Diagnosing Optimizer-Landscape Interaction via Representation in Subset Selection Multitasking: Evidence from Piano Fingering

Ananda Phan Iman
Department of AI Convergence
Gwangju Inst. of Sci. & Tech. (GIST)
South Korea
anandaphan@gm.gist.ac.kr

Tingyang Wei
College of Computing and Data Science
Nanyang Technological University
Singapore
tingyang001@e.ntu.edu.sg

Chang Wook Ahn
Department of AI Convergence
Gwangju Inst. of Sci. & Tech. (GIST)
South Korea
cwan@gist.ac.kr

***Abstract*—Continuous relaxations are commonly used as an encoding scheme for evolutionary multitasking optimization (EMTO), allowing heterogeneous tasks to be optimized within a unified search space. However, these continuous representations fundamentally reshape the induced fitness landscape, and their impact on optimization behavior and knowledge transfer remains underexplored. In this work, we bridge the gap by analyzing the interaction between the representation-induced fitness landscape and optimizer behavior through fitness landscape analysis. Using piano fingering estimation as a representative case of the subset selection problem, we introduce several encoding variants and investigate how their induced landscapes influence search dynamics and solution quality under EMTO. Our analysis indicates that transfer efficacy is strongly associated with changes in representation-induced landscape characteristics, rather than algorithmic factors alone, highlighting the representation design as the key factor of effective knowledge transfer in multitasking optimization.**

***Index Terms*—Art and music, Multi-task optimization, Fitness landscape analysis, Continuous relaxation, Subset selection**

I. Introduction

Evolutionary multitasking (EMTO) [1]–[3] has been considered a powerful paradigm for solving multiple related optimization tasks concurrently by facilitating knowledge transfer through a unified search space. Despite its success in diverse applications [4], the effectiveness of EMTO in the discrete optimization is sensitive to the encoding scheme [1], which remains highly problem-dependent. Specifically, enabling cross-task search requires constructing various continuous search spaces that represent heterogeneous tasks within a unified domain. These efforts span classical combinatorial problems, such as job-shop scheduling problems [5], vehicle routing problems [6], as well as more recent multitask evolutionary machine learning problems [7], [8].

Despite the existing works adopting random-key or specifically designed permutation representations [9], [10] to relax multitask combinatorial optimization, the fundamental impact of such continuous representations on knowledge transfers remains largely unexplored. Since representation directly determines the geometry of the search space, it constrains what an optimizer can exploit [11]. While discrete spaces lack smooth geometry, continuous relaxations provide *soft* solutions with well-defined distances that enable more informative search dynamics. Crucially, different relaxations on the same discrete solution can induce rather different fitness landscapes [12], [13]. Therefore, we are motivated to discuss how these representations affect search and knowledge transfer by investigating their landscape characteristics and how these features relate to multitasking performance.

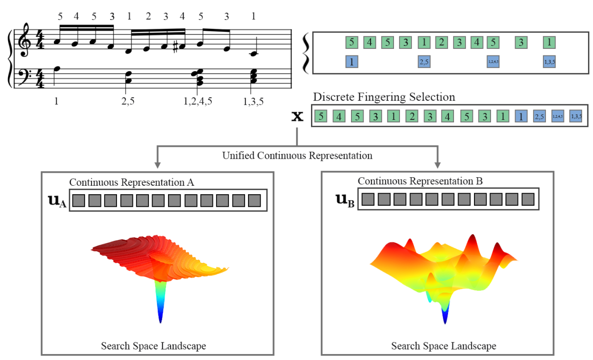


Fig. 1: Illustration of how different unified continuous representations of the same discrete piano fingering selection $\mathbf{x}$ induce distinct search space. Top: A short piano passage with an example of fingering selection. Bottom: Two alternative unified continuous search spaces, within which optimization is performed; color indicates fitness values after decoding.

To empirically investigate these representation-induced effects, we employ piano fingering estimation [14] as a representative instantiation of multitask subset selection [15]. In this problem, the objective is to determine an effective fingering sequence for a musical passage. At each time step t, a subset of k_t fingers is selected from the five available fingers, where k_t is determined by the number of notes played simultaneously. Solving this problem within an EMTO framework thus requires translating the original discrete fingering selections into a unified continuous search space, followed by a decoding process that maps candidate solutions back to valid fingerings.

However, this translation is not unique, as distinct encoding strategies can map the same discrete fingering selections into different continuous representations. Such representation-dependent mappings directly shape the topology and geometry of the resulting search landscape. Consequently, piano fingering provides a suitable environment for diagnosing how different representation-dependent landscapes influence search dynamics and knowledge transfer in evolutionary multitasking.

For illustration, the top part of Fig. 1 shows a short musical passage in Western notation, where green and blue highlight notes played by the right and left hand, respectively. The lower part of the figure illustrates how different continuous representations of the same discrete fingering problem can form distinct landscape geometries. Specifically, the same discrete fingering selection $\mathbf{x}$ admits multiple continuous representations ($\mathbf{u_A}$ and $\mathbf{u_B}$), each defining a different optimization landscape that corresponds to the same underlying discrete solution. As shown in Fig. 1, distinct representation-induced landscapes can possess rather different characteristics, potentially incurring diverse needs for search behavior or knowledge transfer behavior across tasks. Therefore, this illustration highlights the role of continuous relaxation design in shaping the unified space explored by EMTO.

Despite the central role of continuous representation in defining the search landscape and facilitating transfer, to our knowledge, there is no systematic analysis of how different continuous relaxations of subset selection optimization behave under EMTO. Thus, it remains unclear how alternative continuous representations and their associated mapping affect solution quality, search dynamics and transfer performance.

In this work, we address this gap by analyzing how alternative continuous representations reshape the induced search landscape and affect optimizer behavior in subset selection multitasking. Using piano fingering estimation as a representative case, we empirically investigate the interaction between representation-induced search spaces and evolutionary dynamics to provide evidence-based insights into constructing continuous relaxations. While we focus on piano fingering in this paper, our findings offer broader implications for a wide class of subset selection problems, establishing principles for designing representations that facilitate effective search and reliable cross-task knowledge transfer under EMTO.

