i \in \{1, \ldots, k\}, u_i \leq v_i$, and $\exists j \in \{1, \ldots, k\}$ such that $u_j < v_j$ The Pareto set, $X^* \subseteq \mathbf{X}$, is the set of all Pareto-optimal solutions (i.e., solutions not dominated by any other solution in the feasible search space), formally defined as $X^* = \{\mathbf{x}^* \in \mathbf{X} \mid \nexists \mathbf{x} \in \mathbf{X}$ such that $f(\mathbf{x}) \prec f(\mathbf{x}^*)\}$. The image of this set in the objective space, $\mathbf{Z}^* = f(X^*)$, is the Pareto front [12].

B. Multi-objective Neural Architecture Search

The automatic design of CNN architectures has traditionally been performed by human experts with a single-objective focus, namely, maximizing model accuracy. However, metrics beyond accuracy, such as model complexity or the number of learnable parameters, play a crucial role when designing a CNN architecture. NAS refers to the automated process of discovering high-performing architectures. It is a notoriously difficult problem [13]. The foundational work, for example, required thousands of GPU-days to find a single, high-accuracy architecture [14]. This difficulty stems from the discrete, non-convex, and exceptionally large search spaces, combined with the high computational cost of evaluating even one candidate (i.e., full model training). When multiple, conflicting objectives are introduced such as minimizing latency, FLOPs, or memory usage, the underlying complexity increases significantly. The goal is no longer to find a single optimum, but an entire Pareto front of trade-off solutions [1]–[3], [15]. In this paper we focus on the MONAS problem defined as: Minimize $\mathbf{F}(\mathbf{x}) = \big(f_1(\mathbf{x}; w^*(\mathbf{x})),\ f_2(\mathbf{x})\big)^T$ subject to: $w^*(\mathbf{x}) \in \operatorname{argmin} \mathcal{L}(w; \mathbf{x})$. Where f_1 is the classification error and f_2 is the complexity of the CNN. The term f_1 is evaluated on a validation or test dataset, given the optimal weights $w^*(\mathbf{x})$ found by solving the inner optimization problem $\mathcal{L}(w; \mathbf{x})$. These weights are trained using an optimization algorithm, such as Stochastic Gradient Descent [1], [16]. We consider the topological search space from NAST-Bench [9], which is inspired by the success of cell-based representations in NAS.

As shown in Figure 1, the macro-skeleton used in NAST-Bench is composed of 8 sequential components. This starts with a `3x3 conv` then includes 3 main cell-based stages, where each stage repeats a cell $N = 5$ times. These stages are interspersed with residual blocks (stride 2) that act as feature map reducers. The architecture concludes with a global average pooling layer, which flattens the feature map to be used by a fully-connected layer with softmax for the final classification [9]. This relies on a stack of N cells. The internal structure of these cells is *not* fixed; rather, it is sampled from the topological search space, as illustrated in Figure 2. This search space is defined as a directed acyclic graph (DAG) consisting of four nodes. Each edge in this graph represents an operation selected from a predefined, discrete set. This operation set comprises five candidates: `zeroize`, `skip-connect`, `1x1 conv`, `3x3 conv`, and `3x3 avg pool`. The right side of Figure 2 depicts a single candidate cell architecture, which is an instance of this search space where specific operations (represented by the colored edges) have been selected. This cell's DAG is defined by 6 edges, and each edge is selected from the set of 5 predefined operations. This results in a discrete search space of size $5^6 = 15,625$. Every architecture is trained on three different datasets: CIFAR-10 [17], CIFAR-100 [17], and ImageNet-16-120 [18].

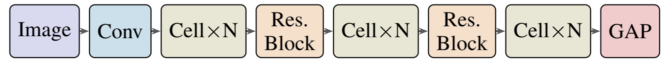

Fig. 1: Macro-skeleton of the NAST-Bench architecture.

