Transfer optimization in complex engineering design

Tan, Alan Wei Min

2019

Tan, A. W. M. (2019). Transfer optimization in complex engineering design. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/105865>

<https://doi.org/10.32657/10220/47873>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**Transfer Optimization in
Complex Engineering Design**

ALAN TAN WEI MIN

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2019

Transfer Optimization in Complex Engineering Design

ALAN TAN WEI MIN

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

04 Feb. 2019

Date

Alan Tan Wei Min

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

04 Feb. 2019

Date

Dr. Ong Yew Soon

Authorship Attribution Statement

This thesis contains material from 2 papers published in the following peer-reviewed journals in which I am listed as an author.

Portions of Chapters 2, 3 and 4 are published as:

1. Alan Tan, Ramon Sagarna, Abhishek Gupta, Yew-Soon Ong, and Chi Keong Goh. Knowledge transfer through machine learning in aircraft design. *IEEE Computational Intelligence Magazine*, 12(4):48–60, 2017.

The contributions of the co-authors are as follows:

- I prepared the manuscript drafts and conducted experimental work in collaboration with Dr. Ramon Sagarna and Dr. Abhishek Gupta.
- Dr. Ong Yew Soon and Dr. Goh Chi Keong provided the initial direction of the paper and assisted in the revision of the manuscript.
- The “real-world dataset” used in the experiments were provided by Dr. Goh Chi Keong.

2. Alan Tan, Yew-Soon Ong, Abhishek Gupta, and Chi-Keong Goh. Multi-problem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems. *IEEE Transactions on Evolutionary Computation*, 23(1):15–28, 2017.

The contributions of the co-authors are as follows:

- The manuscript drafts were prepared in collaboration with Dr. Abhishek Gupta.
- I co-designed the experimental study with Dr. Abhishek Gupta, wrote the associated code and executed the designed experimental study.

- Dr. Ong Yew Soon provided direction throughout the various phases of the project, and assisted in the revision of the numerous drafts.
- Dr. Goh Chi Keong assisted in the revision of the manuscript.

04 Feb. 2019

Date

Alan Tan Wei Min

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my supervisors Dr Ong Yew Soon, Dr Goh Chi Keong, and Dr Cheu Eng Yeow for providing me with the opportunity to pursue a PhD in this field, as well as for their support and guidance; helping me see the forest from the trees. Their insights, and unique perspectives inspired me to widen my research, and consider new ideas and alternatives.

I would also like to thank, Dr Abhishek Gupta, in particular, and Drs Wang Yi, Kelly Yiping, Adams Kong Wai-Kin, Lim Pin and Ramon Sagarna for helping me learn and grow as a researcher as well as an individual. My horizons have certainly been broadened as a result of the time we spent dissecting problems and hunting for solutions. My gratitude goes out also to Rolls-Royce and the Rolls-Royce@NTU Corporate Lab for the financial support, real-world experiences and opportunities to collaborate.

Finally, I would like to thank my wife, Chin Hooi Chuan, and family for their constant love and support, bearing with me through this 4-year journey that could be, at times, bitter-sweet.

Contents

Statement of Originality	i
Supervisor Declaration Statement	ii
Authorship Attribution Statement	iii
Acknowledgement	v
Abstract	x
Lists of Figures	xvi
Lists of Tables	xviii
1 Introduction	1
1.1 Research Background	1
1.2 Research Objectives	3
1.3 Research Contribution	4
1.3.1 Publications	5
1.4 Thesis Organization	6
2 Literature Survey	8
2.1 A Brief Introduction to Machine Learning	8
2.2 Exploring a Complex Engineering Domain: Aircraft Design	10
2.2.1 The General role of Machine Learning in Aircraft Design	13

2.2.2	Future directions for Machine Learning in Aircraft Design . . .	16
2.2.3	Summary of Literature Survey in Aircraft Design	23
2.3	Background in Optimization	24
2.3.1	Single-Objective Optimization	25
2.3.2	Multi-Objective Optimization	25
2.3.3	Evolutionary Optimization	26
2.3.4	Bayesian Optimization	28
2.3.5	Recent Works of Knowledge Transfer for Optimization . . .	30
2.4	Summary	33
3	Transfer Learning for Complex Engineering Design	34
3.1	Chapter Goals	34
3.2	Introduction	35
3.3	A Case Study: Transfer Learning in Aircraft Engine Design	35
3.3.1	About the Use Cases	37
3.3.2	Benchmark Data Generation	38
3.3.3	Initial Experimental Evaluations	39
3.4	Adaptive Homogeneous Transfer Learning Approaches	42
3.4.1	Adaptive Transfer GP (AT-GP) [1]	43
3.4.2	Meta-heuristic Instance Selection for Transfer (MIST) . . .	44
3.4.3	Transfer-Stacking GP (TS-GP) - Multi-Problem Surrogate .	45
3.5	An Adaptive Heterogeneous Transfer Learning Approach	51
3.5.1	Manifold Transfer Gaussian Process (<i>Mani</i> -TGP)	51
3.6	Revisiting the Use Cases	57
3.6.1	Use Case 2 - Transferring Across Homogeneous Domains . .	57
3.6.2	Use Case 3 - Transferring Across Heterogeneous Domains .	61
3.7	Results on Real-World Aircraft Engine Data	61
3.8	Summary	63

4	Transfer Evolutionary Multiobjective Optimization of Computationally Expensive Problems with Multi-Problem Surrogates	65
4.1	Chapter Goals	65
4.2	Introduction	66
4.3	Basic Framework of TEMO-MPS	69
4.4	An Instantiation of TEMO-MPS	70
4.4.1	The Expected Improvement Figure of Merit	71
4.5	Experiments	71
4.5.1	Test Functions	72
4.5.2	Hypervolume Measure	73
4.5.3	Experimental Details	73
4.5.4	Determining the Initial DOE (Population) Size in ParEGO and TEMO-MPS	74
4.5.5	TEMO-MPS versus ParEGO and NSGA-II	76
4.5.6	Investigating the Effects of Source-Target Similarity	80
4.6	Practical Examples	81
4.6.1	A Case Study in Global Black-Box Optimization in Engineering Process Design augmented by Homogeneous Knowledge Transfer	82
4.6.2	Accelerated Automatic Hyperparameter Tuning of Machine Learning Models using Homogeneous Knowledge Transfer	85
4.7	Summary	87
5	Fast Bayesian Optimization Utilizing Heterogeneous Experiences	89
5.1	Chapter Goals	89
5.2	Introduction	90
5.3	Background	92
5.4	Manifold Transfer Bayesian Optimization	93
5.4.1	Regret Bounds Analysis of <i>Mani</i> -TBO	95

5.5	Evaluating the <i>Mani</i> -TBO Algorithm	97
5.5.1	Experimental Details of <i>Mani</i> -TBO	98
5.5.2	Experimental Results of <i>Mani</i> -TBO on Synthetic Functions	99
5.6	Real World Examples	103
5.6.1	Accelerated AutoML using Heterogeneous Knowledge Transfer	103
5.6.2	A Case Study in Global Black-box Optimization in Materials Engineering augmented by Heterogeneous Knowledge Transfer	105
5.7	Summary	107
6	Conclusion	109
6.1	Conclusions	109
6.2	Open Problems	113
	References	115

Abstract

The optimization task is an important element in most real-world design processes. It facilitates the construction of robust, high performance machines that are essential in today’s dynamic, safety-focused world. However, in the domain of complex engineering design, the objective functions to be optimized generally do not have closed-form solutions, are expensive to evaluate, and is non-convex. Consequently, traditional optimization algorithms that typically require up to several thousand exact function evaluations to obtain near-optimal solutions, are problematic to cost-effectively utilize. The solution to this issue has traditionally been in the form of surrogate-assisted optimization, where approximations of the expensive functions are constructed and, subsequently, searched. However, despite the improved results afforded by this methodology, a fairly substantial amount of initial evaluations is still required in order to build sufficiently accurate representations of the problem under investigation.

The improvements to designs in most practical settings are often incremental, with progress made gradually over time. Consequently, there is an abundance of knowledge from previous (or in-progress) design efforts that are related (but distinct), that can be utilized to augment the optimization task in order to achieve accelerated performance. Additionally, modern design cycles are often broken down into manageable pieces that are addressed by numerous teams working together in tandem. As such, relevant information can become available at various stages of an ongoing optimization task. Careful exploitation of this knowledge can be extremely beneficial in real-world settings, particularly for computationally

expensive optimization problems. In other words, in such cases, approaching the optimization task in a *tabula rasa* manner increases the duration (and thereby, cost) of the design stage.

This notion of knowledge transfer, while fairly well-researched in machine learning, has been comparatively unexplored in the domain of optimization problem solving. In general, the majority of existing methodologies of incorporating automatic knowledge transfer in optimization are found to be strongly reliant on the self-adaptiveness of evolutionary selection pressure to gradually cull out transferred information that may be irrelevant or harmful. While this approach has been proven successful in many occasions, there is a chance that excessive harmful transfer across completely unrelated tasks can severely disrupt the search performance leading to decelerated convergence. Accordingly, in this thesis, a study of knowledge transfer for optimization is conducted with the goal of expanding this field of study with a particular focus on the application domain of complex engineering design.

Specifically, as a means of exploring the primary mechanism of knowledge transfer used in this thesis i.e. transfer learning, a case study illustrating its potential for a particular complex engineering domain (i.e. aircraft design) is presented. Thereafter, three novel supervised transfer learning methodologies for regression are proposed that provides the capability to automatically adapt to various degrees of similarities between source (e.g. past design exercises) and target tasks; namely, the *Metaheuristic Instance Selection for Transfer* (MIST), *Transfer-Stacking Gaussian Process - Multi-Problem Surrogate* (TS-GP), and *Manifold Transfer Gaussian Process* (Mani-TGP).

Subsequently, incorporating the explored transfer learning approaches to optimization, a novel *Transfer Evolutionary Multiobjective Optimization* framework utilizing the idea of *Multi-Problem Surrogates* (TEMO-MPS) is presented. The proposed framework provides the capability to acquire and spontaneously transfer multiple learned models, adaptively, across problems, facilitating efficient global

multiobjective optimization. Finally, exploring the potential of heterogeneous knowledge transfer for optimization, an efficient *Manifold Transfer Bayesian Optimization* (*Mani-TBO*) framework is proposed that transfers data from a related but distinct source that may differ from the perspective of input feature spaces (i.e. heterogeneous transfer), and input-output functional relationships.

List of Figures

2.1	Illustration of transfer learning applied to a generic optimization algorithm. In the design effort of Project A, knowledge from a related (but distinct) Project B is used to enhance optimization performance.	18
2.2	Illustration of multi-task learning applied to generic optimization algorithms for two related projects. Both Projects A and B are assisted by a multi-task surrogate model constructed from a combination of data from both projects.	20
2.3	Illustration of multi-view learning applied to a generic optimization algorithm. Data from different perspectives of the system under study are used to build a multi-view learning enhanced surrogate model in order to obtain a more robust and accurate approximation of reality.	22
3.1	Diagrams representing use cases UC1 (top), UC2 (middle), and UC3 (bottom).	37
3.2	LEFT: <i>RMSE</i> values of GP and NT-GP for each output on UC1. RIGHT: <i>RMSE</i> for each output with increasing number of simulations for training a GP model. The red segments show the <i>RMSE</i> of NT-GP trained with 20 (target) simulations, and the red stars highlight the points where GP starts outperforming NT-GP.	40

3.3	<i>RMSE</i> values of GP and NT-GP for each output on UC2 (Left Panel) and UC3 (Right Panel).	41
3.4	<i>RMSE</i> values of GP, NT-GP, AT-GP, MIST and TS-GP for each output on UC2 for the EngineSim dataset (top panel) and the real world dataset (bottom panel). The results are averaged across 20 independent runs.	58
3.5	TOP ROW: Percentage of source data employed (dark colored) for training an NT-GP model with MIST for each output in UC2 for the EngineSim (TOP LEFT) and Real World (TOP RIGHT) datasets. BOTTOM ROW: Transfer stacking coefficients for the TS-GP model for each output in UC2 for the EngineSim (BOTTOM LEFT) and Real World (BOTTOM RIGHT) datasets.	60
3.6	<i>RMSE</i> values of GP, NT-GP, <i>Mani</i> -TGP for each output on UC3 for the EngineSim dataset (left panel) and the real-world dataset (right panel). Different configurations of <i>Mani</i> -TGP are examined, namely, (1) the LOO log predictive probability (Eq. 3.24) and log marginal likelihood (Eq. 3.23) criteria for hyperparameter tuning and (2) learned and random feature transformation (RT). The results are averaged across 20 independent runs.	63
4.1	Illustration of several design teams utilizing TEMO-MPS in their respective design exercises. These teams are based at various geographical locations and progress at different rates. Nevertheless, knowledge transfer between teams occurs - periodically uploading problem-specific data and leveraging available knowledge through the multi-problem surrogate of the TEMO-MPS framework.	68

4.2	Hypervolume vs. Function Evaluations for DTLZ3b–10, 0 for ParEGO (in blue) and TEMO-MPS (in red). Source models for TEMO-MPS is assumed to be drawn from DTLZ3b–0, 0 and DTLZ1b–0, 0. Figures (a)-(d) corresponds to $n_0 = 2(d + 2), 5(d + 2), 8(d + 2)$, or $11(d + 2)$ initial DOE data points, respectively. The results are averaged across 30 runs.	75
4.3	Illustration of the experimental procedure for TEMO-MPS on an example target optimization problem DTLZ1b–10, 0.	76
4.4	Hypervolume vs. Function Evaluations comparison between ParEGO, TEMO-MPS, and NSGA-II for DTLZ1b (TOP-ROW), DTLZ3b (BOTTOM-ROW), and various values of δ_1, δ_2 . The results in the figure are averaged across 30 independent runs.	78
4.5	(a) Pearson correlation coefficient of DTLZ1b–0, 0 and DTLZ1b–10, δ_2 . (b) aggregated objective values of DTLZ1b–0, 0, DTLZ1b–10, 1, DTLZ1b–10, 2 and DTLZ1b–10, 3 given the same inputs. (c) Hypervolume vs. Pearson correlation coefficient for TEMO-MPS at the specified function evaluation intervals (FEs). (d) Generalization performance gain (in %) of MPS vs. Pearson correlation coefficient.	81
4.6	An illustration of a typical RTM process workflow for fiber-reinforced polymer composite parts [2].	84
4.7	Performance comparison between TEMO-MPS, ParEGO, and NSGA-II for the optimization of the simulation-based RTM cycle. For the target task, a 1 m diameter composite disc with a fibre volume fraction of 35% is considered, while the source task involves a 0.8 m diameter composite disc with a fibre volume fraction of 50%. The results presented are based on a reference point of (40s, 40N) and is an average of 30 repeated runs. The shaded area spans 1/2 standard deviation on either side of the mean.	85

4.8	Evolution of Hypervolume vs. Function Evaluations for ParEGO and TEMO-MPS on a multiobjective formulation of the hyperparameter tuning problem of a feedforward neural network applied to a yeast gene functional multi-output classification problem. All results are averaged across 30 independent runs and based on a reference point of [1,1,1]. The shaded area spans 1/2 standard deviation on either side of the mean.	87
5.1	(a)-(d) Optimization search convergence trend comparison between <i>Mani</i> -TBO, traditional BO (from scratch) - using both UCB and EI acquisition functions, and random search for the 5D Griewank, 5D Rastrigin, 2D Six-Hump Camel and 5D Sphere functions, respectively. Results are averaged across 20 independent runs.	100
5.2	Random samples from the USPS (Left - Source) and MNIST (Right - Target) datasets with identical scaling.	103
5.3	Workflow illustration of a generic <i>Liquid Composite Moulding</i> process for fiber-reinforced polymer composite parts [2].	105
5.4	Optimization search convergence trend comparison between <i>Mani</i> -TBO, BO and random search for: (a) automatic hyperparameter tuning (5 target and 6 source dimensions), and (b) materials engineering (4 target and 6 source dimensions) problems.	106

List of Tables

3.1	Input (design) and output parameters for the source and target studies for UC1, UC2 and UC3.	38
4.1	NSGA-II Parameter Settings	74
4.2	TEMO-MPS and ParEGO Parameter Settings	74
4.3	Comparing the mean performance and standard deviation of ParEGO, NSGA-II and TEMO-MPS (see Section 4.5.5). Statistically significant superior results evaluated using a Wilcoxon signed rank test with a 95% confidence interval are presented in bold.	79
4.4	Upper and lower bounds of the design variables in the RTM manufacturing cycle	84
4.5	Upper and lower bounds of the input hyperparameters to the neural network.	87
5.1	Parameter settings for the GA used to optimize the acquisition function for both <i>Mani</i> -TBO and traditional BO.	99
5.2	The characteristics of the synthetic functions used in the experimental evaluation of <i>Mani</i> -TBO.	99
5.3	The configuration of the synthetic test functions used in this numerical study. Note that the <i>random features</i> correspond to the number of randomly inserted features that do not affect the value of the objective function.	100

5.4	Comparing the mean performance and standard deviation of <i>Mani-TBO</i> and BO using the Expected Improvement (EI) and Upper Confidence Bound (UCB) acquisition functions, with learned and random (RT) source feature transformations. Statistically significant superior results relative to BO for the particular acquisition function evaluated using a Wilcoxon signed rank test with a 95% confidence interval are presented in bold.	102
5.5	Upper and lower bounds of hyperparameters in the source and target neural networks of the accelerated AutoML case study.	103
5.6	Upper and lower bounds of design variables of the I/C-LCM and RTM processes.	106

Chapter 1

Introduction

1.1 Research Background

The optimization task is an important element in most real world design processes, facilitating the construction of robust, high performance machines that play a critical role in today's safety-focused, competitive world. Its objective is to identify a solution or a set of solutions that minimizes, or maximizes, the function(s) that describe the phenomena under investigation. More specifically, a generic single-objective optimization task can be defined as follows:

$$\mathbf{x}^{opt} = \arg \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \quad (1.1)$$

where $f : \mathcal{X} \rightarrow \mathbb{R}$ is the objective function on a compact subset $\mathcal{X} \subseteq \mathbb{R}^d$. Many algorithms have been developed to solve these types of problems, ranging from approaches that leverage gradient information to guide the search process [3, 4], to nature-inspired evolutionary-based methods [5, 6].

However, while demonstrably effective in many problem settings, these approaches generally require several hundreds to several thousand evaluations of the objective function to arrive at a suitable solution. This problem is particularly apparent when considering problems in complex engineering design. Tasks in this

application domain are generally characterized by tightly-coupled interactions between multiple processes or phenomena such that fast, analytical solutions are not easily available. As such, common solutions involve detailed numerical simulations that can be extremely expensive. For example, real world fluid flow problems are generally solved by numerically solving variants of the Navier-Stokes equations [7]; a task that can take several minutes to several days to solve depending on the complexity of the geometry. More precisely, the objective function f in these application domains generally do not have a closed-form solution, is expensive to evaluate, and is non-convex. A commonly adopted approach for addressing this challenge, as has been previously studied in various works [8–14], is the use of surrogates or meta-models to approximate the expensive functions, and subsequently search the approximation targeting more *efficient global optimization* (EGO) [15].

At this juncture, it is important to note that real-world problems seldom exist in isolation. Any practically useful system, especially one in an industrial setting, is expected to tackle a large number of problems over a lifetime, many of which will either be repetitive or at least display some domain-specific idiosyncrasies. Taking this cue, in the domain of machine learning, the idea of utilizing data from various *source* tasks to improve the learning on a related *target* task has seen considerable success under the label of *transfer learning* [16]. In contrast, it is observed that similar efforts in the domain of *optimization* have been scarce, with most solvers starting the search from scratch assuming a zero prior knowledge state. It can be argued that for many real-world problems, especially those entailing a high cost of evaluation, overlooking the knowledge available from experience may lead to deleterious overheads in the re-exploration of essentially similar search spaces.

To address this issue, this dissertation aims to explore the advantages of incorporating transfer learning capabilities into optimizers. The main motivation behind the approach is the innate ability of human experts to exploit what has been seen in the past to better address a new problem at hand. In particular, this study aims to adaptively leverage past knowledge in order to achieve accelerated

optimization performance.

1.2 Research Objectives

The overall goal of this thesis is to enhance the performance of surrogate-assisted optimization algorithms by leveraging knowledge from related (but distinct) tasks using TL methods. More specifically, the research objectives of this dissertation are as follows:

1. Many of the existing methodologies for adaptive knowledge transfer in optimization have been reliant on the intrinsic-nature of the evolutionary selection pressure to gradually cull out transferred information that may be relevant or harmful [17]. While this approach has been proven successful in many occasions, there is a chance that excessive harmful transfer across completely unrelated tasks can severely disrupt the search performance leading to decelerated convergence. As such, one objective of this research is to explore novel transfer learning frameworks for optimization that are adaptive and do not rely on selection pressure to cull out harmful transfer.
2. While there are methods that explicitly model the correlation between optimization problems but are restricted to single-objective optimization [18,19]. The need to simultaneously optimize a set of potentially competing objectives is a common occurrence complex engineering design. Consequently, another objective is to explore novel transfer optimization frameworks that can handle multiobjective optimization problems.
3. The complexity of the design process often necessitates the simplification of the problem under investigation. To adequately explore the design space, multiple variations of the problem have to be investigated, resulting the availability of numerous sources of information that might be beneficial for transfer. As such, another objective of this research is to explore the possi-

bility of adaptive transfer from multiple sources in order to enrich the target problem of interest, and further enhance optimization performance.