The relevant source code can be found in the link: https://github.com/anandaphan/pfmt-representation

II. Related Work

A. Solution Representation in Evolutionary Multitasking

Evolutionary Multitasking (EMTO), pioneered by [1], aims to solve multiple tasks simultaneously through the implicit parallelism of evolutionary algorithms, formulated as:

$$\min f_q(\mathbf{x}_q), \quad \text{s.t. } \mathbf{x}_q \in \Omega_q, \quad q \in \{1, \ldots, Q\} \tag{1}$$

where Ω_q represents the decision space and f_q denotes the objective function of the q-th task. While the original paradigm [1] facilitates cross-task search within a unified search space, constructing an effective solution representation via *explicit transfer* [16] has proven crucial for performance. Representation learning for transfer in EMTO has been explored via alignment-based approaches, including subspace alignment [17] and manifold alignment [18].

However, these innovations primarily focus on continuous optimization. When $f_q(\cdot)$ in Eq. (1) is a discrete optimization problem, representation learning becomes non-trivial and highly sensitive to problem-specific contexts [1]. To facilitate implicit parallelism in discrete domains, EMTO traditionally employs random-key encoding [1], [19]. Alternative representations, such as the permutation-based approach proposed by [9], have demonstrated superior performance over random keys in specific domains like routing. Subsequent works have further refined these problem-dependent representations by incorporating domain heuristics [20], [21] and geometric distributions [22].

In this work, we focus on a specific subclass of discrete optimization: subset selection [15]. Formally, the subset selection problem seeks a binary decision vector $\mathbf{x} \in \{0,1\}^n$ that minimizes f under a constraint k:

$$\min_{\mathbf{x} \in \{0,1\}^n} f(\mathbf{x}) \quad \text{s.t.} \quad \sum_{i=1}^{n} x_i \leq k \tag{2}$$

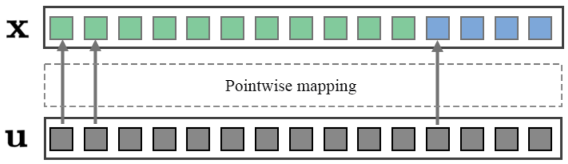
While this formulation is prevalent in multitask feature selection, the impact of solution representation in this context remains underexplored. For instance, some multitask optimizers operate directly on bitstrings [23], while others utilize continuous random keys in $[0, 1]$ decoded via a simple threshold θ [7], [8]. Consequently, unlike continuous or permutation-based optimization, the impact of solution representation in subset selection multitasking remains unclear, particularly regarding how alternative continuous relaxations reshape the induced search landscape and influence transfer performance [12], [13].

B. Solution Representation in Piano Fingering Estimation

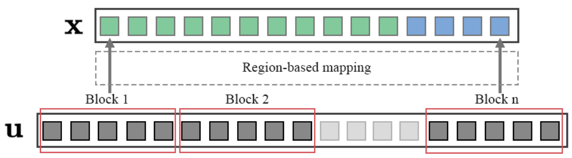
Piano fingering estimation provides an ideal case study for subset selection multitasking. At each time step, it instantiates an exact-k subset selection problem: a specific subset of k fingers must be selected to play k notes simultaneously. Formally, this modifies the subset selection problem defined in Eq. (2) by enforcing a cardinality constraint, $\sum_{i=1}^{n} x_i = k$.

Traditional solutions to this problem include dynamic programming [24], tabu search [25], and variable neighborhood search [26], differential evolution [27], hidden Markov models [28], [29], and reinforcement learning [14], [30], [31]. While these methods operate in discrete fingering spaces, we relax the problem into continuous representations to investigate the resulting search landscapes in our study. This continuous relaxation approach draws inspiration from random-key encodings [19], [32]–[34]. It also relates to latent space optimization [13], [35], [36], where discrete structures are embedded into continuous manifolds. Unlike stochastic methods such as Gumbel-Softmax [37] or Gumbel-Sinkhorn [38]. In contrast, our focus is on deterministic landscapes suitable for evolutionary search.

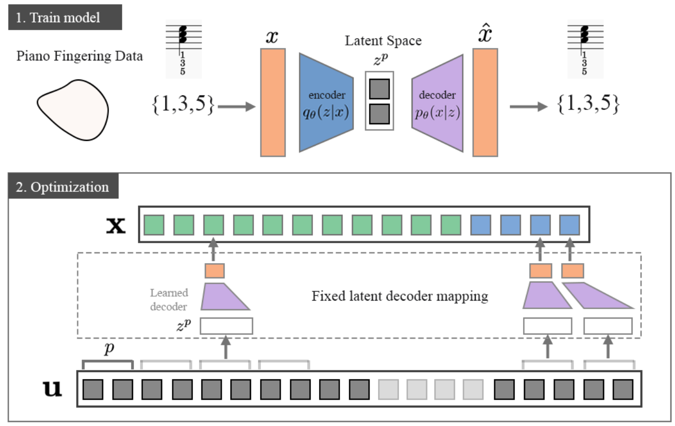
Piano fingering can be solved as multitasking by coupling multiple musical passages altogether. This allows us



(1) R1: Direct (threshold-based)



(2) R2–R3: Modular (block-structured)



(3) R4–R5: Indirect (latent-based)

Fig. 2: Illustration of five representation variants(R1–R5) grouped by their underlying representation mechanisms. (1) R1 (Direct): each continuous variable is deterministically mapped to a discrete fingering selection. (2) R2–R3 (Modular): Fingering selections are controlled by blocks of continuous variables. (3) R4–R5 (Indirect): discrete fingerings are embedded in a low-dimensional latent space learned offline, with a fixed latent-to-discrete mapping during optimization.

to diagnose how representation-dependent landscapes influence search dynamics and cross-task transfer efficiency under EMTO. To our knowledge, no study exists on multitasking music-relevant applications like piano fingering estimation. This absence highlights the novelty of our study as well.