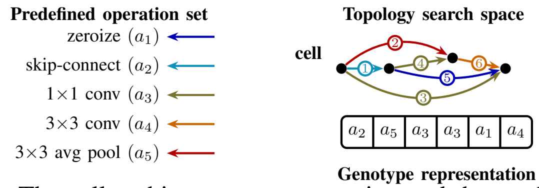

Fig. 2: The cell architecture representation and the predefined operation set. The architecture is a DAG with four nodes and six edges. Each edge is assigned an operation from the set ($a_1 \ldots a_5$), allowing for a genotypic representation [9].

III. Multi-Objective Landscape Analysis of NAS

Given the complexity of MONAS, we study the difficulty of its search space through a systematic landscape analysis.

A. Landscape definition

A Multi-objective Landscape is a triplet $(\mathbf{X}, \mathcal{N}, f)$ such that $\mathbf{X}$ is the solution space, $\mathcal{N} : \mathbf{X} \mapsto 2^{\mathbf{X}}$ is a neighborhood relation, and $f : \mathbf{X} \to \mathbf{Z}$ is the vector of objective functions, and $\mathbf{Z}$ is the objective space. In our work, f is composed of two objective functions: f_1, minimizing classification error, and f_2, minimizing model complexity (measured in FLOPs). The search space $\mathbf{X}$ is the one defined in Figure 2. It consists of a directed acyclic graph (DAG) with 4 nodes and 6 edges, which can be represented by an integer vector of length 6; each position corresponding to an operation from the predefined set.

While $\mathbf{X}$ and f are defined by the benchmark, the landscape is not fully specified until the neighborhood structure $\mathcal{N}$ is chosen. Different neighborhoods induce different landscapes, so we focus on analyzing the impact of the neighborhood on the resulting landscape. We therefore consider the following six neighborhood structures.

a) N1 (Unidirectional Increment): The neighborhood N_1 defines neighbors of a solution $s \in \mathcal{S}$ by incrementing a single component by one unit, with toroidal wrapping at the upper bound. It is a unidirectional neighborhood with minimal Hamming distance.

$$N_1(s) = \{s' \in \mathcal{S} \mid H(s, s') = 1 \wedge \\ \exists! i \in [1, d] : s'_i = (s_i + 1) \bmod k\} \tag{1}$$

where H denotes the Hamming distance, and each dimension i is wrapped cyclically within its discrete range $[l_i, u_i]$.

b) N2 (Bidirectional Increment): The neighborhood N_2 extends N_1 by allowing both $+1$ and -1 perturbations on a single component under toroidal wrapping.

$$N_2(s) = \{s' \in \mathcal{S} \mid H(s, s') = 1 \wedge \\ \exists! i \in [1, d] : s'_i = (s_i \pm 1) \bmod k\} \tag{2}$$

c) N3 (Hamming-2 Increment): The neighborhood N_3 generates all solutions that differ from the current vector s in exactly two components, where each selected position is increased by one unit under toroidal boundary conditions.

$$N_3(s) = \Big\{ s' \in \mathcal{S} \ \Big| \ H(s, s') = 2 \ \wedge \exists I_2 \subset [1, d], \\ |I_2| = 2 : \forall i \in I_2, s'_i = (s_i + 1) \bmod k_i \Big\} \tag{3}$$

d) N4 (Pairwise Swap): This operator defines neighbors through pairwise exchanges. Each neighbor s' is generated by selecting two distinct positions $i, j \in [1, d]$ and swapping their contents. This preserves the multiset of component values while altering their order, as commonly used in permutation-based domains.

$$N_4(s) = \{ \, s' \in \mathcal{S} \mid \exists i < j : s' = \text{swap}(s, i, j) \, \} \tag{4}$$

e) N5 (Insertion): This operator generates neighbors by removing an element at position i and inserting it at position j. This modifies multiple positional dependencies, producing more asymmetric and directed transitions in the search space.

$$N_5(s) = \{ \, s' \in \mathcal{S} \mid \exists i \neq j : s' = \text{insert}(s, i, j) \, \} \tag{5}$$

f) N6 (Hybrid Increment–Swap): N_6 combines the local refinement of N_1 with the reordering capability of N_4.

