Bayesian Forward-Inverse Transfer for Multiobjective Optimization

Tingyang Wei[1[0000−0001−6848−031X]], Jiao Liu[1[0009−0003−3690−7923]], Abhishek Gupta[2[0000−0002−6080−855X]], Puay Siew Tan[3], and Yew-Soon Ong[1,4[0000−0002−4480−169X]]

[1] College of Computing and Data Science, Nanyang Technological University, Singapore
TINGYANG001@e.ntu.edu.sg, {jiao.liu, asysong}@ntu.edu.sg
[2] School of Mechanical Sciences, Indian Institute of Technology, Goa, India
abhishekgupta@iitgoa.ac.in
[3] Singapore Institute of Manufacturing Technology (SIMTech), Agency for Science, Technology and Research, Singapore
pstan@simtech.a-star.edu.sg
[4] Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research, Singapore

Abstract. We present an evolutionary optimizer incorporating knowledge transfer through forward and inverse surrogate models for solving multiobjective problems, within a stringent computational budget. Forward knowledge transfer is employed to fully exploit solution-evaluation datasets from related tasks by building Bayesian forward multitask surrogate models that map points from decision to objective space. Inverse knowledge transfer via Bayesian inverse multitask models makes possible the creation of high-quality solution populations in decision space by mapping back from preferred points in objective space. In contrast to prior work, the proposed method can improve the overall convergence performance to multiple Pareto sets by fully exploiting information available for diverse multiobjective problems. Empirical studies conducted on benchmark and real-world multitask multiobjective optimization problems demonstrate the faster convergence rate and enhanced inverse modeling accuracy of our algorithm compared to state-of-the-art algorithms.

Keywords: Evolutionary algorithms · Bayesian optimization · multiobjective optimization · inverse models

1 Introduction

In various real-world scenarios, decision-makers (DMs) confront the challenges of simultaneously considering multiple conflicting criteria when choosing a specific optimal solution [27, 33, 34]. Such problems are commonly formulated as multiobjective optimization problems (MOPs) [4]. The defining characteristic of MOPs is the absence of a single solution that universally outperforms others across all criteria. Thus, the primary aim in addressing an MOP is to identify

Pareto optimal solutions, mapping to the Pareto front (PF) of optimal performance trade-offs in the objective space. Evolutionary algorithms (EAs), with the inherent population-based nature, are well suited for MOPs [9], through evolving and maintaining a solution set iteratively. The obtained solution set can approximate the PF with well-distributed solutions, enabling DMs to assess performance trade-offs and subsequently select the preferred solution(s) *a posteriori* [29, 39].

While EAs have effectively addressed MOPs, a notable drawback is their data-hungry nature, necessitating massive evaluations to converge upon a set of nondominated solutions. This characteristic renders EAs unsuitable to directly tackle expensive MOPs, especially those demanding time-consuming computer simulations or intricate real-world experiments. To overcome this challenge, leveraging inexpensive surrogate models can aid algorithmic evolution, reducing the budget for expensive evaluations. Notable algorithms in this domain include ParEGO [28], MOEA/D-EGO [48], EHVI [15], and SMS-EGO [37]. However, these algorithms still require a sufficient number of evaluated samples to construct informative surrogate models, thus necessitating a generous function evaluation budget. This necessity has been termed as the *cold start* issue.

Integrating knowledge transfer capabilities into algorithms has emerged as a promising strategy to address cold starts. By allowing algorithms to learn from optimization experiences across related tasks, they often exhibit faster convergence [10, 21]. Notably, transfer evolutionary optimization and evolutionary multitasking have attracted significant attention as effective approaches in this regard [2, 17, 21, 45]. These methodologies recognize that real-world problems rarely exist in isolation, allowing knowledge from various tasks to be adaptively reused, thereby enhancing convergence. The efficacy of these algorithms is heavily influenced by the strategy employed for knowledge transfer. Over the past decades, several knowledge transfer techniques have been proposed and incorporated into the EA to enhance performance. This paper focuses on a recent approach termed *inverse transfer*, which has shown promise in addressing expensive MOPs [32]. This technique leverages Bayesian inverse models that map points on the PF back to their corresponding nondominated solutions in the decision space to achieve knowledge transfer. For instance, Liu et al. [32] introduced invTrEMO, a novel transfer optimization algorithm showcasing the capability to expedite convergence by harnessing implicit knowledge inherent in the source data. Moreover, invTrEMO generates accurate inverse models as a byproduct, facilitating the generation of nondominated solutions tailored to user preferences on demand. However, it is crucial to note that Bayesian inverse models typically assume one-to-one mappings from the PF to Pareto optimal solutions, grounded in the Karush–Kuhn–Tucker conditions [7]. This assumption constrains Bayesian inverse models to be trained solely based on nondominated samples, limiting the utilization of the information embedded in dominated samples across different tasks, and potentially impeding the overall efficacy of the approach.

To overcome the limitations of invTrEMO, we introduce a novel extension, *forward-inverse transfer evolutionary multiobjective optimizer* (F-invTrEMO).

This enhanced method integrates both forward and inverse transfer techniques to maximize information utilization. By combining these two approaches, F-invTrEMO harnesses the strengths of both forward and inverse transfer, enabling more comprehensive knowledge transfer across multiple optimization tasks. The forward transfer component leverages all evaluated samples to capture correlations in function landscapes across all task domains, enhancing the accuracy of Bayesian forward models to effectively guide the optimization process. Meanwhile, the inverse transfer aspect collaboratively utilizes nondominated samples across related tasks, enabling the Bayesian inverse models to map preference vectors from the objective space to the search distribution [24,40]. This distribution facilitates the generation of promising offspring solutions by harnessing knowledge from multiple tasks, thus guiding the evolutionary process. This forward-inverse transfer mechanism can hopefully upgrade the ability to exploit inter-task information across related MOPs [25] including composites manufacturing [23], path planning [1], model compression [8] and hyperspectral unmixing [46].

In this paper, we primarily focus on addressing the more challenging multitask multiobjective optimization problems with F-invTrEMO, rather than the sequential transfer optimization problems [22]. This broader focus aims to further validate the effectiveness of the proposed method. The main contributions of this paper can be summarized as follows:

- We present a novel algorithm named F-invTrEMO. A distinctive feature of F-invTrEMO is its incorporation of both forward and inverse transfer techniques to steer the optimization process. This approach allows F-invTrEMO to capitalize on information from both dominated and nondominated evaluated samples across diverse tasks, thereby maximizing the utilization of available data for improved optimization outcomes.
- We demonstrate that integrating both forward and inverse transfer mechanisms not only accelerates convergence but also yields more precise Bayesian inverse models for approximating the Pareto fronts of individual tasks.
- We validate the superiority of our method across a spectrum of multitask optimization problems, encompassing benchmark problems, a set of engineering optimization problems, and hyperparameter optimization problems.

The remainder of this paper is organized as follows. Section 2 introduces the basic concepts and the related work. Section 3 introduces the proposed F-invTrEMO framework. The effectiveness of the proposed framework is justified by experimental analysis in Section 4, and Section 5 concludes the paper.