III. Methodology

A. Representation Variants

In this section, we introduce several representation variants for mapping fingering subset selections into unified continuous search spaces. Given a task T_i with total musical time step $d = d_{\text{right}} + d_{\text{left}}$, a discrete fingering individual $\mathbf{x} = [x_1, x_2, \ldots, x_d]$, where the search space for each gene is heterogeneous since each x_j involves k_j simultaneously played notes. Specifically, let $\mathcal{F} = \{1, 2, 3, 4, 5\}$ denote the set of finger indices. Then for the j-th gene with k_j notes, the associated fingering search space $\mathcal{X}_j$ is given by the set of all k_j-subsets of $\mathcal{F}$:

$$\mathcal{X}_{i,j} = [\mathcal{F}]^{k_j} = \{A \subseteq \mathcal{F} \mid |A| = k_j\} \tag{3}$$

Moreover, different continuous representations map $\mathbf{x}$ into an n-dimensional unified search vector $\mathbf{u} \in [0, 1]^n$, where $n = \sum_j dim(u_j)$. Although all representations operate over the same nominal domain, different continuous-discrete mappings induce distinct effective geometries. Accordingly, we treat each representation as a unified design consisting of continuous encoding together with the mapping operator that translates candidate solutions into discrete fingering selections for evaluations. Based on these characteristics, we organize the representations into three types: direct, modular, and indirect.

Each representation variant is defined by a pair of $(\mathcal{U}, g)$ where $\mathcal{U} \in [0, 1]^n$ denotes the unified continuous search space and $g : \mathbf{u} \to \mathcal{X}_i$ is a deterministic mapping that translates continuous candidate solutions into discrete fingering selections for fitness evaluations. Given a candidate $\mathbf{u} \in \mathcal{U}$, the corresponding discrete fingering solution is $\mathbf{x} = g(\mathbf{u})$. The fitness function $f(x)$ is defined using the fingering difficulty level [14].

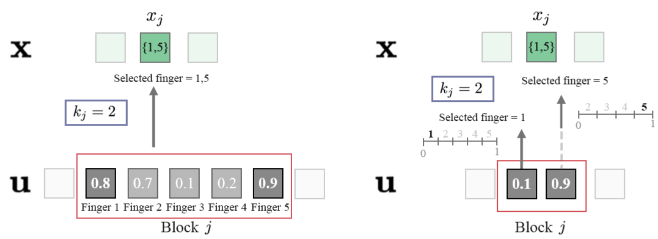
$$f(\mathbf{x}) = \frac{1}{d} \sum_{j=1}^{d} \sum_{l=1}^{L} r_l(x_j) \tag{4}$$

where $r_l(x_j)$ denotes the penalty incurred by the l-th fingering rule when applied to the j-th gene, reflecting the difficulty of executing the corresponding fingering. We adopt L fingering rules defined in [31] throughout this study. Hence, different representations induce distinct fitness landscapes and search behaviors, even when the optimizer is unchanged.

1) Direct Representation (R1): Direct representation employs a *pointwise mapping*, in which each continuous gene u_j corresponds to a single discrete fingering gene x_j, as illustrated in Fig. 2 (1). For the j-th time step requiring k_j fingers, there are $M = \binom{5}{k_j}$ possible finger subsets. These subsets are ordered in a lexicographic order, and the unit interval $[0, 1]$ is partitioned into M contiguous subintervals of equal length, each associated with one subset. A continuous value u_j is then mapped to the fingering subset associated with the unique interval $\left[\frac{m-1}{M}, \frac{m}{M}\right)$ containing u_j.

Due to hard partition boundaries, small perturbations of u_j near these boundaries can induce abrupt changes in the resulting fingering. As a result, this representation exhibits minimal neutrality and high sensitivity, making it suitable as a baseline for analyzing the effects of a pointwise relaxation.

2) Modular Representation (R2–R3): Motivated by the sensitivity of pointwise mappings to hard partition boundaries in R1, we consider modular representations that provide alternative continuous relaxations of the fingering selection problem. Modular representations employ a *region-based mapping*, in which each fingering decision is controlled by a block of continuous variables rather than a single scalar, as illustrated



(a) (b)

Fig. 3: Example of region-based mapping for homogeneous and heterogeneous blocks. (a) homogeneous block with top-k_j truncation mapping, (b) heterogeneous block with sequential selection without replacement.

in Fig. 2 (2). As a result, modular representations naturally induce neutral regions in the fitness space.

In this work, we consider two types of block structures: *homogeneous* and *heterogeneous* blocks, which implement different selection rules illustrated in Fig. 3.

a) Homogeneous blocks (R2): The homogeneous block representation follows a random-key-style encoding [19], where each block $b_j \in [0,1]^5$ contains one value per finger. Given a required cardinality of k_j, mapping is performed at a block level. In R2, a *truncation mapping* [19] selects the indices of top k_j entries in b_j and returns the corresponding fingers subset.

b) Heterogeneous blocks (R3): In contrast, a heterogeneous block $b_j \in [0,1]^{k_j}$ contains one value per simultaneously played note. Mapping is performed by *sequential selection without replacement* over the five fingers. Let $\mathcal{F}_0 = (1,2,3,4,5)$ be the ordered list of available fingers. For $\ell = 1,\ldots,k_j$, $m_\ell = \min\{\lfloor b_{j,\ell}|\mathcal{F}_{\ell-1}|\rfloor,\ |\mathcal{F}_{\ell-1}|-1\}$, select the m_ℓ-th element of $\mathcal{F}_{\ell-1}$ and remove it to obtain $\mathcal{F}_\ell$. The selected fingers are then sorted to form an unordered k_j-subset and thus always k_j distinct fingers.

3) Indirect (Latent) Representation (R4–R5): The indirect representation introduces a low-dimensional continuous encoding of discrete fingering configurations using real fingering data from the PIG Dataset [29]. Each discrete fingering subset x_j is represented by a compact latent vector $\mathbf{z}_j \in \mathbb{R}^p$, enabling search to be conducted in a semantically structured latent space rather than the discrete domain.

Specifically, we employ a variational autoencoder (VAE) [39], where the encoder $q_\phi(\mathbf{z}_j|x_j)$ maps x_j to $\mathbf{z}_j \in \mathbb{R}^p$, and a decoder $p_\theta(x_j|\mathbf{z}_j)$ reconstructs the corresponding fingering. The VAE model is trained offline by minimizing the standard evidence lower bound (ELBO). After training, the encoder is discarded, and the decoder is fixed and used as a mapping during optimization.