$$N_6(s) = N_1(s) \cup N_4(s) \tag{6}$$

B. Pareto Local Optimal Networks

To study the structure of a given landscape $(\mathbf{X}, \mathcal{N}, f)$, we rely on formal analysis tools. In single-objective optimization, the distribution and connectivity of local optima are known to define a landscape's searchability, a concept effectively captured by Local Optimal Networks (LONs) [19]. This concept extends to the multi-objective domain, where the relationships and connectivity between Pareto local optima are critical. The Pareto Local Optimal Networks (PLOS-net) model was introduced to capture this structure, allowing the landscape to be analyzed as a complex network [10].

A PLOS-net is defined as an unweighted, undirected graph $G = (V, E)$. The set of nodes $V \subseteq \mathbf{X}$ consists of all Pareto Local Optimal solutions (PLOS). A solution $x \in \mathbf{X}$ is defined as PLOS if it is not dominated by any solution within its neighborhood: $\forall x^* \in \mathcal{N}(x), \neg(x^* \prec x)$. The set E contains the edges connecting these local optima. An edge $e_{ij} \in E$ exists from a PLOS solution $x^i \in V$ to another PLOS solution $x^j \in V$ if x^j is in the neighborhood of x^i (i.e., $x^j \in \mathcal{N}(x^i)$) or x^i is in the neighborhood of x^j (i.e., $x^i \in \mathcal{N}(x^j)$), see Definition 2 in [10]. Once the PLOS-net G is constructed, we can analyze it using a set of network metrics [10], so to quantify the landscape's connectivity, fragmentation, and internal structure, which are key to understanding its searchability. Table I summarizes the considered metrics.

IV. Experimental results

A. Setup

NATS-Bench is a benchmark comprising two search spaces: the Topology Search Space, which contains $5^6 = 15,625$ neural architectures, and the Size Search Space, which focuses on channel number configurations and contains $8^5 = 32,768$ architectures. We focus exclusively on the Topology Search Space. The architectures in NATS-Bench were trained on three datasets: CIFAR-10 [17], CIFAR-100 [17], and ImageNet-16-120 [18]. Performance metrics were recorded at 12, 90, and 200 epochs. This entire training process was executed three times using three different random seeds, with results captured for the training, validation, and test sets.

We consider two conflicting objectives: the classification error and the computational complexity. The latter is quantified using FLOPs (Floating Point Operations). Following previous works [4], [6], [9], we use the average of these metrics (across the three random seeds) for each neural architecture.

Our methodology is as follows. First, the PLOS-nets were constructed. Given that we use the NATS-Bench benchmark with small space, the PLOS-net $G = (V, E)$ for each neighborhood $\mathcal{N}$ was generated by full enumeration. Table II provides a summary of these neighborhoods. Subsequently, these obtained graphs are analyzed using the standard structural features summarized in Table I to gain a deeper understanding of the differences induced by the underlying neighborhoods.

Finally, to validate our structural analysis, we correlate the landscape metrics with the empirical performance of a search algorithm to evaluate the practical impact of these neighborhoods. We considered two variants of Pareto Local Search (PLS) [11] as a well-established multi-objective baseline solver for this purpose. This experiment, involving 100 independent runs for each neighborhood and dataset, allows us to study how the structural properties of the landscapes (e.g., connectivity, path length) can correlate with performance.

B. Analysis of the PLOS-net features

The PLOS-nets as described in the Section IV-A, are visualized in Figure 3, where nodes are positioned according to their classification error (X-axis) and computational complexity (Y-axis), the latter reported in millions of FLOPs. It is important to note that some edges may appear strictly horizontal or vertical. This is a visual artifact stemming from the scale and resolution of the objective space; small discrepancies in either error rates or complexity values become indistinguishable at this plotting scale. Furthermore, although some

TABLE I: PLOS-net structural metrics [10].

