Exploring Generative AIs as Population Variation Operator in Multi-objective Optimization Problems

1st Gerardo Ibarra-Vazquez
School of Engineering and Sciences
Tecnologico de Monterrey
Monterrey, Mexico
gerardo.ibarra.v@tec.mx

2nd Hugo Terashima-Marín
School of Engineering and Sciences
Tecnologico de Monterrey
Monterrey, Mexico
terashima@tec.mx

3rd Carlos A. Coello Coello
Departamento de Computacón
CINVESTAV-IPN
Mexico City, Mexico
ccoello@cs.cinvestav.mx

Abstract—**In recent years, evolutionary computation has significantly advanced in processes related to machine learning. However, the reciprocal integration of machine learning techniques into evolutionary computation remains relatively unexplored. Machine learning can substantially enhance the understanding of processes within Multi-Objective Evolutionary Algorithms (MOEAs) by harnessing its proficiency in identifying patterns and employing data-driven approaches. Existing studies lack a comprehensive understanding of the intricate interaction between machine learning models and evolutionary algorithms, necessitating prioritized investigation to ensure the efficacy, reliability, and compatibility of integrated models within optimization frameworks. This paper addresses this gap by examining the behavior of using Generative Artificial Intelligence (AI) models as a population variation operator in Multi-objective Optimization Problems. Our experimental results reveal that Generative AI, particularly Distributional Adversarial Networks (DANs), surpasses the performance of a traditional Generative Adversarial Network. Furthermore, DANs improve the population by generating novel non-dominated solutions and augmenting overall performance and diversity. This study reveals the potential of the integration of Generative AI in evolutionary computation, presenting a pathway for advancements in addressing common challenges within multi-objective optimization problems.**

Index Terms—**Generative Artificial Intelligence, Population Variation Operator, Machine Learning, Multi-objective Optimization**

I. Introduction

Multi-objective Evolutionary Algorithms (MOEAs) have been a powerful tool to handle uncertainty while finding global and robust optimal solutions in Multi-objective Optimization Problems (MOPs) [1]. Being population-based search approaches, they inherently have the capability for parallel computing and can generate a set of solutions in a single run. Nevertheless, the increasing complexity of MOPs poses a multifaceted challenge, marked by factors such as the high cost of function evaluations [2], numerous objectives [3], and large-scale search spaces [4]. Despite substantial progress in MOEAs in recent decades, the challenges of MOPs have limited the effectiveness of conventional optimization approaches, requiring more sophisticated approaches. MOEAs have synergized with Machine Learning (ML) to foster integration between these two research domains [5], leveraging the strengths of both approaches and combining evolutionary principles for optimization with the adaptability and learning capabilities intrinsic to ML.

Integrating MOEAs into ML frameworks has ushered in a new era of improved model robustness and adaptability, fostering breakthroughs in optimizing machine learning model parameters, enhancing feature selection processes, fine-tuning hyperparameters, and optimizing ensemble learning strategies [6]. Nevertheless, more exploration is still needed to incorporate ML techniques seamlessly into MOEAs. Recently, ML techniques have been employed to support and enhance the evolutionary components of MOEAs [7]. For example, employing machine learning methods for systematically analyzing feasible solutions produced by the exploration of an MOEA in the variable space that generates a substantial volume of data for a deeper understanding of the search behavior, enhancing the MOEA's future search capability [8]. Dimensionality reduction and spatial transformation techniques can simplify the objective and search spaces [9]. Reinforcement learning can identify suitable evolutionary operators (actions) based on any parent (state), thereby guiding the generator in producing offspring of high quality [10]. Therefore, MOEAs with machine learning have received extensive attention in Evolutionary Computation because they promise to create more adaptive, intelligent algorithms capable of learning and adapting to dynamic environments.

Traditional genetic operators like mating selection, crossover, and mutation in Multi-Objective Evolutionary Algorithms (MOEAs) may need help to fully capture the diversity and complexity of the solution space. Generative Artificial Networks (GANs) offer a solution by autonomously generating data, images, and content with high fidelity [11]. By integrating GANs into MOEAs, we can overcome these limitations. Traditional GANs analyze individual sample points, whereas Distributional Adversarial Networks (DANs) distinguish themselves by having discriminators analyze samples containing more than one example ($n > 1$), enhancing the ability to understand the distribution of data [12]. Figure 1 shows the use of a Generative AI model as a population variation operator in a MOEA.

C.A. Coello Coello is also (on a sabbatical leave) with the Faculty of Excellence of the School of Engineering and Sciences, Tecnologico de Monterrey, Monterrey.

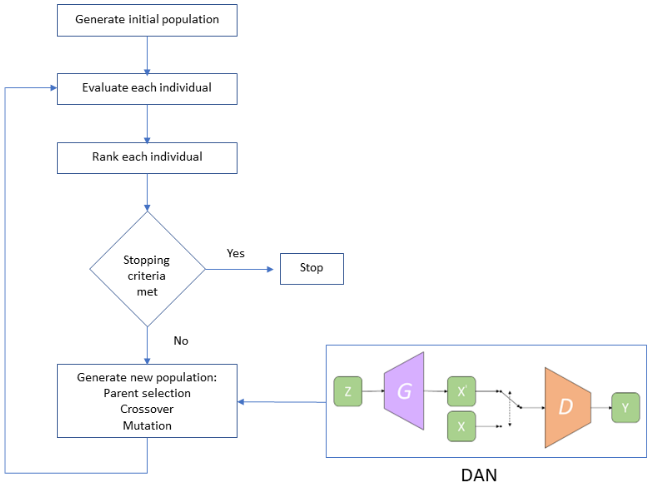

Generate initial population
Evaluate each individual
Rank each individual
Stopping criteria met
Yes
Stop
No
Generate new population: Parent selection Crossover Mutation
Z
G
X
X
D
Y
DAN

Fig. 1: Generative AI as a population variation operation in a MOEA.