4. A limitation of current methods that attempt to instil such human-like capabilities as knowledge transfer is the assumption of *homogeneity*, i.e., the source and target optimization problems have overlapping search spaces with common features spanning equal dimensions. Therefore, an objective of this study is to investigate the concept of knowledge transfer across heterogeneous domains, where either the input feature spaces or input-output functional relationships may differ.

1.3 Research Contribution

The core contribution of this dissertation are three folds. Firstly, three novel adaptive methodologies for supervised transfer learning is presented. Secondly, a transfer evolutionary framework for multi-objective optimization is proposed. Thirdly, a Bayesian optimization framework that can facilitate transfer across heterogeneous domains is presented. Detailed descriptions of these contributions are as follows:

- The capability of adaptively leveraging knowledge from previous design exercises is a critical component in algorithms that can be reliably utilized in real world scenarios. In this dissertation, three such algorithms for regression are proposed:
 - **Meta-heuristic Instance Selection for Transfer:** A model-independent supervised transfer learning algorithm that adaptively selects the most relevant subset of data points from the source dataset in order to maximize predictive accuracy.
 - **Transfer-Stacking Gaussian Process - Multi-Problem Surrogate:** A model-based supervised transfer learning algorithm that

combines multiple base-models using transfer stacking in order to facilitate the incorporation of knowledge from multiple sources.

- **Manifold Transfer Gaussian Process:** A supervised transfer learning algorithm that provides the capability to transfer knowledge across heterogeneous domains through a source feature transformation learned jointly with model parameters.

- **Transfer Evolutionary Multiobjective Optimization - Multi-Problem**

Surrogates: In this study, an adaptive knowledge reuse framework for surrogate-assisted multiobjective optimization of computationally expensive problems, based on the novel idea of multi-problem surrogates, is proposed. The modern day design cycle which is typically distributed in nature often consists of multiple teams working on associated ideas in tandem. In such an environment, vast amounts of related information can become available at various stages of the search process corresponding to some ongoing target optimization exercise. This proposed idea provides the capability to acquire and spontaneously transfer learned models across problems as they become available, facilitating efficient global optimization.

- **Manifold Transfer Bayesian Optimization:** In this study, an efficient Manifold Transfer Bayesian Optimization framework is proposed. In design environments, relevant data is abundant. However, the dynamic nature of the application requires frequent innovative modifications that may lead to a change in the feature space. As such, the key novelty of the proposed framework lies in its capability to transfer data from a related but distinct source that may differ from the perspective of input feature spaces (heterogeneous transfer), and input-output functional relationships.

1.3.1 Publications

The research contributions described above have led to the following publications:

- **Published:** Alan Tan, Ramon Sagarna, Abhishek Gupta, Yew-Soon Ong, and Chi Keong Goh. Knowledge transfer through machine learning in aircraft design. *IEEE Computational Intelligence Magazine*, 12(4):48–60, 2017 [20].
- **Published:** Alan Tan, Yew-Soon Ong, Abhishek Gupta, and Chi-Keong Goh. Multi-problem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems. *IEEE Transactions on Evolutionary Computation*, 23(1):15–28, 2017 [21].
- **Submitted:** A journal paper that has been submitted to IEEE Transactions in Neural Network Learning Systems on the 25th of July, 2018 entitled “Fast Bayesian Optimization Utilizing Heterogeneous Experiences”.

1.4 Thesis Organization

The remainder of the thesis will be organized as follows:

- Chapter 2 presented a literature survey of a particular complex engineering design domain, namely aircraft design. In particular, the role of Machine Learning within the application area is investigated and several potential future directions addressing the problem of data-scarcity (in the optimization of computationally expensive problems) are described. Subsequently, a brief overview of several key elements of optimization relevant to this study is presented followed by an investigation of past efforts to facilitate knowledge transfer for optimization. [20]
- Chapter 3 presents a case study in which transfer learning is applied to several aircraft engine design use cases. An initial evaluation highlighting the limitations of traditional methods will be carried out, followed by the description of several supervised transfer learning algorithms for regression (both for homogenous and heterogenous transfer scenarios) that are based on

the Gaussian process model. Three in particular, are novel; the *Transfer-Stacking Gaussian Process - Multi-Problem Surrogate* (TSGP-MPS), the *Meta-heuristic Instance Selection for Transfer* algorithm (MIST), and the *Manifold Adaptive-Transfer Gaussian Process* (*Mani*ATGP). Subsequently, the use cases are re-evaluated using the described models in order to demonstrate the improvements in predictive performance that are possible by incorporating transfer learning ideas. [20,21]

- Chapter 4 presented a novel *Transfer Evolutionary Multi-Objective Optimization* framework using *Multi-Problem Surrogates* (TEMO-MPS) that facilitates spontaneous knowledge transfer from models origination from related design exercises that have been completed or are in-progress. The efficacy of the proposition is demonstrated on a series of synthetic benchmark functions, as well as two practical case studies. [21] .
- Chapter 5 presents a novel *Manifold Transfer Bayesian optimization* framework (*Mani*-TBO) that can transfer data from a related but distinct source that may differ from the perspective of input feature spaces, and input-output functional relationships. Empirical evaluations of the proposition will be carried out on a series of synthetic functions and real-world case studies.
- Chapter 6 concludes this dissertation and describes several potential future research directions.

Chapter 2

Literature Survey

2.1 A Brief Introduction to Machine Learning

Machine learning (ML) is a cornerstone of artificial and computational intelligence that can be loosely defined as a means of learning approximations from data. The existing methods in the literature have been successfully applied in a host of practical domains spanning natural language processing [22], computer vision [23, 24], recommender systems [25], etc. In general, ML can be broadly grouped into 4 categories: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning [26].

The first of these categories, supervised learning, utilizes a dataset, $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$, in which the outputs (or ground truth) $y^{(i)}$ are available. Here, $\mathbf{x}^{(i)}$ denotes the corresponding set of inputs (or feature vector) within a dataset of size n . As an example, in the context of person identification, $x_1^{(i)}$ could represent weight in kilograms, $x_2^{(i)}$ could be height in meters, $x_3^{(i)}$ the shoe size, $x_4^{(i)}$ the gender, and so on. Each data point, $\mathbf{x}^{(i)} = x_1^{(i)}, x_2^{(i)}, \dots, x_D^{(i)}$ would correspond to a particular individual. The labels $y^{(i)}$ could, for instance, be an element of a finite set of classes (e.g. country of origin -corresponding to a classification task) or a

Portions of this chapter are derived from the following publication: Alan Tan, Ramon Sagarna, Abhishek Gupta, Yew-Soon Ong, and Chi Keong Goh. Knowledge transfer through machine learning in aircraft design. *IEEE Computational Intelligence Magazine*, 12(4):48–60, 2017. [20]

continuous number (e.g. annual income - corresponding to a regression task). Ultimately, the goal of supervised learning is to utilize the dataset in order to learn a relationship between the inputs and the ground truth. Subsequently, given an unknown input vector $\mathbf{x}^{(*)}$, this model can be used to make predictions of the output $\hat{y}^{(*)}$.

The second of these categories, unsupervised learning, assumes a dataset $\{(\mathbf{x}^{(i)})\}_{i=1}^n$ where no ground truth is available i.e. only the input vectors are available for model learning. Examples of such datasets include the numerous images produced on social media platforms, and collected sensor readings from within a manufacturing facility. The classical unsupervised learning task, clustering, aims to group these unlabeled data points (i.e. assigning a cluster identifier) based solely on the characteristics of the input vector alone. In dimensionality reduction, the task aims to identify a compact representation of the input vector (i.e. $D_{compact} < D$).

The third of these categories, semi-supervised learning, represents a combination of the two aforementioned learning strategies. Specifically, the dataset consists of both labeled and unlabeled examples. Such a situation commonly occurs in many practical scenarios where only a subset of available data would be annotated (typically, by hand), leaving a relatively large portion of the data unlabeled. Examples include image data where a research organization would manually label facial images to be either male or female, or data-mined movie reviews from online sites that are manually labeled to be either positive or negative. In either case, a large portion would remain unlabeled due to budgetary or time constraints. The goal of this approach is to utilize both these datasets in order to construct a more accurate model relative to either supervised or unsupervised learning in isolation.

Finally, reinforcement learning, a fairly new learning paradigm, involves, as part of the learning methodology, interactions with the environment under study. The methodology involves learning a policy function which determines an ac-

tion to be executed given a state (i.e. a particular configuration of the environment). This policy function is similar, in many respects, to the supervised learning paradigm, where a relationship is learned between the state (i.e., input) and the corresponding action (i.e., the output) that, typically, maximizes the expected reward. Examples of reinforcement learning applications include games such as chess and go, and controls tasks (e.g. robotics, equipment load optimization).

In the section that follows, a brief review of a complex engineering design domain is presented. In particular, the focus will be on an extremely critical application area, namely, aircraft design. The overarching objective of this study is to determine the current state-of-the-art in this application domain with respect to role of machine learning, as well as to identify potential avenues for improvement.

2.2 Exploring a Complex Engineering Domain: Aircraft Design

The modern-day society is heavily dependent on the aircraft. Its ubiquity facilitates fast, efficient long distance travel as well as growth in many important areas such as medicine, economics, and international politics. Its functionality involves a variety of mutually interdependent systems that heavily interact. Further, it is tied to extremely strict safety standards, and high performance requirements [27]. Even so, significant technological advancements have occurred since the Wright brothers' first wood and fabric-based construction. Since then, in all technical areas, significant improvements have been achieved - including better understanding of the fundamental science behind flight (e.g. lift, drag, weight and flight performances). Computational methods and tools have also seen improvements, in particular, due to the availability of empirical data for verification and calibration. In spite of this, the design of an aircraft remains time-consuming and difficult, often requiring up to 6 years from initial conception to product delivery [28].

The aircraft is essential a multidisciplinary design optimization (MDO) task,

involving complex analysis, large search spaces, and multiple interacting disciplines. If the disciplinary boundaries are ignored, an MDO is essentially a nonlinear programming task; optimizing the objectives subject to a set of constraints. The difficulty lies in the fact that the disciplinary analyses that model the system behaviour are mutually interdependent. Consequently, it is important to account for the interaction between components through couplings variables or scenarios where the outputs of one analysis are inputs to another. In some cases, the objectives and constraints may themselves be dependent on the the outputs of multiple disciplinary analyses. Therefore, the key challenge is to manage the couplings of the system effectively and efficiently. A simplified alternative is to considered the problem as a set of single-discipline optimization tasks that occur in parallel or sequentially. However, proper consideration of the interdependencies of a multidisciplinary system will generally result in a more accurate representation of reality [29].

The complexity of the aircraft necessitates the involvement of specialists and MDO architectures to manage the interdependencies between disciplines. As an extremely active field of research, the domain has seen a variety of applications (e.g. wind turbines [30], buildings [31], and aircraft [32]), as well as MDO architecture development (as summarized in surveys: [33–36]) These architectures can be categorized into monolithic, where the MDO task is considered as a single-optimization problem, and distributed architectures, where the problem is decomposed into smaller, more management, subproblems. For further information, the interested reader is directed to a relatively recent publication, by Martings and Lambe, where a comprehensive summary, and comparison of all existing architectures was presented [29].

Regardless of the chosen architecture, one of the most important component is the engineering analyses itself. Past efforts often involve expensive physical experimentation and simplified analytical theory [37]. More recently, however, these approaches have largely been replaced by computational engineering meth-

ods due to the availability of modern high-performance computational resources. These methods utilize physics-based numerical simulations that solve mathematical models in a discretized manner. These simulations, while numerous in their benefits over past approaches, can sometimes take minutes to weeks for a single simulation [38]. In an MDO task, thousands of simulations may be required before a suitable solution is found. Consequently, alternative approaches that speed up these complex analyses would be a tremendous boon to the industry.

In aircraft design, machine learning has played a key role in reducing the impact of the aforementioned challenge. Predominantly, the technologies stemming from this field of study have been used for approximating the expensive physics-based simulations using supervised regression models, or more commonly, surrogate models. Its application has been widespread throughout various stages of aircraft design. However, despite the promising outcomes achieved thus far, it is deemed that the full capabilities of machine learning are yet to be fully unveiled and exploited in this domain. In the section of the chapter, a summary is presented of the efforts by the aerospace community to apply machine learning in aircraft design, and highlight the new opportunities presented by recent advances in the field to further its impact. Specifically, it is noted that in current practices, surrogate models are typically built from scratch assuming zero prior knowledge, only relying on data sampled from the ongoing target problem of interest. However, it is contended that any practically useful intelligent system in an industrial setting will be faced with a large number of problems over a lifetime, with the problems likely sharing domain-specific overlaps.

With this in mind, three relatively advanced machine learning technologies are described that are especially developed to enable automatic knowledge transfer as a means of improving upon existing *tabula rasa* efforts. This dissertation aims to unveil meta-machine learning as a promising approach to enhance the efficiency of aircraft design and facilitate the realization of a more *agile* design process [39,40]. Specifically, improving design procedures and methodologies so as to allow for

rapid adaptation to change, lower development costs, and shorter time-to-market. The remainder of this portion of the chapter is organized as follows: Section 2.2.1 provides an overview of the general role of machine learning in aircraft design; Section 2.2.2 sheds light on several potential directions for enhancing the future impact of machine learning through the incorporation of automatic knowledge transfer; Section 2.2.3 summarizes the literature survey of this complex engineering domain.

2.2.1 The General role of Machine Learning in Aircraft Design

Within the domain of engineering, machine learning (ML) has predominantly been used in the form of supervised regression models, or more commonly, surrogate models [14, 41]. Its application has been well-documented in numerous surveys [42–44], including detailed descriptions of its nuances e.g. cross-validation procedures, design of experiments methodologies, and the algorithms themselves. The range of algorithms used include kriging or Gaussian process [45–47], co-kriging [48], neural networks [49–51], support vector regression [52, 53], radial basis functions [51], and polynomial regression (2nd order variant in particular) [54–56].

Next, the use of machine learning in two areas of aerospace applications that have drawn considerable research interest is briefly discussed: (1) data-driven surrogate models of physical phenomena, and (2) machine-learning complemented physics simulations. The list of works described herein is not comprehensive but is rather a representative subset meant to provide a general impression of the concentration of efforts in this domain. Notably, a common facet of existing works is their rather myopic view of an ongoing target problem at hand; often overlooking the possibility of automatically extracting and transferring knowledge across different problems (that may well share underlying commonalities) to enhance the outcome of machine learning. This observation illuminates a key deficiency of current practices that is holding back the efficacy of machine learning in aircraft design, thereby motivating the future research directions that are sketched

in Section 2.2.2.

2.2.1.1 Data-driven Surrogate Models of Physical Phenomena

Due to its computational cost, a prime candidate for the application of machine learning regression models is to approximate the physics-based numerical simulators used heavily in engineering design [57]. For example, a two-stage supersonic turbine optimization task involving $O(10)$ design variables was presented by Papilla et al. [58]. Their approach involved a combination of radial-basis neural networks and simpler polynomial regression as a means of approximating computationally expensive CFD simulations. Similarly, Huang et al. [59] reduced the computational cost of an aeroengine turbine disc analysis under thermal and mechanical loads using kriging regression models that approximate the structural mechanics analysis of an engine component. In Sellar et al. [60], non-discipline level information (i.e. other subsystem outputs) was approximated utilizing neural networks within an MDO framework to improve overall optimization performance.

Surrogate models are also typically used to facilitate the use of variable-fidelity physics-based simulations. To elaborate, highly-accurate high-fidelity simulations are generally extremely expensive. Low-fidelity simulations, on the other hand, are less accurate but considerably cheaper. One solution to bridge this gap, as presented in [61], is to approximate the difference between these two models. Specifically, the authors built a kriging model to approximate the difference between low and high-fidelity CFD simulations and subsequently used this model along with the low-fidelity simulator in the optimization process. The results show improved accuracy relative to surrogate models built on the CFD data directly. Along a similar vein, in a work by Jaeggi et al. [62], the divergence between low and high-fidelity CFD simulations was predicted utilizing sparse pseudo-input GP models to serve as *simulation risk* in a multiobjective optimization task. The objective of this approach is to guide engineers in the development of designs that are minimally affected by simulation error and assist in the efficient utilization of

computational resources. In a recent publication by Minisci and Vasile [63], neural networks were used to calculate forces acting upon an unmanned spacecraft instead of a more expensive CFD simulation. Specifically, a multi-fidelity approach was used when obtaining data for network training, i.e. high-fidelity models for local refinement and low fidelity models for global sampling.

2.2.1.2 Machine Learning-complemented Physics Simulation

Fluid flow is governed by a set of highly complex partial differential equations (PDEs) known as the Navier-Stokes (NS) equations. These PDEs do not have analytical solutions, and must, therefore, be solved numerically. While directly solving these equations numerically are possible, they can be extremely expensive, particularly for turbulent flows that are common in aerospace applications. Instead, a simplification known as the Reynolds-Averaged Navier-Stokes (RANS) equations is more commonly used in CFD simulators. A key element of these equations is the turbulence modeling closure term that is typically modeled using ad-hoc heuristics that may produce inaccuracies in certain flow regimes. In [64], the authors proposed the use of ML to build a more accurate approximation of this closure term as means of enhancing the stability and accuracy of this simplified variant of the NS equations. As a proof-of-concept, a neural network regressor trained using data from a RANS-based CFD simulation with the Spalart-Allmaras turbulence model was constructed. Subsequently, the approximation is used in place of the aforementioned turbulence model and was shown to reproduce the original set of results. In [65, 66], the authors presented a methodology that learned a correction model that accounts for the deficiencies in a simpler low-fidelity CFD simulation (e.g. RANS) relative to high-fidelity direct numerical simulations of the NS equations. The correction factors were identified based on several variations of the problem. Subsequently, a GP regressor was used to learn this factor as a function of several problem-specific variables, and then injected into the simulator, producing more accurate results.

The finite element and finite volume methods are among the most popular techniques for solving PDEs, which is the mathematical form of the equations that govern many physical phenomena. Consequently, these numerical methodologies are commonly used for applications such as fluid flow and structural mechanics analyses. However, for accurate results, the physical domain under investigation must first be appropriately discretized in order to form a *mesh*. Specifically, regions of high interest require a dense mesh with tightly packed elements so as to capture all the relevant physics, while other regions can be made sparse in order to reduce computational cost [67]. A mesh that is improperly designed can result in a significant reduction in simulation accuracy. In some cases, the mesh is adaptively refined during the simulation process itself, which can be computationally expensive. In [67, 68], the authors proposed the construction of an expert system, based on the mesh design of a variety of similar geometries, in order to automatically determine the resolution of the mesh for the target problem of interest. In [69, 70], the authors proposed the use of self-organizing neural networks for automatic mesh-generation, based on a given mesh density (e.g. by an expert system) while meeting user-specified requirements (e.g. mesh type, element proportions, interior angles, etc).

2.2.2 Future directions for Machine Learning in Aircraft Design

The application of machine learning in the domain of aircraft design has led to significant advancements in the overall efficiency of the design process. However, a closer inspection reveals the presence of a particular shortcoming that is a key motivator for this dissertation. In particular, it is important to note that many learning algorithms require hundreds to thousands of data points in order to build a sufficiently accurate model for most real-world applications. However, the cost of obtaining a single data point using numerical simulation tools or physical experimentation is, generally, extremely high, leading to the scenario of data-scarcity [71]. Consequently, these existing methodologies continue to suffer from

the *cold start* problem, i.e. the significant time and resources required to generate the initial dataset for new problems [29].

With that in mind, it is noted that in many real-world settings, due to the incremental nature of progress, relevant data from past design projects are likely to be abundant. It is this capability to leverage knowledge from past experiences that differentiate an expert from an amateur. Therefore, a critical step towards further improving aircraft design using machine learning is the development of algorithms that can provide the capability to improve the efficiency of the design process through the adaptive incorporation of knowledge from related design projects. For example, in [72], the utilization of GP facilitating cross-domain knowledge exchange was shown to result in an improvement of 40% relative to more traditional approaches without transfer [18]. Further investigation of such ideas leading to significant cost savings would clearly be a benefit to aerospace applications and serve as motivation for the discussions that follow.

