Inverse Multiobjective Optimization by Generative Model Prompting

Jiao Liu*, Abhishek Gupta†, Yew-Soon Ong*, Puay Siew Tan‡
*College of Computing & Data Science, Nanyang Technological University (NTU), Singapore
†School of Mechanical Sciences, Indian Institute of Technology, Goa, India.
‡ Singapore Institute of Manufacturing Technology (SIMTech), A*STAR, Singapore
jiao.liu@ntu.edu.sg, abhishekgupta@iitgoa.ac.in, asysong@ntu.edu.sg, and pstan@simtech.a-star.edu.sg

***Abstract*—The integration of multiobjective optimizers with inverse models—that map points on the Pareto front to corresponding nondominated solutions—has drawn attention. These inverse models serve a dual purpose, not only facilitating the generation of candidate solutions during the optimization process, but also offering insights for multiobjective decision-making upon completion of optimization. However, today's inverse models mainly serve to capture one-to-one mapping relations, restricting them to learn only from nondominated solution samples. As a result, the information embedded in dominated samples is not fully utilized. In this paper, we introduce a novel approach of building conditional *inverse generative models* (invGMs) from optimization data, making the most of both nondominated and dominated solution samples during training. Different from standard inverse models, decision-makers can query such invGMs with prompts expressed in the form of *any* desired objective function values, leading them to produce a corresponding solution. Through iterative prompting, invGMs are shown to accelerate the creation of diverse sets of high-quality solutions even during the course of multiobjective optimization runs. Empirical studies on three industrial optimization problems highlight the proposed method's faster convergence rate and improved inverse modeling accuracy.**

***Index Terms*—Multiobjective optimization optimization, conditional generative models, inverse models.**

I. Introduction

Multiobjective optimization problems (MOPs) refer to optimization problems with multiple conflicting objectives to be optimized simultaneously [1]. Such problems are pervasive in various industrial fields, including engineering design [2] and manufacturing operations [3]. Unlike standard single-objective optimization problems, a distinctive characteristic of MOPs is that they yield a Pareto set (PS) of trade-off optimal solutions, known as the *Pareto optimal solutions* or *nondominated solutions*, instead of a single optimum [4]. The central goal in solving MOPs is to identify a well-distributed set of optimized solutions that map to the Pareto front (PF), representing the optimal performance trade-offs in the objective space. Subsequently, decision-makers (DMs) can scrutinize the performance trade-offs and select the most preferable solution(s) from the obtained results.

Recently, there has been a growing interest in the integration of *inverse models*—models that map points on the PF back to the corresponding nondominated solutions in the decision space—into the multiobjective optimization process [5]–[9]. These inverse models serve a dual purpose, not only facilitating the generation of candidate solutions during the online optimization process to enhance convergence [7], [10] but also offering insights for on-demand multiobjective decision-making upon completion of the optimization [11]. Commonly, inverse models are assumed to capture one-to-one mappings from the PF to PS. This assumption is based on the Karush–Kuhn–Tucker conditions, implying that both the PS and the PF are $(m-1)$-dimensional piecewise manifolds for continuous decision spaces [7]. However, this assumption confines inverse models to be trained solely based on nondominated samples, limiting the complete utilization of information embedded in dominated samples.

In response, this paper introduces an innovative approach that leverages conditional *inverse generative models* (invGM)—utilizing generative models conditioned by the objective function values as the inverse models—to enhance multiobjective optimization, denoted as invGM-MO. The noteworthy advantage of this strategy lies in that invGM can make the most use of both nondominated and dominated solution samples during training, thus resulting in more accurate inverse models. Moreover, different from standard inverse models, decision-makers can query such invGMs with prompts expressed in the form of *any* desired objective function values, leading them to produce corresponding solutions. This makes it possible to accelerate the creation of diverse sets of high-quality solutions even during the course of multiobjective optimization runs through iterative prompting the invGM. It is noteworthy that, while some studies have introduced the use of conditional generative models to assist in multiobjective optimization [12]–[14], invGM-MO stands out by utilizing invGM to achieve both improved convergence and on-demand multiobjective decision-making.

The remainder of the paper is organized as follows. Section II introduces the basic concepts of MOPs. The proposed method is elucidated in Section III. Section IV presents the experimental studies, and Section V concludes the paper.