Integrating ML models into MOEAs has become increasingly prevalent in Evolutionary Computation. However, there is a noticeable gap in the literature regarding thoroughly analyzing these models' behavior before their integration. Existing studies often focus solely on technical aspects, overlooking the intricate interaction between machine learning models and evolutionary algorithms. Therefore, there is a critical need for research prioritizing a comprehensive understanding of these models' behavior to ensure the integrated model's efficacy, reliability, and compatibility within the optimization framework. This study addresses this gap by examining the behavior of using Generative AI models as a population variation operator in MOPs. It evaluates the behavior of a DAN and GAN model using four key indicators: the C-metric, the Hypervolume, the IGD+ indicator, and the Riesz S-energy function. Additionally, the study employs a one-tailed Wilcoxon ranksum test to validate the statistical confidence of the results obtained by the models.

Throughout the subsequent sections of this paper, we provide a succinct overview of the theoretical background in Section II. The experimental setup is outlined in Section III. Subsequently, Section IV presents and discusses the obtained results. Lastly, conclusive insights are drawn in Section V.

II. Theoretical Background

In this section, we establish the theoretical background for MOEAs assisted by ML. Before delving into the specifics, it is essential to provide a concise overview of the general flow of an MOEA from the perspective of ML. This brief introduction will show the fundamental operational aspects and the interplay between evolutionary algorithms and machine learning within multi-objective optimization. The starting step in a typical MOEA begins with creating an initial parent population composed of randomly sampled solutions. Next, a population of offspring solutions is generated through a generative or reproductive model composed of genetic operators, such as mating selection, crossover, and mutation [13]. The newly generated solutions are assessed through a function evaluator to determine the value obtained in each objective function. Then, a merged population of parents and offspring is introduced into a discriminative or selective model, which ranks and filters the best solutions that will survive for the next generation. Usually, these two models are commonly referred to as the evolutionary generator and discriminator. The evolutionary generator aims to create offspring with superior qualities over their parents, while the evolutionary discriminator endeavors to discern the qualities between parent and offspring solutions and then select the best individuals for the next generation (Fig. 2). Therefore, by exploring various solutions that gradually converge, an MOEA is able to approximate the optimal solutions in MOPs driven by these two processes (generation and discrimination).

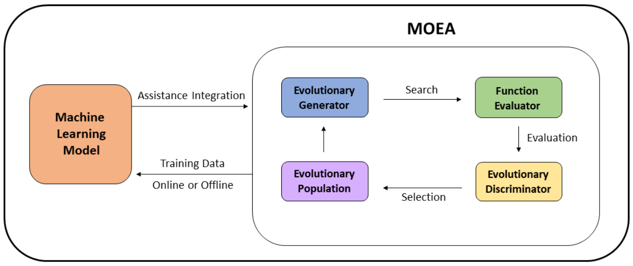


Fig. 2: Evolutionary process assisted by Machine Learning.

In this context, ML can have a fundamental role in enhancing the evolutionary process [7]. Two main research avenues have emerged from this challenge: (1) Evolutionary Generators: Existing genetic operators deteriorate their performance in large-scale spaces, almost becoming a random process emphasizing the importance of learning strategies to enhance the generator's search capability for effectively solving Large-Scale Multi-Objective Problems [14]; and (2) Evolutionary Discriminators: As the number of objectives increases, the effectiveness of evaluation and selection strategies diminishes. Therefore, the discriminator's ability to converge and maintain diversity needs improvement in MOPs [15].

On one hand, random-based genetic operators frequently struggle to find a proper direction to speed up the search, causing the population to progress in a coarse and sluggish manner, a challenge that becomes complex with larger search spaces [16]. Therefore, there is a raise in the computing resources required to approximate optimal solutions in MOPs. On the other hand, Covariance Matrix Adaptation evolution strategies may struggle with multi-objective optimization due to their single-objective focus and the exponential computational overhead of maintaining the covariance matrix, potentially limiting scalability in high-dimensional problems [17].

Generative models in machine learning have gained popularity for their ability to learn and represent the statistical distribution of training data, allowing the generation of new samples based on learned patterns [18]. Among these models, GANs have emerged as a prominent approach. GANs consist of two neural networks (see Fig. 3), a Generator (G) and a discriminator (D), engaged in an adversarial training process [19]. The generator produces synthetic data by considering samples from the dataset, while the discriminator distinguishes between real and synthetic data, driving the generator to produce more realistic samples.

GANs have succeeded in various applications, with remarkable achievements in computer vision [20]. Their contributions in image-related domains span from synthesizing and enhancing images to performing transformative tasks like super-resolution [21], deep fake technology for image generation [22], among others. Even the generation of synthetic data for training machine learning models when real data is limited or difficult to obtain [23].

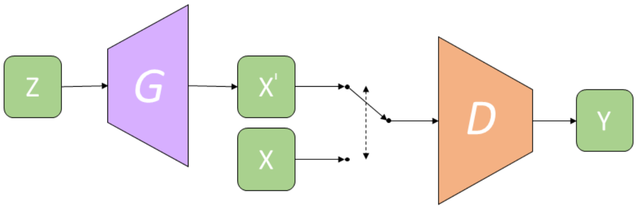


Fig. 3: GAN architecture composed of the generator (G) and discriminator (D) models. Z denotes the noise source, X' is the synthetic data sample, X is the original data sample, and Y is the label (real≈ 1, fake≈ 0).

Despite achieving success, GANs face several well-recognized challenges, with their training process being a focal point of concern [24]. During training iterations, a tendency for oscillation between the generator and discriminator occurs, and even a subtle imbalance in their respective capacities may contribute to the divergence of the training process. Another challenge occurs when the generator learns a distribution focusing on a limited set of modes within the original data distribution, neglecting the remaining space. This issue, called mode collapse, becomes apparent in the generation of synthetic data that, despite their realism, exhibit a remarkable deficiency in diversity, converging toward a limited array of prototypes [25].

A training framework that uses distributional adversaries called Distributional Adversarial Network (DAN) [12] mitigates the mode collapse problem, and for each prediction, it adapts the discriminator to consider an entire sample rather than a singular sample point. Specifically, the discriminator is defined as a set function denoted by $M : 2^{\mathbb{R}^d} \to \mathbb{R}$, designed to operate on a sample $x_1, \ldots, x_n$ with potentially varying sizes.