The notion of knowledge transfer has, however, been largely unexplored by the members of the aerospace community, with few notable exceptions. In [73], Han and Görtz proposed a hierarchical kriging model that approximates a high-fidelity function by using a sampled lower-fidelity function for its model trend. Valenzuela et al. [74] proposed a multi-fidelity Gaussian process model that enhances predictions of new concepts using observations from a previous concept with similar trends. These works mark the beginnings of initiatives intending to exploit knowledge transfer within the domain of aircraft design. Doubtless, the members of the aerospace community can further benefit from the large body of research and mathematical rigor provided by machine learning. Accordingly, in this section, three relatively modern technologies are highlighted that can help make significant strides in aircraft design by facilitating automatic knowledge transfer across problems and domains in surrogate modeling. The need to consider advanced techniques, in addition to more traditional incremental learning strategies (where problem-specific data appears over time [75]), shall be clarified through

Figure 2.1: Illustration of transfer learning applied to a generic optimization algorithm. In the design effort of Project A, knowledge from a related (but distinct) Project B is used to enhance optimization performance.

numerical case studies in Chapter 3. In particular, it will be shown that the direct (naïve) reuse of past knowledge may at times turn out to be worse than no transfer at all if the knowledge is not properly adapted - this is because the features, data distribution, and/or characteristics of target functions (such as the mean and scale parameters) need not precisely match across distinct problems/domains [76].

2.2.2.1 Transfer Learning

The development of complex designs draws heavily from prior knowledge, as *tabula rasa*-style design efforts would be extremely difficult (or impossible) to conduct. This reliance is especially significant in aircraft design, where the enormity of the task necessitates simplifying assumptions made based on technical knowledge and past experiences. This fact, coupled with the incremental nature of design and the preferred decomposition approach of MDO, indicates that there is a great deal of knowledge from past (and in-parallel) projects that can be exploited in order to improve the quality of designs. For instance, an engineering team designing a turbine for an aircraft engine would use, as reference, past designs that have been

successful and modify them accordingly to suit the current application. Relevant transferable knowledge may also originate from different parts of the aircraft that may fill a similar role. The fan blades from a low-pressure compressor and a high-pressure compressor of a gas turbine engine, for example, may have similar design characteristics despite the differences in the environments in which they operate.

However, this notion of knowledge transfer in optimization has largely been unexplored with only a few related works published fairly recently [77–82]. While prior knowledge incorporation can occur through appropriate problem formulations, the optimization algorithm itself is conducted from scratch i.e. without reusing knowledge (e.g. data or regression models) gathered from previous optimization efforts. As such, a possible research direction is the application of transfer learning as a means to exploit knowledge embedded in sources of information, such as data or regression models, obtained from related but distinct design efforts.

As illustrated in Figure 2.1, the objective of transfer learning is to improve the learning of the predictive function of a target task using knowledge from a source task, where either the domains or the tasks differ [16]. One suitable use case would be an optimization task of an aircraft engine in which source information would originate from a design effort to optimize a different engine configuration comprising several recurring parameters and outputs. Another potential use case would be to investigate the addition of new design parameters or an altogether different set of design parameters relative to a prior (source) task.

2.2.2.2 Multi-Task Learning

In real-world aircraft designs, it is not uncommon for the total number of design parameters throughout the entire system to number in the thousands. Consider, for example, a generic turbofan engine used by most commercial airliners. Its functionality depends on a host of interdependent systems such as the compressors, turbine, and combustion chamber - each of which is itself reliant on thou-

Figure 2.2: Illustration of multi-task learning applied to generic optimization algorithms for two related projects. Both Projects A and B are assisted by a multi-task surrogate model constructed from a combination of data from both projects.

sands of design variables. Due to effects such as the curse of dimensionality [83] coupled with the high computational cost of precision simulations, a monolithic optimization of the entire design space would be prohibitively expensive.

As such, a common practice is to analyze only a small subset of these design parameters (which constitutes a subproblem). Consequently, there are often different variations of these subproblems that are conditioned on the specific choice of all remaining design parameters. A typical example in aircraft design of such a situation would be the optimization of an aircraft wing. In each variation, the design team may aim to optimize performance parameters (e.g. lift, drag, and weight) as a function of the geometry of the wing [9], conditioned on certain fixed variables (e.g. materials, and flight conditions) subsequently varied as part of distinct studies. To elaborate, one variation might study a wing utilizing composite material A while another, composite material B; each with its own unique set of material properties.

The domain of multi-task learning is uniquely suited for these types of prob-

lems. In contrast to transfer learning which leverages on *past* experiences, multi-task learning assumes that there are multiple related tasks that can be *simultaneously* solved in order to facilitate knowledge transfer [18, 72, 84]. In the context of multi-task regression, the model aims to learn a mapping as follows: $f : \mathcal{X} \rightarrow \mathbb{R}^m$ where $\mathcal{X}$ represents a common set of input parameters and the m outputs are considered to belong to different tasks [72].

By learning related tasks together using a shared representation, the generalization performance is contended to improve by exploiting the shared knowledge in the training signals of the related task(s). As illustrated in Figure 2.2, from an aircraft design perspective, a single multi-task regression model built from data originating from various related design efforts can be used to boost multiple optimization algorithms at the same time - in the spirit of multi-task optimization [2, 18, 85, 86]. Approaching the problem in this manner facilitates knowledge transfer and reduces the overall computational cost of the design effort as the same regions of the parameter space need not be searched repeatedly [18], thereby resulting in the need for fewer expensive numerical (finite element or finite volume) simulations. It is worth noting here that, although the dimensionality for each of the subproblems is expected to be significantly lower than for the entire design space, it could be the case that the common set of input parameters defines a large space for the multi-task regression. In such scenarios, while the arguments in favor of savings in simulations would still hold, the process of multitasking does not rule out the application of classic approaches, such as space reduction techniques [87] (the beginning of Chapter 3 elaborates more on this point).

2.2.2.3 Multi-View Learning

Due to the computationally expensive nature of disciplinary analyses, a common strategy is to approximate the numerical simulations using models of varying fidelity. In certain situations, physical phenomena or structures of interest can be analyzed using either a full 3-D view or even a restricted 2-D/1-D view, depending

Figure 2.3: Illustration of multi-view learning applied to a generic optimization algorithm. Data from different perspectives of the system under study are used to build a multi-view learning enhanced surrogate model in order to obtain a more robust and accurate approximation of reality.

on the types of statistics the engineer is looking for. For example, while analyzing plate-like structures (which are common in aircraft), one may consider a 3-D discretization if through-thickness effects are of interest, or may emphasize on the 2-D discretization if only the in-plane effects are of paramount importance [88]. Thus, data that provides an understanding of the mechanics from different perspectives is made available in situations where computational budget is a prime limitation.

The domain of multi-view learning is uniquely suited to analyzing multifaceted data-streams of the aforementioned type. This fairly new paradigm jointly learns a function for each perspective in order to exploit the redundant views of the same domain and, consequently, improves prediction performance [89]. This form of learning is characterized by heterogeneous features that can be partitioned into groups (i.e. views). Typical examples of views, in the commonly cited context of person identification, include fingerprints, photographs, and personal identi-

fication documents - each with its own set of features. While there have been substantial efforts put into advancing the domain of multi-view learning (e.g. regression [90, 91] and clustering [92]), the application of this particular approach in aircraft design has, thus far, been unexplored.

Multiple viewpoints of a particular component or subsystem generally occur in the form of numerical simulations that emphasize on a specific facet of that component or subsystem (as described previously in the context of plate-like structures). This is often achieved by skewing the fidelity of the simulation in favor of phenomena of interest. For example, in the design optimization effort of a compressor, engineers could model the object with all its 3-dimensional intricacies, detailing the complex shape of each blade, the effects of gravity, and many other carefully defined minutiae. As these simulations would be extremely expensive, engineers may often restrict the scope of the model by emphasizing only those physical phenomena that are temporarily of the most interest. As illustrated in Figure 2.3, instead of treating each view independently of the others, multi-view learning could be exploited to provide a much more robust, comprehensive, and consequently accurate approximation of the overall reality.

2.2.3 Summary of Literature Survey in Aircraft Design

Many important aspects of today's modern society rely heavily on the aircraft. It connects cities across continents, and facilitates long distance collaborations, encourages tourism and provides the means to bring aid where it is needed. Its design, however, is a difficult task involving many mutually-interdependent systems, riddled with difficult to analyse responses. Machine learning, loosely defined as means to build approximations from data, has provided the means to reduce the computational cost associated with the design cycle.

As a consequence of the complex nature of the task, improvements to the construction of an aircraft are incremental; engineers rely heavily on acquired experiences and technical understanding. As such, there is an abundance of relevant

knowledge than can be exploited in order to improve the design process. In this section, a brief overview of the role in which ML has played within the domain of aircraft design, is presented. In general, the industry has come to rely quite heavily on ML in the form of surrogate models, but typically do not utilize data beyond what is sampled for the problem currently under study. Subsequently, three advanced ML fields of study, namely *Transfer Learning*, *Multi-Task Learning*, and *Multi-View Learning*, have been described. These ML technologies provide the means to facilitate knowledge transfer across design tasks.

Transfer learning aims to augment the approximation of a function of interest by utilizing data from a related source domain. In the case of the design of aircraft, such a source domain could be a previously completed or currently in-progress design project. On the other hand, multi-task learning simultaneously approximates a variety of domains that share similarities; achieving improved performance through shared knowledge within a unified model representation. Generally, the input spaces are homogeneous i.e. unaltered. In aircraft design, this scenario may occur during the optimization of a component (e.g. the wing) while varying inputs that are not easily parameterized (e.g. material properties), leading to multiple variants of a particular task. Finally, multi-view learning, utilizes data originating from different perspectives of a system, with correspondingly unique input spaces, in order to achieve improved predictive accuracy. In aircraft design, numerical simulations of varying fidelity would constitute different perspectives from which to draw knowledge from.

2.3 Background in Optimization

Optimization is an important component in many industries, particularly complex engineering design. It facilitates the construction of robust, high performance machines that play integral roles in the modern society. In this section, a brief overview of several key elements of optimization that are relevant to this disserta-

tion will be presented along with recent efforts that apply the notion of knowledge transfer within this domain.

2.3.1 Single-Objective Optimization

The optimization task involves identifying a solution that minimizes or maximizes the function under investigation. More generally, a generic single-objective optimization task can be defined as follows:

$$\begin{aligned} \text{Minimize: } & f(\mathbf{x}), \\ \text{subject to: } & \mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathcal{X}, \end{aligned} \tag{2.1}$$

where $f : \mathcal{X} \rightarrow \mathbb{R}$ is the objective function on a compact subset $\mathcal{X} \subseteq \mathbb{R}^d$. As an extremely well-studied field, a wide range of methods have been developed to solve this problem. A detailed discussion on the various aspects of solving such optimization problems is presented in [93].

2.3.2 Multi-Objective Optimization

Without loss of generality, the basic statement of a multi-objective optimization problem for a minimization task can be stated as follows:

$$\begin{aligned} \text{Minimize: } & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})), \\ \text{subject to: } & \mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathcal{X}, \end{aligned} \tag{2.2}$$

Unlike the single-objective variant, solving multiobjective optimization problems (MOPs) results in a diverse set of solutions that represent the best possible trade-offs between the different objectives. These solutions are traditionally labelled as Pareto optimal solutions [94]. A detailed discussion on various aspects of MOPs and associated optimization techniques can be found in [94]. Additionally, as a

well-studied field, there have been numerous methods presented in the literature to tackle MOPs [95–99].

2.3.3 Evolutionary Optimization

Evolutionary algorithms (EAs) are biologically inspired approaches that are loosely based around the concept of evolution. They are characterized by a population of solutions that are subjected to natural selection and mutation, resulting in a gradual improvement of the overall population fitness over time [100]. This family of algorithms have seen a great deal of use in the field of computer science for a variety of reasons, particularly its ability to perform global optimization on highly complex, non-linear functions that may have non-differentiable points, without making any assumptions about the underlying fitness landscape. EAs are not without drawbacks, however. The associated computational costs can be a limiting factor as a fairly large number of function evaluations are often required before a suitably optimal solution can be found; an issue that is especially prominent when dealing with computationally expensive problems.

The procedure of an EA applied to a single-objective optimization problem can be summarized as follows [101]. The algorithm begins with the initialization of a population of solution which are, subsequently, evaluated using the function under investigation. The resultant values constitute the fitness of the solutions. Next, a set of child solutions are generated using evolutionary operators e.g. the simulated binary crossover [102] and the polynomial mutation [103] operators. The fittest solutions, selected, for example, according to the $(\mu + \lambda)$ -ES [104] selection strategy, then becomes the next population of solutions. The procedure repeats until some stopping criteria is reached.

When dealing with MOPs, however, the complexity of the problem increases. As discussed in the previous subsection, there is no longer a single optimal solution, but rather a set of solutions that represent the best possible compromise between the objective functions under consideration. Consequently, differ-

ent strategies are employed relative to the single-objective variant. For example, NSGA-II [95], a popular algorithm for MOPs, selects the next generation utilizing a relation describing Pareto dominance and a neighbourhood density measure to encourage diversity of solutions found at the Pareto front. Another popular approach is the utilization of aggregation techniques to decompose the multiobjective problem into a set of single-objective problems [105, 106].

2.3.3.1 Objective Aggregation Procedures

The utilization of objective aggregation approaches when dealing with MOPs yields several notable advantages. These include computational efficiency, and the option to leverage the numerous single-objective optimization methods that have been well-studied [107]. Two aggregation methods that have seen considerable use in the literature, namely the Tchebycheff and weighted sum approaches, are described next:

Weighted Sum Approach: This method is based on minimizing convex combinations of the objectives, which can be stated as:

$$\text{Minimize: } y(\mathbf{x}|\boldsymbol{\omega}) = \sum_{j=1}^m \omega_j f_j. \quad (2.3)$$

Here, y is the aggregation of all objectives, f_j is the j^{th} objective function, and $\boldsymbol{\omega}$ represents a vector of strictly positive weights such that:

$$\boldsymbol{\omega} = (\omega_1, \omega_2, \dots, \omega_m) | \sum_{j=1}^m \omega_j = 1. \quad (2.4)$$

Optimal solutions found using the weighted sum aggregation approach can potentially cover the entire Pareto set (given a sufficiently diverse set of weight vectors) as long as the Pareto front is convex. However, this approach is not compatible with nonconvex Pareto fronts [98], which leads us to the next (generally more

flexible) technique.

Tchebycheff Approach: In this case, the transformed problem after objective aggregation is stated as:

$$\text{Minimize : } y(\mathbf{x}|\boldsymbol{\omega}) = \max_{j=1}^m \{\omega_j f_j - z_j^*\}, \quad (2.5)$$

where z_j^* is a reference point with respect to the j^{th} objective, and $y, \boldsymbol{\omega}$ are interpreted in the manner stated previously. This approach has been found to be well-suited for both convex and non-convex Pareto fronts [108]. While there is a possibility that non-differential points occur, EAs, that can readily deal with such issues, provide an effective solution. Consequently, when dealing with MOPs where little is known ahead of time, the use of EAs and the Tchebycheff aggregation method is an effective combination.

2.3.4 Bayesian Optimization

Bayesian optimization (BO) is a sequential surrogate-assisted optimization strategy that provides an appealing solution to the exploration-exploitation trade-off. It is ideally suited for expensive black-box functions due to its excellent performance while remaining efficient in terms of the number of function evaluations required [109].

Traditional, the search is initiated with a randomly sampled dataset which is augmented at every subsequent iteration with a newly evaluated solution chosen in a principled manner. To elaborate, let the dataset after n exact evaluations be $\mathcal{D}^{(n)} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$, where $y^{(i)} = f(\mathbf{x}^{(i)}) + \epsilon$ is the observed value of the possibly noisy objective function. Note that ϵ represents an i.i.d noise term such that $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. Given $\mathcal{D}^{(n)}$, a probabilistic model $p(f|\mathcal{D}^{(n)})$ is learned, and the posterior uncertainty of $p(f|\mathcal{D}^{(N)})$ is used to define a cheap acquisition function $\alpha(\mathbf{x}; p(f|\mathcal{D}^{(n)}))$ that quantifies the utility of candidate solutions for the next (exact) evaluation of f . With this, the next solution to be evaluated is

determined as:

$$\mathbf{x}^{(n+1)} = \arg \max_{\mathbf{x} \in \mathcal{X}} \alpha(\mathbf{x}; p(f|\mathcal{D}^{(n)})). \quad (2.6)$$

The augmented dataset $\mathcal{D}^{(n+1)} = \{\mathcal{D}^{(n)}, (\mathbf{x}^{(n+1)}, y^{(n+1)})\}$ is then used for building the probabilistic model in the subsequent iteration. The process continues until some pre-defined stopping condition, such as the consumption of available computational budget, is satisfied.

2.3.4.1 Gaussian Process Regression

Due to its unique properties, the probabilistic Gaussian process (GP) is often used in BO. A standard GP is a stochastic non-parametric approach for regression. It describes a distribution over functions as:

$$f \sim GP(m, k) \quad (2.7)$$

where m is the mean (typically $m = 0$) and k is a covariance kernel [110]. They have been widely accepted as effective approximation methods that provide predictions together with uncertainty estimates, as a measure of statistical confidence. In particular, they facilitate the development of computationally cheap acquisition functions (or figures of merit) α , such as the expected improvement measure, that provide a balance between exploration and exploitation during the optimization search.

Given training data that consists of n pairs of inputs and outputs $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$ - where $\mathbf{x}^{(i)}$ is the i th row of the $n \times d$ -dimensional input matrix $\mathbf{X}$ and $y^{(i)}$ is the i th instance in the vector of output labels $\mathbf{y}$ - the predictive distribution at an unknown input, $\mathbf{x}^{(*)}$, is given by the following:

$$\hat{y}(\mathbf{x}^{(*)}) = K_*(K + \sigma_n^2 I)^{-1} \mathbf{y}, \quad (2.8)$$

$$\hat{\sigma}^2(\mathbf{x}^{(*)}) = K_{**} - K_*(K + \sigma_n^2 I)^{-1} K_*^T, \quad (2.9)$$

where $\hat{y}$ and $\hat{\sigma}^2$ are the mean and posterior variance, respectively, of the predictive distribution at $\mathbf{x}^{(*)}$. Further, $K_* = [k(\mathbf{x}^{(*)}, \mathbf{x}^{(1)}), \dots, k(\mathbf{x}^{(*)}, \mathbf{x}^{(n)})]$ and $K_{**} = k(\mathbf{x}^{(*)}, \mathbf{x}^{(*)})$, where k represents the covariance function, K is the associated covariance matrix, and σ_n is a noise parameter. One popular option is the automatic relevance determination (ARD) squared exponential covariance function defined as follows:

$$k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \theta_f \exp\left(\sum_{D=1}^d \frac{-(x_D^{(i)} - x_D^{(j)})^2}{2\theta_D^2}\right), \quad (2.10)$$

where $\boldsymbol{\theta} = \{\sigma_n, \theta_f, \theta_1, \theta_2, \dots, \theta_d\}$ are $d + 2$ hyperparameters to be learned by minimizing the negative logarithm of the marginal likelihood function:

$$l = -\frac{1}{2} \log|K + \sigma_n^2 I| - \frac{1}{2} \mathbf{y}^T (K + \sigma_n^2 I)^{-1} \mathbf{y} - \frac{n}{2} \log 2\pi. \quad (2.11)$$

2.3.5 Recent Works of Knowledge Transfer for Optimization

The notion of knowledge transfer, as discussed previously in Section 2.2.2.1, is an attractive notion as it can facilitate the reuse of knowledge obtained from past design exercises that are, generally, plentiful. While fairly well-researched in machine learning (e.g. [16]), this concept has been comparatively unexplored in the domain of optimization. In this section, a brief review is presented on the application of knowledge transfer within the domain of numerical optimization.

The concept of automatic transfer in global black-box optimization, or *transfer optimization*, was recently surveyed and summarized in [111]. In particular, the authors defined transfer optimization as means to facilitate improved performance for a new target task $\mathcal{T}$ given a knowledge-base $\mathcal{M}_S$ gathered from past optimization efforts. Specifically, for a minimization problem, this improvement can be expressed as:

$$Q_n(\mathcal{T}) - Q_n(\mathcal{T}|\mathcal{M}_S) \geq 0 \quad (2.12)$$

where $Q_n = f(\mathbf{x}^*) : \mathbf{x}^* \in \mathcal{D}^{(n)} \wedge (\nexists \mathbf{x} \in \mathcal{D}^{(n)} : f(\mathbf{x}) < f(\mathbf{x}^*))$, and n is the available computational budget. Further, the authors classified the paradigm into two categories: (1) sequential transfer, where problems are tackled one after another, and (2) multi-tasking, where multiple tasks are simultaneously solved. In this thesis, the focus is on the first of these categories, i.e. *sequential transfer optimization*.