Metric	Description
`Node prop`	Proportion of solutions in $\mathbf{X}$ that are PLOs.
`Degree avg`	Average node out-degree in the graph G.
`comp prop`	Number of connected components (proportional to node count).
`comp max size`	Size of the largest connected component.
`path length`	Average path length between all reachable node pairs.
`path pos ext`	Proportion of nodes with a directed path to a global POS.
`path len pos`	Average path length from a node to its nearest global POS.
`assort deg`	Assortativity by degree (i.e., connectivity to similar-degree nodes).
`assort pos`	Assortativity by optimality (PLO connecting to PLO vs. POS).
`assort rank`	Assortativity by rank (tendency to connect to similar-rank nodes).

TABLE II: Neighborhood sizes for $d = 6$ and $k = 5$.

Neighborhood	Operation Definition	Calculation	Size
N1	Choose 1, apply $+1$	$\binom{6}{1}$	**6**
N2	Choose 1, apply $+1$ or -1	$\binom{6}{1} \times 2$	**12**
N3	Choose 2, apply $+1$ to both	$\binom{6}{2}$	**15**
N4	Choose 2, swap	$\binom{6}{2}$	**15**
N5	Choose i, insert at j ($i \neq j$)	6×5	**30**
N6	$N_1 \cup N_4$	$6 + 15$	**21**

neighborhood operators (N_1, N_3, N_5, N_6) are inherently non-symmetric where a transition from x to x^* does not imply the reverse, the PLOS-nets [10] are defined as undirected graphs to reveal the topological backbone of the landscape. This allows to study the structural connectivity and reachability between local optima independently of the search direction.

The structural analysis of the constructed PLOS-net graphs is based on the ten metrics defined in Table I. These characteristics are summarized in the heatmaps shown in Figure 4. The heatmaps illustrate that the choice of neighborhood operator induces fundamentally different landscape topologies, a finding that is highly consistent across all three datasets.

As a general observation, the set of PLO solutions changes substantially depending on the neighborhood. For instance, the `Node prop` metric (proportion of solutions that are PLOs) is consistently high for N1 and N3 (0.7–1.0), but very low for N4, N5, and N6 (0.0–0.2), while N2 shows an intermediate value (0.2–0.4). This confirms that the number of local optima is highly sensitive to the operator. Furthermore, the connectivity of local optima varies significantly not only with the neighborhood definition but also, to a lesser extent, with the dataset. Further insights into the landscape are presented hereafter.

a) Connectivity and Fragmentation: A primary distinction appears in graph connectivity. N1 and N3 imply highly unified landscapes, characterized by a very low `comp prop` and a high `comp max size`, with connected components covering the majority of the nodes. In contrast, N4, N5, and N6 imply highly fragmented landscapes. As seen in Heatmaps, N5 consistently results in the highest `comp prop` ($\approx$ 0.95-1.0), indicating that the PLOS-net has effectively isolated and disconnected nodes. N2 represents an intermediate structure, being moderately fragmented.

b) Relationship to the Pareto Set: The landscapes also differ in their connection to the Pareto Optimal Set (POS). In the N1 and N3 landscapes, the `path pos exist` characteristic is high ($>$ 0.92), showing that most PLO nodes belong to a component connecting to the POS. For N2, this connectivity is partial (0.30-0.49), while for the fragmented N4, N5, and N6 landscapes, this connection is virtually non-existent.

c) Path Structure: The internal topologies of the graphs vary. While N1, N2, and N3 have relatively short `path length`, their paths to the global front differ. N3 exhibits the longest `path length pos`, suggesting a circuitous route to the POS, whereas N1 provides more direct paths.

d) Assortativity: The assortativity describes the internal local organization of the landscapes. N1, N2, and N3 all show high `assort degree` (> 0.64), indicating a non-random, hub-like structure. However, their organization by solution quality (`assort rank`) differs. N1 and N2 show moderate rank assortativity (0.42-0.56), implying high-quality nodes tend to connect to other high-quality nodes. N3 shows the opposite, with a very low `assort rank` (0.0-0.14), suggesting high-quality solutions are connected to lower-ranked solutions.