In contrast to the conventional GAN discriminator, DANs employs a data-driven technique known as the Deep Mean Encoder (DME). This approach utilizes the mean as a pivotal statistic for discerning between distributions, playing a crucial role in integrating the Maximum Mean Discrepancy (MMD) [26]. MMD stands out as a robust metric with strong theoretical foundations, specifically designed for measuring the discrepancy between distributions. Rather than employing explicit mapping, the Deep Mean Encoder (DME) adopts a more sophisticated approach by learning this function through the parameterization of ϕ as a neural network. The resulting expression for the DME, denoted as μ, assumes the following form:

$$\mu(\mathbb{P}) = \mathbb{E}_{\mathbf{X}\sim\mathbb{P}}[\phi(\mathbf{x})] \tag{1}$$

In practical applications, μ is constrained to access $\mathbb{P}$ through finite-sized samples, consequently adopting the following form:

$$\mu(\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}) = \frac{1}{n}\sum_{i=1}^{n} \phi(\mathbf{x}_i) \tag{2}$$

The main feature of the DAN model is the distributional encoder (DME). However, to fully integrate eq. (2) as part of the discriminator in the adversarial training involves employing a classifier. Specifically, when presented with a vector encoding, a sample originating from either the data distribution $\mathbb{P}_x$ or the generated distribution $\mathbb{P}_G$, the classifier φ_s assigns a label of 1 to indicate the sample's origin from $\mathbb{P}_x$ and 0 otherwise. The quantification of the distribution discrepancy can be realized by assessing the classifier's confidence and employing a logistic loss. This formulation obtains the following objective function:

$$d_s(\mathbb{P}_0, \mathbb{P}_1) = log(\varphi_s(\mu(\mathbb{P}_1)) + log(1 - \varphi_s(\mu(\mathbb{P}_1)) \tag{3}$$

This formulation bears similarity to the original GAN objective but diverges in a pivotal aspect: the expectation is encapsulated within φ_s. In essence, while in GANs the loss of a sample is defined as the expected loss across sample points, in DAN, the sample loss is a singular value derived from the deep mean embedding of its expected value. The full model (sample classifier) is denoted as $M_s := \varphi_s \cdot \mu$, encompassing the DME and the classifier.

III. Experimental setup

This section presents the experimental setup to evaluate the Generative Artificial Intelligence (AI) models as population variation operators. Our objective is to empirically assess the performance of DAN and compare it with a traditional GAN in the context of MOPs.

The experiment began by the empirical evolution of populations from NSGA-II in an intermediate step, where the Pareto front is shaped, and the distribution of individuals is defined. Subsequently, GANs and DANs were employed to generate new individuals. The objective of this experiment was to study the behavior of these models across various geometries derived from benchmark problems. For the performance evaluation, we adopted the c-metric to measure the coverage of the newly generated individuals compared to the original set in the objective space. After that, the two sets were merged, and applied non-dominated sorting with crowding distances to select the best individuals to match the original population size. Then, the Hypervolume was employed to quantify the volume of the objective space that is dominated by a particular set of solutions, the IGD+ indicator measured the inverted generational distance between the population and the Pareto front, and the Riesz S-energy function provided insight into the dispersion or concentration of individuals within the objective space. Additionally, an one-tailed Wilcoxon ranksum test was used to compare the results obtained by DAN and GAN.

The experiment implementation was made using the Pymoo library [27], with TensorFlow [28] for implementing Generative Adversarial Networks (GANs) and Distributional Adversarial Networks (DANs). Population size was set to 100 individuals. For each algorithm for each test problem, we performed 30 independent runs.

A. Benchmark

Based on the review from [29], and considering recent benchmarks such as the ZCAT test problems [30]. We selected the Zitzler, Deb, and Thiele (ZDT) Test Suite for our analysis, considering its widespread use in the literature. This benchmark aligns with our goal of conducting an initial exploration with a limited number of variables and objectives. This constraint facilitates the visualization of Pareto fronts and streamlines our initial analysis.

The ZDT test suite comprises six MOPs (ZDT1-6) and stands out as one of the most widely utilized benchmarks in the literature [31]. ZDT5 is often omitted from analysis in the literature due to its binary encoding. These problems, representative of diverse challenges, encompass features such as multimodality, leading to phenomena like Pareto many-to-one problems (as seen in ZDT6), disconnected problems (as in ZDT3), and multifrontal problems (as presented in ZDT4). All the ZDT test problems are characterized by employing only one position parameter, underscoring their dependency on a single variable. Although widely embraced, the ZDT test suite must provide a comprehensive evaluation framework for the diverse landscape of multi-objective optimization problems.

TABLE I: ZDT Benchmark

Problem	Variables	Objective	Geometry
ZDT1	30	2	convex
ZDT2	30	2	concave
ZDT3	30	2	disconnected
ZDT4	10	2	convex
ZDT6	10	2	concave

In Tab. I, we present a comprehensive overview of the ZDT test problems, highlighting key characteristics such as the number of variables, objectives, and their geometry. In our experimental setup, we used the ZDT test suite (ZDT 1-6, excluding ZDT5) to evaluate the proposed methodology.

B. Indicators

In this section, we will outline the indicators adopted in the evaluation process and their definitions. We use the c-metric from [32] that considers two Pareto front approximations $\mathbf{A}, \mathbf{B} \subseteq \mathbf{X}$. The C-metric maps the ordered pair $(\mathbf{A}, \mathbf{B})$ to the interval [0;1]:

$$C(\mathbf{A}, \mathbf{B}) = \frac{|\{\mathbf{b} \in \mathbf{B} \mid \exists\, \mathbf{a} \in \mathbf{A} : \mathbf{a} \preceq \mathbf{b}\}|}{|\mathbf{B}|} \tag{4}$$

When $C(\mathbf{A}, \mathbf{B}) = 1$, it indicates that all elements in $\mathbf{B}$ are dominated by or equal to the elements of $\mathbf{A}$. Contrarily, when $C(\mathbf{A}, \mathbf{B}) = 0$, it means that all elements in $\mathbf{B}$ strictly dominate the elements in set $\mathbf{A}$. It is worth to say that $C(\mathbf{A}, \mathbf{B}) \neq 1 - C(\mathbf{A}, \mathbf{B})$. This indicator quantifies the ratio of points in a Pareto set approximation $\mathbf{A}$ dominated by the Pareto set approximation $\mathbf{B}$.