In [17, 112], the transfer of knowledge is promoted by simply utilizing the final population from a GA optimization of the source task as the initial population of the target task. In a similar effort but for the domain of controller design in evolutionary robotics, Mashaiov and Tal [113] applied this idea of *genetic transfer* to great success. In their work, they found that the traditional initial population generated at random is very unlikely to find a suitable design. A better approach is to leverage, as part of the initial set of solutions, solutions from a related source task in order to effectively initiate the search process. In a relatively recent effort, the reuse of “building-blocks” of knowledge from source tasks for genetic programming (in this case, code fragments) was proposed by Iqbal et al. [80, 114]. Their proposition was demonstrated on boolean problems [80] and classification problems [114] resulting in promising outcomes. Exploring along a different path, in [85], the paradigm of evolutionary multi-tasking was introduced providing the means to simultaneously address multiple tasks (that are related) using a single population of evolving solutions. Within this framework, knowledge transfer took place via the sharing of genetic material, resulting in improved convergence characteristics on a variety of practical optimization problems [2, 86].

The aforementioned techniques facilitated knowledge transfer primarily through the transfer of genetic material. In contrast, the following approaches utilize, in conjunction with EAs, higher-order model-based transfer. In [77], for example, knowledge transfer took place for an optimization and, simultaneously, a problem learning paradigm. The proposed EA is assisted by a classifier in order to extract the source task feasibility structure, and reused to avoid infeasible solutions when

addressing the target task. In another work, a memetic computation paradigm was proposed by Feng et al. [79, 115] for graph-based problems that captures transferable knowledge in the form of a learned distance metric.

As an alternative approach to evolutionary-based knowledge transfer, there have been some recent attempts to incorporate knowledge transfer in Bayesian optimization by explicitly modelling the correlation between the optimization problems [18, 19] - instead of placing the onus entirely on evolution. In [76, 116], knowledge transfer was facilitated by the construction of a common response surface for source ($\mathcal{S}$) and target ($\mathcal{T}$) optimization exercises, allowing a shared GP model to be reused across tasks. In contrast, [19] put forward a meta-learning initialization scheme to bias the initially sampled dataset towards optimized solutions of similar problems solved previously. The similarity was learned by a regression model mapping the numerical descriptors (or metafeatures) of pairs of tasks to the Spearman’s rank correlation between their respective objective functions.

In [18], the transfer of knowledge was accomplished by using a multi-task GP to build a joint probabilistic model $p(f_{\mathcal{T}}, f_{\mathcal{S}} | \mathcal{D}_{\mathcal{T}}, \mathcal{D}_{\mathcal{S}})$ for the objectives of the target and source optimization tasks. In particular, the correlation between tasks was captured by using a product covariance function for solution and task pairs. A novel acquisition function in the context of the above was proposed in [117]. Further scalability enhancements in multi-task settings were carried out in [118] by using Bayesian neural networks instead of GPs.

In [119], a similar methodology was proposed, where models from related source optimization tasks are transferred through a construction of an ensemble in which each model is weighted according to the probability that it is the best available model based on a ranking loss function. In [120], two additional methodologies were proposed to determine the weights of the ensemble. The first is the product of experts model where the joint likelihood of the ensemble is a weighted product of the likelihoods of individual GP models built on single datasets. The

second approach is kernel regression, where the weights are set to the similarity between source and target datasets, described according a set of user-defined meta-features. In [121], the authors utilized a feedforward neural network to combine Bayesian linear regression models built for each dataset as a means of facilitating knowledge transfer across optimization tasks.

2.4 Summary

In general, the majority of existing methodologies of incorporating automatic knowledge transfer in evolutionary computation are found to be strongly reliant on the self-adaptiveness of evolutionary selection pressure to gradually cull out transferred information that may be irrelevant or harmful [17]. While this approach has been proven successful in many occasions, there is a chance that excessive harmful transfer across completely unrelated tasks can severely disrupt the search performance leading to decelerated convergence. In response to this research gap, there have been efforts to explore model-based knowledge transfer but these are few in number and relatively restricted in scope. In particular, the works in the literature do not consider knowledge transfer for optimization from multiple sources, nor do they utilize data from heterogeneous sources, where the dimensionalities of the tasks may differ.

Chapter 3

Transfer Learning for Complex Engineering Design

3.1 Chapter Goals

Many of the important application domains that define our lives in the modern era fall under the characterization of complex engineering design, where the combination of strict requirements and the unavailability of cheap analytical solutions necessitates the widespread utilization of computationally expensive high-fidelity simulations. Consequently, a common strategy is to enhance the designs gradually over time. As such, there exists a vast amount of related knowledge that can be leveraged in order to improve the design process. In this chapter, the goal is to develop novel transfer learning algorithms for complex engineering design that can facilitate the fulfilment of Research Objectives 1 through 4 (see [Section 1.2](#)). Specifically, adaptive methodologies that can handle (1) homogeneous and heterogeneous transfer (where the input feature spaces of the source and target domains may differ), and (2) transfer from multiple sources.

Portions of this chapter are derived from the following publications: Alan Tan, Ramon Sagarna, Abhishek Gupta, Yew-Soon Ong, and Chi Keong Goh. Knowledge transfer through machine learning in aircraft design. *IEEE Computational Intelligence Magazine*, 12(4):48–

3.2 Introduction

Complex engineering design is an application domain where the enormity of the task necessitates incremental design enhancements. Further, its complexity generally requires heavy utilization of expensive numerical simulations that can take a significant amount of time to complete. In such cases, transferring knowledge between related design tasks, in order to reduce the amount of training data required, is extremely desirable. The field of study that facilitates this capability (i.e. *Transfer Learning* - see Section 2.2.2.1) aims to improve the learning of a target task using knowledge from a source task where either the tasks themselves or the domains differ [16].

In Section 3.3, a case study for transfer learning is presented that outlines three use case along with a preliminary evaluation. Subsequently, adaptive homogeneous and heterogeneous transfer learning approaches will be described in Sections 3.4 and 3.5, respectively. Armed with these methodologies, the use cases will be revisited in Section 3.6. To further explore the applicability of the TL methods, they are again tested on real-world datasets provided by Rolls-Royce in Section 3.7. Finally, the chapter is summarized in Section 3.8.

3.3 A Case Study: Transfer Learning in Aircraft Engine Design

The potential of the ML paradigms described in Section 2.2.2 is demonstrated in this section through a case study illustrating the effects of transfer-learning-based data reuse (Section 2.2.2.1) for aircraft engine design.

The initial assessment of new design ideas is an important element of the development cycle of a new aircraft engine. Generally, the estimation of critical engine attributes (e.g. thrust, weight, fuel consumption), based on operational

60, 2017 [20], and Alan Tan, Yew-Soon Ong, Abhishek Gupta, and Chi-Keong Goh. Multi-problem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems. *IEEE Transactions on Evolutionary Computation*, 23(1):15–28, 2017 [21].

parameters (e.g. pressure, temperature, altitude), are done through simulation. The aim, at this stage, is to arrive at a suitable initial design which will then be further refined at later stages of component-level analyses using higher-fidelity simulation techniques. Due to the high dimensionality of the design space and the high cost of the aforementioned simulations, surrogate models play a crucial role in this task [122].

More precisely, overall engine attributes, such as weight, thrust, or specific fuel consumption, are estimated through simulation for the given design performance parameters of a different kind; e.g. pressures, temperatures, velocities, etc. This space is typically explored with the aim of resolving a good starting design point which will be further refined upon at the component-level analyses that are often based on high-fidelity physics-based numerical simulations. The adoption of surrogate modeling to assist engineers in this search comes as no surprise if the sparseness of the design space that is caused by the (typically) high dimensionality and the significant cost of numerical simulations [122] is considered.

In large search spaces, insufficient data, assuming no additional information, may lead to a poorly performing surrogate model. Consequently, any analysis performed on a poor approximation may result in suboptimal designs or inaccurate design evaluations. The aim, therefore, is to improve the fit of the model (e.g. by obtaining additional data through more simulations), while carefully taking into account the overall computational cost. This motivates the need for approaches that can better leverage any available data to provide acceptable accuracies, even with small dataset sizes.

A common approach to these issues is the utilization of space reduction techniques [87]. For example, a family of techniques aim to refine the feasible design space adaptively. Though effective, additional data points are still required at each iteration of the approach. Another complementary approach is to reduce the dimensionality of the problem, either by eliminating unnecessary design variables, or finding compressed representation of the problem under investigation.

Figure 3.1: Diagrams representing use cases UC1 (top), UC2 (middle), and UC3 (bottom).

In contrast, the approach advocated in this thesis is to leverage data generated for problems (previously completed or in-progress) that are related (but distinct) to the current problem of interest. If the transfer learning model of choice can appropriately exploit underlying similarities between design projects, then additional data can become available essentially *for free*. It should be noted, however, that these approaches are complementary with those previously described. In fact, their careful combination has led to promising results in ML works such as [123, 124].

3.3.1 About the Use Cases

In this case study, three use cases are devised that represent scenarios design engineers commonly face in practice and which readily accommodate a data transfer approach. Represented schematically in Figure 3.1, these use cases are as follows:

- **Use Case 1 (UC1):** This use case involves two studies of a particular engine configuration with identical simulator outputs. The primary aim is

to examine the effect of considering an additional input/design parameter (e.g. pressure ratio of the compressor) to the target study.

- **Use Case 2 (UC2):** In this use case, the source and target studies consider identical design parameters and objectives but with different engine configurations. However, it should be noted that the engines are still related (e.g. a turbojet with or without an afterburner).
- **Use Case 3 (UC3):** This use case simulates the scenario where both the engine specifications and the set of design parameters differ between the previous and current studies. In particular, the current study considers one fewer design parameter than the previous study. As with UC2, some underlying relationship between both engines exist.

In the remainder of this section, an empirical assessment is presented that illustrates the possible improvements to surrogate models that are well within reach by simply incorporating transfer learning ideas into the aforementioned use cases.

Table 3.1: Input (design) and output parameters for the source and target studies for UC1, UC2 and UC3.

Input Parameters	UC1		UC2		UC3		Output Parameters
	Source	Target	Source	Target	Source	Target	
	CF6 Turbofan	CF6 Turbofan	J85 Turbojet	CF6 Turbofan	J85 Turbojet	CF6 Turbofan	
Altitude-ft	✓	✓	✓	✓	✓	✓	O1: Net Thrust
Fan Bypass Ratio	✓	✓	✗	✗	✗	✗	O2: Fnet/W
Pressure Ratio (Compressor)	✓	✓	✓	✓	✓	✗	O3: Fuel Flow
Pressure Ratio (Burner)	✓	✓	✓	✓	✓	✓	O4: TSFC
Efficiency (Turbine)	✓	✓	✓	✓	✓	✓	O5: Core Airflow
A8/A2 (Nozzle)	✗	✓	✓	✓	✓	✓	O6: Weight

3.3.2 Benchmark Data Generation

In this study, data for each use case was generated using a popular simulator developed by NASA Glenn Research Center known as EngineSim (v1.8a) [125]. This open-source tool is a quick to evaluate and relatively simple simulator of

aircraft propulsion designed for educational purposes. Additionally, real-world data for use cases 2 and 3 are also provided by Rolls-Royce.

The configuration of the use cases are summarized in Table 3.1. In particular, for UC1, the source and target datasets are generated from a CF6 Turbofan model, with 5 and 6 inputs, respectively. UC2, on the other hand, considers a J85 Turbojet and a CF6 Turbofan for the source and target datasets, respectively, but with 5 identical inputs. In UC3, the source and target datasets are generated from the same pair of (distinct) engine models as with UC2, but 4 and 5 input parameters, respectively. In all use cases, the same set of objectives or model responses are considered.

For these experiments, 500 engine configurations are generated for the source and target datasets using the latin hypercube (LHS) DOE procedure, and normalized to a mean of 0 and standard deviation of 1. GP models are then constructed from these datasets to serve as a surrogate of the EngineSim simulator in order to provide greater flexibility in the evaluation process. All subsequent datasets used for the specific use cases are then sampled from these GP-based approximation surfaces.

3.3.3 Initial Experimental Evaluations

The objective of TL is to improve the learning of a target task using knowledge from a related (but distinct) source task. In order to properly evaluate the achievable improvements of TL, a comparison to a conventional surrogate modelling approach is conducted. For fairness of comparison, the TL methods discussed in this chapter utilize GP (Section 2.3.4.1) as the underlying base-model. This implementation of GP utilizes a squared exponential covariance function, and the hyperparameters of the model are optimized using a conjugate gradient approach with a computation budget of 500 evaluations. For UC1, 20 and 300 instances are sampled using LHS from the target and source surrogates, respectively. For UC2, 20 target and 50 source instances are sampled and, for UC3, 7 target and

Figure 3.2: **LEFT:** $RMSE$ values of GP and NT-GP for each output on UC1. **RIGHT:** $RMSE$ for each output with increasing number of simulations for training a GP model. The red segments show the $RMSE$ of NT-GP trained with 20 (target) simulations, and the red stars highlight the points where GP starts outperforming NT-GP.

50 source instances are sampled. For testing, 200 target instances are sampled. The error is calculated using the root mean squared error (RMSE) measure.

3.3.3.1 Evaluation of Use Case 1

A simple approach to TL is to apply a conventional regression model to the combination of the two datasets. In UC1, this Naïve-Transfer GP (NT-GP) is utilized, but with the 6th (missing) parameter of the source dataset filled with mean of the equivalent input parameter from the target. A comparison of the traditional GP and NT-GP is presented in the left panel of Figure 3.2. The results indicate that NT-GP improves upon GP for every output, with significant improvements in four. This observation suggests that when the functional relationship between the input and output parameters are identical (in this case, the datasets were sampled from identical engines), simply combining the two datasets can result improvements in predictive accuracy.

In order to approach the performance achieved by NT-GP, a larger volume of target data will have to be made available to the GP [126]. The right panel of Figure 3.2 substantiates this observation. The plot shows the RMSE of GP when trained with an increasing number of data points. A red star (*) indicates

Figure 3.3: $RMSE$ values of GP and NT-GP for each output on UC2 (**Left Panel**) and UC3 (**Right Panel**).

the dataset size in which NT-GP begins to be outperformed by GP - providing an estimation of the amount of additional data that must be provided to GP before its performance equals NT-GP. Specifically, NT-GP saves roughly 55 and 40 simulations for outputs 1 and 6, respectively. In the remaining outputs, however, NT-GP saves up to 190 simulations. It should be noted that the impact of such savings depends on the simulation cost. Several hundred simulations saved may be inconsequential for simpler models, but when high fidelity CFD analyses are involved, just a few simulations saved may well have a remarkable effect in the cost.

3.3.3.2 Evaluation of Use Case 2

In UC2, the left panel of Figure 3.3 shows generally inferior results of NT-GP relative to GP. In some cases, the inherent dissimilarities between source and target datasets, when used unaltered, may result in *negative transfer*. However, as previously discussed, the choice of engine configurations for this use case suggests that there are underlying similarities that can be exploited. In fact, many TL methodologies proposed in the literature implicitly assumes that the source and target datasets are related in some way [16]. Consequently, in order to properly utilize available knowledge from the source dataset and avoid negative transfer, approaches that adaptively accounts for the similarities are required [16, 127]. In

Section 3.4, three such adaptive TL approaches are evaluated.

3.3.3.3 Evaluation of Use Case 3

As previously discussed, UC3 differs from UC2 in that the dimensionalities of the source and target datasets are not the same, i.e., the source considers 5 dimensions, while the target considers 4. Consequently, adaptive methodologies intended for homogeneous search spaces are expected to perform poorly. Nevertheless, as an initial evaluation, NT-GP is considered. Similar to UC1, for NT-GP, the 2nd (missing) parameter (Pressure Ratio - Compressor) is manually added and filled with the mean of the 2nd parameter of the source dataset. As is shown in the right panel of Figure 3.3, this approach generally does not result in improved predictive performance - an outcome that is indicative of negative transfer. Accordingly, a novel knowledge adaptation methodology for transfer, termed *Mani-TGP*, that provides the capability to leverage heterogeneous source data in a supervised manner is presented in Section 3.5.

3.4 Adaptive Homogeneous Transfer Learning Approaches

Methodologies that can automatically adapt to source datasets that have varying degrees of similarity to the target problem is an important capability. As has been demonstrated in Section 3.3.3.2, improper transfer of knowledge can often lead to a dramatic decrease in model accuracy. In the following sections, three adaptive TL models for homogeneous transfer will be presented: The *Adaptive Transfer Gaussian Process* [1], the *Meta-heuristic Instance Selection for Transfer*, and the *Transfer Stacking Gaussian Process - Multi-Problem Surrogate*. Note that the latter two are, as far as we know, novel.

3.4.1 Adaptive Transfer GP (AT-GP) [1]

GP provides a means for a non-parametric, flexible modelling and subsequent inference on the set of response functions of a domain of interest. In this particular case study, the domain corresponds to the design outputs and its associated input parameters. The correlations among output values at distinct input locations are captured in the covariance matrix resulting from the evaluation of a covariance function $k(\cdot, \cdot)$ on the set of input pairs. In general, inputs that are close to each other, according to the similarity measure defined by k , will induce high correlations in the output space. Thus, through the choice of covariance function, different properties (e.g. smoothness, length scales, etc.) of the response functions can be modelled. In a scenario with two domains, where one of them is the target, a direct use of k on all the available inputs may lead the model to endorse deceptive dependencies if the source points are not consistent with the correspondence between inputs similarity and output dependencies in the target domain. This suggests that the similarities between the domains should also be taken into account. AT-GP accounts for the relatedness between the source and target domains through a transfer covariance function explicitly capturing their correlation. More precisely, denoting the source and target domains by $\mathcal{S}$ and $\mathcal{T}$ respectively, and given two inputs $\mathbf{x}^{(i)}, \mathbf{x}^{(j)}$ from either domain, AT-GP employs the following covariance function:

$$\ddot{k}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \begin{cases} \lambda k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}), & \mathbf{x}^{(i)} \in X_{\mathcal{T}} \& \mathbf{x}^{(j)} \in X_{\mathcal{S}} \\ & or \mathbf{x}^{(i)} \in X_{\mathcal{S}} \& \mathbf{x}^{(j)} \in X_{\mathcal{T}} \\ k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}), & otherwise. \end{cases} \quad (3.1)$$

where $k(\cdot, \cdot)$ is any valid covariance function and λ is a hyperparameter measuring the correlation between points from source ($\mathcal{S}$) and target ($\mathcal{T}$). Note that, for $\ddot{k}$ to represent a valid covariance function, it must be positive semi-definite. This is fulfilled whenever $-1 \leq \lambda \leq 1$ [1].

The hyperparameter λ , through appropriate learning, captures the similarities between source and target tasks. In particular, if data from a highly correlated (negatively or positively) source is incorporated into the model, the magnitude of λ would be close to 1, resulting in source instances that contribute significantly to model inference. On the other hand, if the magnitude of λ is close to 0, due to the utilization of an unrelated source, then the source instances would be less important. As such, the degree of which transfer occurs is automatically modulated as a result of this hyperparameter.

3.4.2 Meta-heuristic Instance Selection for Transfer (MIST)

Different from ATGP, MIST is a transfer learning strategy that is independent of the choice of the underlying base-model [71]. The primary aim of the approach is to adaptively select the subset of source instances that are most relevant to the target problem of interest. Subsequently, the source subset and the target data are combined using a transfer learning base-model of the user’s choice. In this study, for ease of comparison, the chosen base-model is the NT-GP.

For UC2, for example, given identical DOE sampling approaches for both target and source, the approach taken by MIST is to select the source instances so that the conditional distribution of the source is identical to that of the target. To elaborate, MIST aims to select the subset of the source instances that, in combination with the target instances, forms the best representation of the target distribution. In some cases, MIST may determine that an empty subset is the best available choice; implying that the source data is not relevant to the target problem, and, consequently, transfer does not occur.

During the search process, MIST evaluates the quality of each candidate solution (i.e. a source subset) by calculating the associated generalization error. It should be noted that, as there is a reliance on the choice of the base-model, the objective function landscape of the optimization problem may vary quite significantly. Consequently, to address this ill-defined optimization scenario, MIST

utilizes a meta-heuristic population-based algorithm [128] where a set of candidate solutions (i.e. the population) is iteratively updated until a user-defined stopping criteria is achieved. The aforementioned updates involve the selection of high-performing candidates, followed by the incorporation of these subsets into a model, and, finally, sampling said model to generate new solutions that are evaluated, and added to the population based on solution quality. Examples of such algorithms are genetic algorithms, and differential evolution. In this study, the search aspect of MIST [71] involves the utilization of an estimation of distribution algorithm [129], where the probability distribution of the current population is estimated using a tree-structured Bayesian network which facilitates the capture of order-1 dependencies from promising source instances.

In these experiments, MIST estimates the generalization error using an internal 3-fold cross-validation scheme, with a population size of 400, and a stopping criteria of 100 iterations. These experiments are repeated 20 times in order to properly estimate the capabilities of the proposed algorithm.