To obtain a discrete fingering solution, $\mathbf{u}$ is divided into d segments $\mathbf{u}_j \in [0,1]^p$ which are transformed into latent space via a bijective transformation $h(\cdot)$, and subsequently mapped using the fixed VAE decoder $g_\theta(\cdot)$:

$$\begin{aligned} h &: [0,1]^p \to \mathbb{R}^p, & z_j &= h(u_j) \\ g_\theta &: \mathbb{R}^p \to \mathcal{X}_{i,j}, & x_j &= g_\theta(z_j) \end{aligned} \tag{5}$$

TABLE I: Representation Variant Summary

ID	Type	Mapping
R1	Direct	Partition-based pointwise mapping
R2	Modular (hom.)	top-k_j truncation mapping
R3	Modular (het.)	Sequential selection without replacement
R4	Indirect	Fixed VAE decoder, $p = 2$
R5	Indirect	Fixed VAE decoder, $p = 1$

The indirect representation is illustrated in Fig. 2 (3). We consider two latent dimensionalities, $p = 2$ (R4) and $p = 1$ (R5), to examine the effect of latent compression in the induced search landscape. By construction, nearby latent points tend to decode to identical or similar fingering selections, inducing a landscape informed by real pianistic data.

Together, the three types of representations span a spectrum from one-to-one deterministic to data-driven, each inducing a distinct continuous search space and interacting differently with evolutionary search and knowledge transfer under EMTO.

B. Fitness Landscape Analysis

To characterize how continuous representations reshape the search space explored by EMTO, we analyze the representation-induced landscapes using random walks. Although a wide range of landscape measures exists [40], we focus on ruggedness and neutrality as they directly capture fitness variation and invariance under local perturbations, and are widely used in fitness landscape analysis (FLA) [41]–[43].

For each representation with unified dimension n, we perform random walks on $[0,1]^n$ using normalized Gaussian steps, $x_{t+1} = \mathcal{B}(x_t + \eta z_t/\|z_t\|_2)$ with $z_t \sim \mathcal{N}(0, I)$, step size η, and reflection bounds. Each visited point is decoded into a fingering solution via $g(\cdot)$ and evaluated to obtain the fitness trace $F_t = f(g(x_t))$.

a) Ruggedness: Ruggedness reflects how rapidly fitness changes under small perturbations [40]. We quantify ruggedness via random walk autocorrelation of $\{F_t\}$, using the lag-1 autocorrelation $\rho(1)$ and autocorrelation length τ [41], [44]. Higher $\rho(1)$ indicates smoother landscapes.

b) Neutrality: Neutrality captures plateau-like behavior where local perturbations do not change the mapped objective [40], [42]. We compute neutrality ν by measuring ε-neutrality, $\nu_\varepsilon = \frac{1}{W}\sum_{t=0}^{W-1} \mathbb{I}(|F_{t+1} - F_t| \le \varepsilon)$, where W is the total visited point. We set ε per representation as $\varepsilon = \gamma(\max \Delta F - \min \Delta F)$ with $\gamma = 0.001$, where $\Delta F = F_{t+1} - F_t$.

Together, ruggedness and neutrality provide a principled basis for analyzing the fitness landscapes induced by each representation.

IV. Experimental Study

A. Experiment Setting

We construct eight task pairs from PIG dataset [29] with controlled musical relatedness. Relatedness is defined by key-signature agreement and tonal relationship (same key, relative key, different keys, or unrelated), inducing different levels of structural overlap between tasks (complete/partial/none). Cases

1-4 use major keys and cases 5-8 use minor keys, with cases 3 and 7 involving relative key pairings. Thus, for each task pair $T = \{T_1, T_2\}$, we set population size per task to $N = 50$ and 10^5 evaluations per task, with fitness computed using Eq. (4).

To assess representation effects across optimizers, we evaluate the recent EMTO algorithms spanning the three canonical operator families: GA, DE, and ES. Specifically, *GA-based* methods include AT-MFEA [45], MFEA-AKT [46], MFEA-DGD [47], MFMP [48], and MTEA-AD [49]. *DE-based* methods include BLKT-DE [50] and MKTDE [51]. Lastly, *ES-based* methods include MTES-KG [52]. These algorithms span both multifactorial and multipopulation paradigms, and we use the MTO Platform [53] implementation and adapted our piano fingering tasks to their evaluation interface without modifying the algorithmic operators or default hyperparameters.

We evaluate five representations summarized in Table I. For the VAE-based representations (R4-R5), we use a VAE with two hidden MLP layers (64 and 32 units, LeakyReLU), trained for 40 epochs (batch size 512) with Adam (lr $= 5 \times 10^{-3}$). The latent mapping $h(\cdot)$ linearly rescales $u_j \in [0, 1]^p$ to $z_j \in [-3, 3]^p$ via $z_j = 6u_j - 3$.

For each representation-algorithm pair, 32 independent runs are conducted to characterize both optimization performance and behavior. Performance is evaluated using the average of the best objective value on each task and the aggregate metrics. To implement the fitness landscape analysis described in Section III-B, for each representation, we perform 50 independent random walks in the unified space, each with 5000 visited points, using a normalized Gaussian neighborhood ($\eta = 0.05$). Fitness values along the walks are used to compute the lag-1 autocorrelation $\rho(1)$, autocorrelation length τ, and neutrality ν, which are then correlated with EMTO performance.

B. Landscape Characterization Across Representations

Fig. 4 summarizes the distribution of representation-induced landscape features. All landscapes (R1-R5) exhibit smooth landscapes with high lag-1 autocorrelation ($\rho(1) > 0.90$) and long correlation length ($\tau > 95$), indicating coherent neighborhoods in the unified space. The main differentiator is neutrality ν, which controls the prevalence of plateaus and thus the balance between directed search and neutral drift.

Although all landscapes are smooth, correlation persistence differs across representations. R2 exhibits the longest autocorrelation lengths, forming wide and coherent basins that support efficient long-range exploitation. In contrast, R3 shows the shortest τ despite being locally smooth, resulting in faster correlation decay. R1, R4, and R5 lie between these extremes.

Neutrality separates the representations most clearly: R2 is dominated by extended neutral plateaus, making it favorable for drift-based and covariance-based optimizers. This suggests that homogeneous blocks in R2 induce plateau-dominated landscapes. Conversely, R1, R3, and R5 exhibit low neutrality, resulting in more decision-forced and slope-driven landscapes. Furthermore, lowering the latent size from R4 to R5 leads to a decrease in neutrality.