C. Analysis of PLS search performance

To validate the findings from our structural landscape analysis, we now correlate the topological properties with the empirical performance of a search algorithm. We use Pareto Local Search (PLS) for this purpose, as its behavior is fundamentally guided by the neighborhood structure [11]. However, the specific strategy used to exploit this neighborhood is also a critical factor. Therefore, to understand the full impact of the landscape, we analyze two distinct PLS variants (**PLS-1** and **PLS-2**).Both variants are initialized with the same random solution (x_0) and use the same neighborhood structure, but differ in their exploration strategy, as depicted in Algorithm 1. **PLS-1** employs an exhaustive neighborhood exploration. In each step, it evaluates *all* neighbors of a solution and adds all non-dominated, unvisited ones to the archive. **PLS-2** evaluates neighbors only until the first non-dominated, unvisited solution is found. This single solution is added to the archive, and the neighborhood search is immediately aborted. A characteristic of this PLS strategy is its sensitivity to the neighborhood's internal evaluation order, which clearly impacts the search trajectory.

This dual experimental setup allows us to examine how landscape topologies (N1–N6) influence both exhaustive and first-improvement search strategies. Table III reports results from 100 runs. PLS-2 achieved a statistically better mean hypervolume in 12 of 18 cases and was equivalent in the remaining six. In all cases, PLS-2 required significantly fewer average Function Evaluations (FEvals), often substantially (e.g., N2 on CIFAR-10 drops from 305 to 237 FEvals). These results show that the first-improvement strategy can find better solutions while reducing computational cost, whereas exhaustive neighborhood search wastes evaluations without necessarily improving global performance. The second key finding is that the choice of neighborhood operator determines the landscape's searchability, which closely aligns with our previous topological analysis. The 18 test cases can be