2 Preliminaries

2.1 Multitask Multiobjective Optimization

Considering the synergies inherent in MOPs, multitask multiobjective optimization concurrently solves a set of MOPs by leveraging information across these problem domains [16,22]. Let $f_{k,i}(\cdot)$, where $k \in \{1, \ldots, K\}$ and $i \in \{1, \ldots, m\}$, be the ith objective on the kth optimization task, without loss of generality, the multitask MOP is defined as follows:

$$\begin{aligned} &\forall T_k, k \in \{1, \ldots, K\}, \\ &\min : \mathbf{F}_k(\mathbf{x}_k) = (f_{k,1}(\mathbf{x}_k), \ldots, f_{k,m}(\mathbf{x}_k)), \quad \text{s.t. } \mathbf{x}_k \in \Omega_k \subset \mathbb{R}^d \end{aligned} \tag{1}$$

where T_k denotes the kth task setting, $\mathbf{F}_k(\cdot)$ represents the objective function vector corresponding to the kth task setting, $\mathbf{x}_k = (x_{k,1}, \ldots, x_{k,d})$ denotes the solution corresponding to the kth task setting, and Ω_k signifies the decision space corresponding to the kth task setting. In this paper, we assume that each objective function is expensive to evaluate, imposing a constraint on the evaluation budget allocated for each task. The key concepts [4] associated with the formulation in (1) are explained as follows:

- (**Pareto Dominance**) Solution $\mathbf{x}_k^{(a)}$ is said to Pareto dominate another solution $\mathbf{x}_k^{(b)}$ on the kth task setting, if $\forall i \in \{1, 2, \ldots, m\}$, $f_{k,i}(\mathbf{x}_k^{(a)}) \leq f_{k,i}(\mathbf{x}_k^{(b)})$ and $\exists i' \in \{1, 2, \ldots, m\}$ such that $f_{k,i'}(\mathbf{x}_k^{(a)}) < f_{k,i'}(\mathbf{x}_k^{(b)})$.
- (**Pareto Optimal Solutions**) Solution $\mathbf{x}_k^*$ is said to be Pareto optimal on the kth task setting if there are no other candidate solutions that can dominate $\mathbf{x}_k^*$.
- (**Pareto Set**) Pareto set (PS) consists of all the Pareto optimal solutions.
- (**Pareto Front**) The image of the Pareto set in the objective space is referred to as the Pareto front (PF).

The result of (1) can be represented as a set of Pareto optimal solution sets:

$$PS_k = \{\mathbf{x}_k^{(1)*}, \mathbf{x}_k^{(2)*}, \mathbf{x}_k^{(3)*}, \ldots\}, \quad \mathcal{S} = \bigcup_{k=1}^{K} PS_k, \tag{2}$$

where $\mathbf{x}_k^{(\cdot)*}$ is a Pareto optimal solution of the kth task, PS_k is a Pareto optimal solution set of the kth task setting, and $\mathcal{S}$ is the optimal set of solution sets.

2.2 Decomposition-Based Multiobjective Optimization

The fundamental concept behind decomposition-based multiobjective optimization is to decompose a MOP into a series of single-objective subproblems through the utilization of a specific scalarization function [47]. By solving these subproblems individually, a finite set of Pareto optimal solutions can be obtained. In this paper, we concentrate on augmented Tchebycheff scalarization [28, 30], selected for its simplicity and suitability for non-convex Pareto fronts. This technique combines objective functions using the following expression, which remains consistent across tasks:

$$f^{tch}(\mathbf{x}|\mathbf{w}) = \max_{1 \leq i \leq m} \{w_i \cdot (f_i(\mathbf{x}) - (z_i^* - \epsilon))\} + \rho \sum_{i=1}^{M} w_i f_i(\mathbf{x}), \tag{3}$$

where $f_i(\mathbf{x})$ is the ith objective of a specific MOP, $\mathbf{w}$ is known as the *preference vector* sampled from a $(m-1)$ dimensional simplex (i.e., $\mathbf{w} \in \mathcal{W} = \{(w_1, w_2, ..., w_m)^T | \sum_{i=1}^{m} w_i = 1, w_i \in [0,1]\}$), z_i^* is the ideal objective function value (i.e., minimal possible value) for the ith objective, the coupled term $(z_i^* - \epsilon)$ is the utopia objective value by maintaining ϵ as a small positive value for the ith objective, and ρ is a sufficiently small scalar weight.

2.3 Gaussian Process

To address expensive objective functions, Gaussian Processes (GPs) [38] are commonly employed as the surrogate to facilitate a data-efficient problem-solving process. This method has been widely applied in various multiobjective optimization techniques, including ParEGO [28], MOEA/D-EGO [48], EHVI [15], and SMS-EGO [37]. In this paper, we also opt for GP as the surrogate model, where we assume $f \sim \mathcal{GP}(\mu, \kappa(\mathbf{x}, \mathbf{x}'))$, where μ represents the mean function with a zero prior, and κ is a kernel function that encapsulates the correlation between input data points. Given sampled data $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ with noise $\epsilon^{(i)} \sim \mathcal{N}(0, \sigma_\epsilon^2)$, the predicted posterior distribution of the GP model, $\mathcal{N}(\mu(\mathbf{x}^{(*)}), \sigma^2(\mathbf{x}^{(*)}))$, for a query $\mathbf{x}^{(*)}$ can be computed as follows:

$$\mu(\mathbf{x}^{(*)}) = \mathbf{k}_*^\intercal(\boldsymbol{\Sigma} + \sigma_\epsilon^2 \mathbf{I}_N)^{-1}\mathbf{y}, \tag{4}$$

$$\sigma(\mathbf{x}^{(*)}) = \kappa_{**} - \mathbf{k}_*^\intercal(\boldsymbol{\Sigma} + \sigma_\epsilon^2 \mathbf{I}_N)^{-1}\mathbf{k}_*. \tag{5}$$

where $\kappa_{**} = \kappa(\mathbf{x}^{(*)}, \mathbf{x}^{(*)})$, $\mathbf{k}_*$ is the kernel vector between $\mathbf{x}^{(*)}$ and the data in $\mathcal{D}$, $\boldsymbol{\Sigma}$ is an $N \times N$ matrix with elements $\Sigma_{p,q} = \kappa(\mathbf{x}^{(p)}, \mathbf{x}^{(q)})$, $p, q \in \{1, \dots, N\}$, $\mathbf{I}_N$ is a $N \times N$ identity matrix, and $\mathbf{y}$ is the vector of the noisy observations.