3.4.3 Transfer-Stacking GP (TS-GP) - Multi-Problem Surrogate

The transfer-stacking multi-problem surrogate is a model-based supervised transfer learning regression model that can facilitate the incorporation of knowledge from multiple sources. This is achieved through transfer stacking which combines multiple base surrogate models using a meta-regression approach. In this instantiation of the multi-problem surrogate (MPS), the transfer stacking module, shown in Algorithm 1, utilizes a linear meta-regression for this purpose.

Given $D_{\mathcal{T}} = \{X_{\mathcal{T}}, \mathbf{y}_{\mathcal{T}}\} = \{(\mathbf{x}_{\mathcal{T}}^{(i)}, y_{\mathcal{T}}^{(i)})\}_{i=1}^{n_{\mathcal{T}}}$ solutions evaluated so far, together with B pretrained source models, the mixture coefficients of meta-regression $\mathbf{a} = (a_{\mathcal{S},1}, \dots, a_{\mathcal{S},B}, a_{\mathcal{T}})$ are obtained by solving the following optimization problem minimizing the squared error of *out-of-sample* predictions:

Algorithm 1 Transfer Stacking Gaussian Process MPS with Linear Meta-Regression

Input: $D_{\mathcal{T}}$, B Source Models

Output: h_{MPS}

Step 1 - Building Target Model: Learn an initial target Gaussian process model $h_{\mathcal{T}}$ using all of $D_{\mathcal{T}}$.

for $i = 1, 2, \dots, n_{\mathcal{T}}$ **do**

Step 2 - Generating the Out-of-Sample Target Feature Vector ($\hat{\mathbf{y}}_{\mathcal{T}}$): Construct a Gaussian process model tmp_model_i using $\boldsymbol{\theta}$, $X_{\mathcal{T}}[!i]$, and $\mathbf{y}_{\mathcal{T}}[!i]$. Here, $\boldsymbol{\theta}$ represents learned hyperparameters of $h_{\mathcal{T}}$ from Step 1, which are directly reused without retraining. Further, $[!i]$ signifies that the i th instance $(\mathbf{x}_{\mathcal{T}}^{(i)}, y_{\mathcal{T}}^{(i)})$ is dropped from $X_{\mathcal{T}}$ and $\mathbf{y}_{\mathcal{T}}$ following a *leave-one-out scheme*. Thereafter, the target feature $\hat{\mathbf{y}}_{\mathcal{T}}^{(i)}$ is given by the mean of the predictive distribution of tmp_model_i at $\mathbf{x}_{\mathcal{T}}^{(i)}$, conditioned on $X_{\mathcal{T}}[!i]$ and $\mathbf{y}_{\mathcal{T}}[!i]$.

end for

Step 3 - Generating Source Features ($\hat{\mathbf{y}}_{\mathcal{S}}$): For each *Source Model*, calculate the output corresponding to target points in $X_{\mathcal{T}}$. The observed outputs are stored as $\hat{y}_{\mathcal{S},1}^{(i)}, \dots, \hat{y}_{\mathcal{S},B}^{(i)}$ for $i = 1, \dots, n_{\mathcal{T}}$.

Step 4 - Calculating the Meta-Regression Coefficients: Using the target and source features obtained in Steps 2 and 3, respectively, solve the optimization problem shown in Eq. 3.2 to obtain the mixture coefficients ($\mathbf{a}$).

Step 5 - Output: Save $(\mathbf{a}, \text{Source Models}, h_{\mathcal{T}})$ to form h_{MPS} .

$$\begin{aligned}
 \text{Minimize: } SE(\mathbf{a}) &= \sum_{i=1}^{n_{\mathcal{T}}} \left(\sum_{j=1}^B a_{\mathcal{S},j} \hat{y}_{\mathcal{S},j}^{(i)} + a_{\mathcal{T}} \hat{y}_{\mathcal{T}}^{(i)} - y^{(i)} \right)^2 \\
 \text{Subject to: } \sum_{j=1}^B a_{\mathcal{S},j} + a_{\mathcal{T}} &= 1 \\
 a_{\mathcal{S},j} &\geq 0, \text{ for } j = 1, \dots, B \\
 a_{\mathcal{T}} &\geq 0
 \end{aligned} \tag{3.2}$$

where $\hat{y}_{\mathcal{S},1}^{(i)}, \dots, \hat{y}_{\mathcal{S},B}^{(i)}$ are predictions by the source models $h_{\mathcal{S},1}, \dots, h_{\mathcal{S},B}$ at $\mathbf{x}_{\mathcal{T}}^{(i)}$, and $\hat{y}_{\mathcal{T}}^{(i)}$ is an out-of-sample prediction of an initial target model at $\mathbf{x}_{\mathcal{T}}^{(i)}$ - see Algorithm 1 for details. By considering the out-of-sample predictions, the intention is to

leverage the source models to help improve generalization performance. Once the coefficients of the meta-regression model are calculated, the final prediction of the MPS model at an unknown input $\mathbf{x}_{\mathcal{T}}^{(*)}$ is given by the following:

$$\hat{y}(\mathbf{x}_{\mathcal{T}}^{(*)}) = \sum_{j=1}^B a_{\mathcal{S},j} \hat{y}_{\mathcal{S},j}(\mathbf{x}_{\mathcal{T}}^{(*)}) + a_{\mathcal{T}} \hat{y}_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{(*)}) \quad (3.3)$$

3.4.3.1 Source-Target Similarity Capture

The *adaptiveness* of the proposed algorithm is a crucial advantage. This feature provides the capability to, in principle, account for similarities between $\mathcal{S}$ and $\mathcal{T}$ tasks. To elaborate, when a strongly correlated model $h_{\mathcal{S},j}$ (relative to the target) is used in the transfer stacking algorithm, the coefficient $a_{\mathcal{S},j}$ will generally be assigned a large value - providing greater importance to $h_{\mathcal{S},j}$. Conversely, the assigned coefficient will likely be small when the source model in question is irrelevant to the target. Consequently, this adaptive feature provides a means to automatically account for negative transfer, a common problem in transfer learning [16], without the need for the intervention of the practitioner.

To substantiate these claims, herein a theoretical examination of the relationship between the estimated source-target similarity and the learned mixture coefficient values $a_{\mathcal{S},j}$ (for $j = 1, \dots, B$) and $a_{\mathcal{T}}$ for a transfer stacking model is conducted. The analysis is initiated by defining a symmetric $(B+1) \times (B+1)$ correlation matrix C . The (p, q) th element of the matrix is given as $C_{pq} \equiv \sum_{i=1}^{n_{\mathcal{T}}} \epsilon_p^{(i)} \epsilon_q^{(i)}$, where

$$\epsilon_q^{(i)} = \begin{cases} y_{\mathcal{T}}^{(i)} - \hat{y}_{\mathcal{S},q}^{(i)} & \text{if } q \leq B \\ y_{\mathcal{T}}^{(i)} - \hat{y}_{\mathcal{T}}^{(i)} & \text{for } q = B + 1 \end{cases} \quad (3.4)$$

is the prediction error of the q th surrogate model on the i th training input. For the mathematical derivations below, following assumptions are made that simplify the analysis.

- The prediction error of the q th surrogate model has zero mean when averaged across the $n_{\mathcal{T}}$ target inputs, i.e., $\sum_{i=1}^{n_{\mathcal{T}}} \epsilon_q^{(i)} = 0$. Note that no restriction is placed on $|\epsilon_q^{(i)}|$.
- The prediction errors of all (source + target) surrogate models are mutually uncorrelated. In other words, $C_{pq} = 0$, $\forall (p, q)$ s.t. $p \neq q$.

Based on Equation 3.4, the similarity between the q th source and the target problem are interpreted to be represented by the generalization error of the q th model measured on the entire (target) training set, i.e., $C_{qq} = \sum_{i=1}^{n_{\mathcal{T}}} [\epsilon_q^{(i)}]^2$. If C_{qq} is large then the q th source model is intuitively expected to be less relevant to the target. On the other hand, if C_{qq} is small, then the q th source is likely closely related to the target.

Following Equation 3.3, and based on the constraints of Equation 3.2, the squared prediction error of the MPS model on the i th out-of-sample training input can be expressed in terms of the $\epsilon_q^{(i)}$'s as:

$$(y_{\mathcal{T}}^{(i)} - \hat{y}_{\mathcal{T}}^{(i)})^2 = \left(\sum_{q=1}^{B+1} a_q \epsilon_q^{(i)} \right)^2 \quad (3.5)$$

where,

$$a_q = \begin{cases} a_{\mathcal{S},q} & \text{if } q \leq B \\ a_{\mathcal{T}} & \text{for } q = B + 1 \end{cases}. \quad (3.6)$$

Enforcing the assumption that $C_{pq} = 0$, $\forall p \neq q$, the squared error (SE) of MPS measured over the entire training set, as given in Equation 3.2, can be rewritten as [130]:

$$SE(\mathbf{a}) = \sum_{q=1}^{B+1} (a_q^2 C_{qq}). \quad (3.7)$$

Using the method of Lagrange multipliers to solve for the coefficients of the transfer stacking model, we have,

$$\frac{\partial}{\partial a_p} \left[\sum_{q=1}^{B+1} (a_q^2 C_{qq}) - 2\lambda \left(\sum_{q=1}^{B+1} a_q - 1 \right) \right] = 0. \quad (3.8)$$

Simplifying this equation, we arrive at the condition:

$$a_p = \frac{\lambda}{C_{pp}}. \quad (3.9)$$

This result provides the insight that the coefficient a_p of the p th surrogate model is inversely proportional to the generalization error of the model calculated on the entire (target) dataset. Equivalently, it can be said that as the source-target similarity decreases, the value of the corresponding mixture coefficient a_p also decreases. As a result, its impact on the overall generalization performance of the transfer stacking model is automatically reduced.

3.4.3.2 Stepwise description of transfer stacking

The steps for constructing a TS-GP MPS model utilizing a linear meta-regressor, as outlined in Algorithm 1, is described below. It should be noted that this procedure is similar to that of the classical stacked generalization algorithm [131], but extended to account for the incorporation of models from related but distinct sources.

Initially, a GP model $h_{\mathcal{T}}$ is built from $D_{\mathcal{T}}$ (Section 2.3.4.1). Subsequently, in the second step, a target feature vector ($\hat{\mathbf{y}}_{\mathcal{T}}$), comprising of out-of-sample predictions $\hat{y}_{\mathcal{T}}^{(i)}$ at $\mathbf{x}_{\mathcal{T}}^{(i)}$, for $i = 1, \dots, n_{\mathcal{T}}$, is generated. This is achieved through a variant of the *leave-one-out* cross-validation scheme [15]. To elaborate, the out-of-sample prediction at $\mathbf{x}_{\mathcal{T}}^{(i)}$ is given by the mean of the predictive distribution of an intermediate Gaussian process model tmp_model_i which employs the same hyperparameter settings as $h_{\mathcal{T}}$ (without retraining). The only distinction between tmp_model_i and $h_{\mathcal{T}}$ is that the i th instance $(\mathbf{x}_{\mathcal{T}}^{(i)}, y_{\mathcal{T}}^{(i)})$ is left-out from the dataset of tmp_model_i . Thus, for all practical purposes, the input $\mathbf{x}_{\mathcal{T}}^{(i)}$ is unseen to tmp_model_i , with its predictive distribution conditioned on the remaining $n_{\mathcal{T}} - 1$

samples. The rationale behind using this leave-one-out scheme is that it provides legitimate out-of-sample predictions without the need for costly retraining of the intermediate Gaussian process models. Note that the theoretical validity of the approach stems from the observation in [15] that the log marginal likelihood (used for tuning the hyperparameters) is typically not affected significantly by the exclusion of a single (left-out) data point.

In the third step, source features ($\hat{\mathbf{y}}_{\mathcal{S}}$), i.e. the predictions $\hat{y}_{\mathcal{S},1}^{(i)}, \dots, \hat{y}_{\mathcal{S},B}^{(i)}$ by the source models $h_{\mathcal{S},1}, \dots, h_{\mathcal{S},B}$ on the points in $X_{\mathcal{T}}$, are generated. With $\hat{\mathbf{y}}_{\mathcal{S}}$ and $\hat{\mathbf{y}}_{\mathcal{T}}$ in hand, the mixture coefficients ($\mathbf{a}$) are then calculated in step four by solving Equation 3.2 using any convex optimization method. Finally, in step five, the mixture coefficients, source models, and target model are saved to form h_{MPS} , which is subsequently used for the target optimization task.

3.4.3.3 Salient feature of transfer stacking

It is important to note that the calculation of the acquisition function used in Bayesian optimization (see Section 2.3.4), generally requires the standard deviation of the predictive distribution in addition to the mean. In this regard, an important feature of transfer stacking with Gaussian process based surrogates and linear meta-regression emerges from the following fact:

Fact: Let $Y_1, Y_2, \dots, Y_N$ be independent normal random variables that are specified by means $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N$ and variances $\hat{\sigma}_1^2, \hat{\sigma}_2^2, \dots, \hat{\sigma}_N^2$. Then, any linear combination of these variables, given by $\sum_{i=1}^N a_i Y_i$, is also normally distributed as $\mathcal{N}(\sum_{i=1}^N a_i \hat{y}_i, \sum_{i=1}^N a_i^2 \hat{\sigma}_i^2)$.

Accordingly, the posterior variance of the predictive distribution at $\mathbf{x}^{(*)}$ obtained via transfer stacking is given by:

$$\hat{\sigma}^2(\mathbf{x}_{\mathcal{T}}^{(*)}) = \sum_{j=1}^B a_{\mathcal{S},j}^2 \hat{\sigma}_{\mathcal{S},j}^2(\mathbf{x}_{\mathcal{T}}^{(*)}) + a_{\mathcal{T}}^2 \hat{\sigma}_{\mathcal{T}}^2(\mathbf{x}_{\mathcal{T}}^{(*)}) \quad (3.10)$$

where $\hat{\sigma}_{S,1}(\mathbf{x}_{\mathcal{T}}^{(*)}), \dots, \hat{\sigma}_{S,B}(\mathbf{x}_{\mathcal{T}}^{(*)})$ and $\hat{\sigma}_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{(*)})$ are the standard deviations of the predictive distributions of the source and target Gaussian process models, respectively, at $\mathbf{x}_{\mathcal{T}}^{(*)}$.

3.5 An Adaptive Heterogeneous Transfer Learning Approach

The capability to transfer knowledge across heterogeneous domains is challenging to realize but extremely valuable. In real world scenarios, particularly in aircraft design, it is common to evaluate subsets of design parameters in order to simplify the analysis - leading to a wide variety of datasets in which the dimensionalities do not necessarily coincide with the current problem of interest. In the following section, a novel, adaptive TL approach that can facilitate heterogeneous transfer, i.e. the *Manifold Transfer Gaussian Process*, is presented.

3.5.1 Manifold Transfer Gaussian Process (*Mani*-TGP)

A standard GP is a stochastic non-parametric approach for regression. It describes a distribution over functions as:

$$f \sim \mathcal{GP}(m, k) \quad (3.11)$$

where m is the mean (typically $m = 0$) and k is a covariance kernel. Given the dataset $\mathcal{D} = \{X, \mathbf{y}\} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$, the predictive distribution of the GP at a new data point $\mathbf{x}^{(*)}$ is given by:

$$p(f^{(*)}|X, \mathbf{y}, \mathbf{x}^{(*)}) \sim \mathcal{N}(\hat{\mu}(\mathbf{x}^{(*)}), \hat{\sigma}^2(\mathbf{x}^{(*)})) \quad (3.12)$$

$$\hat{\mu}(\mathbf{x}^{(*)}) = \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{y} \quad (3.13)$$

$$\hat{\sigma}^2(\mathbf{x}^{(*)}) = k_{**} - \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{k}_* \quad (3.14)$$

where, $K(X, X)$ is an $n \times n$ matrix whose ij th entry is $k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$; $\mathbf{k}_* = K(\mathbf{x}^{(*)}, X)$; $k_{**} = k(\mathbf{x}^{(*)}, \mathbf{x}^{(*)})$.

In the case of transfer learning, a source domain $\mathcal{S}$ is considered, typically with a sufficient amount of training data, to be used to improve the learning in a related target domain $\mathcal{T}$ in which there is only limited data. Let $\mathcal{D}_{\mathcal{S}} = \{X_{\mathcal{S}}, \mathbf{y}_{\mathcal{S}}\} = \{(\mathbf{x}_{\mathcal{S}}^{(i)}, y_{\mathcal{S}}^{(i)})\}_{i=1}^{n_{\mathcal{S}}}$ and $\mathcal{D}_{\mathcal{T}} = \{X_{\mathcal{T}}, \mathbf{y}_{\mathcal{T}}\} = \{(\mathbf{x}_{\mathcal{T}}^{(i)}, y_{\mathcal{T}}^{(i)})\}_{i=1}^{n_{\mathcal{T}}}$ denote the source and target datasets, respectively. The combination of the two datasets is represented as $X = (X_{\mathcal{S}}, X_{\mathcal{T}})$ and $\mathbf{y} = (\mathbf{y}_{\mathcal{S}}, \mathbf{y}_{\mathcal{T}})$. As proposed in [1], a promising approach to achieve *adaptive* knowledge transfer across distinct learning tasks is to simply redefine the covariance function $k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ in the following manner:

$$\ddot{k}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \begin{cases} \lambda k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}), & \mathbf{x}^{(i)} \in X_{\mathcal{T}} \text{ \& } \mathbf{x}^{(j)} \in X_{\mathcal{S}} \\ & or \ \mathbf{x}^{(i)} \in X_{\mathcal{S}} \text{ \& } \mathbf{x}^{(j)} \in X_{\mathcal{T}} \\ k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}), & otherwise. \end{cases} \quad (3.15)$$

Here $-1 \leq \lambda \leq 1$ is a parameter measuring source-target similarity, and $k(.,.)$ can be any valid kernel. Note that this equation is identical to Equation 3.1. At this juncture, it is important to note that for *any* valid covariance kernel to be *directly* used within the aforementioned approach, the feature spaces $\mathcal{X}_{\mathcal{S}}$ and $\mathcal{X}_{\mathcal{T}}$, from which the datasets $X_{\mathcal{S}}$ and $X_{\mathcal{T}}$ are drawn, must be homogeneous; which implies that their dimensionality must be the same. This observation leads us to the exploration of manifold feature transformations as a way to enable transfer learning across problems that may belong to heterogeneous domains.

3.5.1.1 Source Feature Transformation

Let $\mathcal{X}_{\mathcal{S}} \subset \mathbb{R}^{d_{\mathcal{S}}}$ and $\mathcal{X}_{\mathcal{T}} \subset \mathbb{R}^{d_{\mathcal{T}}}$. Further, it is assumed that $d_{\mathcal{S}} \neq d_{\mathcal{T}}$. In such a setting, the alignment of the source features with that of the target calls for some

form of feature mapping (M). In this paper, M is considered to be realized as a deterministic, parametrized function. In particular, a new *latent* representation is induced for the source input features through an l -layer neural network. The i th layer of the network performs the following transformation:

$$\psi_i(Z_i) = \phi(W_i \cdot Z_i^T + \mathbf{b}_i)^T \quad (3.16)$$

where, Z_i is the input of the i th layer (note: $Z_1 = X_S$, and $Z_{i+1} = \psi_i(Z_i)$), W_i is the weight matrix, $\mathbf{b}_i$ is the bias vector, and ϕ is some activation function. With this, the overall transformation can be written as the following composition:

$$X'_S = M(X_S) = (\psi_l \circ \dots \circ \psi_1)(X_S). \quad (3.17)$$

The new representation of the source features is guaranteed to have the same dimensionality as the target by requiring that the matrix W_l to have $d_{\mathcal{T}}$ rows, and $\mathbf{b}_l$ to be a $d_{\mathcal{T}}$ -dimensional vector. In other words, the l th layer of the neural network must have $d_{\mathcal{T}}$ hidden neurons.

In light of the feature transformation procedure, the transfer kernel introduced in Equation 3.15 is modified as:

$$\tilde{k}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \begin{cases} \lambda k(M(\mathbf{x}^{(i)}), \mathbf{x}^{(j)}), & \text{case } A \\ \lambda k(\mathbf{x}^{(i)}, M(\mathbf{x}^{(j)})), & \text{case } B \\ k(M(\mathbf{x}^{(i)}), M(\mathbf{x}^{(j)})), & \text{case } C \\ k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}), & \text{otherwise} \end{cases} \quad (3.18)$$

where, case $A := (\mathbf{x}^{(i)} \in X_S; \mathbf{x}^{(j)} \in X_{\mathcal{T}})$; case $B := (\mathbf{x}^{(i)} \in X_{\mathcal{T}}; \mathbf{x}^{(j)} \in X_S)$; case $C := (\mathbf{x}^{(i)}, \mathbf{x}^{(j)} \in X_S)$.