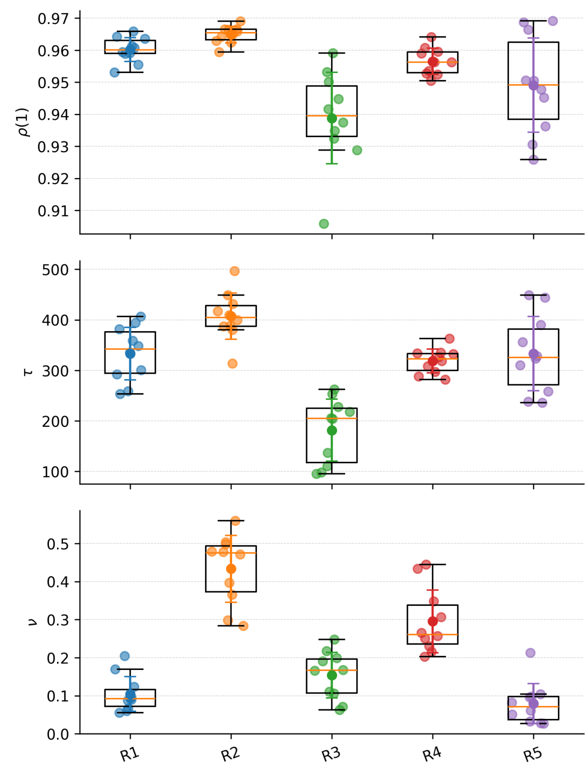


Fig. 4: Distribution of representation-induced landscape features of R1-R5 across tasks, including lag-1 autocorrelation $\rho(1)$, autocorrelation length τ, and neutrality ν.

C. Representation Effect on Multitasking Performance

Table II summarizes the relative transfer effects and Fig. 5 shows the performance rank distributions across representations and algorithms. Overall, multitasking performance depends strongly on both representation and optimizer, and positive transfer is not guaranteed.

Particularly, under AT-MFEA, MTEA-AD, BLKT-DE, and MKTDE, R2 achieved up to $+7\%$ (GA-based) and $+11$-12% (DE-based). Moreover, R3 shows particularly strong gain for DE-based multitasking, achieving improvements of $+27.27\%$ in BLKT-DE and $+13.70\%$ under MKTDE. In contrast, R1 and R5 yield only small positive gains for AT-MFEA and MTEA-AD, with the improvements noticeably smaller than those observed in R2-R3. Both also experience severe performance degradation under MKTDE, while the others show improvements. This suggests that R1 and R5 provide limited transferable structure and only weakly support multitasking. Furthermore, MFEA-AKT, MFMP, and MTES-KG consistently underperform across all representations with degradation ranging from -6% to -106%, indicating systematic negative transfer.

Beyond the performance improvement, Fig. 5 reveals an important difference in performance consistency and dominance through rank distributions. Notably, R1 frequently attains the best rank (rank 1) across multiple optimizers and tasks, reflecting its strong task efficiency and aggressive search

TABLE II: Median percent transfer effect Δ (%) across 8 tasks, $\Delta\ (\%) = (f^{\text{STO}} - f^{\text{MTO}})/f^{\text{STO}} \times 100$ (minimization, matched STO per optimizer family). $\Delta < -100$ indicates $f^{\text{MTO}} > 2f^{\text{STO}}$. Bold indicates the largest Δ (best transfer effect) per family.

		Δ (%)						Δ (%)			Δ (%)
Rep.	GA	AT-MFEA	MFEA-AKT	MFEA-DGD	MFMP	MTEA-AD	DE	BLKT-DE	MKTDE	CMA-ES	MTES-KG
R1	—	+0.78%	−6.54%	−136.34%	−9.80%	+1.06%	—	+8.11%	−3.82%	—	−26.27%
R2	—	+7.01%	−26.35%	−106.73%	−12.35%	+7.26%	—	+11.56%	+2.74%	—	−106.03%
R3	—	+5.62%	−15.02%	−98.69%	−56.29%	**+9.26%**	—	**+27.27%**	+13.70%	—	−43.83%
R4	—	+3.50%	−13.11%	−101.10%	−6.34%	+4.82%	—	+5.15%	+0.89%	—	−20.93%
R5	—	+0.23%	−7.64%	−116.24%	−11.27%	+1.71%	—	+5.69%	−9.33%	—	−6.06%

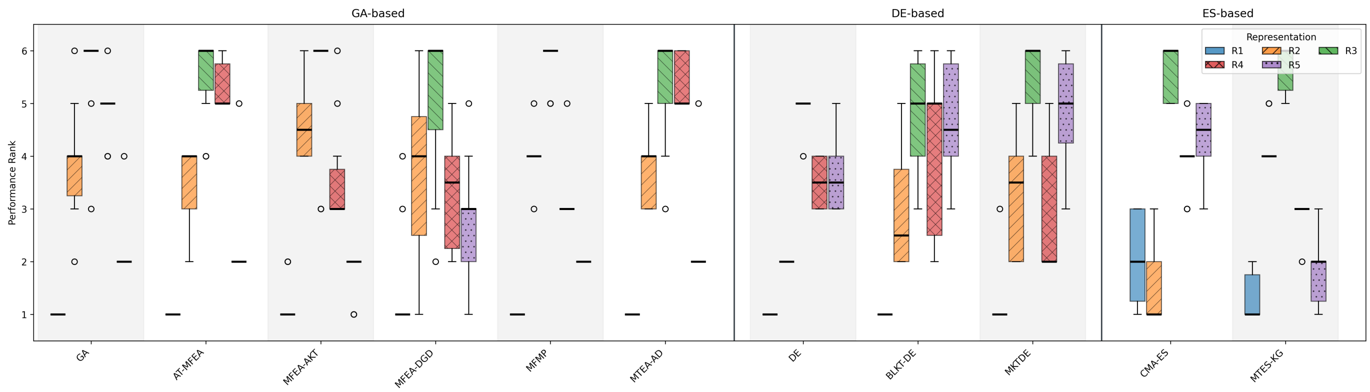


Fig. 5: Performance rank distributions across representations and optimizer. Boxplots show the median and interquartile range (IQR) across all tasks and runs. Collapsed boxes indicate zero IQR (stable central performance). Optimizers are grouped by their family; lower ranks indicate better performance.

behavior. However, this dominance is not uniformly preserved across multitasking strategies. Particularly, under the ES-based multitasking optimizer MTES-KG, the strong performance of R2 observed under CMA-ES deteriorates substantially, with ranks dropping to 4-5 and performance degrading by up to -106%, consistent with a breakdown in transfer effectiveness when optimizer-landscape compatibility is violated.