For the HV, we use the implementation from [33], which is described as the volume of the space in the objective space dominated by the Pareto front approximation S and delimited from above by a reference point $r \in \mathbb{R}^m$ such that for all $z \in S, z \prec r$. The mathematical expression for the HV indicator is defined as follows:

$$HV(S, r) = \lambda_m(\bigcup_{z \in S} [z; r]) \tag{5}$$

where λ_m is the m-dimensional Lebesgue measure. For $IGD+$, we used the original definition from [34]. This particular modification enhances the IGD indicator, rendering it weakly Pareto compliant and defined by the following mathematical expression:

$$IGD^{+}(\mathbf{A}) = \frac{1}{|Z|} \left(\sum_{i=1}^{|Z|} {d_i^{+}}^{2} \right)^{1/2} . \tag{6}$$

In the context of minimization, the modified distance $d_i^+ = \max\{a_i - z_i, 0\}$ denotes the distance from z_i to the nearest solution in set $\mathbf{A}$ with the associated value a_i. Pair-potential functions, such as Riesz s-energy function, have been employed in MOEAs to quantify point distribution based on pairwise interactions, aiming to enhance population diversity in the objective space [35]. For calculating the s-energy [36] of a set of approximation points $\mathbf{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_N\}$, where $\mathbf{a}_i \in \mathbb{R}^m$ is given by:

$$E_s(\mathbf{A}) = \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} \frac{1}{\|a_i + a_j\|^s} \tag{7}$$

where $\|\cdot\|$ denotes the Euclidean distance, and $s \geq 0$ is a parameter that determines the power to which the distances are raised.

IV. Results

In Figs. 4 and 5, we present visual comparisons depicting examples of the original and the generated populations for each of the ZDT test problems using the GAN and DAN models. In the displayed image, the *x*-axis represents the number of decision variables, the *y*-axis denotes individual instances, and the intensity of each pixel corresponds to the respective variable value. Our observations show that the generated populations exhibit discernible patterns derived from the original individuals. However, the DAN model shows its capability to construct more precise individuals and to establish a more remarkable similarity with the original population. This comparative analysis reveals that while the generated populations from the GAN model learn some information from the original individuals, the DAN model surpasses this by constructing more precise populations using distributional information.

Table II presents the results of the C-metric and the HV indicators. Regarding the C-metric, DAN consistently outperforms GAN, demonstrating superiority by yielding nearly seven times more non-dominated solutions in the best-case scenario (ZDT4). In the worst-case scenario, DAN achieved a mean value of 0.2048 having more coverage than GAN (0.1096). The intricate nature of ZDT6, a many-to-one problem where multiple solutions in the decision space map to the same solution in the objective space, makes both models unable to produce non-dominated solutions. Regarding HV, the mean value for the problems ZDT1-4 increased when individuals from DAN were integrated into the population. The one-tailed Wilcoxon ranksum test also indicates significant differences, showing that the DAN results are significantly better than those of the GAN.

TABLE II: Statistical results (mean and standard deviation) of the C metric and HV values on the ZDT test suite.

	C-metric		HV		
	GAN	DAN	Original	GAN	DAN
Problem	Mean (Std.)	Mean (Std.)	Mean (Std.)	Mean (Std.)	Mean (Std.)
ZDT1	0.1975 (0.2496)	**0.4095 (0.1904)**∗	0.6554 (0.0360)	0.6568 (0.0360)	**0.6586 (0.0359)**∗
ZDT2	0.1902 (0.2371)	**0.2535 (0.2165)**∗	0.3698 (0.0751)	0.3708 (0.0753)	**0.3714 (0.0752)**∗
ZDT3	0.1042 (0.1096)	**0.2048 (0.1261)**∗	0.9499 (0.0465)	0.9514 (0.0443)	**0.9518 (0.0456)**∗
ZDT4	0.0699 (0.1950)	**0.5015 (0.3119)**∗	0.6282 (0.1555)	0.6289 (0.1560)	**0.6381 (0.1536)**∗
ZDT6	0 (0)	0 (0)	0.3419 (0.0213)	0.3419 (0.0213)	0.3419 (0.0213)

The best result is shown in gray. The symbol ∗ is placed when the DAN result is statistically different from the GAN result based on a one-tailed Wilcoxon ranksum test with a significance level of 5%.

In Table III, we present the results from applying the IGD+ and the Riesz S-energy indicators to the original set and the merged sets, which include individuals generated by both GAN and DAN models. Regarding IGD+, we observe lower values when DAN individuals are merged with the original population. Despite being a small contribution to the indicator, this decrease underscores the meaningful impact on both convergence and coverage. In the context of the Riesz S-energy function, we compute the logarithm of the indicator to make the resulting values easier to interpret. We note a consistent trend where incorporating DAN individuals reduces the indicator value, indicative of a more evenly distributed population. The one-tailed Wilcoxon ranksum test shows a significant superiority of DAN over GAN in most cases; however, for ZDT2 and ZDT3, where the IGD+ indicator was used, no significant differences were found, possibly due to the concave nature of ZDT2 and the disconnected Pareto front in ZDT3, which are challenging for optimization algorithms.

In Figures 6 and 7, we provide visual comparisons of benchmark problems, including the Pareto front, original population, and generated individuals by the GAN and DAN models. We classify the generated solutions as dominated or non-dominated by the original population, offering insights into the coverage as evaluated by the C-metric. Remarkably, DAN excels over GAN in generating new non-dominated solutions, while both models face challenges dealing with disconnected shapes, as shown in ZDT3. The many-to-one problem, exemplified by ZDT6, poses a substantial challenge. Although both models generate similar populations, as depicted in Figures 4 and 5, a considerable portion of these populations converges to the same location in objective space. This underscores the complexity of addressing many-to-one problems in multi-objective optimization.