2.4 Inverse Modeling in multiobjective Optimization

Inverse modeling has emerged as a powerful tool in the domain of multiobjective optimization, offering innovative strategies for navigating optimization and enabling DMs to generate preferred trade-off solutions on demand. Under the premise that the Karush–Kuhn–Tucker conditions hold, PS and PF are both $(m-1)$-dimensional piecewise manifolds under mild conditions [14], enabling inverse models to facilitate the mapping of solutions from the objective space back into the decision space. It is noteworthy that the input for inverse models can encompass not only objective function values [7, 24] but also preference vectors [20, 30, 31]. In this paper, we specifically explore the mapping from preference vectors to decision variables. This approach allows DMs to articulate their preferences more easily without delving deeply into the intricacies of the PF topology [40]. GP models are utilized to map a preference vector $\mathbf{w}$ to the jth decision variable as $\Psi_j^{-1}(\mathbf{w})$, where $\Psi_j^{-1} \sim \mathcal{GP}(\mu_j, \kappa_j(\mathbf{w}, \mathbf{w}'))$. By amalgamating all d mappings, the inverse prediction $[\Psi_1^{-1}(\mathbf{w}), \Psi_2^{-1}(\mathbf{w}), \dots, \Psi_d^{-1}(\mathbf{w})]$ is obtained. To train the inverse GP models, a dataset $\mathcal{D}^{inv} = \{(\mathbf{w}^{(l)}, \mathbf{x}^{(l)})\}_{l=1}^{N^{nd}}$ should be provided first, where $\mathbf{w}^{(l)}$ is the preference vector corresponding to the nondominated solution $\mathbf{x}^{(l)}$. It's worth noting that the multiobjective optimizer used may not be decomposition-based, and therefore, well-defined subproblems with associated preference vectors may not be specified beforehand. As a general strategy, the following formulation can be employed to derive $\mathbf{w}^{(l)}$ for $\mathbf{x}^{(l)}$ based on objective function values:

$$w_i^{(l)} = \frac{c_i^{(l)}}{\sum_{i=1}^{M} c_i^{(l)}}, i \in \{1, \ldots, M\}, \tag{6}$$

where $c_i^{(l)} = \frac{\sum_{v=1}^{m}(f_v(\mathbf{x}^{(l)})-(z_v^*-\epsilon))}{(f_i(\mathbf{x}^{(l)})-(z_i^*-\epsilon))}$, ensuring the preference vector $\mathbf{w}^{(l)}$ can correspond to the aggregated solution in the objective space.

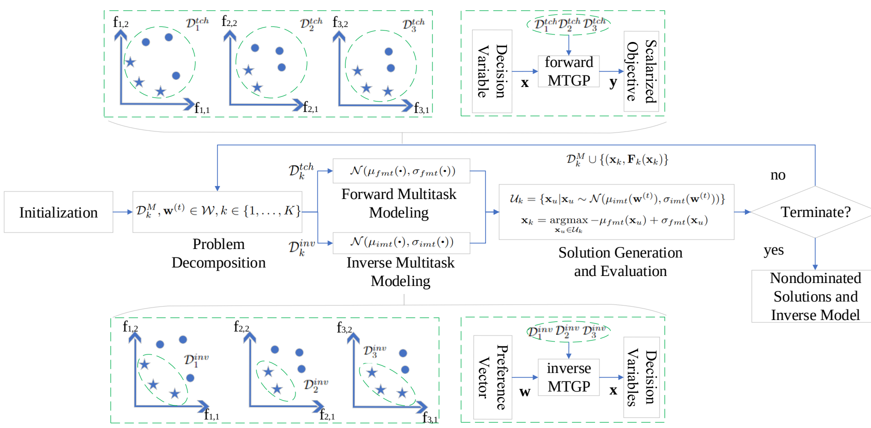


Fig. 1: Workflow of the proposed F-invTrEMO. Inverse modeling considers only nondominated solutions while forward modeling considers all the solutions.

3 Forward-Inverse Transfer for Multitask Multiobjective Optimization

This section elucidates the methodology of the proposed F-invTrEMO. We commence by delineating the workflow of the proposed method and demonstrate how to incorporate forward and inverse transfer into the optimization process. Subsequently, we introduce multitask Gaussian Process (MTGP) models [3] including forward MTGP and inverse MTGP that enable forward and inverse transfer.

3.1 General Framework of F-invTrEMO

The general workflow of the F-invTrEMO can be detailed in Fig. 1 and **Algorithm 1**, and the steps are explained as follows:

- **Initialization**: In this step, N_{init} solutions are generated and evaluated for each task, thus forming the initial datasets $\{\mathcal{D}_1^M, \dots, \mathcal{D}_K^M\}$, where $\mathcal{D}_k^M = \{(\mathbf{x}_k^{(l)}, \mathbf{F}_k(\mathbf{x}_k^{(l)}))\}_{l=1}^{N_k}$ and $N_k = N_{init}$.
- **Problem Decomposition**: In each iteration, we sample a randomly selected preference vector $\mathbf{w}^{(t)} \in \mathcal{W}$ and scalarize the MOPs into single-objective optimization problems based on (3). The resultant dataset for task k can be denoted as $\mathcal{D}_k^{tch} = \{\mathbf{X}_k, \mathbf{y}_k\} = \{(\mathbf{x}_k^{(l)}, y_k^{(l)})\}_{l=1}^{N_k}$, and $y_k^{(l)} = f_k^{tch}(\mathbf{x}_k^{(l)}) + \epsilon^{(l)}$ represents a noisy observation of the scalarized output corresponding to the k-th task, with $\epsilon^{(l)}$ being additive Gaussian noise with zero mean.
- **Forward Multitask Modeling**: In each iteration, a forward MTGP model is constructed to guide the optimization process for each task. Notably, in the context of F-invTrEMO, the forward model prediction for each task is designed to approximate the augmented Tchebycheff function as defined in (3) and the training set for the forward MTGP model is $\mathcal{D}_k^{tch}, k \in \{1, \dots K\}$. *It is important to note that the datasets for the forward MTGP models contain both dominated and nondominated samples, enabling the algorithm to fully utilize information from all the data.*
- **Inverse Multitask Modeling**: To construct inverse models, the dataset $\mathcal{D}_k^{inv}$ is generated for the kth task by only leveraging the nondominated samples within $\mathcal{D}_k^M$. To be specific, given multitask MOPs with d-dimensional decision variable for each task, d datasets are included in dataset $\mathcal{D}_k^{inv}$ as $\mathcal{D}_{k,j}^{inv} = \{\mathbf{W}_k, \mathbf{X}_{k,j}\} = \{(\mathbf{w}_k^{(l)}, \mathbf{x}_{k,j}^{(l)})\}_{l=1}^{N_k^{nd}}$, $j \in \{1, \dots, d\}$, containing pairs of nondominated solutions $\mathbf{x}_k^{(l)}$ and corresponding preference vectors $\mathbf{w}_k^{(l)}$ obtained according to (6). Subsequently, d independent inverse MTGP models are trained based on $\{\mathcal{D}_1^{inv}, \dots \mathcal{D}_K^{inv}\}$ to map the preference vectors to nondominated solutions in the decision space.
- **Evolutionary Solution Generation**: For each solving task, a candidate solution is generated, and the corresponding dataset is updated. For the kth task, given $\mathbf{w}^{(t)}$, the inverse MTGP provides a search distribution characterized by a Gaussian distribution. Subsequently, a set of offspring solutions $\mathcal{U}_k$ is produced by sampling according to this search distribution. We utilize the upper confidence bound (UCB) acquisition function to select the most promising solution $\mathbf{x}_k^{(t)}$ from $\mathcal{U}_k$ for the kth task. In the case of a minimization problem, the UCB function is formulated as $-\mu_{fmt}(\mathbf{x}_u) + \beta \cdot \sigma_{fmt}(\mathbf{x}_u)$, where β is a predefined parameter balancing exploration and exploitation, $\mu_{fmt}(\mathbf{x}_u)$ and $\sigma_{fmt}(\mathbf{x}_u)$ are the predictive mean and standard deviation for a given query $\mathbf{x}_u$ by the forward MTGP. The selected $\mathbf{x}_k^{(t)}$ is then evaluated by the objective function of the kth task, denoted as $\mathbf{F}_k(\mathbf{x}_k^{(t)})$. Finally, the dataset $\mathcal{D}_k^M$ is updated by appending $\{(\mathbf{x}_k^{(t)}, \mathbf{F}_k(\mathbf{x}_k^{(t)}))\}$ to it.