To assess whether the observed differences indicate a globally dominant representation, a Friedman Test was conducted across all representations and the multitasking optimizer. The results indicate a significant global performance separation among representations. Furthermore, pairwise Wilcoxon signed-rank tests ($\alpha = 0.05$) show that R1 achieves statistically superior performance compared to the remaining representations when aggregated across all settings. Taken together, the variability in improvement and rank stability across optimizers suggests that these performance differences are driven not solely by representation choice, but also by optimizer–landscape interaction.

D. Optimizer-Landscape Interaction in STO and MTO

The performance trends observed in the previous subsection indicate that the same representation-optimizer pair can lead to markedly different performance outcomes once transfer is introduced. This observation suggests that multitasking optimization (MTO) alters the relationship between optimizer dynamics and the representation-induced landscape. To better understand this behavior, we analyze this relationship via Spearman correlations between landscape features and performance under single task optimization (STO) and MTO as shown in Fig. 6.

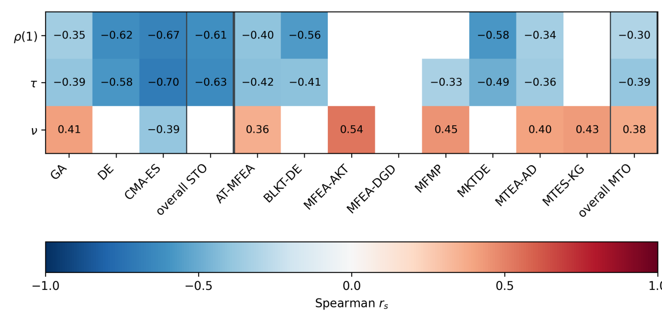


Fig. 6: Heatmap of Spearman correlation (r_s) between algorithm performance and landscape, including lag-1 autocorrelation $\rho(1)$, autocorrelation length τ, and neutrality ν. Columns grouped by optimizer (STO and MTO), and only statistically significant with Holm-corrected ($\alpha = 0.05$) are shown; others are blanks.

Under STO, ruggedness-related features $\rho(1)$ and τ dominate across optimizers, while neutrality plays a secondary and optimizer-dependent role. Higher neutrality tends to degrade GA performance, strongly benefits CMA-ES, and is largely irrelevant for DE. This behavior is consistent with the predominantly local nature of STO, where progress is driven by exploitation of local structure.

Under MTO, neutrality becomes substantially more predictive and ruggedness remains relevant, indicating that successful transfer requires representations that can preserve useful partial structures under cross-task variation and a coherent local neighborhood in the unified space. Thus, MTO performance is influenced by *both* ruggedness and neutrality.

In our MTO setting, mapping-induced neutral plateaus can buffer cross-task transfer, allowing offspring to inherit transferable components without immediate degradation, thereby mitigating negative transfer. Although all representations yield relatively smooth traces (high $\rho(1)$), smoothness can still limit post-transfer refinement: representations with stronger local coherence enable faster exploitation after mixing. Hence, effective MTO requires representations that support safe information exchange and maintain usable local structure for exploitation, helping explain why improvements arise only when both properties are present.

At the algorithmic level, MTO reveals a clear difference in how different optimizers respond to representation-induced landscape features. AT-MFEA and MTEA-AD exhibit the strongest coupling to landscape structure, with performance significantly associated with both ruggedness-related metrics and neutrality, suggesting they benefit from neutral buffers for exchange while still relying on coherent local neighborhoods for refinement. In contrast, methods such as MFEA-AKT and MTES-KG are primarily associated with neutrality, with smoothness metrics showing little explanatory power. Their performance improves as neutrality decreases, consistent with smaller degradation observed for R1 and R5 under these optimizers. MFMP is sensitive to τ and ν but not $\rho(1)$, suggesting it responds to correlation length and neutrality rather than purely to lag-1 continuity. DE-based multitasking optimizers still remain strongly associated with smoothness-related metrics and limited sensitivity to neutrality, indicating that DE operators continue to rely on local continuity even under multitasking. On the other hand, MFEA-DGD shows no consistent associations with any landscape metric, coinciding with its consistently poor performance and suggesting limited ability to exploit induced landscape structure.

Overall, AT-MFEA and MTEA-AD are the most structure-aligned optimizers, as they are the only methods whose performance ranks are significantly associated with both smoothness and neutrality-related landscape metrics.

E. Discussion

This study shows that MTO alters the relationship between optimizer dynamics and representation-induced landscape structure. While STO is largely explained by the ruggedness-related features, MTO increases the importance of neutrality together with ruggedness. This suggests that effective multitasking requires both neutral regions and coherent local neighborhoods. This distinction clarifies why representation-optimizer pairs that perform well in isolation may fail or even degrade once multitasking is introduced. Practically, representation choice should be optimizer-aware, since transfer changes which landscape properties are predictive.

Our findings further suggest that optimizer design embodies implicit assumptions about the geometry of the induced landscape in both single and multitasking settings. When transfer is introduced, cross-task recombination induces larger and less local perturbations in the unified space, which can invalidate these assumptions and lead to negative or catastrophic transfer. Consequently, no representation can be universally optimal across optimizers. Effective multitasking emerges only when the transfer mechanism is compatible with the induced landscape structure. These results indicate that representation should be treated as a first-order design factor in subset selection multitasking optimization, as it fundamentally shapes the landscape properties exploited during transfer.

However, several limitations should be noted. First, the analysis is conducted on subset selection problems. Further validation is thus required to assess its generality across other problem domains. Second, the landscape-performance relationship is established through correlation and does not imply direct causality. Third, although multiple optimizer families are examined, we do not exhaustively cover all transfer mechanisms. Finally, extending these findings to adaptive or co-evolving representations remains an important direction for future work.