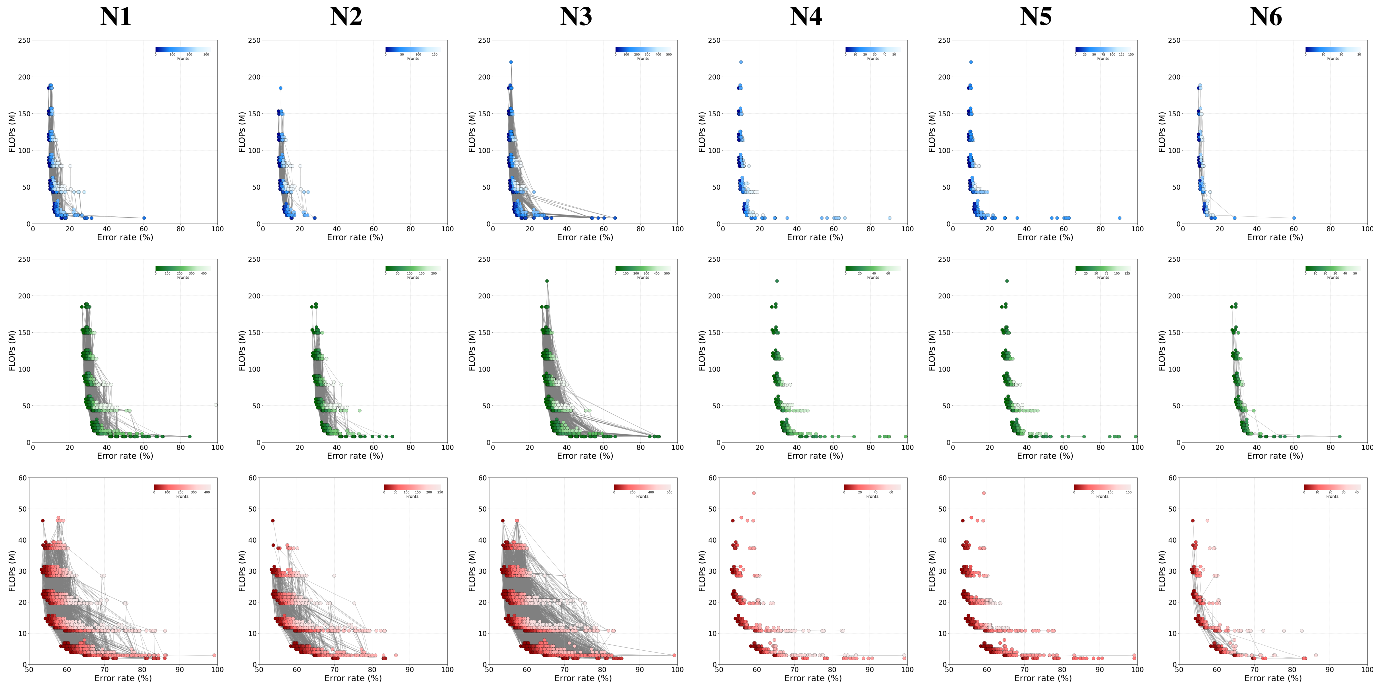

Fig. 3: Landscape analysis results. Columns show neighborhoods N1–N6 and rows show datasets: CIFAR-10 (top), CIFAR-100 (middle), and ImageNet-16-120 (bottom). N1–N3 exhibit strong connectivity across datasets, with N3 showing the highest node proportion, highlighting the effectiveness of small, local-step operators. In contrast, N4 and N5 produce highly fragmented landscapes with many isolated nodes, indicating that large, disruptive operators (`swap` and `insertion`) hinder traversal of the solution space. N6 further illustrates the limitations of operator unions, resulting in a sparse and weakly connected landscape.

Algorithm 1: Pareto Local Search PLS-1 and PLS-2

$A \leftarrow \{x_0\}$; $V \leftarrow \emptyset$; `// Initialize archive with an initial solution` x_0
while $A \setminus V \neq \emptyset$ **do**
  Select a random solution x from $A \setminus V$;
  $V \leftarrow V \cup \{x\}$; `// Mark` x `as visited`
  for *each* $x' \in \mathcal{N}(x)$ **do**
    if $\neg(x \prec x')$ **then** `// If` x `does not dominate` x'
      if $x' \notin V$ **then** `// And` x' `has not been visited`
        $A \leftarrow$ Non-Dominated-Filter$(A \cup \{x'\})$; `// Update Archive`
        if *PLS-1* **then continue**;
        else if *PLS-2* **then break**;
Return A

clearly divided into two groups. In the case of N1, N2, N3, N6 neighborhoods consistently produce high-quality solutions (e.g., Mean HV > 0.98 on CIFAR-10 for N1, N2, N3, N6; > 0.93 on CIFAR-100). This is attributed to the fact that these operators induce connected, searchable landscapes, as identified in our PLOS-net analysis. N1 is the most efficient, consistently requiring the fewest FEvals (120-147). It finds high-quality solutions quickly, correlating with a landscape structure that provides short paths to the global front. N2 and N6 are the best neighborhoods. As marked by the † in Table III, these neighborhoods produce the best-in-dataset solutions (N6 on CIFAR-10, N2 on CIFAR-100 and ImageNet). Specifically, in the case of neighborhood N6, which is the union of N1 and N4, the PLS-2 algorithm first traverses the N1 neighborhood. It only advances to N4 if no admissible solution in N1 is found, which provides the search with added flexibility.

V. Conclusions

This research shows that the searchability of a MONAS landscape is not an intrinsic property of the search space, but is largely determined by the neighborhood operator, making operator design critical in NAS. We show that success depends not on the landscape alone, but on the interaction between the operator-induced topology and the behavior of the search algorithm. Our study highlights that operator efficiency and searchability in NAS is non-trivial and opens new perspectives for algorithm design. It suggests that future work should emphasize the co-design of operators that ensure connectivity. Besides, the PLS-2 strategy demonstrates that greedy, cost-effective algorithms can outperform exhaustive approaches when the landscape is well-structured. Finally, this study raises several open research questions. Our analysis relies on