II. Multiobjective Optimization Problems

Commonly, a MOP can be formulated as:

$$\begin{aligned} \min : & \ \mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})\} \\ \text{s.t. } & \mathbf{x} \in \mathcal{X} \end{aligned} \tag{1}$$

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00142

Algorithm 1 invGM-MO

Input: Target MOP objective functions $\mathbf{f}(\mathbf{x})$; decision space $\mathcal{X}$; *evaluation budget*;

Output: The invGM and the obtained nondominated solutions;

1: Sample N_{init} solutions $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N_{init})}$ in $\mathcal{X}$;
2: $\mathcal{D} \leftarrow \{(\mathbf{x}^{(l)}, \mathbf{y}^{(l)})\}_{l=1}^{N=N_{init}}$, where $\mathbf{y}^{(l)} = \mathbf{f}(\mathbf{x}^{(l)})$;
3: **while** $N <$ *evaluation budget* **do**
4: Build the invGM $p(\mathbf{x}|\mathbf{y})$ based on $\mathcal{D}$, where the objective function values are considered as the prompts;
5: Select the best N^q samples from $\mathcal{D}$ according to the nondominated sorting, and record their objective function vectors as $\mathcal{Y}_{nd} = \{\mathbf{y}_{nd}^{(1)}, \dots, \mathbf{y}_{nd}^{(N^q)}\}$;
6: Generate the prompts $\{\mathbf{y}_{prompt}^{(1)}, \cdots, \mathbf{y}_{prompt}^{(N^{(q)})}\}$ according to equation (2);
7: Sample N^q candidate solutions $\{\mathbf{x}_{cand}^{(1)}, \dots, \mathbf{x}_{cand}^{(N^q)}\}$ from $p(\mathbf{x}|\mathbf{y})$ prompted by $\{\mathbf{y}_{prompt}^{(1)}, \dots, \mathbf{y}_{prompt}^{(N^{(q)})}\}$;
8: Evaluate all of the candidate solutions and update $\mathcal{D}$;
9: $N \leftarrow N + 1$;
10: **end while**

where $f_i(\mathbf{x}), (i \in \{1, \dots, m\})$ is the ith objective function, m is the number of objectives, $\mathbf{x}$ is the decision vector, $\mathcal{X} = \{\mathbf{x} = (x_1, \dots, x_d) | L_j \leq x_j \leq U_j,\ j = 1, \dots, d\}$ is the decision space, d is the dimensions of the decision vector, and L_j and U_j are the lower and upper bounds of the jth decision variable x_j. Some key concepts associated with the MOP are introduced as follows [15]:

- *Pareto Dominance*: For decision vectors $\mathbf{x}_a$ and $\mathbf{x}_b$, if $\forall i \in \{1, 2, \dots, m\}$, $f_i(\mathbf{x}_a) \leq f_i(\mathbf{x}_b)$ and $\exists i' \in \{1, 2, \dots, m\}$, $f_{i'}(\mathbf{x}_a) < f_{i'}(\mathbf{x}_b)$, $\mathbf{x}_a$ is said to Pareto dominate $\mathbf{x}_b$.
- *Pareto Optimal Solution*: If no decision vector in $\mathcal{X}$ Pareto dominates $\mathbf{x}_a$, then $\mathbf{x}_a$ is a Pareto optimal solution.
- *Pareto Set (PS)*: The set of all Pareto-optimal solutions forms the PS in decision space.
- *Pareto Front (PF)*: The image of the PS in the objective space forms the PF.

III. Proposed Method

The steps of invGM-MO are shown in **Algorithm 1** and explained in what follows.

- **Data initialization** (steps 1-3): We randomly generate N_{init} candidate solutions within the decision space. These generated solutions are then evaluated using the objectives of the solving MOP, forming an initial dataset denoted as $\mathcal{D} = \{(\mathbf{x}^{(l)}, \mathbf{y}^{(l)})\}_{l=1}^{N_{init}}$.
- **Building the conditional generative model** (steps 5): An invGM, denoted as $p(\mathbf{x}|\mathbf{y})$, is built based on the dataset $\mathcal{D}$, where the objective function values are considered as the prompts.
- **Solutions generation** (steps 6-10): During this process, a set of candidate solutions is generated based on $p(\mathbf{x}|\mathbf{y})$:
  1) Do nondominated sorting [16] for all of the samples in $\mathcal{D}$ and record the objective function values of the best N^q samples as $\mathcal{Y}_{nd} = \{\mathbf{y}_{nd}^{(1)}, \dots, \mathbf{y}_{nd}^{(N^q)}\}$
  2) Generate a set of desired objective function values using the following equation for prompting the invGM:
  $$\begin{aligned} \mathbf{y}_{prompt}^{(l)} &= \mathbf{y}_{nd}^{(l)} - |\boldsymbol{\sigma}_{nd}^{(l)} \cdot \mathbf{s}^{(l)}|, \\ l &\in \{1, \dots, N^q\}, \end{aligned} \tag{2}$$
  where $\mathbf{s}^{(l)}$ is a random value generated according to the standard Gaussian distribution and $\boldsymbol{\sigma}_{nd}^{(l)}$ is the standard deviation corresponding to $\mathbf{y}_{nd}^{(l)}$. In this paper, $\boldsymbol{\sigma}_{nd}^{(l)}$ is adaptively set to the standard deviation of the five vectors closest to $\mathbf{y}_{nd}^{(l)}$ in $\mathcal{Y}_{nd}$.
  3) Sample N^q candidate solutions $\{\mathbf{x}_{cand}^{(1)}, \dots, \mathbf{x}_{cand}^{(N^q)}\}$ from $p(\mathbf{x}|\mathbf{y})$ by employing $\{\mathbf{y}_{prompt}^{(1)}, \dots, \mathbf{y}_{prompt}^{(N^{(q)})}\}$ to prompt the invGM.