TABLE III: Statistical results (mean and standard deviation) of the IGD+ and Riesz S-energy function values on the ZDT test suite.

	IGD+			Riesz S-energy		
	Original	GAN	DAN	Original	GAN	DAN
Problem	Mean (Std.)	Mean (Std.)	Mean (Std.)	Mean (Std.)	Mean (Std.)	Mean (Std.)
ZDT1	0.1472 (0.0250)	0.1462 (0.0250)	**0.1446 (0.0248)**∗	14.9596 (0.9782)	14.6738 (0.9712)	**14.0325 (0.9613)**∗
ZDT2	0.0964 (0.0230)	0.0959 (0.0229)	**0.0955 (0.0228)**	14.9693 (0.9934)	14.5667 (0.8940)	**14.1700 (1.0696)**∗
ZDT3	0.1808 (0.0334)	0.1798 (0.0340)	**0.1792 (0.0339)**	14.7601 (1.2790)	14.5081 (1.3737)	**14.1491 (1.1388)**∗
ZDT4	0.16 (0.1585)	0.1595 (0.1587)	**0.1594 (0.1612)**∗	15.3285 (2.1976)	15.1156 (2.0662)	**14.1642 (2.3520)**∗
ZDT6	0.1237 (0.0170)	0.1237 (0.0170)	0.1237 (0.0170)	15.9503 (1.3602)	15.9503 (1.3602)	15.9503 (1.3602)

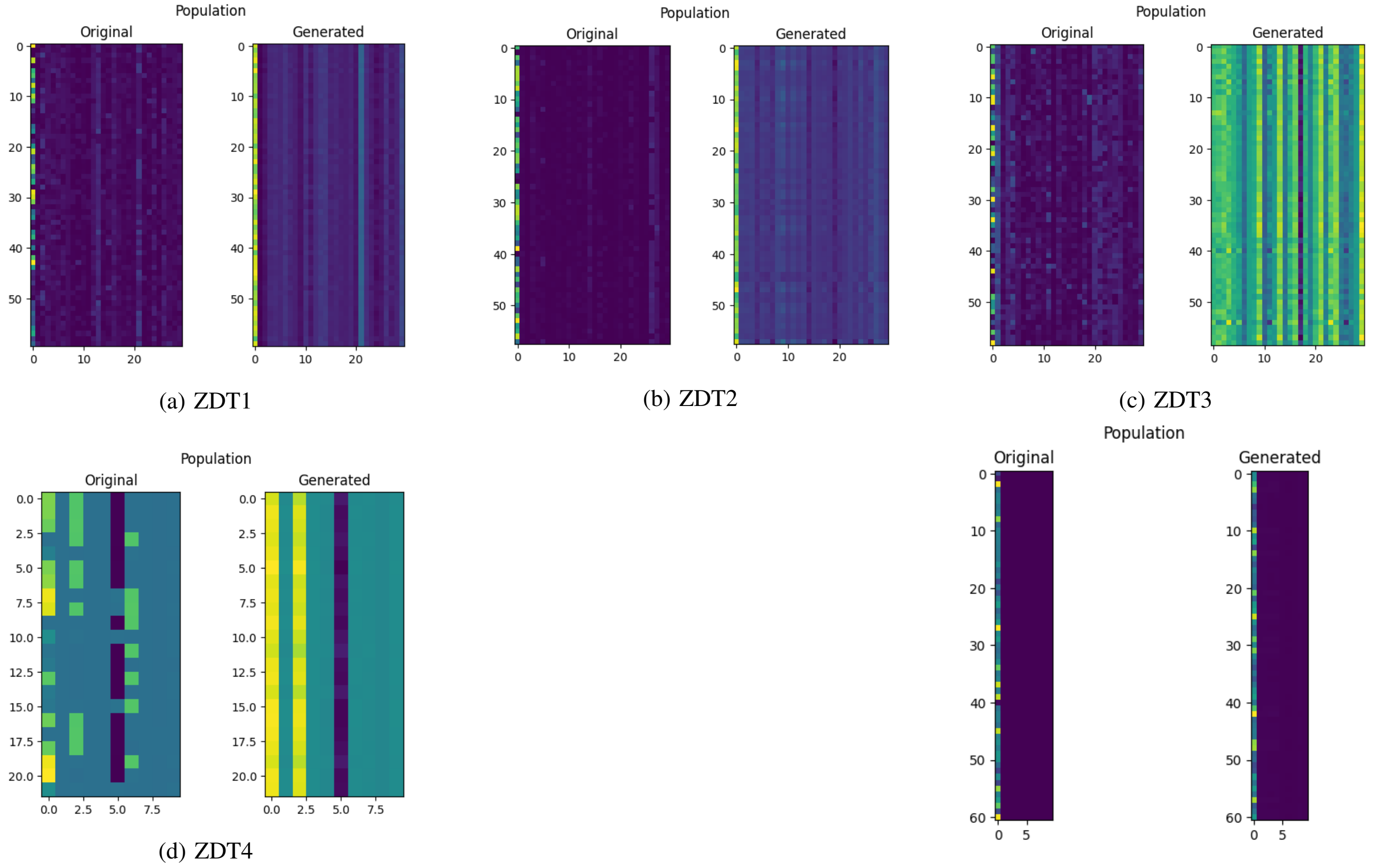
The best result is shown in gray. The symbol ∗ is placed when the DAN result is statistically different from the GAN result based on a one-tailed Wilcoxon ranksum test with a significance level of 5%.

V. Conclusions

This research provides valuable insights into the role of Generative AI as a population variation operator within Multi-objective Problems. Generative AI showed effectiveness, particularly in addressing more straightforward problems such as ZDT1, ZDT2, and ZDT4, where it successfully captures and reproduces the underlying distribution of the population. However, challenges arise when dealing with more complex problems, such as the disconnected shape in ZDT3 or the multifrontality leading to Pareto many-to-one problems in ZDT6, highlighting the limitations faced by these models.