Steps 5 to 20 in **Algorithm 1** are repeated until the termination condition is met. The algorithm then returns the inverse MTGP models and the nondominated solutions corresponding to each optimization task.

Algorithm 1: The workflow of F-invTrEMO

Input: Task size K, Initialization budgets N_{init}, Dimension size d, Sample size of offspring solutions N_S, Objective functions $\mathbf{F}_k$ for each task k, Predefined preference vector set $\mathcal{W}$.
Output: Pareto optimal solutions for each task, Inverse model for each dimension

foreach *task* k **do**
  Initialize N_{init} solutions for the kth task, thus forming dataset $\mathcal{D}_k^M = \{(\mathbf{x}_k^{(l)}, \mathbf{F}_k(\mathbf{x}_k^{(l)}))\}_{l=1}^{N_k=N_{init}}$
end
$t \leftarrow 0$
while $t < t_{max}$ **do**
  Sample a preference vector $\mathbf{w}^{(t)}$ randomly from $\mathcal{W}$
  foreach *task* k **do**
    Scalarize $\mathcal{D}_k^M$ based on (3) and $\mathbf{w}^{(t)}$ to obtain dataset $\mathcal{D}_k^{tch} = \{(\mathbf{x}_k^{(l)}, y_k^{(l)})\}_{l=1}^{N_k}$
  end
  Build a forward MTGP model $\mathcal{N}(\mu_{fmt}(\mathbf{x}^{(*)}), \sigma^2_{fmt}(\mathbf{x}^{(*)}))$ using $\cup_{k=1}^{K}\mathcal{D}_k^{tch}$
  foreach *dimension* j **do**
    Generate k datasets $\mathcal{D}_{k,j}^{inv} = \{(\mathbf{w}_k^{(l)}, \mathbf{x}_{k,j}^{(l)})\}_{l=1}^{N_k^{nd}}, (k \in \{1, \ldots, K\})$ based on nondominated samples in $\mathcal{D}_k^M$ and (6)
    Build an inverse MTGP model $\mathcal{N}(\mu_{imt,j}(\mathbf{w}_{k^*}^{(*)}), \sigma^2_{imt,j}(\mathbf{w}_{k^*}^{(*)}))$ using $\{\mathcal{D}_{1,j}^{inv}, \ldots, \mathcal{D}_{K,j}^{inv}\}$ for the j-th decision variable
  end
  foreach *task* k **do**
    Let $\mathbf{w}_k^{(*)} = \mathbf{w}^{(t)}$, then sampling a set $\mathcal{U}_k$ of N_S offspring solutions based on the prediction provided by the inverse MTGP models
    Select solution $\mathbf{x}_k^{(t)} = \mathrm{argmax}_{\mathbf{x}_u \in \mathcal{U}_k} -\mu_{fmt}(\mathbf{x}_u) + \sigma_{fmt}(\mathbf{x}_u)$
    $\mathcal{D}_k^M \leftarrow \mathcal{D}_k^M \cup \{(\mathbf{x}_k^{(t)}, \mathbf{F}_k(\mathbf{x}_k^{(t)}))\}$
  end
  $t \leftarrow t + 1$
end

3.2 Multitask Gaussian Process as forward and inverse Models

Within the F-invTrEMO framework, the incorporation of MTGP models in both forward and inverse modeling [5] aims to thoroughly exploit sampled data across multiple tasks, thus enhancing information utilization. The iterative refinement of both forward and inverse modeling mutually benefits each other, resulting in improved approximation of the PF during and after optimization.

Distinguishing itself from canonical GP models outlined in equations (4) and (5), the primary innovation of MTGP lies in the formulation of a multitask kernel

[5] The training process of the MTGP model can be found in [19].

function [3], as expressed below:

$$\kappa_{mt}((\mathbf{x}_{input}, a), (\mathbf{x}'_{input}, b)) = \kappa_{tasks}(a, b) \cdot \kappa(\mathbf{x}_{input}, \mathbf{x}'_{input}) \tag{7}$$

Here, $\mathbf{x}_{input}$ and $\mathbf{x}'_{input}$ represent arbitrary input vectors, κ_{tasks} assesses the similarity among tasks, and κ mirrors the kernel functions in standard GPs, as delineated in equation (5). Within the context of this paper, the task indices a and b refer to the ath and bth optimization tasks, respectively. Subsequently, we will elucidate on the construction of the forward and inverse MTGP models.

- **Forward MTGP Building**: In each iteration, we construct a forward MTGP for the optimization tasks, mapping decision variables to aggregated objective values based on (3). The training set for task k during forward modeling is $\mathcal{D}_k^{tch}$. Subsequently, the posterior distribution of the forward MTGP can be formulated as follows:

$$\mu_{fmt}(\mathbf{x}_{k^*}^{(*)}) = \bar{\mathbf{k}}_{fmt,*}^{\intercal}(\bar{\boldsymbol{\Sigma}}_{fmt} + \boldsymbol{\Lambda}_{fmt})^{-1}\bar{\mathbf{y}} \tag{8}$$

$$\begin{aligned}\sigma_{fmt}(\mathbf{x}_{k^*}^{(*)}) = \kappa_{fmt}((\mathbf{x}_{k^*}^{(*)}, k^*), (\mathbf{x}_{k^*}^{(*)}, k^*)) - \\ \bar{\mathbf{k}}_{fmt,*}^{\intercal}(\bar{\boldsymbol{\Sigma}}_{fmt} + \boldsymbol{\Lambda}_{fmt})^{-1}\bar{\mathbf{k}}_{fmt,*}\end{aligned} \tag{9}$$

where $\bar{\mathbf{k}}_{fmt,*}$ is the multitask kernel vector between the input $\mathbf{x}_{k^*}^{(*)}$ and all the inputs in $\{\mathbf{X}_1, \ldots, \mathbf{X}_K\}$, $\boldsymbol{\Sigma}_{fmt}$ is the overall multitask kernel matrix corresponding to the forward MTGP, $\bar{\mathbf{y}}$ is the vector with all the weighted outputs in $\{\mathbf{y}_1, \ldots, \mathbf{y}_K\}$ where every single output can be obtained with the preference vector $\mathbf{w}$ and (3), and $\boldsymbol{\Lambda}_{fmt}$ is the additive noise variance matrix of the forward MTGP.
- **Inverse MTGP Building**: For inverse modeling, we establish an inverse MTGP model for each dimension of decision variables across all tasks. Given the training dataset $\{\mathcal{D}_1^{inv}, \ldots \mathcal{D}_K^{inv}\}$, the posterior distribution for a query preference vector $\mathbf{w}_{k^*}^{(*)}$, $\mathcal{N}(\mu_{imt,j}(\mathbf{w}_{k^*}^{(*)}), \sigma_{imt,j}(\mathbf{w}_{k^*}^{(*)}))$ can be computed using the formula (10) and (11) with the kernel function (7).