With this, the covariance matrix of the *Mani*-TGP model takes the following form:

$$\tilde{K} = \begin{pmatrix} K_{SS} & K_{S\mathcal{T}} \\ K_{\mathcal{T}S} & K_{\mathcal{T}\mathcal{T}} \end{pmatrix} \quad (3.19)$$

where K_{SS} and $K_{\mathcal{T}\mathcal{T}}$ are the covariance matrices of the data in the source and target, respectively, and $K_{S\mathcal{T}} (= K_{\mathcal{T}S}^T)$ is the covariance matrix across tasks.

3.5.1.2 Learning the *Mani*-TGP Model

Training the *Mani*-TGP model involves *jointly* learning the following set of parameters:

$$\theta = \{\theta_{\tilde{k}}, \sigma_S^2, \sigma_{\mathcal{T}}^2, \lambda, W_1, \dots, W_l, \mathbf{b}_1, \dots, \mathbf{b}_l\}. \quad (3.20)$$

Here, in addition to the parameters previously defined, $\theta_{\tilde{k}}$ represents the parameters of the modified transfer kernel, and $\sigma_S^2, \sigma_{\mathcal{T}}^2$ are the source and target measurement noise variances, respectively. In order to optimize the target data distribution under the influence of the source, the following conditional probability is considered [1]:

$$p(\mathbf{y}_T | \mathcal{D}_S, X_{\mathcal{T}}) \sim \mathcal{N}(\boldsymbol{\mu}_{\mathcal{T}}, C_{\mathcal{T}}) \quad (3.21)$$

where,

$$\begin{aligned} \boldsymbol{\mu}_T &= K_{\mathcal{T}S}(K_{SS} + \sigma_S^2 I)^{-1} \mathbf{y}_S \\ C_T &= (K_{\mathcal{T}\mathcal{T}} + \sigma_{\mathcal{T}}^2 I) - K_{\mathcal{T}S}(K_{SS} + \sigma_S^2 I)^{-1} K_{S\mathcal{T}} \end{aligned} \quad (3.22)$$

The resultant maximization of the (log) marginal likelihood of the observed target data is formulated as:

$$\text{maximize: } -\frac{1}{2} \ln |C_{\mathcal{T}}| - \frac{1}{2} (\mathbf{y}_T - \boldsymbol{\mu}_T)^T C_{\mathcal{T}}^{-1} (\mathbf{y}_T - \boldsymbol{\mu}_T). \quad (3.23)$$

Alternatively, a leave-one-out (LOO) log predictive probability can be used for learning the *Mani*-TGP hyperparameters [110]:

maximize:

$$\begin{aligned} & \sum_{i=1}^{n_{\mathcal{T}}} \ln p(y_{\mathcal{T}}^{(i)} | X_{\mathcal{T}}, \mathbf{y}_{\mathcal{T}}^{(-i)}, \mathcal{D}_{\mathcal{S}}, \theta) \\ &= \sum_{i=1}^{n_{\mathcal{T}}} \left(-\frac{1}{2} \ln \sigma_{LOO}^{(i)2} - \frac{(y_{\mathcal{T}}^{(i)} - \mu_{LOO}^{(i)})^2}{2\sigma_{LOO}^{(i)2}} - \frac{1}{2} \ln 2\pi \right) \end{aligned} \quad (3.24)$$

where the notation $\mathbf{y}_{\mathcal{T}}^{(-i)}$ refers to all target outputs except for instance i . Here, the LOO cross-validation (CV) predictive mean $\mu_{LOO}^{(i)}$ and variance $\sigma_{LOO}^{(i)2}$ are computed according to Eqs. 3.26 and 3.27, with training sets $\{X_{\mathcal{T}}^{(-i)}, \mathbf{y}_{\mathcal{T}}^{(-i)}\}$ and $\mathcal{D}_{\mathcal{S}}$.

To reduce the computational cost, the LOO-CV predictive mean and variance can be calculated using the inverse of the covariance matrix $(\tilde{K} + \Lambda)$, where $\Lambda = \begin{pmatrix} \sigma_{\mathcal{S}}^2 I_{n_{\mathcal{S}}} & 0 \\ 0 & \sigma_{\mathcal{T}}^2 I_{n_{\mathcal{T}}} \end{pmatrix}$, with a single row and column removed in turn. Specifically, using the method of inversion by partitioning,

$$\begin{aligned} \mu_{LOO}^{(i)} &= y_{\mathcal{T}}^{(i)} - [(\tilde{K} + \Lambda)^{-1} \mathbf{y}]_i / [(\tilde{K} + \Lambda)^{-1}]_{ii} \\ \sigma_{LOO}^{(i)2} &= 1 / [(\tilde{K} + \Lambda)^{-1}]_{ii}. \end{aligned} \quad (3.25)$$

where $[(\tilde{K} + \Lambda)^{-1}]_{ii}$ is the $(i + n_{\mathcal{S}})$ th element along the $(\tilde{K} + \Lambda)^{-1}$ diagonal, and $[(\tilde{K} + \Lambda)^{-1} \mathbf{y}]_i$ is the $(i + n_{\mathcal{S}})$ th element of the $(\tilde{K} + \Lambda)^{-1} \mathbf{y}$ vector.

Note that while employing shallow networks for source feature transformations, model parameter combinations that perform consistently can be found with off-the-shelf optimizers [132]. As a simplified alternative, in the spirit of the random vector functional-link net [133], in the numerical studies, the case in which the parameters $\{W_1, \dots, W_l, b_1, \dots, b_l\}$ of the feature mapping M are randomly initial-

ized and then fixed is considered; such that only the parameter set $\{\theta_k, \sigma_S^2, \sigma_T^2, \lambda\}$ need be optimized via Eqs. 3.23 or 3.24.

3.5.1.3 *Mani*-TGP Predictions

For a given *Mani*-TGP model, the mean ($\tilde{\mu}$) and variance ($\tilde{\sigma}^2$) of the predictive distribution $p(f^{(*)}|\mathcal{D}_S, \mathcal{D}_T, \mathbf{x}_T^{(*)})$ is similar to that of a standard GP.

$$\tilde{\mu}(\mathbf{x}_T^{(*)}) = \tilde{\mathbf{k}}_*^T (\tilde{K} + \Lambda)^{-1} \mathbf{y} \quad (3.26)$$

$$\tilde{\sigma}^2(\mathbf{x}_T^{(*)}) = \tilde{k}_{**} - \tilde{\mathbf{k}}_*^T (\tilde{K} + \Lambda)^{-1} \tilde{\mathbf{k}}_* \quad (3.27)$$

It is worth highlighting that the effect of source feature transformation can be more than merely addressing the dimensionality mismatch between the target and source. In fact, the transformation may lead to a new feature representation that is better suited for unveiling the hidden source-target similarities even in the case of homogeneous transfer optimization.

3.5.1.4 Exploratory Experiments with Toy Functions

In this section, two common synthetic benchmark functions in optimization literature (i.e. Rosenbrock and Rastrigin [134,135]) are used to examine three aspects of the proposed *Mani*-TGP. Specifically, the response of the model when (1) source and target tasks are different, (2) source and target tasks are identical, and (3) random features are appended to an identical source task, are investigated.

For these experiments and the ones in subsequent sections of this chapter, the baseline GP is obtained from the *gpml* package [136]. In the current implementation of *Mani*-TGP, scarcely available data in typical BO applications is catered to by evolving (small) single-layer neural networks with rectified linear activation function for the source feature transformations. Additionally, randomized feature transformations are also considered (i.e. randomized weights for the single-layer

neural network with a sigmoid activation function) as an alternative strategy to reduce learning complexity. Both the standard GP model and *Mani*-TGP employ the square-exponential kernel. Furthermore, the traditional (log) marginal likelihood (LML - Eq. 3.23) as well as the LOO log predictive probability (LOO-LPP - Eq. 3.24) as alternative criteria is considered for tuning the *Mani*-TGP hyperparameters.

First, the scenario where the source and target functions (2D Rosenbrock with 100 samples and 2D Rastrigin with 20 samples, respectively) are different is considered. In this case, *Mani*-TGP learns a λ of ~ 0.75 . In contrast, when source and target samples are drawn from the same function (i.e., 2D Rastrigin), a λ of ~ 0.98 is learned. This result suggests that *Mani*-TGP can correctly identify and, subsequently, regulate the contributions of the source dataset depending on its similarity to the target.

Further, for scenarios where both the source and target domains are exactly the same, the identity matrix can be consistently learned as the feature transformation, at least in the low-dimensional 2D/3D cases. Furthermore, near zero neural network weights can be inferred for any additional randomly inserted source features that have no effect on the output values.

3.6 Revisiting the Use Cases

3.6.1 Use Case 2 - Transferring Across Homogeneous Domains

As the top panel of Figure 3.4 shows, the three adaptive homogeneous TL methods considered here are able to outperform GP and, more noticeably, NT-GP in almost all the cases. TS-GP achieves the largest and most consistent improvement in all outputs. MIST, on the other hand, shows the second largest improvement of the three adaptive TL methodologies - showing superior performance over standard GP in 4 of 6 outputs with roughly equivalent results in 2. Finally, AT-GP exhibits improvements in 2 of 6 outputs, performs equally well in 1, but is marginally worse

Figure 3.4: $RMSE$ values of GP, NT-GP, AT-GP, MIST and TS-GP for each output on UC2 for the EngineSim dataset (**top panel**) and the real world dataset (**bottom panel**). The results are averaged across 20 independent runs.

in 3.

Of particular note is the performance of the adaptive TL methodologies on outputs where NT-GP is an improvement over standard GP - indicating that the objective landscapes of the source and target have a larger degree of similarity. With the exception of TS-GP, both MIST and AT-GP do not successfully outperform NT-GP in such cases, but still outperforms GP. Conversely, in cases where NT-GP is far worse than GP - indicating that the source-target similarities are small, the performance of the adaptive TL methodologies range from only slightly worse to markedly better. These results suggests that the approaches are, generally, able to leverage source knowledge where applicable to provide superior predictive performance and minimize the effects of negative transfer otherwise.

The top left panel of Figure 3.5 presents the averaged proportion of the source dataset selected after the search in MIST for each output. As can be observed, the percentage does not increase further than about 60%, suggesting that a portion of the source data may not be viable for reuse, while there indeed exists a significant subset that is beneficial. Along a similar vein, the bottom left panel of Figure 3.5 shows the averaged transfer stacking coefficient values for the TS-GP model for each output. The relative magnitudes of the source coefficient roughly coincides with the proportions observed for MIST. Further, it can be observed that for outputs 3 and 6, where NT-GP outperforms standard GP, the two methodologies correctly assigns a greater weight or proportion to the source data.

These figures provide a means to explain the poor performance of NT-GP in outputs 1, 2, 4 and 5. In this study, the primary difference between NT-GP and MIST is the choice of source instances that are reused during the GP model training; i.e. the former utilizes all available source data, while MIST intelligently chooses which to keep. This result suggests that by the simple appropriate selection of source instances, the negative transfer can be made positive, leading to dramatic improvements in surrogate modelling. Overall, this result further reinforces the need for the investigation of *what knowledge to transfer*.

Figure 3.5: **TOP ROW:** Percentage of source data employed (dark colored) for training an NT-GP model with MIST for each output in UC2 for the EngineSim (TOP LEFT) and Real World (TOP RIGHT) datasets. **BOTTOM ROW:** Transfer stacking coefficients for the TS-GP model for each output in UC2 for the EngineSim (BOTTOM LEFT) and Real World (BOTTOM RIGHT) datasets.

3.6.2 Use Case 3 - Transferring Across Heterogeneous Domains

As shown in the left panel of Figure 3.6, in 4 of 6 of the objectives, the *Mani*-TGP models were able to achieve noticeably superior generalization performance relative to traditional GP for the EngineSim dataset. As seen previously, NT-GP proved to be inferior in all but objective 3 where it showed relatively similar performance to other models. Further observations also indicated that the Log Marginal Likelihood (LML) function (Eq. 3.23) performed better than the LOO Log Predictive Probability (LOO-LPP) function (Eq. 3.24) in 4 of 6 cases with learned M , but only 2 of 6 cases with random M . However, a small percentage of the *Mani*-TGP runs using the LML function and learned M were unstable; an indication of the difficulty of the learning problem when using the aforementioned criteria, particularly when optimizing a large number of parameters.

3.7 Results on Real-World Aircraft Engine Data

In order to further explore their applicability, the TL methods above are tested on real-world data provided by Rolls-Royce. The datasets was generated from data sampled using a latin hypercube DOE procedure (LHS) from two different engines, where the number of source and target simulations were 274 and 253 respectively. Both datasets have the same 5 input parameters and 6 outputs. As with the EngineSim datasets, GP regression models are built for each output of the source and target datasets to allow for greater flexibility in the evaluation process. All subsequent datasets generated for UC2 and UC3 are then sampled from these GP-based approximation surfaces.

For UC2, 15 and 50 data points were sampled using LHS for the target and source datasets, respective. For testing, 200 data points were sampled from the target models. The experiments were repeated 20 times. As is shown in the bottom panel of Figure 3.4, the significant difference between the performance of GP and NT-GP is a further indication that a more rigorous and adaptive approach

is required. Note that this disparity is much larger than was observed in UC2 on EngineSim data (top panel of Figure 3.4). Further evidence of this fact is presented in the right column of Figure 3.5, where relatively small proportions of source data are selected by MIST and small transfer stacking coefficients were assigned to the source model in TS-GP. In spite of this, the adaptive methods clearly overcome the deficiencies of NT-GP. As before, the performance of MIST is moderate, exhibiting superior results in some objectives, but marginally worse than GP in others. AT-GP, however, is able to achieve equivalent or superior performance in all but the 3rd objective. Finally, TS-GP shows a dramatic increase in the accuracy of the predictions for all the outputs.

It should be noted that while the strategies employed by MIST can be successful in some cases, a clear problem with the approach is that it relies on a wrapper strategy in order to find the (sub)optimal subset of source data. While this may be advantageous if the goal is to minimize the prediction error, it also limits the scalability of the method. At the other end of the spectrum, TS-GP transfers knowledge in the form of pre-trained source models, which is a strategy that is relatively inexpensive. This factor is an important consideration when evaluating strategies if efficiency is an issue.

In UC3, 7 and 50 data points were sampled for the target and source datasets, respectively. As with UC2, 200 data points were sampled from the target models for testing. To simulate the UC3 scenario, where the dimensionality of the domains differ, when sampling from the target GP model, the 5th parameter was fixed to the mean of the 5th parameter of the original target dataset. Consistent with the results of UC3 for EngineSim, as shown in the right panel of Figure 3.6, NT-GP is a generally inferior approach, suggesting that the source-target similarities are low. Yet, even in this case, *Mani*-TGP is able to show superior or equivalent performance relative to GP, adaptively leveraging source data to improve the predictive performance of the model. A closer inspection indicates that the learned M outperforms random M in 10 of 12 cases. However, as with the EngineSim

Figure 3.6: $RMSE$ values of GP, NT-GP, *Mani*-TGP for each output on UC3 for the EngineSim dataset (**left panel**) and the real-world dataset (**right panel**). Different configurations of *Mani*-TGP are examined, namely, (1) the LOO log predictive probability (Eq. 3.24) and log marginal likelihood (Eq. 3.23) criteria for hyperparameter tuning and (2) learned and random feature transformation (RT). The results are averaged across 20 independent runs.

dataset, instability is observed for the LML function with learned M . Consequently, in the heterogeneous optimization experiments that follow (Chapter 5), the LOO-LPP criteria is used to learn model hyperparameters.

In light of the promising outcomes of these experiments, the investigation of various refinement options, such as a more cost effective alternative to the wrapper strategy of MIST, become an attractive avenues for future work. In general, the field of TL is ripe with various algorithms with different strengths and weaknesses. Aircraft surrogate modelling could be boosted further by leveraging these techniques to address more advanced scenarios or by designing tailored TL methods upon aircraft design-specific domain knowledge.

3.8 Summary

Transfer learning aims to utilize data from distinct (but possibly related) sources as a means of enhancing the approximation of a target function. In aircraft design, these source domains could refer to design projects that have been completed or are in-progress. While a great deal of work have been presented in the literature

regarding various applications of these “advanced” ML techniques, relatively few efforts are in the domain of aircraft design. In this chapter of the dissertation, in partial fulfilment of Research Objectives 1 through 5 (see Section 1.2), an investigation of adaptive transfer learning for complex engineering design, specifically aircraft engine design, has been presented.

Empirical observations indicate that simplistic strategies (e.g. concatenating the source and target datasets) can work in certain situations, while others require more robust, adaptive strategies. Four such methodologies were described in some detail: Adaptive Transfer GP, Metaheuristic Instance Selection for Transfer (MIST), Transfer Stacking GP Multi-Problem Surrogate (TSGP) and Manifold Transfer GP (Mani-TGP) - from which the latter 3 are novel. In particular, the TSGP Multi-Problem Surrogate provides the capability to transfer knowledge from multiple sources (Research Objective 3 - Section 1.2), and the Mani-TGP model facilitates transfer across heterogeneous domains (Research Objective 4 - Section 1.2). The surrogate models augmented by TL methodologies have been shown to outperform its traditional counterparts in several surrogate model building use cases within the domain of aircraft engine design. In subsequent chapters, these TL methodologies are evaluated within the context of optimization in order to evaluate its influence on convergence characteristics.

Chapter 4

Transfer Evolutionary Multiobjective Optimization of Computationally Expensive Problems with Multi-Problem Surrogates

4.1 Chapter Goals

In most real-world settings, designs are often gradually adapted and improved over time. Consequently, there exists knowledge from distinct (but possibly related) design exercises, which have either been previously completed or are currently in-progress, that may be leveraged to enhance the optimization performance of a particular target optimization task of interest. In such environments,

Portions of this chapter are derived from the following publication: Alan Tan, Yew-Soon Ong, Abhishek Gupta, and Chi-Keong Goh. Multi-problem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems. *IEEE Transactions on Evolutionary Computation*, 23(1):15–28, 2017 [21].

vast amounts of related information can become available at various stages of the search process corresponding to some ongoing target optimization exercise. Further, the complexity of real world problems generally necessitates the involvement of numerous potentially competing objectives. Accordingly, as part of the Research Objectives 1, 2 and 3 (see Section 1.2), the goal of this chapter is the investigation of an *adaptive knowledge reuse* framework for surrogate-assisted multiobjective optimization of computationally expensive problems that provides the capability to spontaneously transfer multiple learned models across problems, facilitating *efficient global optimization*.

4.2 Introduction

The quality of a solution is often a function of more than a single criterion in many real-world scenarios. These multiobjective optimization problems (MOPs) result in a series of solutions that represent a trade-off between the objectives, i.e. Pareto optimal solutions. The ubiquity of such problems has led to significant interest, among evolutionary computation researchers and otherwise, towards the development of algorithms that can effectively deal with multiple objectives simultaneously. Evolutionary algorithms (EAs) have been generally accepted as effective approaches to solve such problems in parallel, converging on the entire Pareto optimal solution set concurrently [57, 137, 138]. Several commonly used multiobjective evolutionary algorithms (MOEAs) include PAES [96], HypE [99], SPEA2 [97], MOEA/D [98], DMEA I & II [139, 140], NSGA-II [95], etc.

A common issues with MOEAs as well as other population-based metaheuristic approaches, is that it generally requires a significant amount of function evaluations (FEs) before a suitably optimal solution can be found. This is a problem, in particular, for real-world problems where each evaluation can be extremely expensive. A typical solution is the use of surrogate models (or meta-models) to approximate these expensive functions, and, subsequently, search these ap-

proximations - resulting in a more efficient search process (e.g. Efficient Global Optimization - EGO [15]). One such approach based on the EGO algorithm, but designed specifically for MOPs, is ParEGO [141]. In the spirit of *Bayesian optimization* (BO), the approach leverages the probabilistic nature of GP regression models to induce a cheap acquisition function (e.g. expected improvement), balancing exploration and exploitation, that is subsequently searched. As described in Section 2.3.4, BO provides a principled approach towards the selection of points to be evaluated by the expensive function, providing greater efficiency relative to the randomized schemes of traditional MOEAs.

A related idea was also presented in [108], where Gaussian process models in conjunction with the MOEA/D algorithm were used to process several candidate solutions in parallel to converge towards the Pareto set. More recently, in [142], a Gaussian process surrogate-assisted extension of the reference vector-guided EA was shown to have particular efficacy for computationally expensive *many-objective* problems. Alternatively, in [143], the S-metric selection-based EGO (SMS-EGO) was proposed to optimize the S-metric using covariance matrix adaptation evolution strategies. In [144], radial basis functions were preferred to create the surrogate response surfaces. Notably, by selecting points for evaluation through a combination of various metrics, the balance between exploration and exploitation was maintained. Additionally, a large amount of works have been proposed that exploit function approximation in some form within the framework of evolutionary computation; summarized in [145, 146].