Our findings suggest that the integration of the generated individuals into the evolving population has the potential to yield remarkable improvements, as evidenced by changes in several performance indicators. Tables II and III highlight the improvements obtained by Generative AI models. The DAN model shows a superior performance compared to a traditional GAN, offering more non-dominated solutions, better coverage (C-metric), and improvements in HV for ZDT1-4, while also having a positive impact on diversity, obtaining lower IGD+ values and a more evenly distributed population according to the Riesz S-energy indicator. Additionally, the one-tailed Wilcoxon ranksum test also showed significant differences, indicating that the DAN results are significantly better than those of the GAN. These changes highlight how Generative AIs significantly impact the overall population, improving the performance across several problem instances.

In conclusion, the findings from our exploration of Generative AI as a population variation operator in MOPs reveal promising avenues for further research and practical imple-



(a) ZDT1

(b) ZDT2

(c) ZDT3

(d) ZDT4

(e) ZDT6

Fig. 4: Visual Comparison between original and generated population using a GAN.

mentation. The observed results in performance indicators underscore the potential benefits of integrating generated individuals into the population. As part of our future work, we envision extending this research to recent benchmarks that can handle large-scale multi-objective optimization problems where existing genetic operators deteriorate their performance.

References

[1] C. A. C. Coello, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.

[2] L. V. Santana-Quintero, A. A. Montano, and C. A. C. Coello, "A review of techniques for handling expensive functions in evolutionary multi-objective optimization," *Computational intelligence in expensive optimization problems*, pp. 29–59, 2010.

[3] Z.-H. Zhan, L. Shi, K. C. Tan, and J. Zhang, "A survey on evolutionary computation for complex continuous optimization," *Artificial Intelligence Review*, pp. 1–52, 2022.

[4] Y. Tian, L. Si, X. Zhang, R. Cheng, C. He, K. C. Tan, and Y. Jin, "Evolutionary large-scale multi-objective optimization: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–34, 2021.

[5] Z. Fan, K. Hu, F. Li, Y. Rong, W. Li, and H. Lin, "Multi-objective evolutionary algorithms embedded with machine learning—a survey," in *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2016, pp. 1262–1266.

[6] Z.-H. Zhan, J.-Y. Li, and J. Zhang, "Evolutionary deep learning: A survey," *Neurocomputing*, vol. 483, pp. 42–58, 2022.

[7] S. Liu, Q. Lin, J. Li, and K. C. Tan, "A survey on learnable evolutionary algorithms for scalable multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, 2023.

[8] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, "A survey of multiobjective evolutionary algorithms for data mining: Part i," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 4–19, 2013.

[9] R. Jiao, B. H. Nguyen, B. Xue, and M. Zhang, "A survey on evolutionary multiobjective feature selection in classification: approaches, applications, and challenges," *IEEE Transactions on Evolutionary Computation*, 2023.

[10] M. M. Drugan, "Reinforcement learning versus evolutionary computation: A survey on hybrid algorithms," *Swarm and evolutionary computation*, vol. 44, pp. 228–246, 2019.

[11] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," *IEEE transactions on knowledge and data engineering*, vol. 35, no. 4, pp. 3313–3332, 2021.

[12] C. Li, D. Alvarez-Melis, K. Xu, S. Jegelka, and S. Sra, "Distributional adversarial networks," *CoRR*, vol. abs/1706.09549, 2017. [Online]. Available: http://arxiv.org/abs/1706.09549

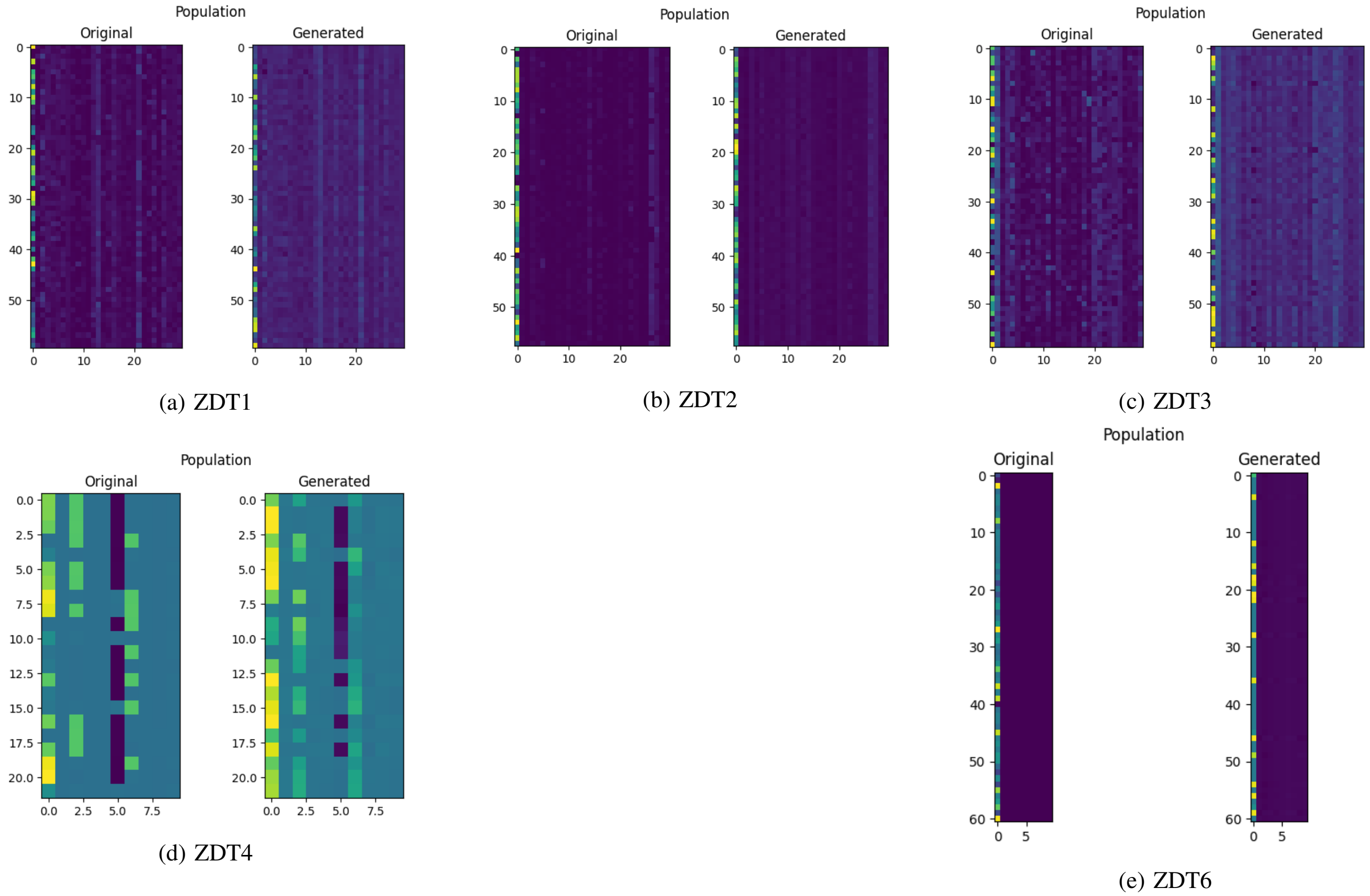
[13] P. P. Bonissone, R. Subbu, N. Eklund, and T. R. Kiehl, "Evolutionary algorithms+ domain knowledge= real-world evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 256–280, 2006.