V. Conclusion

This work presents a systematic analysis of how representation-induced landscape structure interacts with the optimizer dynamics in evolutionary multitasking optimization. Using piano fingering as the representative case of subset selection multitasking, we examine the performance of various continuous representations under both single-task and multitasking optimization. Our results show that multitasking changes which landscape properties influence optimization behavior. While ruggedness dominates STO, MTO performance depends on both neutrality and ruggedness. These findings highlight the importance of optimizer-landscape compatibility and demonstrate that representation plays a central role in multitasking optimization.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01842, Artificial Intelligence Graduate School Program (GIST)).

References

[1] A. Gupta, Y. S. Ong, and L. Feng, "Multifactorial evolution: Toward evolutionary multitasking," *IEEE Trans. on Evol. Comput.*, vol. 20, 2016.

[2] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan, "Multiobjective multifactorial optimization in evolutionary multitasking," *IEEE Trans. on Cybern.*, vol. 47, no. 7, pp. 1652–1665, 2017.

[3] T. Wei, J. Liu, A. Gupta, P. S. Tan, and Y.-S. Ong, "(θ_l,θ_u)-parametric multi-task optimization: Joint search in solution and infinite task spaces," *IEEE Trans. on Evol. Comput.*, pp. 1–1, 2025.

[4] T. Wei, S. Wang, J. Zhong, D. Liu, and J. Zhang, "A review on evolutionary multitask optimization: Trends and challenges," *IEEE Trans. on Evol. Comput.*, vol. 26, no. 5, pp. 941–960, 2022.

[5] R. Li, L. Wang, W. Gong, and F. Ming, "An evolutionary multitasking memetic algorithm for multiobjective distributed heterogeneous welding flow shop scheduling," *IEEE Trans. on Evol. Comput.*, vol. 29, no. 6, pp. 2287–2298, 2025.

[6] Y. Huang, W. Zhou, Y. Wang, M. Li, L. Feng, and K. C. Tan, "Evolutionary multitasking with centralized learning for large-scale combinatorial multiobjective optimization," *IEEE Trans. on Evol. Comput.*, vol. 28, no. 5, pp. 1499–1513, 2024.

[7] N. Zhang, A. Gupta, Z. Chen, and Y. S. Ong, "Evolutionary machine learning with minions: A case study in feature selection," *IEEE Trans. on Evol. Comput.*, vol. 26, 2022.

[8] Q.-T. Yang, X.-X. Xu, Z.-H. Zhan, J. Zhong, S. Kwong, and J. Zhang, "Evolutionary multitask optimization for multiform feature selection in classification," *IEEE Trans. on Cybern.*, vol. 55, no. 4, pp. 1673–1686, 2025.

[9] Y. Yuan, Y.-S. Ong, A. Gupta, P. S. Tan, and H. Xu, "Evolutionary multitasking in permutation-based combinatorial optimization problems: Realization with tsp, qap, lop, and jsp," in *2016 IEEE Region 10 Conference (TENCON)*, pp. 3157–3164, 2016.

[10] L. Feng, L. Zhou, A. Gupta, J. Zhong, Z. Zhu, K. C. Tan, and K. Qin, "Solving generalized vehicle routing problem with occasional drivers via evolutionary multitasking," *IEEE Trans. on Cybern.*, vol. 51, 2021.

[11] G. Ochoa, S. Verel, F. Daolio, and M. Tomassini, *Local Optima Networks: A New Model of Combinatorial Fitness Landscapes*, pp. 233–262. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[12] Y. Ichikawa, "Controlling continuous relaxation for combinatorial optimization," in *Proceedings of the 38th International Conference on Neural Information Processing Systems*, 2024.

[13] S. Bartunov, V. Nair, P. Battaglia, and T. Lillicrap, "Continuous latent search for combinatorial optimization," in *Learning Meets Combinatorial Algorithms at NeurIPS2020*, 2020.

[14] A. P. Iman and C. W. Ahn, "A model-free deep reinforcement learning approach to piano fingering generation," in *2024 IEEE Conference on Artificial Intelligence (CAI)*, pp. 31–37, IEEE, 2024.

[15] C. Qian, Y. Yu, and Z.-H. Zhou, "Subset selection by pareto optimization," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[16] L. Feng, L. Zhou, J. Zhong, A. Gupta, Y. S. Ong, K. C. Tan, and A. K. Qin, "Evolutionary multitasking via explicit autoencoding," *IEEE Trans. on Cybern.*, vol. 49, 2019.

[17] Z. Liang, H. Dong, C. Liu, W. Liang, and Z. Zhu, "Evolutionary multitasking for multiobjective optimization with subspace alignment and adaptive differential evolution," *IEEE Trans. on Cybern.*, vol. 52, no. 4, pp. 2096–2109, 2022.

[18] Z. Chen, Y. Zhou, X. He, and J. Zhang, "Learning task relationships in evolutionary multitasking for multiobjective continuous optimization," *IEEE Trans. on Cybern.*, vol. 52, no. 6, pp. 5278–5289, 2022.

[19] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *INFORMS J. Comput.*, vol. 6, pp. 154–160, 1994.

[20] Y. Huang, L. Feng, M. Li, Y. Wang, Z. Zhu, and K. C. Tan, "Fast vehicle routing via knowledge transfer in a reproducing kernel hilbert space," *IEEE Trans. on Systems, Man, and Cybern.: Systems*, vol. 53, no. 9, pp. 5404–5416, 2023.

[21] L. Feng, Y. Huang, L. Zhou, J. Zhong, A. Gupta, K. Tang, and K. C. Tan, "Explicit evolutionary multitasking for combinatorial optimization: A case study on capacitated vehicle routing problem," *IEEE Trans. on Cybern.*, vol. 51, no. 6, pp. 3143–3156, 2021.

[22] R. Lim, L. Zhou, A. Gupta, Y.-S. Ong, and A. N. Zhang, "Solution representation learning in multi-objective transfer evolutionary optimization," *IEEE Access*, vol. 9, pp. 41844–41860, 2021.

[23] J. Lin, Q. Chen, B. Xue, and M. Zhang, "Evolutionary multitasking for multiobjective feature selection in classification," *IEEE Trans. on Evol. Comput.*, vol. 28, no. 6, pp. 1852–1866, 2024.