$$\mu_{imt,j}(\mathbf{w}_{k^*}^{(*)}) = \bar{\mathbf{k}}_{imt,*j}^{\intercal}(\bar{\boldsymbol{\Sigma}}_{imt,j} + \boldsymbol{\Lambda}_{imt,j})^{-1}\bar{\mathbf{X}}_j \tag{10}$$

$$\begin{aligned}\sigma_{imt,j}(\mathbf{w}_{k^*}^{(*)}) = &\kappa_{imt,j}((\mathbf{w}_{k^*}^{(*)}, k^*), (\mathbf{w}_{k^*}^{(*)}, k^*)) - \\ &\bar{\mathbf{k}}_{imt,*j}^{\intercal}(\bar{\boldsymbol{\Sigma}}_{imt,j} + \boldsymbol{\Lambda}_{imt,j})^{-1}\bar{\mathbf{k}}_{imt,*j}\end{aligned} \tag{11}$$

where $\bar{\mathbf{k}}_{imt,*j}$ is the multitask kernel vector between the input $\mathbf{w}_{k^*}^{(*)}$ and all the inputs in $\{\mathbf{W}_1, \ldots, \mathbf{W}_K\}$, $\boldsymbol{\Sigma}_{imt,j}$ is the multitask kernel matrix corresponding to the jth inverse MTGP, $\bar{\mathbf{X}}_j$ contains all the outputs in $\{\mathbf{X}_{1,j}, \ldots, \mathbf{X}_{K,j}\}$ and $\boldsymbol{\Lambda}_{imt,j}$ is the additive noise variance matrix corresponding to the jth inverse MTGP.

Table 1: IGD results on benchmark problems and real-world problems.

Problems	Tasks	ParEGO	PSL-MOBO	F-invTrEMO
mDTLZ-1	Task-1	0.0897 (0.0214) −	0.1307 (0.0512) −	**0.0611 (0.0051)**
	Task-2	0.1538 (0.0780) ≈	0.1877 (0.0699) ≈	**0.1528 (0.1200)**
	Task-3	0.1282 (0.0281) −	0.1876 (0.0955) −	**0.0731 (0.0099)**
	Task-4	0.1630 (0.0331) −	0.2571 (0.0812) −	**0.1182 (0.0718)**
mDTLZ-2	Task-1	0.1305 (0.0113) −	0.1708 (0.0149) −	**0.1135 (0.0083)**
	Task-2	0.1656 (0.0207) −	0.1798 (0.0185) −	**0.1489 (0.0174)**
	Task-3	0.1598 (0.0184) −	0.2537 (0.0836) −	**0.1328 (0.0150)**
	Task-4	0.3054 (0.0590) −	0.4187 (0.1235) −	**0.1439 (0.0154)**
mDTLZ-3	Task-1	0.4371 (0.1679) −	0.7681 (0.0535) −	**0.2726 (0.0277)**
	Task-2	0.5621 (0.1276) −	0.8203 (0.0347) −	**0.3059 (0.0399)**
	Task-3	0.4907 (0.1474) −	0.8142 (0.0624) −	**0.2833 (0.0356)**
	Task-4	0.5843 (0.1082) −	0.8922 (0.0571) −	**0.3241 (0.0305)**
EO	Task-1	0.0152 (0.0024) −	0.0549 (0.0100) −	**0.0095 (0.0011)**
	Task-2	0.0159 (0.0020) −	0.0518 (0.0132) −	**0.0102 (0.0017)**
	Task-3	0.0142 (0.0026) −	0.0526 (0.0090) −	**0.0104 (0.0013)**
HPO-1	Task-1	0.0208 (0.0030) −	0.0180 (0.0013) −	**0.0170 (0.0028)**
	Task-2	0.0222 (0.0023) −	0.0264 (0.0022) −	**0.0169 (0.0012)**
	Task-3	**0.0418 (0.0041)** ≈	0.0606 (0.0226) −	0.0420 (0.0134)
HPO-2	Task-1	0.0277 (0.0043) −	0.0338 (0.0021) −	**0.0220 (0.0053)**
	Task-2	0.0227 (0.0027) −	0.0226 (0.0026) −	**0.0160 (0.0016)**

4 Experimental Studies

In this section, we assess the effectiveness of F-invTrEMO using various multitask multiobjective optimization problems, encompassing benchmark problems [32,40], an engineering optimization problem [41], and hyperparameter optimization problems [36]. We compare F-invTrEMO with the classical ParEGO, which only incorporates single-task forward GP models, and the state-of-the-art PSL-MOBO [30], which includes both single-task forward and inverse models. For all the methods, the evaluation budget for each task in optimizing the multitask MOPs is set to 100 evaluations, with an initialization budget N_{init} of 20. During scalarization in each algorithm iteration, the weight ρ is fixed at 0.05. For the UCB function, the parameter β is set to 0.5. Due to the page limit, we introduce more experiments for comparison with a recent multiobjective Bayesian optimization method [35] and experiments as per advanced performance indicators including IGD$^+$ [26] and hypervolume [49] in the supplementary file [44].

4.1 Problem Settings

Benchmark Problems Three base problems are formulated based on the well-established DTLZ benchmark [11], following the methodology outlined in prior research [40]. By adjusting the parameters in these base problems, three sets of multitask modified DTLZ (referred to as mDTLZ) problems can be generated [32,40]. In our experiment, the dimension of the decision space for all mDTLZ optimization tasks is set to six. Additional detailed information about the base problems can be found in the supplementary file [44], with insights into the corresponding parameters provided in [32] and [40].

Real-world Problems Our real-world problems include an engineering optimization problem (EO) and two hyperparameter optimization problems (HPO).

It is commonplace to seek optimal engineering designs for components under diverse conditions, such as varying load cases or production materials. In such scenarios, obtaining high-quality solutions corresponding to each environment is imperative. Our case study centers on an engineering optimization problem known as the four-bar truss design problem [6]. We endeavor to identify a set of optimal designs for the four-bar truss under three distinct load cases. Further details regarding the parameter settings of the design problem are available in the supplementary file [44], while the formulation of objectives can be found in [41].

Hyperparameter optimization is a classical topic in the field of machine learning [18]. The configuration of hyperparameters for machine learning models can impact the model performance, computational resources, and interpretability to decision-makers. In this paper, we apply F-invTrEMO to generate optimal hyperparameter sets concurrently across diverse related tasks. We consider the following two scenarios. The first problem set (HPO-1) contains three hyperparameter optimization problems. The three optimization problems adjust the hyperparameters of the same machine-learning model, XGBoost [5], but on three distinct classification tasks including credit approval, medical diagnosis, and speech classification problems. The second problem set (HPO-2) contains two hyperparameter optimization problems. The two optimization problems adjust the hyperparameters of distinct but related machine learning models (i.e., XGBoost in 'gbtree' mode and 'dart' mode.) on the same classification task, medical diagnosis problem [36].

All the hyperparameter optimization case studies are implemented using the framework of YAHPO Gym [36]. Each task for a problem set is a tri-objective hyperparameter tuning, targetting at classification accuracy, model RAM, and model interpretability. Further details can be found in the supplementary file [44] and OpenML problems [43].