  Subsequently, the generated candidate solutions undergo evaluation using the objective functions of the solving MOP and are appended to the dataset $\mathcal{D}$.

Steps 5-10 are repeated until a prescribed evaluation budget is exhausted. Once the termination condition is met, the nondominated solutions in the acquired dataset $\mathcal{D}$ and the invGM are returned to the user for assessment.

IV. Experimental Studies

A. Implementation Details

In our implementation, we utilize the conditional variational autoencoder (CVAE) [17] as the generative model due to its simplicity, which helps reduce training time and computational resource consumption. The generative process of CVAE begins with generating a set of latent variables, $\mathbf{z}$, from a prior distribution, $p(\mathbf{z}|\mathbf{y})$. Subsequently, data $\mathbf{x}$ in the decision space is generated, conditioned on both $\mathbf{z}$ and $\mathbf{y}$ using the decoder $p(\mathbf{x}|\mathbf{z}, \mathbf{y})$. During training, we employ the stochastic gradient variational Bayes framework [18], using the variational lower bound of the log-likelihood as the loss function:

$$\log p(\mathbf{x}|\mathbf{y}) \geq -KL\left(q(\mathbf{z}|\mathbf{x}, \mathbf{y}) || p(\mathbf{z}|\mathbf{y})\right) + \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p(\mathbf{x}|\mathbf{z}, \mathbf{y})\right]. \tag{3}$$

In (3), $KL(\cdot||\cdot)$ represents the Kullback-Leibler divergence, and $q(\mathbf{z}|\mathbf{x}, \mathbf{y})$ denotes the encoder introduced to approximate the true posterior. The parameters of both the encoder and the decoder are estimated by optimizing the variational lower bound. The value of the variational lower bound is estimated using Monte Carlo based on the dataset $\mathcal{D}$.

In our experiments, both N^{init} and N^q are set to 50, and the evolution budget is set to 500. This evaluation budget is notably lower than that of typical multiobjective optimization algorithms, highlighting the efficient convergence of the proposed invGM-MO approach. This demonstrates invGM-MO's potential for managing even computationally expensive objectives. In terms of the CVAE, we assume the predictions of both the encoder and the decoder are Gaussian distributions, set the dimension of the latent vector $\mathbf{z}$ to 10, and design the

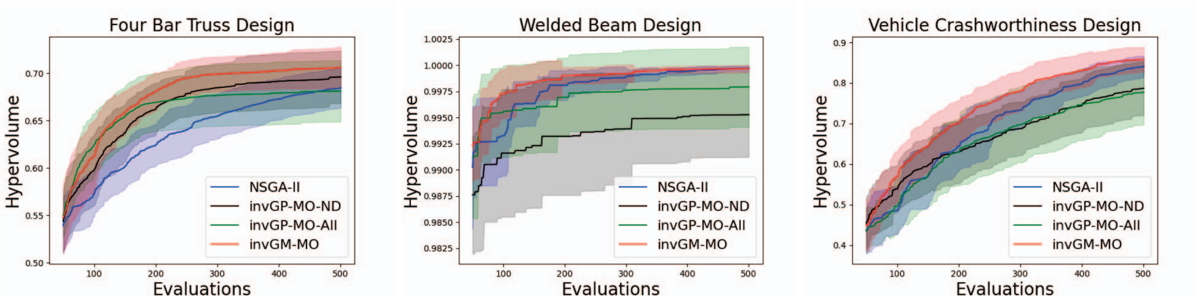

Fig. 1. Comparison of HV convergence trends averaged over 20 independent runs of NSGA-II, invGP-MO, invGP-MO-All, and invGM-MO. Shaded areas represent one standard deviation on either side of the mean. (a) Four bar truss design. (b) Welded beam design. (c) Vehicle crashworthiness design.