[14] O. Ramos-Figueroa, M. Quiroz-Castellanos, E. Mezura-Montes, and R. Kharel, "Variation operators for grouping genetic algorithms: A review," *Swarm and Evolutionary computation*, vol. 60, p. 100796, 2021.

[15] C. A. Coello Coello, S. González Brambila, J. Figueroa Gamboa, M. G. Castillo Tapia, and R. Hernández Gómez, "Evolutionary multiobjective optimization: open research areas and some challenges lying ahead," *Complex & Intelligent Systems*, vol. 6, pp. 221–236, 2020.

[16] A. Vié, "Qualities, challenges and future of genetic algorithms," *Available at SSRN 3726035*, 2020.

[17] Y. Akimoto and N. Hansen, "Diagonal acceleration for covariance matrix



(a) ZDT1

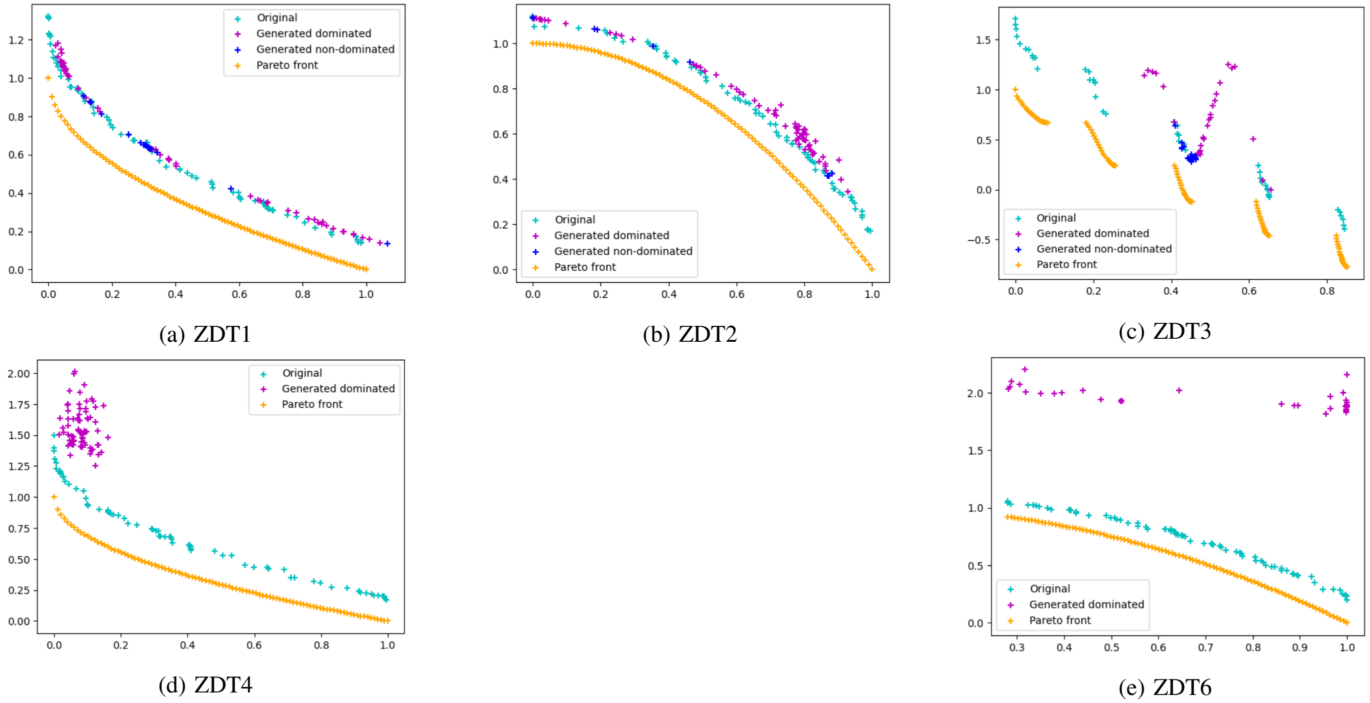
(b) ZDT2

(c) ZDT3

(d) ZDT4

(e) ZDT6

Fig. 5: Visual Comparison between original and generated population using a DAN.