4.2 Performance Metrics

We involve four metrics to assess the performance of the algorithms, including the inverted generational distance (IGD) [42], root mean square error (RMSE),

The Inverted Generational Distance (IGD) is a commonly utilized metric in multiobjective optimization. It quantifies the Euclidean distance between the objective function values of the obtained nondominated solution sets and the true PF. For the kth task, the corresponding IGD value is computed as follows:

$$\begin{aligned} IGD_k = \frac{1}{N^r} \sum_{r=1}^{N^r} \min\{&||\mathbf{F}_k(\mathbf{x}_{opt,k}^{(r)}) - \mathbf{F}_k(\mathbf{x}_k^{(1)})||_2, \\ &\ldots, ||\mathbf{F}_k(\mathbf{x}_{opt,k}^{(r)}) - \mathbf{F}_k(\mathbf{x}_k^{(N_k^{nd})})||_2\} \end{aligned} \tag{12}$$

Table 2: RMSE results on benchmark problems and real-world problems.

Problems	Tasks	PSL-MOBO	F-invTrEMO
mDTLZ-1	Task-1	0.3021 (0.0744) −	**0.0721 (0.0173)**
	Task-2	**0.3828 (0.0770)** ≈	0.4250 (0.2263)
	Task-3	0.3790 (0.0837) −	**0.1592 (0.0780)**
	Task-4	0.4019 (0.0787) −	**0.3568 (0.1691)**
mDTLZ-2	Task-1	0.3501 (0.1321) −	**0.0679 (0.0295)**
	Task-2	0.2997 (0.0146) −	**0.1436 (0.0219)**
	Task-3	0.7010 (0.1819) −	**0.1700 (0.0454)**
	Task-4	0.9420 (0.1361) −	**0.3248 (0.1767)**
mDTLZ-3	Task-1	1.0219 (0.0310) −	**0.6946 (0.0392)**
	Task-2	1.0430 (0.0429) −	**0.7412 (0.0325)**
	Task-3	1.0562 (0.0457) −	**0.7830 (0.0361)**
	Task-4	1.1166 (0.0747) −	**0.8301 (0.0423)**
EO	Task-1	0.2211 (0.0242) ≈	**0.2173 (0.0054)**
	Task-2	0.2290 (0.0261) −	**0.2144 (0.0067)**
	Task-3	0.2225 (0.0372) ≈	**0.2046 (0.0053)**
HPO-1	Task-1	0.1818 (0.1143) −	**0.0943 (0.0721)**
	Task-2	0.1663 (0.1558) −	**0.0626 (0.0125)**
	Task-3	0.1881 (0.0545) −	**0.1273 (0.0411)**
HPO-2	Task-1	0.1855 (0.1075) −	**0.0811 (0.0476)**
	Task-2	0.1572 (0.1612) −	**0.0694 (0.0260)**

Where $\{\mathbf{x}_{opt,k}^{(1)}, \ldots, \mathbf{x}_{opt,k}^{(N^r)}\}$ represents a set of true Pareto optimal solutions that correspond to well-distributed points along the Pareto front, and $\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(N_k^{nd})}$ denote the nondominated solutions obtained in task k by the optimizer. In this paper, N^r is set to 2000.[6]

The RMSE aims to measure the accuracy of the obtained inverse models. Given a test set $\mathcal{D}_{opt,k} = \{(\mathbf{w}_{opt,k}^{(r)}, \mathbf{x}_{opt,k}^{(r)})\}_{r=1}^{N^{test}}$, where $\mathbf{w}_{opt,k}^{(r)}$ is the preference vector corresponding to $\mathbf{x}_{opt,k}^{(r)}$. Since the goal of the inverse model is to satisfy DM's preferences articulated in the objective space, we calculate the RMSE of the kth task in the objective space as:

$$RMSE_k = \sqrt{\frac{\sum_{r=1}^{N^r} ||\mathbf{F}_k(\mathbf{x}_{opt,k}^{(r)}) - \mathbf{F}_k(\mathbf{x}_{pred,k}^{(r)})||_2^2}{N^r}} \tag{13}$$

where $\mathbf{x}_{pred,k}^{(r)}$ is the prediction of the inverse models with respect to $\mathbf{w}_{opt,k}^{(r)}$. The predicted means of the inverse models are taken as the prediction $\mathbf{x}_{pred,k}^{(r)}$.

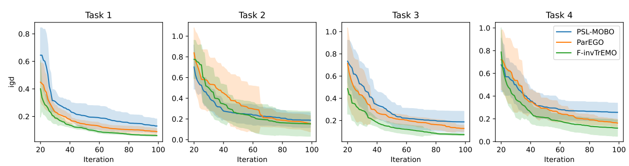
4.3 Results

In Tables 1 to 3, numbers with indicators (+), (−) and (≈) imply that the compared algorithm is better than, worse than, or similar to F-invTrEMO at 95% confidence level as per the Wilcoxon signed-rank test.

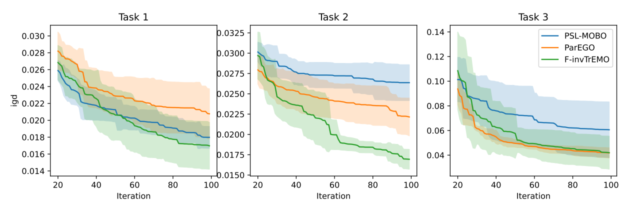
[6] For benchmark problems, the ground truth solutions serve as the reference solution set. For real-world problems, bi-objective and tri-objective reference solution sets are obtained using NSGA-II [13] and NSGA-III [12] with population size 500 and generation size 1000, similar to [40].

Table 3: IGD and RMSE results on benchmark problems for F-invTrEMO with and without forward multitask GP (invTrEMO).

Problems	Tasks	IGD		RMSE	
		invTrEMO	F-invTrEMO	invTrEMO	F-invTrEMO
mDTLZ-1	Task-1	0.0820 (0.0126) −	**0.0611 (0.0051)**	0.1174 (0.0358) −	**0.0721 (0.0173)**
	Task-2	**0.1264 (0.0699)** ≈	0.1528 (0.1200)	0.4857 (0.2244) ≈	**0.4250 (0.2263)**
	Task-3	0.1203 (0.0301) −	**0.0731 (0.0099)**	0.3181 (0.1851) −	**0.1592 (0.0780)**
	Task-4	0.1457 (0.0427) ≈	**0.1182 (0.0718)**	0.4911 (0.2010) −	**0.3568 (0.1691)**
mDTLZ-2	Task-1	0.1157 (0.0074) ≈	**0.1135 (0.0083)**	**0.0637 (0.0244)** ≈	0.0679 (0.0295)
	Task-2	**0.1284 (0.0118)** +	0.1489 (0.0174)	0.1466 (0.0484) ≈	**0.1436 (0.0219)**
	Task-3	0.1350 (0.0120) ≈	**0.1328 (0.0150)**	**0.1556 (0.0273)** ≈	0.1700 (0.0454)
	Task-4	0.2192 (0.1015) −	**0.1439 (0.0154)**	0.3360 (0.1605) ≈	**0.3248 (0.1767)**
mDTLZ-3	Task-1	0.3090 (0.1115) −	**0.2726 (0.0277)**	0.7355 (0.0633) −	**0.6946 (0.0392)**
	Task-2	0.4589 (0.1682) −	**0.3059 (0.0399)**	0.8288 (0.1004) −	**0.7412 (0.0325)**
	Task-3	0.3103 (0.0743) −	**0.2833 (0.0356)**	0.7837 (0.0650) ≈	**0.7830 (0.0361)**
	Task-4	0.3941 (0.1475) −	**0.3241 (0.0305)**	0.8403 (0.0803) ≈	**0.8301 (0.0423)**



(a)



(b)

Fig. 2: The IGD convergence trends of the mDTLZ-1 problem with four tasks and the HPO-1 problem with three tasks provided by ParEGO, PSL-MOBO, and F-invTrEMO (a) The mDTLZ-1 problem. (b) The HPO-1 problem.