While surrogate-assisted optimization approaches have achieved promising results, these methods only consider data gathered from the target problem of interest. In actuality, there exists many other rich sources of information that can be leveraged to augment the search process. This approach is particularly beneficial for computationally expensive problems, where data is scarce and difficult to obtain, especially during the initial stages of the search (i.e. the *cold start* problem). Fortunately, real-world problems draw heavily from past design efforts,

Figure 4.1: Illustration of several design teams utilizing TEMO-MPS in their respective design exercises. These teams are based at various geographical locations and progress at different rates. Nevertheless, knowledge transfer between teams occurs - periodically uploading problem-specific data and leveraging available knowledge through the multi-problem surrogate of the TEMO-MPS framework.

leading to a plethora of closely related knowledge that can be leveraged in order to improve optimization efficiency. In machine learning, the concept of knowledge transfer between one or more source tasks in order to enhance the learning of some target task is known as transfer learning [16]. This field of study has seen considerable success in many application areas. In optimization however, this concept is relatively unexplored. As such, in this section of the thesis, an efficient approach termed *Transfer Evolutionary Multiobjective Optimization utilizing Multi-Problem Surrogates* (TEMO-MPS) is proposed that provides the capability to leverage knowledge from multiple surrogate models from related but distinct design tasks in order to enhance the current problem under investigation.

The key feature of the proposed approach, as shown in Figure 4.1, is its adaptiveness. More specifically, surrogate models from a collection of sources, e.g. previously completed or in-progress design projects, are taken into consideration; some of which are more related to the target problem than others. The multi-problem surrogate (MPS) will, in such settings, automatically modulate the extent of which transfer occurs, and, in principle, guarantee the success of the search.

Notably, the viability of the proposition from a practical point-of-view is supported by the widespread availability of modern cloud-computing platforms and communication facilities.

Of particular note, significant advantages have been demonstrated through the utilization of surrogate models, such as Gaussian process or Kriging (Section 2.3.4.1), in conjunction with MOEAs. However, this is the first work in the literature that has incorporated the MPS within the context of multiobjective optimization. Additionally, while sharing some underlying similarities with ensemble methods [147], the MPS is conceptually unique in that it facilitates the incorporation of knowledge from a potentially diverse set of sources.

The remaining sections of this chapter are as follows: In Section 4.3, the proposed TEMO-MPS framework is presented. Subsequently, in Section 4.4, a particular instantiation of the framework is discussed, followed by experimental results in Sections 4.5, and 4.6 in which the efficacy of the framework is demonstrated on a series of synthetic benchmark functions and real-world case studies. Finally, Section 4.7 concludes the chapter.

4.3 Basic Framework of TEMO-MPS

Existing surrogate-assisted optimization approaches do improve upon traditional MOEAs, but, generally, only consider data from the target problem of interest; despite the fact that this information is, for computationally expensive problems, scarce and expensive to obtain. The proposed TEMO-MPS framework aims to address this issue by facilitating the incorporating of knowledge embedded in surrogate models from related (but distinct) sources in an adaptive fashion. The proposed approach, summarized in Algorithm 2, is as follows:

First, an initial set of n_0 solutions are sampled using a DOE procedure or randomly, and then evaluated using the expensive objective function and stored as $\mathcal{D}_T$. Next, the data points are aggregated according to a randomly sampled

weight-vector $\omega^{(i)}$, resulting in $y_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}|\omega^{(i)})$. Subsequently, the multi-problem surrogate h_{MPS} is built using the dataset $D_{\mathcal{T}}$ and available source models at iteration t . Next, an appropriate acquisition function induced from h_{MPS} , $\alpha(\mathbf{x}_{\mathcal{T}}|h_{MPS})$, is maximized using an EA within TEMO-MPS. The resulting solution $\mathbf{x}_{\mathcal{T}}^{(t+1)}$ is then evaluated and added to the dataset $D_{\mathcal{T}}$.

Algorithm 2 General TEMO-MPS Framework

Inputs: *Source Models*, weight-vectors $\omega^{(1)}, \dots, \omega^{(n_{\omega})}$

Output: Nondominated solutions of $D_{\mathcal{T}}$.

1. Initialization: Evaluate an initial set of n_0 solutions and store as $D_{\mathcal{T}}$.

while stopping conditions are not met **do**

2. Aggregation: Aggregate the function values of $D_{\mathcal{T}}$ according to a randomly sampled weight-vector $\omega^{(i)}$ to obtain $y_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}|\omega^{(i)})$.

3. Multi-Problem Surrogate: Build multi-problem surrogate model h_{MPS} that approximates $y_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}|\omega^{(i)})$ using aggregated solutions in $D_{\mathcal{T}}$ and available *Source Model*(s) at time t (e.g. Algorithm 1).

4. Evolutionary Optimization: Conduct evolutionary search using h_{MPS} surrogate, which ascertains the next point to evaluate using the true (expensive) objective functions. The evolved solution $\mathbf{x}_{\mathcal{T}}^{(t+1)}$ is evaluated and appended to $D_{\mathcal{T}}$.

end while

In the following sections, $y_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}|\omega^{(i)})$ will be replaced, simply, with $y_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}})$ or $y_{\mathcal{T}}$ for the sake of brevity, as every iteration, the aforementioned aggregation procedure based on a randomly sampled $\omega^{(i)}$ is repeated.

4.4 An Instantiation of TEMO-MPS

While there can be different variations of the framework described in Section 4.3, one possible instantiation is presented here. This variation utilizes a Tchebycheff aggregation approach (see Equation 2.5), the transfer-stacking Gaussian process surrogate models with linear meta-regression (TS-GP - Section 3.4.3), and an *expected improvement* measure to quantify the merit of evaluating a new point using the expensive objective functions. In the subsections that follow, details regarding

the various components that characterize this instantiation will be described.

4.4.1 The Expected Improvement Figure of Merit

The acquisition function $\alpha(\mathbf{x}_{\mathcal{T}})$, in the 4th step of Algorithm 2 (general TEMO-MPS framework), is a figure of merit that provides an effective means of estimating the quality of a candidate solution in a way that accounts for the uncertainties of the predictions by the GP model. Among the many such functions proposed in the literature, the *expected improvement* (EI) is a popular option that balances exploitation and exploration of the search landscape [15, 108, 141]. Consequently, EI is employed in the instantiation of the proposed framework.

Specifically, the improvement provided by a predicted output $\hat{y}_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{(*)})$ (Equation 3.3) of an unknown input $\mathbf{x}_{\mathcal{T}}^{(*)}$, relative to the best available solution found so far y_{min} , can be expressed as follows:

$$I(\mathbf{x}_{\mathcal{T}}^{(*)}) = \max(y_{min} - \hat{y}_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{(*)}), 0) \quad (4.1)$$

Accordingly, the EI measure at $\mathbf{x}_{\mathcal{T}}^{(*)}$ can be expressed as:

$$\begin{aligned} \alpha_{EI}(\mathbf{x}_{\mathcal{T}}^{(*)}) = E[I(\mathbf{x}_{\mathcal{T}}^{(*)})] &= (y_{min} - \hat{y}_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{(*)}))\Phi\left(\frac{y_{min} - \hat{y}_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{(*)})}{\hat{\sigma}_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{(*)})}\right) \\ &+ \hat{\sigma}_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{(*)})\phi\left(\frac{y_{min} - \hat{y}_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{(*)})}{\hat{\sigma}_{\mathcal{T}}(\mathbf{x}_{\mathcal{T}}^{(*)})}\right) \end{aligned} \quad (4.2)$$

where $\hat{\sigma}_{\mathcal{T}}$ (Equation 3.10) is the uncertainty of the predictions of the GP model. $\Phi(\cdot)$ and $\phi(\cdot)$ are the standard normal distribution and density functions, respectively.

4.5 Experiments

In this section, the performance of the proposed TEMO-MPS framework is compared against ParEGO [96] and the popular NSGA-II [95] on a series of synthetic

examples.

4.5.1 Test Functions

Two sets *multimodal* synthetic functions, based on functions found in literature [148], are utilized to illustrate the capabilities of TEMO-MPS. These functions, as shown in Equations 4.3 and 4.4, are modified as follows: (1) In order to reduce the number of local pareto fronts and, consequently, the overall “ruggedness” of the function, the value of 20π within the cosine term of the original function is modified to 2π [141], (2) the function is parameterized by the variables δ_1 and δ_2 in order to facilitate the generation of related tasks of which to transfer. The function, modified as described above, is denoted as DTLZ1b- δ_1, δ_2 . It should be noted that when $\delta_1 = \delta_2 = 0$, the expressions revert to the original test functions shown in [141]. It should be noted that the values of δ_1 and δ_2 affect the similarity between variants of the function. For example, DTLZ1b-10,0 is more similar to DTLZ1b-0,0 relative to DTLZ1b-30,0. This particular aspect will be further discussed in Section 4.5.6.

$$\begin{aligned}
&\text{Minimize } f_1 = \frac{1}{2}x_1(1 + g) \\
&\text{Minimize } f_2 = \frac{1}{2}(1 - x_1)(1 + g) \\
&g = (100 + \delta_1) \left[7 + \delta_2 + \sum_{i \in \{2, \dots, 8\}} (x_i - 0.5)^2 \right. \\
&\quad \left. - \cos(2\pi(x_i - 0.5)) \right] \\
&x_i \in [0, 1], i \in \{1, \dots, d\}, d = 8
\end{aligned} \tag{4.3}$$

The second group is derived from DTLZ3 [148]. As shown in Equation 4.4, DTLZ3b is also parameterized by δ_1 and δ_2 . Resultant functions are denoted by DTLZ3b- δ_1, δ_2 .

$$\begin{aligned}
&\text{Minimize } f_1 = (1 + g) \cos(x_1\pi/2) \cos(x_2\pi/2) \\
&\text{Minimize } f_2 = (1 + g) \cos(x_1\pi/2) \sin(x_2\pi/2) \\
&\text{Minimize } f_3 = (1 + g) \sin(x_1\pi/2) \\
&g = (100 + \delta_1) \left[6 + \delta_2 + \sum_{i \in \{3, \dots, 8\}} (x_i - 0.5)^2 \right. \\
&\quad \left. - \cos(2\pi(x_i - 0.5)) \right] \\
&x_i \in [0, 1], i \in \{1, \dots, d\}, d = 8
\end{aligned} \tag{4.4}$$

4.5.2 Hypervolume Measure

In this study, the quality of the solutions identified by the optimization algorithm is assessed using the hypervolume criteria - a measure that describes, relative to a reference point, the size of the region in objective space dominated by a set of candidate solutions [149, 150]. The specific reference point used in the experiments described in the section that follows is presented in the figure captions or tabulated results.

4.5.3 Experimental Details

In order to assure a fair comparison, the instantiation of TEMO-MPS and ParEGO utilize the EI acquisition function (Equation 4.2), the Tchebycheff aggregation (Equation 2.5), and identical evolutionary operators. This ensures that, when no source models are utilized, TEMO-MPS reverts to ParEGO - facilitating the evaluation of the knowledge transfer functionality of the proposed framework. The algorithms utilize the polynomial mutation [103], simulated binary crossover (SBX) [102], and the $(\mu + \lambda)$ -ES [104] selection strategy (with $\mu = \lambda$). It should be noted that in the original configuration of ParEGO proposed in [96], a steady-state population update strategy was chosen. Comparative studies were conducted en-

sureing that the modifications made in this study yielded similar results. The base-model used is the GP regression model with automatic relevance determination (ARD) squared exponential covariance function (Equation 2.10).

Details of the algorithmic settings are shown in the following: Table 4.1 (NSGA-II), Table 4.2 (ParEGO and TEMO-MPS). Note that as the study considers computationally expensive problems, the budget for NSGA-II is correspondingly small. Each experiment is repeated 30 times and, its statistical significance estimated using a Wilcoxon signed rank test at 95% confidence interval (see Table 4.3).

Parameter	Setting
Population Size	20
Number of Generations	25
Index of Polynomial Mutation	10
Index of SBX	10

Table 4.1: NSGA-II Parameter Settings

Parameter	Setting
Maximum Function Evaluations	500
Number of Weight Vectors, N_w	11 for $m = 2$; 15 for $m = 3$
Number of EA iterations	500
Size of EA Population	20
Mutation Probability	$1/d$
SBX and Mutation Index	10
Crossover Probability	0.9

Table 4.2: TEMO-MPS and ParEGO Parameter Settings

4.5.4 Determining the Initial DOE (Population) Size in ParEGO and TEMO-MPS

As a first step, the size of the initial set of data points n_0 and its effect on the overall performance of the optimization algorithms is examined. Using the DTLZ3b-10,0 as the target task, the search is conducted with $n_0 = 2(d+2), 5(d+2), 8(d+2)$, or $11(d+2)$. The source models utilized in TEMO-MPS for this experiment are drawn from ParEGO runs of DTLZ1b-0,0 and DTLZ3b-0,0 (which

Figure 4.2: Hypervolume vs. Function Evaluations for DTLZ3b-10,0 for ParEGO (in blue) and TEMO-MPS (in red). Source models for TEMO-MPS is assumed to be drawn from DTLZ3b-0,0 and DTLZ1b-0,0. Figures (a)-(d) corresponds to $n_0 = 2(d + 2)$, $5(d + 2)$, $8(d + 2)$, or $11(d + 2)$ initial DOE data points, respectively. The results are averaged across 30 runs.

includes a unique model for every ω).

As shown in Figure 4.2, the convergence characteristics of TEMO-MPS is generally superior to that of ParEGO - requiring fewer function evaluations (FEs) to achieve similar performance to ParEGO. Specifically, TEMO-MPS requires 75 FEs for $n_0 = 2(d + 2)$ (best case), and 132 FEs for $n_0 = 11(d + 2)$ (worse case) to achieve equivalent performance to ParEGO at 250 FEs. This is equivalent to a savings of roughly 70% and 47%, respectively, with respect to ParEGO.

In general, the results of Figure 4.2 show that during the initial stages of the search for TEMO-MPS, rapid improvement is observed (relative to ParEGO); an indication of the benefits afforded by knowledge transfer from the various source models incorporated into the MPS model. In the remainder of this paper, the

Figure 4.3: Illustration of the experimental procedure for TEMO-MPS on an example target optimization problem DTLZ1b–10, 0.

value of n_0 is set to $2(d+2)$ as it provides the largest improvement overall relative to ParEGO, in addition to better scalability when $d_{\mathcal{T}}$ and $d_{\mathcal{S}}$ are large [144]. It should be noted that no significant changes in the findings were observed when extending the maximum number of FEs to 500.

4.5.5 TEMO-MPS versus ParEGO and NSGA-II

The capabilities of TEMO-MPS are showcased in this section through a comparison against ParEGO and NSGA-II on a set of 18 synthetic test functions generated from the DTLZ1b (Equation 4.3) and DTLZ3b (Equation 4.4) functions with $\delta_1 \in \{10, 20, 30\}$ and $\delta_2 \in \{1, 2, 3\}$. In each run, TEMO-MPS utilizes source models from the other 17 test functions (excluding the δ_1, δ_2 pair of the

current target task); this includes a unique model for each ω . This experimental procedure is illustrated in Figure 4.3.

As with the observations from Section 4.5.4, the results presented in Figure 4.4 and Table 4.3 show that TEMO-MPS generally displays superior convergence characteristics relative to ParEGO. Specifically, for DTLZ3b, TEMO-MPS is superior to ParEGO at 50 and 100 FEs in all cases, 7/9 cases at 200 FEs, and 8/9 cases at 500 FEs. The results further indicate that due to the rapid improvement of TEMO-MPS during the initial stages of the search process, ParEGO begins to close the gap only after 500 FEs. NSGA-II, however, exhibits inferior convergence characteristics relative to TEMO-MPS and ParEGO due to the low computational budget and non-surrogate-assisted nature of the algorithm. These promising results demonstrate the capabilities of TEMO-MPS in leveraging knowledge from related source tasks; giving confidence to the efficacy of the proposed framework.

Figure 4.4: Hypervolume vs. Function Evaluations comparison between ParEGO, TEMO-MPS, and NSGA-II for DTLZ1b (**TOP-ROW**), DTLZ3b (**BOTTOM-ROW**), and various values of δ_1, δ_2 . The results in the figure are averaged across 30 independent runs.

Table 4.3: Comparing the mean performance and standard deviation of ParEGO, NSGA-II and Temo-MPS (see Section 4.5.5). Statistically significant superior results evaluated using a Wilcoxon signed rank test with a 95% confidence interval are presented in bold.

Experiments	Bound	50 Generations						100 Generations						200 Generations						500 Generations								
		ParEGO			Temo-MPS			ParEGO			Temo-MPS			ParEGO			Temo-MPS			ParEGO			Temo-MPS			NSGA-II		
		Mean	Std		Mean	Std		Mean	Std		Mean	Std		Mean	Std		Mean	Std		Mean	Std		Mean	Std		Mean	Std	
DTLZ1b - 10,1	75,75	0.2612	0.1109	0.6508	0.0149		0.5139	0.0305	0.6620	0.0118		0.5992	0.0154	0.6709	0.0099		0.6214	0.0114	0.6822	0.0115		0.6444	0.0261					
DTLZ1b - 20,1	75,75	0.1835	0.1091	0.5970	0.0099		0.4738	0.0262	0.6120	0.0065		0.5504	0.0118	0.6165	0.0082		0.5720	0.0128	0.6263	0.0090		0.5993	0.0274					
DTLZ1b - 30,1	75,75	0.1596	0.1124	0.5338	0.0104		0.4204	0.0349	0.5463	0.0065		0.5030	0.0158	0.5549	0.0072		0.5201	0.0106	0.5628	0.0090		0.5063	0.0438					
DTLZ1b - 10,2	120, 120	0.1876	0.1092	0.5050	0.0092		0.4192	0.0231	0.5149	0.0054		0.4851	0.0103	0.5231	0.0065		0.4996	0.0075	0.5291	0.0069		0.5010	0.0249					
DTLZ1b - 20,2	120, 120	0.1455	0.0826	0.4226	0.0081		0.3520	0.0203	0.4379	0.0056		0.4153	0.0113	0.4430	0.0061		0.4263	0.0074	0.4475	0.0061		0.4100	0.0250					
DTLZ1b - 30,2	120, 120	0.1141	0.0660	0.3512	0.0070		0.2945	0.0179	0.3620	0.0058		0.3442	0.0107	0.3665	0.0066		0.3528	0.0070	0.3705	0.0064		0.3264	0.0315					
DTLZ1b - 10,3	200, 200	0.2796	0.1261	0.5795	0.0139		0.5043	0.0213	0.5968	0.0058		0.5612	0.0103	0.6019	0.0080		0.5830	0.0107	0.6070	0.0067		0.6010	0.0169					
DTLZ1b - 20,3	200, 200	0.2167	0.0917	0.5170	0.0096		0.4452	0.0190	0.5297	0.0063		0.5009	0.0091	0.5352	0.0056		0.5206	0.0075	0.5397	0.0067		0.5249	0.0160					
DTLZ1b - 30,3	200, 200	0.1897	0.1056	0.4426	0.0119		0.3888	0.0196	0.4564	0.0056		0.4405	0.0073	0.4630	0.0054		0.4509	0.0071	0.4695	0.0070		0.4496	0.0189					
DTLZ3b - 10,1	150,150,150	0.0283	0.0488	0.4327	0.0540		0.2784	0.0964	0.5027	0.0452		0.4420	0.0429	0.5414	0.0384		0.5187	0.0320	0.5825	0.0307		0.1642	0.0833					
DTLZ3b - 20,1	150,150,150	0.0155	0.0356	0.3416	0.0424		0.1918	0.0743	0.4178	0.0278		0.3426	0.0440	0.4557	0.0251		0.4174	0.0350	0.4868	0.0304		0.1045	0.0606					
DTLZ3b - 30,1	150,150,150	0.0064	0.0156	0.2321	0.0569		0.1225	0.0520	0.3012	0.0236		0.2459	0.0394	0.3319	0.0238		0.3204	0.0201	0.3748	0.0249		0.0492	0.0401					
DTLZ3b - 10,2	300,300,300	0.0699	0.0830	0.4343	0.0529		0.4009	0.0628	0.5329	0.0394		0.5143	0.0394	0.5690	0.0294		0.5562	0.0362	0.6016	0.0274		0.2198	0.0510					
DTLZ3b - 20,2	300,300,300	0.0266	0.0417	0.3455	0.0647		0.2832	0.0572	0.4245	0.0376		0.4025	0.0342	0.4635	0.0268		0.4613	0.0241	0.4949	0.0269		0.1275	0.0511					
DTLZ3b - 30,2	300,300,300	0.0214	0.0384	0.2584	0.0367		0.1871	0.0461	0.3084	0.0195		0.2887	0.0262	0.3324	0.0170		0.3420	0.0249	0.3590	0.0196		0.0703	0.0430					
DTLZ3b - 10,3	450,450,450	0.0854	0.0904	0.4145	0.0785		0.4275	0.0668	0.4980	0.0355		0.5350	0.0349	0.5295	0.0359		0.5871	0.0325	0.5667	0.0353		0.2179	0.0558					
DTLZ3b - 20,3	450,450,450	0.0310	0.0585	0.3209	0.0555		0.2929	0.0890	0.3892	0.0274		0.4187	0.0382	0.4197	0.0258		0.4694	0.0290	0.4524	0.0249		0.1503	0.0346					
DTLZ3b - 30,3	450,450,450	0.0150	0.0325	0.2670	0.0375		0.1914	0.0535	0.3172	0.0161		0.3052	0.0253	0.3347	0.0179		0.3636	0.0261	0.3596	0.0203		0.0860	0.0391					

4.5.6 Investigating the Effects of Source-Target Similarity

In this section, the previous line of inquiry is built upon by investigating the effects of incorporating source models of increasing dissimilarity into TEMO-MPS. Specifically, this experiment involves solving DTLZ1b-0,0 as the target problem while utilizing source models from $\delta_2 \in \{1, 2, 3\}$ with $\delta_1 = 10$, i.e. DTLZ1b-10,1, followed by DTLZ1b-10,2 and, finally, DTLZ1b-10,3.