[24] A. Kasimi, E. Nichols, and C. Raphael, "A simple algorithm for automatic generation of polyphonic piano fingerings.," in *Proceeding of ISMIR 2007*, pp. 355–356, 3 2007.

[25] M. Balliauw, D. Herremans, D. P. Cuervo, and K. Sörensen, "A tabu search algorithm to generate piano fingerings for polyphonic sheet music," 2015.

[26] M. Balliauw, D. Herremans, D. P. Cuervo, and K. Sörensen, "A variable neighborhood search algorithm to generate piano fingerings for polyphonic sheet music," *Int. Trans. Oper. Res.*, vol. 24, pp. 509–535, 2017.

[27] R. De Prisco, G. Zaccagnino, and R. Zaccagnino, "A differential evolution algorithm assisted by ANFIS for music fingering," *Swarm and Evol. Comput.*, pp. 48–56, 2012.

[28] E. Nakamura, N. Ono, and S. Sagayama, "Merged-output hmm for piano fingering of both hands," 2014.

[29] E. Nakamura, Y. Saito, and K. Yoshii, "Statistical learning and estimation of piano fingering," *Information Sciences*, vol. 517, pp. 68–85, 2020.

[30] W. Gao, S. Zhang, N. Zhang, X. Xiong, Z. Shi, and K. Sun, "Generating fingerings for piano music with model-based reinforcement learning," *Applied Sciences*, vol. 13, no. 20, p. 11321, 2023.

[31] A. P. Iman and C. W. Ahn, "Elite episode replay memory for polyphonic piano fingering estimation," *Mathematics*, vol. 13, no. 15, 2025.

[32] J. F. Gonçalves and M. G. C. Resende, "Biased random-key genetic algorithms for combinatorial optimization," *Journal of Heuristics*, vol. 17, pp. 487–525, 2011.

[33] F. Rothlauf, D. E. Goldberg, and A. Heinzl, "Network random keys—a tree representation scheme for genetic and evolutionary algorithms," *Evol. Comput.*, vol. 10, no. 1, pp. 75–97, 2002.

[34] M. A. Londe, L. S. Pessoa, C. E. Andrade, and M. G. Resende, "Biased random-key genetic algorithms: A review," *European Journal of Operational Research*, vol. 321, no. 1, pp. 1–22, 2025.

[35] X. Lu, J. Gonzalez, Z. Dai, and N. Lawrence, "Structured variationally auto-encoded optimization," in *International Conference on Machine Learning*, vol. 80 of *Proceedings of Machine Learning Research*, pp. 3267–3275, 2018.

[36] R. Luo, F. Tian, T. Qin, E. Chen, and T.-Y. Liu, "Neural architecture optimization," in *Advances in Neural Information Processing Systems*, vol. 31, pp. 7827–7838, 2018.

[37] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *International Conference on Learning Representations*, 2017.

[38] G. Mena, D. Belanger, S. Linderman, and J. Snoek, "Learning latent permutations with gumbel-sinkhorn networks," in *International Conference on Learning Representations*, 2018.

[39] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR*, 2014.

[40] 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.

[41] E. Weinberger, "Correlated and uncorrelated fitness landscapes and how to tell the difference," *Biol. Cybern.*, vol. 63, p. 325–336, Sept. 1990.

[42] L. Barnett, "Ruggedness and neutrality—the nkp family of fitness landscapes," in *Proceedings of the Sixth International Conference on Artificial Life*, (Cambridge, MA, USA), p. 18–27, MIT Press, 1998.

[43] L. Vanneschi, P. Collard, S. Verel, M. Tomassini, Y. Pirola, and G. Mauri, "A comprehensive view of fitness landscapes with neutrality and fitness clouds," in *Proceedings of EuroGP 2007*, pp. 241–250, 2007.

[44] P. F. Stadler, *Fitness landscapes*, pp. 183–204. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.

[45] X. Xue, K. Zhang, K. C. Tan, L. Feng, J. Wang, G. Chen, X. Zhao, L. Zhang, and J. Yao, "Affine transformation-enhanced multifactorial optimization for heterogeneous problems," *IEEE Trans. on Cybern.*, vol. 52, no. 7, pp. 6217–6231, 2022.

[46] L. Zhou, L. Feng, K. C. Tan, J. Zhong, Z. Zhu, K. Liu, and C. Chen, "Toward adaptive knowledge transfer in multifactorial evolutionary computation," *IEEE Trans. on Cybern.*, vol. 51, no. 5, pp. 2563–2576, 2021.

[47] Z. Liu, G. Li, H. Zhang, Z. Liang, and Z. Zhu, "Multifactorial evolutionary algorithm based on diffusion gradient descent," *IEEE Trans. on Cybern.*, vol. 54, no. 7, pp. 4267–4279, 2024.

[48] G. Li, Q. Lin, and W. Gao, "Multifactorial optimization via explicit multipopulation evolutionary framework," *Information Sciences*, vol. 512, pp. 1555–1570, 2020.

[49] C. Wang, J. Liu, K. Wu, and Z. Wu, "Solving multi-task optimization problems with adaptive knowledge transfer via anomaly detection," *IEEE Trans. on Evol. Comput.*, pp. 1–1, 2021.

[50] Y. Jiang, Z.-H. Zhan, K. C. Tan, and J. Zhang, "Block-level knowledge transfer for evolutionary multitask optimization," *IEEE Trans. on Cybern.*, vol. 54, no. 1, pp. 558–571, 2024.

[51] J.-Y. Li, Z.-H. Zhan, K. C. Tan, and J. Zhang, "A meta-knowledge transfer-based differential evolution for multitask optimization," *IEEE Trans. on Evol. Comput.*, vol. 26, no. 4, pp. 719–734, 2022.

[52] Y. Li, W. Gong, and S. Li, "Multitask evolution strategy with knowledge-guided external sampling," *IEEE Trans. on Evol. Comput.*, vol. 28, no. 6, pp. 1733–1745, 2024.

[53] Y. Li, W. Gong, T. Zhang, F. Ming, S. Li, Q. Gu, and Y.-S. Ong, "MToP: A MATLAB benchmarking platform for evolutionary multitasking," *arXiv preprint arXiv:2312.08134*, 2023.