Comparison with ParEGO and PSL-MOBO In Table 1, we present a summary of the IGD (mean and standard deviation) obtained by ParEGO, PSL-MOBO, and F-invTrEMO after 100 evaluations with 10 independent runs. Notably, F-invTrEMO outperforms the other methods in terms of IGD results for 18 out of the 20 tasks, indicating a better convergence of F-invTrEMO than its two competitors. Additionally, in Fig.2, we illustrate the convergence trends of ParEGO, PSL-MOBO, and F-invTrEMO on mDTLZ-1 and HPO-1 problems. Fig. 2 demonstrates that F-invTrEMO exhibits faster convergence compared to both ParEGO and PSL-MOBO. These results underscore the capability of F-invTrEMO to achieve superior convergence outcomes.

In addition to evaluating convergence performance, we also assess the accuracy of the obtained inverse models. Since only PSL-MOBO provides inverse models, Table 2 presents the RMSE results obtained by PSL-MOBO and F-invTrEMO. The findings indicate that, for 17 out of the 20 tasks, F-invTrEMO achieves superior RMSE values, suggesting that F-invTrEMO can produce more accurate inverse models through knowledge transfer across multiple tasks.

Ablation Study In F-invTrEMO, we incorporate the forward transfer to maximize the utilization of information from both dominated and nondominated samples across distinct tasks based on the inverse transfer model. This study verifies the forward transfer's effectiveness by comparing F-invTrEMO that fully exploits the interplay between forward transfer and inverse transfer models with invTrEMO that is solely based on inverse transfer models. The obtained IGD and RMSE results are presented in Table 3. Our analysis reveals that the convergence of F-invTrEMO is significantly improved with the inclusion of a forward MTGP model. In terms of both IGD and RMSE results, F-invTrEMO outperforms invTrEMO in 7 out of the 12 tasks on IGD and 5 of 12 tasks on RMSE. These findings suggest that by leveraging information from both dominated and nondominated samples corresponding to different tasks, the algorithm enhances both the convergence of the optimization and the accuracy of the inverse models.

5 Conclusion

In this paper, we incorporate knowledge transfer in both forward and inverse surrogate modelling for multiobjective optimization. It's argued that both forward and inverse models encounter the challenge of data scarcity. Both forward and inverse knowledge transfer are employed to alleviate this problem. Forward knowledge transfer improves convergence performance, while inverse knowledge transfer enhances the quality of approximating PF, both by utilizing the available information across distinct tasks through multitask modeling. The results are compared with those using only surrogate models without forward or inverse knowledge transfer across benchmark problems and real-world problems. These studies indicate that the proposed mechanism can enhance both the convergence performance and PF approximation results in multitask multiobjective optimization under strict computational constraints.

Acknowledgement. This research is partly supported by the National Research Foundation, Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No.: AISG2-GC-2023-010, "Design Beyond What You Know": Material-Informed Differential Generative AI (MIDGAI) for Light-Weight High-Entropy Alloys and Multi-functional Composites (Stage 1a)", the Distributed Smart Value Chain programme which is funded under the Singapore RIE2025 Manufacturing, Trade and Connectivity (MTC) Industry Alignment Fund-Pre-Positioning, (Award No: M23L4a0001), and the Centre for Frontier AI Research (CFAR) under Agency for Science, Technology and Research (A*STAR), and the College of Computing and Data Science, Nanyang Technological University.

Disclosure of Interests The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bali, K.K., Gupta, A., Ong, Y.S., Tan, P.S.: Cognizant multitasking in multiobjective multifactorial evolution: Mo-mfea-ii. IEEE Trans. on Cybern. **51**(4), 1784–1796 (2021)
2. Bali, K.K., Ong, Y.S., Gupta, A., Tan, P.S.: Multifactorial evolutionary algorithm with online transfer parameter estimation: Mfea-ii. IEEE Transactions on Evolutionary Computation **24**(1), 69–83 (2019)
3. Bonilla, E.V., Chai, K., Williams, C.: Multi-task gaussian process prediction. Advances in neural information processing systems **20** (2007)
4. Branke, J.: Multiobjective optimization: Interactive and evolutionary approaches, vol. 5252. Springer Science & Business Media (2008)
5. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 785–794. KDD '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939785
6. CHENG, F.Y., LI, X.S.: Generalized center method for multiobjective engineering optimization. Engineering Optimization **31**(5), 641–661 (1999). https://doi.org/10.1080/03052159908941390, https://doi.org/10.1080/03052159908941390
7. Cheng, R., Jin, Y., Narukawa, K., Sendhoff, B.: A multiobjective evolutionary algorithm using gaussian process-based inverse modeling. IEEE Transactions on Evolutionary Computation **19**(6), 838–856 (2015). https://doi.org/10.1109/TEVC.2015.2395073
8. Choong, H.X., Ong, Y.S., Gupta, A., Chen, C., Lim, R.: Jack and masters of all trades: One-pass learning sets of model sets from large pretrained models. IEEE Computational Intelligence Magazine **18**(3), 29–40 (2023). https://doi.org/10.1109/MCI.2023.3277769
9. Coello, C.A.C.: Evolutionary algorithms for solving multi-objective problems. Springer (2007)
10. Da, B., Gupta, A., Ong, Y.S.: Curbing negative influences online for seamless transfer evolutionary optimization. IEEE Trans. on Cybern. **49**(12), 4365–4378 (2018)
11. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600). vol. 1, pp. 825–830 vol.1 (2002). https://doi.org/10.1109/CEC.2002.1007032
12. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation **18**(4), 577–601 (2014). https://doi.org/10.1109/TEVC.2013.2281535
13. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans. on Evol. Comput. **6**(2), 182–197 (2002)
14. Dellnitz, M., Schütze, O., Hestermeyer, T.: Covering pareto sets by multilevel subdivision techniques. Journal of optimization theory and applications **124**, 113–136 (2005)