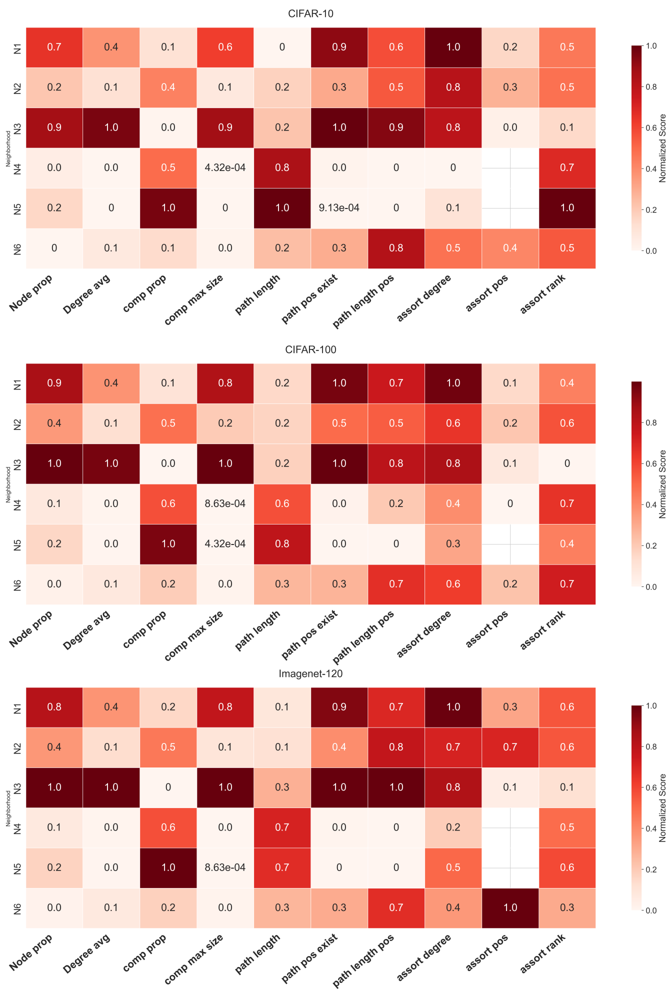

Fig. 4: Heatmaps of PLOS-net structural characteristics for neighborhoods N1–N6 across CIFAR-10 (top), CIFAR-100 (middle), and ImageNet-16-120 (bottom). Color intensity indicates the normalized value of each landscape metric.

an exhaustive benchmark that eliminates evaluation cost and stochasticity. An important next step is to investigate how these landscape properties can be sampled or approximated online, for instance using zero-cost proxies. Another key question concerns the impact of training noise on landscape topology: whether stochasticity smooths the landscape by connecting components or instead introduces more complex local optima?

TABLE III: Hypervolume and average FEvals (100 runs) comparing PLS-1 and PLS-2. Best mean hypervolume is in bold. The last column indicates when PLS-2 is statistically better (+), worse (-), or equivalent (=) to PLS-1 (Mann-Whitney, $p < 0.05$). A † marks the best configuration.