(a) ZDT1

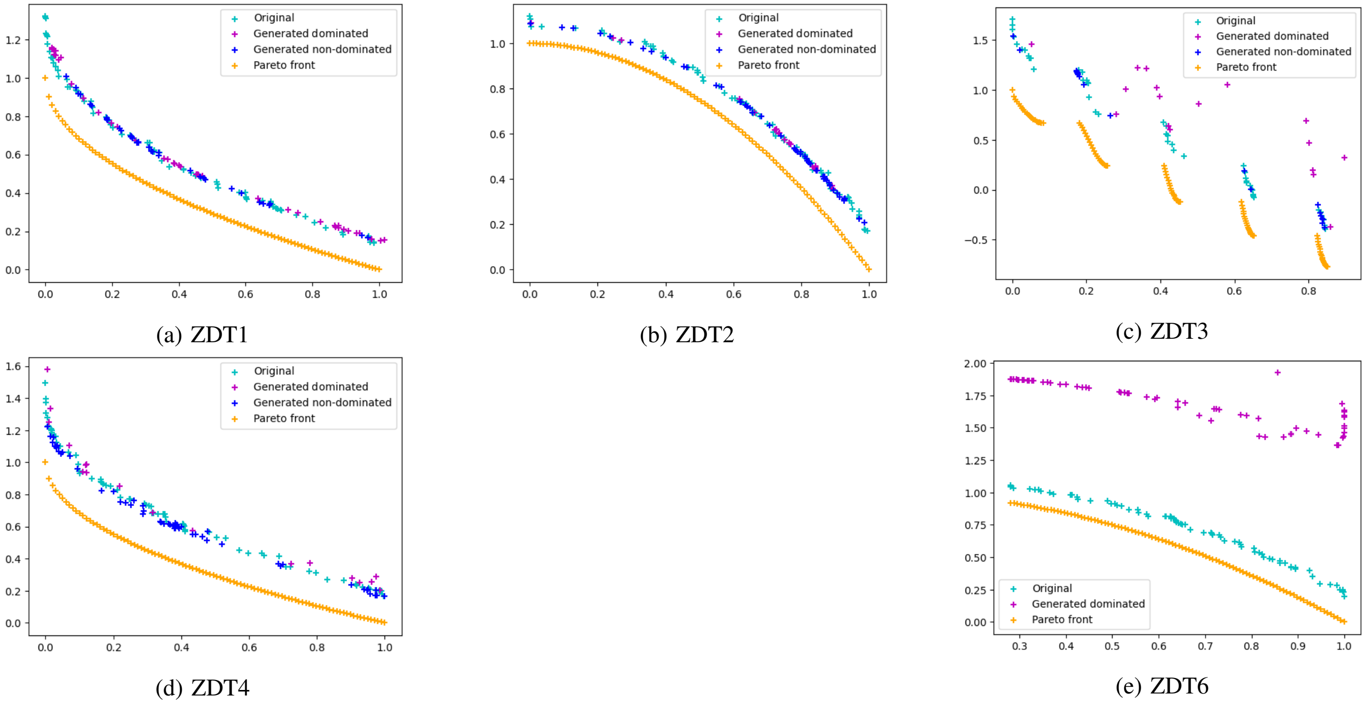
(b) ZDT2

(c) ZDT3

(d) ZDT4

(e) ZDT6

Fig. 6: Visual Comparison between the Pareto front, original population and generated individuals using a GAN divided by dominated and non-dominated solutions.



(a) ZDT1

(b) ZDT2

(c) ZDT3

(d) ZDT4

(e) ZDT6

Fig. 7: Visual Comparison between the Pareto front, original population and generated individuals using a DAN divided by dominated and non-dominated solutions.

adaptation evolution strategies," *Evolutionary computation*, vol. 28, no. 3, pp. 405–435, 2020.

[18] L. Faury, C. Calauzenes, O. Fercoq, and S. Krichen, "Improving evolutionary strategies with generative neural networks," *arXiv preprint arXiv:1901.11271*, 2019.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[20] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE signal processing magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[21] A. Bulat and G. Tzimiropoulos, "Super-fan: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with gans," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 109–117.

[22] T. Zhang, "Deepfake generation and detection, a survey," *Multimedia Tools and Applications*, vol. 81, no. 5, pp. 6259–6276, 2022.

[23] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, p. 100258, 2022.

[24] A. Dash, J. Ye, and G. Wang, "A review of generative adversarial networks (gans) and its applications in a wide variety of disciplines: From medical to remote sensing," *IEEE Access*, 2023.

[25] H. Chen, "Challenges and corresponding solutions of generative adversarial networks (gans): A survey study," *Journal of Physics: Conference Series*, vol. 1827, no. 1, p. 012066, mar 2021. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/1827/1/012066

[26] A. Smola, A. Gretton, L. Song, and B. Scholkopf, "A hilbert space embedding for distributions," in *IFIP Working Conference on Database Semantics*, 2007.

[27] J. Blank and K. Deb, "Pymoo: Multi-objective optimization in python," *Ieee access*, vol. 8, pp. 89 497–89 509, 2020.

[28] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[29] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[30] S. Zapotecas-Martínez, C. A. C. Coello, H. E. Aguirre, and K. Tanaka, "Challenging test problems for multi-and many-objective optimization," *Swarm and Evolutionary Computation*, vol. 81, p. 101350, 2023.

[31] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000.

[32] C. Audet, J. Bigeon, D. Cartier, S. Le Digabel, and L. Salomon, "Performance indicators in multiobjective optimization," *European Journal of Operational Research*, vol. 292, no. 2, pp. 397–422, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221720309620

[33] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, jul 2012.

[34] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *Evolutionary Multi-Criterion Optimization: 8th International Conference, EMO 2015, Guimarães, Portugal, March 29–April 1, 2015. Proceedings, Part II 8*. Springer, 2015, pp. 110–125.

[35] J. G. Falcón-Cardona, E. Covantes Osuna, and C. A. Coello Coello, "An overview of pair-potential functions for multi-objective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2021, pp. 401–412.

[36] S. V. Borodachov, D. P. Hardin, and E. B. Saff, *Discrete energy on rectifiable sets*. Springer, 2019.