15. Emmerich, M., Giannakoglou, K., Naujoks, B.: Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. IEEE Transactions on Evolutionary Computation **10**(4), 421–439 (2006). https://doi.org/10.1109/TEVC.2005.859463
16. Feng, L., Gupta, A., Tan, K.C., Ong, Y.S.: Evolutionary Multi-task Optimization: Foundations and Methodologies. Springer (2023)
17. Feng, L., Zhou, L., Zhong, J., Gupta, A., Ong, Y.S., Tan, K.C.: Evolutionary multitasking via explicit autoencoding. IEEE transactions on cybernetics **49**(9), 3457–3470 (2018)
18. Feurer, M., Hutter, F.: Hyperparameter optimization. Automated machine learning: Methods, systems, challenges pp. 3–33 (2019)
19. Gardner, J., Pleiss, G., Weinberger, K.Q., Bindel, D., Wilson, A.G.: Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018)
20. Giagkiozis, I., Fleming, P.J.: Pareto front estimation for decision making. Evolutionary computation **22**(4), 651–678 (2014)
21. Gupta, A., Ong, Y.S., Feng, L.: Multifactorial evolution: toward evolutionary multitasking. IEEE Trans. on Evol. Comput. **20**(3), 343–357 (2016)
22. Gupta, A., Ong, Y.S., Feng, L.: Insights on transfer optimization: Because experience is the best teacher. IEEE Transactions on Emerging Topics in Computational Intelligence **2**(1), 51–64 (2017)
23. Gupta, A., Ong, Y.S., Feng, L., Tan, K.C.: Multiobjective multifactorial optimization in evolutionary multitasking. IEEE Trans. on Cybern. **47**(7), 1652–1665 (2016)
24. Gupta, A., Ong, Y.S., Shakeri, M., Chi, X., NengSheng, A.Z.: The blessing of dimensionality in many-objective search: An inverse machine learning insight. In: 2019 IEEE International Conference on Big Data (Big Data). pp. 3896–3902 (2019). https://doi.org/10.1109/BigData47090.2019.9005525
25. Gupta, A., Zhou, L., Ong, Y.S., Chen, Z., Hou, Y.: Half a dozen real-world applications of evolutionary multitasking, and more. IEEE Computational Intelligence Magazine **17**(2), 49–66 (2022). https://doi.org/10.1109/MCI.2022.3155332
26. Ishibuchi, H., Masuda, H., Tanigaki, Y., Nojima, Y.: Modified distance calculation in generational distance and inverted generational distance. In: Gaspar-Cunha, A., Henggeler Antunes, C., Coello, C.C. (eds.) Evolutionary Multi-Criterion Optimization. pp. 110–125. Springer International Publishing, Cham (2015)
27. Kim, Y., Pan, Z., Hauser, K.: Mo-bbo: Multi-objective bilevel bayesian optimization for robot and behavior co-design. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 9877–9883 (2021). https://doi.org/10.1109/ICRA48506.2021.9561846
28. Knowles, J.: Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Transactions on Evolutionary Computation **10**(1), 50–66 (2006). https://doi.org/10.1109/TEVC.2005.851274
29. Lai, G., Liao, M., Li, K.: Empirical studies on the role of the decision maker in interactive evolutionary multi-objective optimization. In: 2021 IEEE Congress on Evolutionary Computation (CEC). pp. 185–192 (2021). https://doi.org/10.1109/CEC45853.2021.9504980

30. Lin, X., Yang, Z., Zhang, X., Zhang, Q.: Pareto set learning for expensive multi-objective optimization. In: Advances in Neural Information Processing Systems. vol. 35, pp. 19231–19247. Curran Associates, Inc. (2022)
31. Lin, X., Zhen, H.L., Li, Z., Zhang, Q.F., Kwong, S.: Pareto multi-task learning. Advances in neural information processing systems **32** (2019)
32. Liu, J., Gupta, A., Ong, Y.S.: Inverse transfer multiobjective optimization (2023)
33. Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., Chi, E.H.: Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. p. 1930–1939. KDD '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3219819.3220007, https://doi.org/10.1145/3219819.3220007
34. Ma, P., Du, T., Matusik, W.: Efficient continuous pareto exploration in multi-task learning. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 6522–6531. PMLR (13–18 Jul 2020)
35. Paria, B., Kandasamy, K., Póczos, B.: A flexible framework for multi-objective bayesian optimization using random scalarizations. In: Adams, R.P., Gogate, V. (eds.) Proceedings of The 35th Uncertainty in Artificial Intelligence Conference. Proceedings of Machine Learning Research, vol. 115, pp. 766–776. PMLR (22–25 Jul 2020), https://proceedings.mlr.press/v115/paria20a.html
36. Pfisterer, F., Schneider, L., Moosbauer, J., Binder, M., Bischl, B.: Yahpo gym - an efficient multi-objective multi-fidelity benchmark for hyperparameter optimization. In: Proceedings of the First International Conference on Automated Machine Learning. Proceedings of Machine Learning Research, vol. 188, pp. 3/1–39. PMLR (25–27 Jul 2022)
37. Ponweiser, W., Wagner, T., Biermann, D., Vincze, M.: Multiobjective optimization on a limited budget of evaluations using model-assisted s-metric selection. In: Parallel Problem Solving from Nature–PPSN X: 10th International Conference, Dortmund, Germany, September 13-17, 2008. Proceedings 10. pp. 784–794. Springer (2008)
38. Seeger, M.: Gaussian processes for machine learning. International journal of neural systems **14**(02), 69–106 (2004)
39. Sinha, A., Korhonen, P., Wallenius, J., Deb, K.: An interactive evolutionary multi-objective optimization algorithm with a limited number of decision maker calls. European Journal of Operational Research **233**(3), 674–688 (2014). https://doi.org/https://doi.org/10.1016/j.ejor.2013.08.046
40. Tan, C.S., Gupta, A., Ong, Y.S., Pratama, M., Tan, P.S., Lam, S.K.: Pareto optimization with small data by learning across common objective spaces. Scientific Reports **13**(1), 7842 (May 2023). https://doi.org/10.1038/s41598-023-33414-6
41. Tanabe, R., Ishibuchi, H.: An easy-to-use real-world multi-objective optimization problem suite. Applied Soft Computing **89**, 106078 (2020). https://doi.org/https://doi.org/10.1016/j.asoc.2020.106078
42. Van Veldhuizen, D.A., Lamont, G.B.: Multiobjective evolutionary algorithm research: A history and analysis. Tech. rep., Citeseer (1998)
43. Vanschoren, J., van Rijn, J.N., Bischl, B., Torgo, L.: Openml: Networked science in machine learning. SIGKDD Explor. Newsl. **15**(2), 49–60 (jun 2014). https://doi.org/10.1145/2641190.2641198
44. Wei, T., Liu, J., Gupta, A., Tan, P.S., Ong, Y.S.: Bayesian Forward-Inverse Transfer for Multiobjective Optimization - Supplemen-

tary Materials (Jun 2024). https://doi.org/10.5281/zenodo.11665260, https://doi.org/10.5281/zenodo.11665260
45. Wei, T., Wang, S., Zhong, J., Liu, D., Zhang, J.: A review on evolutionary multitask optimization: Trends and challenges. IEEE Transactions on Evolutionary Computation **26**(5), 941–960 (2022). https://doi.org/10.1109/TEVC.2021.3139437
46. Wei, T., Zhong, J.: Towards generalized resource allocation on evolutionary multitasking for multi-objective optimization. IEEE Computational Intelligence Magazine **16**(4), 20–37 (2021). https://doi.org/10.1109/MCI.2021.3108310
47. Zhang, Q., Li, H.: Moea/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. on Evol. Comput. **11**(6), 712–731 (2007)
48. Zhang, Q., Liu, W., Tsang, E., Virginas, B.: Expensive multiobjective optimization by moea/d with gaussian process model. IEEE Transactions on Evolutionary Computation **14**(3), 456–474 (2010). https://doi.org/10.1109/TEVC.2009.2033671
49. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms — a comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.P. (eds.) Parallel Problem Solving from Nature — PPSN V. pp. 292–301. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)