TABLE I
INVERSE MODEL RMSE RESULTS OF INVGP-MO AND INVGM-MO.

Optimization Tasks	invGP-MO-ND	invGP-MO-All	invGM-MO
	Average RMSE±Std	Average RMSE±Std	Average RMSE±Std
Four Bar Truss Design	0.0845±0.0285	0.1160±0.0581	**0.0834±0.0193**
Welded Beam Design	0.3142±0.0776	0.1195±0.0353	**0.2045±0.0434**
Vehicle Crashworthiness Design	0.0725±0.0564	0.7377±0.2055	**0.0388±0.0296**

structures of the encoder and the decoder follows: "($\mathbf{x}$, $\mathbf{y}$) - 30 - 30 - $(\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z)$" and "($\mathbf{z}$, $\mathbf{y}$) - 30 - 30 - $(\boldsymbol{\mu}_x, \boldsymbol{\sigma}_x)$". Here, $\boldsymbol{\mu}_z$ and $\boldsymbol{\sigma}_z$ are the mean and the standard deviation of the predicted Gaussian distribution of the encoder, and $\boldsymbol{\mu}_x$ and $\boldsymbol{\sigma}_x$ are the mean and the standard deviation of the prediction of the decoder.

To evaluate the performance of invGM-MO, three industrial design problems summarized in [19] are employed to assess the performance of the algorithms, i.e., the four-bar truss design, the welded beam design, and the vehicle crashworthiness design. In terms of the evaluation metric, we assess the quality of the nondominated solutions provided by the algorithms using the hypervolume (HV) [20]. Note that before calculating the HV, we normalize the objective function values of all samples in $\mathcal{D}$ into the region $[0, 1]$. Then, each component of the reference point is set to 1. Moreover, to measure the quality of the inverse modeling, we employ a set of nondominated solutions as the test set $\mathcal{D}^{test} = \{(\mathbf{x}_{test}^{(r)}, \mathbf{y}_{test}^{(r)})\}_{r=1}^{N^{test}}$ to calculate the root mean square error (RMSE), i.e.,

$$RMSE = \sqrt{\frac{\sum_{r=1}^{N^{test}} ||\mathbf{x}_{test}^{(r)} - \mathbf{x}_{pred}^{(r)}||_2^2}{N^{test}}}, \tag{4}$$

where $\mathbf{x}_{pred}^r$ is the prediction of the generative model when $\mathbf{y}_{test}^{(r)}$ is given. Note that, the predicted mean of the decoder is taken as the prediction of $\mathbf{x}_{pred}^r$. Here, $\mathcal{D}^{test}$ is obtained by using NSGA-II [16] with the population size of 200 and the evaluation budget is set to 100000. Such a number of population size and evaluation budget is enough for NSGA-II to provide high quality near-optimal solutions, thus supporting the calculation of the RMSE.

B. Experimental Results

We conduct an empirical evaluation of the invGM-MO algorithm against three other algorithms: NSGA-II, invGP-MO-ND, and invGP-MO-All. NSGA-II is a widely recognized evolutionary multiobjective optimization algorithm and serves as a baseline for our comparison. The invGP-MO-ND is a variant of invGM-MO that utilizes the Gaussian process model [21] for inverse modeling, inspired by [7]. The model is implemented using only the discovered nondominated solutions in each iteration, capturing a one-to-one mapping. The invGP-MO-All is another variant that verifies the effectiveness of invGM-MO. It uses a Gaussian process for inverse modeling, but builds the model using both dominated and nondominated solutions. This approach tests whether directly approximating the non-one-to-one mapping using all data yields good results.

We present the convergence trends of the hypervolume (HV) in Fig.1 and summarize the RMSE results in TableI. The trends shown in Fig. 1 highlight a substantially faster convergence for invGM-MO compared to NSGA-II, invGP-MO-ND, and invGP-MO-All. This convergence trend underscores the efficacy of the proposed approach in accelerating the convergence rate. Moreover, considering the RMSE results, invGM-MO surpasses invGP-MO-ND and invGP-MO-All, indicating its superior ability to harness the information within the dataset $\mathcal{D}$. These findings collectively underscore the effectiveness of invGM-MO.