		PLS-1 (Exh.)		PLS-2 (First)		
Dataset	N.	Mean (Std)	FE.	Mean (Std)	FE.	St.
CIFAR-10	N1	0.9805 (0.0090)	147	**0.9816** (0.0093)	120	=
	N2	**0.9862** (0.0049)	305	0.9861 (0.0036)	237	=
	N3	0.9853 (0.0055)	362	**0.9859** (0.0039)	344	=
	N4	0.7330 (0.1532)	48	**0.8359** (0.1085)	34	+
	N5	0.7419 (0.1461)	76	**0.8228** (0.0104)	50	+
	N6	0.9797 (0.0118)	426	**0.9866** (0.0032)†	401	+
CIFAR-100	N1	0.9307 (0.0225)	131	**0.9389** (0.0142)	112	+
	N2	**0.9582** (0.0114)†	357	0.9576 (0.0100)	305	=
	N3	**0.9554** (0.0093)	411	0.9548 (0.0107)	345	=
	N4	0.6729 (0.1653)	49	**0.7703** (0.1161)	35	+
	N5	0.6625 (0.1598)	79	**0.7860** (0.0992)	50	+
	N6	0.9399 (0.0184)	485	**0.9557** (0.0099)	475	+
ImageNet16-120	N1	0.9005 (0.0587)	144	**0.9151** (0.0297)	120	+
	N2	0.9424 (0.0180)	376	**0.9466** (0.0169)†	359	+
	N3	0.9351 (0.0121)	400	**0.9360** (0.0106)	351	=
	N4	0.6224 (0.1345)	47	**0.7016** (0.0965)	35	+
	N5	0.6054 (0.1259)	80	**0.7290** (0.0898)	52	+
	N6	0.9138 (0.0380)	587	**0.9315** (0.0187)	479	+
+/−/=						**12 / 0 / 6**

References

[1] Z. Lu, I. Whalen, V. Boddeti, Y. Dhebar, K. Deb, E. Goodman, and W. Banzhaf, "NSGA-Net," in *GECCO*, pp. 419–427, 2019.
[2] Z. Lu, K. Deb, E. Goodman, W. Banzhaf, and V. N. Boddeti, "NSGANetV2: Evolutionary Multi-objective Surrogate-Assisted Neural Architecture Search," pp. 35–51, 2020.
[3] C. Garcia-Garcia, A. Morales-Reyes, and H. J. Escalante, "Continuous cartesian genetic programming based representation for multi-objective neural architecture search," *Applied Soft Computing*, vol. 147, 2023.
[4] G. Ochoa and N. Veerapen, "Neural architecture search: A visual analysis," in *PPSN XVII*, pp. 603–615, 2022.
[5] Q. M. Phan and N. H. Luong, "Pareto local search is competitive with evolutionary algorithms for multi-objective neural architecture search," in *GECCO*, p. 348–356, 2023.
[6] S. L. Thomson, G. Ochoa, N. Veerapen, and K. Michalak, "Channel configuration for neural architecture: Insights from the search space," in *GECCO*, p. 1267–1275, 2023.
[7] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," 2019.
[8] C. Garcia-Garcia, H. J. Escalante, and A. Morales-Reyes, "CGP-NAS," in *GECCO Companion*, pp. 643–646, 2022.
[9] X. Dong, L. Liu, K. Musial, and B. Gabrys, "NATS-Bench: Benchmarking nas algorithms for architecture topology and size," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
[10] A. Liefooghe, B. Derbel, S. Verel, M. López-Ibáñez, H. Aguirre, and K. Tanaka, "On pareto local optimal solutions networks," in *PPSN XV*, pp. 232–244, 2018.
[11] L. Paquete, M. Chiarandini, and T. Stützle, "Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study," in *Metaheuristics for multiobjective optimisation*, Springer, 2004.
[12] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2007.
[13] T. Elsken, J. H. Metzen, and F. Hutter, "Neural Architecture Search: A Survey," tech. rep., 2019.
[14] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.
[15] M. Pinos, V. Mrazek, and L. Sekanina, "Evolutionary approximation and neural architecture search," *Genetic Programming and Evolvable Machines*, jun 2022.
[16] C. Garcia-Garcia, B. Derbel, A. Morales-Reyes, and H. J. Escalante, "Speeding up the multi-objective nas through incremental learning," pp. 3–15, 2025.
[17] A. Krizhevsky, "Learning multiple layers of features from tiny images," tech. rep., University of Toronto, 2009.
[18] P. Chrabaszcz, I. Loshchilov, and F. Hutter, "A downsampled variant of imagenet as an alternative to the cifar datasets," *arXiv preprint arXiv:1707.08819*, 2017.
[19] G. Ochoa, M. Tomassini, S. Vérel, and C. Darabos, "A study of nk landscapes' basins and local optima networks," in *GECCO*, p. 555–562, 2008.