V. CONCLUSION

The incorporation of inverse models into the multiobjective optimization process not only shows promise in improving algorithmic convergence but also streamlines on-demand multiobjective decision-making with straightforward queries. However, traditional inverse models are assumed to capture one-to-one mappings, relying solely on nondominated samples for training. This study introduces invGM, which harnesses information from both nondominated and dominated samples, seamlessly integrating it into the multiobjective optimization framework. Our findings reveal that decision-makers can effectively query invGM with prompts specifying any desired objective function values, leading to the generation of corresponding solutions. Moreover, through iterative prompting, invGMs accelerate the generation of diverse sets of high-quality solutions even during ongoing multiobjective optimization runs. Empirical studies on three industrial optimization problems highlight the proposed method's superior convergence speed and improved inverse modeling accuracy.

Acknowledgment

This research is partly supported by the Distributed Smart Value Chain programme which is funded in part by the Singapore RIE2025 Manufacturing, Trade and Connectivity (MTC) Industry Alignment Fund-Pre-Positioning (Award No: M23L4a0001), the Ramanujan Fellowship from the Science and Engineering Research Board, Government of India (Grant No. RJF/2022/000115), and the College of Computing & Data Science, Nanyang Technological University.

References

[1] J. Branke, *Multiobjective optimization: Interactive and evolutionary approaches*. Springer Science & Business Media, 2008, vol. 5252.

[2] A. Osyczka, "An approach to multicriterion optimization problems for engineering design," *Computer Methods in Applied Mechanics and Engineering*, vol. 15, no. 3, pp. 309–333, 1978.

[3] J.-q. Li, H.-y. Sang, Y.-y. Han, C.-g. Wang, and K.-z. Gao, "Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions," *Journal of Cleaner Production*, vol. 181, pp. 584–598, 2018.

[4] M. Ehrgott, *Multicriteria optimization*. Springer Science & Business Media, 2005, vol. 491.

[5] D. Lim, Y.-S. Ong, and B.-S. Lee, "Inverse multi-objective robust evolutionary design optimization in the presence of uncertainty," in *Proceedings of the 7th annual workshop on genetic and evolutionary computation*, 2005, pp. 55–62.

[6] D. Lim, Y.-S. Ong, Y. Jin, B. Sendhoff, and B. S. Lee, "Inverse multi-objective robust evolutionary design," *Genetic Programming and Evolvable Machines*, vol. 7, pp. 383–404, 2006.

[7] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using gaussian process-based inverse modeling," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 838–856, 2015.

[8] A. Gupta, Y.-S. Ong, M. Shakeri, X. Chi, and A. Z. NengSheng, "The blessing of dimensionality in many-objective search: An inverse machine learning insight," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 3896–3902.

[9] J. Liu, A. Gupta, and Y.-S. Ong, "Inverse transfer multiobjective optimization," *arXiv preprint arXiv:2312.14713*, 2023.

[10] X. Lin, Z. Yang, X. Zhang, and Q. Zhang, "Pareto set learning for expensive multi-objective optimization," *Advances in Neural Information Processing Systems*, vol. 35, pp. 19 231–19 247, 2022.

[11] C. S. Tan, A. Gupta, Y.-S. Ong, M. Pratama, P. S. Tan, and S. K. Lam, "Pareto optimization with small data by learning across common objective spaces," *Scientific Reports*, vol. 13, no. 1, p. 7842, 2023.

[12] A. Suresh, D. Shah, A. S. Admasu, D. Upadhyay, and K. Deb, "Multi-objective evolutionary design of microstructures using diffusion autoencoders," in *AI for Accelerated Materials Design-NeurIPS 2023 Workshop*, 2023.

[13] P. R. Kalehbasti, M. D. Lepech, and S. S. Pandher, "Augmenting high-dimensional nonlinear optimization with conditional gans," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2021, pp. 1879–1880.

[14] U. Garciarena, R. Santana, and A. Mendiburu, "Expanding variational autoencoders for learning and exploiting latent representations in search distributions," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018, pp. 849–856.

[15] K. Deb, *Multi-objective optimisation using evolutionary algorithms: an introduction*. Springer, 2011.

[16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[17] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, 2015.

[18] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[19] R. Tanabe and H. Ishibuchi, "An easy-to-use real-world multi-objective optimization problem suite," *Applied Soft Computing*, vol. 89, p. 106078, 2020.

[20] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications," *Theoretical Computer Science*, vol. 425, pp. 75–103, 2012.

[21] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.