In the first step of this analysis, the Pearson correlation coefficient is calculated. This measure provides an indication of the linear correlation between the aggregated landscapes for both source and target tasks, where a decreasing Pearson correlation is associated with decreasing source-target similarity. As presented in Figure 4.5(a), such a trend is observed. Specifically, DTLZ1b-10,3 is found to be the least correlated to the target function (DTLZ1b-0,0), while DTLZ1b-10,1 is the most. To further visualize this decrease in correlation the aggregated outputs for all tasks is sorted based on the magnitude of the DTLZ1b-0,0 outputs. The resulting Figure 4.5(b) demonstrates that as the Pearson correlation coefficient decreases, the spread of the aggregated outputs become larger due to the diminishing correlation, and is located further relative to the DTLZ1b-0,0 curve.

Next, Figure 4.5(c), the hypervolume measure of TEMO-MPS at 50, 75 and 100 FEs are shown against the corresponding Pearson correlation coefficient. The results confirms the expectations that as the source-target similarities increase, there is a corresponding monotonic improvement in the overall optimization performance. To further verify the assertion that this improvement is a consequence of improved modelling accuracy (due to knowledge transfer), rather than, for example, the blessings of uncertainty [151], the generalization error (RMSE) of the TS-GP model is compared with GP. Specifically, as shown in Figure 4.5(d), there is an observed improvement in the *Generalization Performance Gain* $= (RMSE_{GP} - RMSE_{MPS})/RMSE_{GP}$ as a consequence of increasing source-target similarity (expressed in the form of the Pearson correlation coefficient). Note that *GP* subscript in the “Generalization Performance Gain” function refers

Figure 4.5: **(a)** Pearson correlation coefficient of DTLZ1b-0,0 and DTLZ1b-10, δ_2 . **(b)** aggregated objective values of DTLZ1b-0,0, DTLZ1b-10,1, DTLZ1b-10,2 and DTLZ1b-10,3 given the same inputs. **(c)** Hypervolume vs. Pearson correlation coefficient for TEMO-MPS at the specified function evaluation intervals (FEs). **(d)** Generalization performance gain (in %) of MPS vs. Pearson correlation coefficient.

to the no-transfer case.

4.6 Practical Examples

To further verify the capabilities of the proposed framework, TEMO-MPS is applied to two practical case studies; namely, a complex engineering design problem, and a hyperparameter optimization exercise of a machine learning model.

4.6.1 A Case Study in Global Black-Box Optimization in Engineering Process Design augmented by Homogeneous Knowledge Transfer

Complex engineering design is a domain that generally achieves progress through gradual, incremental enhancements - making it a key beneficiary of the proposed framework. The availability of large amounts of information from related sources facilitates the incorporation of knowledge transfer as a means to reduce the exorbitant design costs. To demonstrate, to identify a suitable solution in this application domain, a traditional EA would require several thousand FEs. Further, a high-fidelity simulation can take several hours or days per evaluation. Consequently, improving the efficiency of the optimization algorithm is extremely desirable.

In this section, the proposed TEMO-MPS is applied to a real-world process engineering design problem from the composites manufacturing industry based on expensive finite-element simulations. Speeding-up the optimization task in such a computationally expensive scenario can be of tremendous benefit to the industrial domain. Specifically, herein, the optimization of the *resin transfer moulding* (RTM) process for two geometrically similar but ultimately distinct composite parts is considered.

As depicted in Figure 4.6, the RTM is a popular methodology for manufacturing composite parts of high-quality and is used particularly in the aeronautics and automobile industries. Details of the composite manufacturing processes (in addition to RTM) can be found in [152, 153]. The RTM process involves the utilization of metallic mold machined to match the geometry of the composite part being manufactured. This process begins by first placing the fibre-reinforced polymer within the mould cavity. It is subsequently compressed to the desired thickness before the liquid thermosetting resin is injected at high pressure into the closed mould so as to fully saturate all remaining voids.

The process itself is a function of four critical design variables: velocity of mold closure $V_{closure}$, resin injection pressure P_{inj} , temperature of the mold T_{mold} , and temperature of the resin T_{resin} . In the initial design stages of a particular composite part, the primary aim is to identify the set of design parameters that minimizes the associated costs and, simultaneously, minimize the capital layout. Specifically, the aforementioned costs are addressed through the manipulation of the peak internal forces generated during the manufacturing process, which influences the size and cost of the peripheral equipment (e.g. hydraulic press). The MOP formulation of this problem [154] is as follows:

$$\begin{aligned} &\text{Minimize: (Peak Internal Force, Mold Filling Time)} \\ &\text{Subject to: } Variables \in Feasible\ Design\ Space \end{aligned} \tag{4.5}$$

In this case study, two distinct but geometrically similar epoxy composite discs (reinforced by glass-fibre) is considered. The target task is the manufacture of a 1 m diameter disc with 35% fibre volume fraction, while the source task considers a smaller 0.8 m diameter disc with a larger volume fraction of 50%. Additional details regarding its material properties, and the governing PDEs is described in detail in [153]. Note that as a consequence of the geometric similarities, some underlying synergy exists between the two MOPs. The bounds of the design variables are stated in Table 4.4, and the algorithm specific-parameters (i.e. of TEMO-MPS, ParEGO and NSGA-II) are identical to those utilized in the previous section. In addition to the standard procedures for multiobjective optimization utilized thus far, a simulation of the resulting effect of utilizing data streaming in continuously from the related source task is conducted. To elaborate, as TEMO-MPS performs its search for the target task, the source task is executed in parallel (using ParEGO). Initially, the source task contains 100 instances. After 10 iterations of TEMO-MPS, the source task is assumed to now have 110 instances, with 120 data points at iteration 20, and so on. In reality, the

Figure 4.6: An illustration of a typical RTM process workflow for fiber-reinforced polymer composite parts [2].

Table 4.4: Upper and lower bounds of the design variables in the RTM manufacturing cycle

Design Variable	Lower Bound	Upper Bound
$V_{closure}$	1 mm/min	10mm/min
P_{inj}	1 MPa	10 MPa
T_{mold}	293 K	348 K
T_{resin}	293 K	348 K

viability of sharing data in this manner is made possible through the utilization of cloud-computing platforms.

The results of this case study indicate that, as shown in Figure 4.7, TEMO-MPS is able to successfully leverage source information in order to achieve rapid improvement (relative to ParEGO) during the initial stages of the search process. Specifically, TEMO-MPS achieves a hypervolume of 0.25 after only ~ 6700 seconds relative to the ~ 9950 seconds required by ParEGO i.e. resulting in a savings of 32% or ~ 3250 seconds. From the previous analysis, this encouraging results is attributed to the improved approximation of the target landscape as a consequence of incorporating source models from a related task. Further, as with previous experiments, the performance of NSGA-II is inferior to TEMO-MPS or ParEGO due to the limited computational budget imposed upon the algorithm.

Figure 4.7: Performance comparison between TEMO-MPS, ParEGO, and NSGA-II for the optimization of the simulation-based RTM cycle. For the target task, a 1 m diameter composite disc with a fibre volume fraction of 35% is considered, while the source task involves a 0.8 m diameter composite disc with a fibre volume fraction of 50%. The results presented are based on a reference point of (40s, 40N) and is an average of 30 repeated runs. The shaded area spans 1/2 standard deviation on either side of the mean.

4.6.2 Accelerated Automatic Hyperparameter Tuning of Machine Learning Models using Homogeneous Knowledge Transfer

For ML models, an important task is the identification of the optimal set of hyperparameters for a particular task of interest. This activity is often quite computationally expensive as it requires repeated evaluations of a particular ML model of choice. For instance, neural networks include hyperparameters such as regularization coefficient, number of epochs, learning rate, choice of activation function, number of hidden units, etc. In some cases, the difference between a state-of-the-art result and one that is subpar can be attributed to the appropriate selection of these hyperparameters. Unfortunately, particularly in deep learning scenarios where the number of parameters can number in the millions, the evaluation of a single set of hyperparameters can often be daunting.

The task at hand is a multi-class classification problem solved using a multi-layer fully-connected neural network. The objective of this study is to evaluate

the potential of utilizing a subset of the original dataset as a knowledge source in order to accelerate the exploration of the hyperparameter settings [18]. In particular, the dataset is a yeast gene functional classification problem with 2417 instances, and considers 103 features, and 14 classes that are non-exclusive. The hyperparameters of the neural network tuned in this case study are shown in Table 4.5 (along with the relevant bounds).

The multi-class classification problem is formulated as a multiobjective optimization problem, where the accuracy of each class is an objective to be optimized. The motivation behind this approach is the possibility that there might exist architectures that are preferentially beneficial to certain outputs relative to others. Consequently, performance trade-offs may have to be evaluated, a task that would be difficult if the problem is formulated as a single objective optimization of a unified accuracy measure. To simplify the study, and remain consistent with the experiments presented thus far, the yeast gene classification problem is restricted to 3 outputs (i.e. 3-objective optimization problem). The neural network is trained using a scaled conjugate gradient backpropagation algorithm with a mean squared error criteria.

Subsequently, it is demonstrated that by utilizing a subset of the original dataset from which to draw source models from, TEMO-MPS can leverage the knowledge to perform rapid optimization relative to ParEGO. Specifically, the source dataset utilizes 500 randomly sampled instances for training and the remaining 1917 for validation - an approach that simulates rapid low-cost exploratory analysis. Subsequently, a 30% hold-out scheme is employed for the training of the model for the target task. As with previous experiments, both TEMO-MPS and ParEGO are constrained to a budget of 500 FEs (i.e. 500 hyperparameter configurations evaluated) and averaged across 30 independent runs.

As shown in Figure 4.8, TEMO-MPS achieves superior convergence characteristics relative to ParEGO. Specifically, TEMO-MPS requires 347 FEs to achieve the hypervolume of ParEGO at 500 FEs (i.e. approximately 30.6% saved). This

Table 4.5: Upper and lower bounds of the input hyperparameters to the neural network.

Hyperparameters (Input)	Lower Bound	Upper Bound
n_{hidden}	1	10
n_{nodes}	1	100
γ	0	1
max_epochs	1	500

Figure 4.8: Evolution of Hypervolume vs. Function Evaluations for ParEGO and TEMO-MPS on a multiobjective formulation of the hyperparameter tuning problem of a feedforward neural network applied to a yeast gene functional multi-output classification problem. All results are averaged across 30 independent runs and based on a reference point of $[1,1,1]$. The shaded area spans 1/2 standard deviation on either side of the mean.

encouraging result demonstrates the capabilities afforded by TEMO-MPS in the context of accelerated hyperparameter tuning of ML models.

4.7 Summary

In this chapter, as part of Research Objectives 1, 2, and 3 (see Section 1.2), a novel adaptive knowledge transfer framework for multiobjective optimization of computationally expensive problems is presented. Proposed in this thesis for the first time, the *multi-problem surrogate* (MPS) (see Section 3.4.3) facilitates the capability to leverage information from multiple related but distinct sources to

enhance optimization performance of a particular target task.

Based on experimental results on synthetic functions, a complexing engineering process design case study, and an automated hyperparameter tuning case study, TEMO-MPS shows superior convergence characteristics relative to ParEGO and NSGA-II - saving up to 70% of the FEs. A closer analysis show that the overall achievable improvements of TEMO-MPS is a function of the similarity between source and target tasks. In particular, the adaptive component of the MPS is theoretically shown to regulate the contributions from source models. Specifically, the utilization of highly correlated source models result in rapid improvements during the initial stages of the search process due to relevant knowledge exploitation. However, if unrelated source models are used instead, then the negative effects are automatically minimized.

Despite the promising outcomes, the framework relies on a key assumption, namely, homogeneity between the sources and target models, i.e. the input feature spaces are identical. In the subsequent chapter, the feasibility of knowledge transfer for optimization across heterogeneous domains is investigated.

Chapter 5

Fast Bayesian Optimization Utilizing Heterogeneous Experiences

5.1 Chapter Goals

Optimization problems are frequently conducted from scratch assuming no prior knowledge. However, this approach is often practically intractable, particularly for computationally expensive problems that require large amounts of function evaluations before a suitable solution can be found. In contrast, humans possess the innate ability to learn from past problem-solving experiences and seamlessly transfer the knowledge to quickly address related scenarios in the future. Inspired by this idea, recent notable efforts have been made to incorporate such capabilities in Bayesian Optimization solvers, demonstrating substantial performance speed-ups in many applications as compared to traditional *tabula-rasa* approaches. However, a limitation of current methods is that they generally consider transfer between problems with homogeneous search spaces. Thus, little effort is made to discern the presence of hidden similarities between problems that may appear different on the surface, i.e. possessing non-homogeneous search spaces. With

this in mind, in this chapter, as part of Research Objectives 1, and 4 (see Section 1.2), a Bayesian Optimization framework that can facilitate knowledge transfer across seemingly disparate optimization exercises with heterogeneous domains is investigated.

5.2 Introduction

Real-world problems seldom exist in isolation. Any practically useful system, especially one in an industrial setting, is expected to tackle a large number of problems over a lifetime, many of which will either be repetitive or at least display some domain-specific idiosyncrasies. Taking this cue, in the domain of machine learning, the idea of utilizing data from various *source* tasks to improve the learning on a related *target* task has seen considerable success under the label of transfer learning. In contrast, it is observed that similar efforts in the domain of *optimization* have been scarce, with most solvers starting the search from scratch assuming a zero prior knowledge state. It can be argued that for many real-world problems, especially those entailing a high cost of evaluation, overlooking the knowledge available from experience may lead to deleterious overheads in the re-exploration of essentially similar search spaces.

For dealing with computationally expensive *black-box* optimization problems, *Bayesian optimization* (BO) has emerged as an attractive option due to its high sample efficiency [109]. For a given task, the method proceeds as a sequential model-based optimization approach that iterates over the following steps: (a) learn a probabilistic model $p(f)$ - typically a *Gaussian process* (GP) - approximating the objective function f ; (b) use $p(f)$ to induce a relatively cheap *acquisition function* that leverages the uncertainty in the posterior to guide the exploration; (c) optimize the acquisition function to determine the next point to evaluate; (d) augment the dataset with the newly evaluated point. Regardless of its known sample efficiency, a challenge BO continues to face is that of *cold start*; indicating

the difficulty of gathering sufficient data to build an informative model that can effectively guide the search. To address this issue, recently proposed methods have shed light on the advantages of incorporating transfer learning capabilities into optimizers [18, 19]. The main inspiration behind the proposed approach is drawn from the innate ability of human experts to exploit what has been seen in the past to more quickly address a never before seen problem at hand. It is indeed the ability of humans to take advantage of past experiences that sets apart an expert from a novice. While instilling such human-like capabilities within BO has shown promising results, a limitation of current methods is the assumption of *homogeneity*, i.e., the source and target optimization problems have overlapping search spaces with common features spanning equal dimensions.

With the above in mind, in this chapter, a novel generalization of the idea of *transfer Bayesian optimization* (TBO) is put forward to address *heterogeneous* optimization problems that have similar, but not necessarily the same, search space; and in turn accounting for any differences in dimensionality. In order to harness hidden source-target correlations in such settings, a novel knowledge adaptation strategy for transfer is presented, inspired by concepts from *manifold learning*, such that the data accumulated from heterogeneous past experiences can be transformed to a new representation that is better aligned with the target task of interest. The transformation can either be a random projection (in the simplest case), or be learned jointly [132] with an *adaptive transfer* GP model that is capable of capturing the degree of correlation between the domains [1, 155]. The resultant *manifold transfer GP* (*Mani-TGP*) is subsequently incorporated within the framework of BO, making it possible to realize the transfer of knowledge across heterogeneous optimization problems. Accordingly, the overall proposal is labelled as *manifold transfer Bayesian optimization* (*Mani-TBO*). To the best of my knowledge, thesis represents the first work that develops a BO framework that is augmented by knowledge transfer capabilities across heterogeneous optimization problems.

There are expected to be several real-world applications of *Mani*-TBO. For instance, in the field of engineering design, products are typically adapted and improved over time, where new parameters (features) are added and redundant parameters removed (or fixed) as the design evolves depending on the requirements of a particular application [21]. Consequently, despite the fundamental relationship between the baseline and the modified designs, their features are not necessarily the same. Such scenarios provide an ideal environment for the transfer of knowledge across heterogeneous design optimization problems to be explored. While some heuristic strategies for tackling such problems have been studied recently [86], the present thesis is one of the first attempts to address the problem via a principled TBO approach. It is further noted that the empirically evidenced sample efficiency of BO (and *Mani*-TBO in particular), in comparison to state-of-the-art meta-heuristic methods, has significant implications for computationally expensive problems in the domain of machine learning as well. The potential of *Mani*-TBO in this regard shall be illustrated through a case study in meta-learning for automatic model selection (*AutoML* [156, 157]). Specifically, the search for optimal hyperparameters for machine learning models is augmented using adaptive knowledge transfer across distinct model architectures.

A detailed exposition of the ideas discussed heretofore are organized in the remainder of the chapter as follows: Section 5.3 briefly reiterates a description of traditional BO. In Section 5.4, the TBO framework incorporating *Mani*-TGP (see Ch. 3.5.1), i.e. *Mani*-TBO, is described. In Sections 5.5 and 5.6, the proposed algorithm is numerically investigated on a diverse test bed. Finally, Section 5.7 concludes the chapter.

5.3 Background

In this chapter, the following target optimization problem is considered:

$$\mathbf{x}^{opt} = \arg \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \quad (5.1)$$

where $\mathbf{x}$ is a d -dimensional solution vector in the search space $\mathcal{X}$ which is a bounded subset of $\mathbb{R}^d$, and f is a computationally expensive *black-box* objective function.

In traditional Bayesian optimization (BO), the search is initiated with a randomly sampled dataset which is augmented at every subsequent iteration with a newly evaluated solution chosen in a principled manner. To elaborate, let the dataset after n exact evaluations be $\mathcal{D}^{(n)} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$, where $y^{(i)} = f(\mathbf{x}^{(i)}) + \epsilon$ is the observed value of the possibly noisy objective function. Note that ϵ represents an i.i.d noise term such that $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. Given $\mathcal{D}^{(n)}$, a probabilistic model $p(f|\mathcal{D}^{(n)})$ is learned, and the posterior uncertainty of $p(f|\mathcal{D}^{(n)})$ is used to define a cheap acquisition function $\alpha(\mathbf{x}; p(f|\mathcal{D}^{(n)}))$ that quantifies the utility of candidate solutions for the next (exact) evaluation of f . With this, the next solution to be evaluated is determined as:

5.4 Manifold Transfer Bayesian Optimization

The *Mani*-TBO framework is, at its core, an enhancement of traditional BO. Thus, the basic iterative procedure of (probabilistic) model building, followed by candidate solution identification through acquisition function optimization, is retained. The key improvement lies in the incorporation of the *Mani*-TGP model for probabilistic approximation of the objective function, so as to enable knowledge transfer between source and target tasks that may be heterogeneous.

The overall framework, summarized in Algorithm 1, consists of three primary steps. First, the initial set of solutions $\mathbf{x}_{\mathcal{T}}^{(1)}, \dots, \mathbf{x}_{\mathcal{T}}^{(n_{\mathcal{T}})}$ are evaluated using the expensive function to obtain the dataset $\mathcal{D}_{\mathcal{T}}$. Next, using the datasets $\mathcal{D}_{\mathcal{S}}$ and $\mathcal{D}_{\mathcal{T}}$, the probabilistic *Mani*-TGP model $p(f|\mathcal{D}_{\mathcal{S}}, \mathcal{D}_{\mathcal{T}})$ is learned. Subsequently, the next solution to be evaluated $\mathbf{x}_{\mathcal{T}}^{(*)}$ is identified by maximizing an appropriate acquisition

Algorithm 3 Pseudocode of *Mani-TBO*

Inputs: Expensive function evaluator; computational budget; search space $\mathcal{X}_{\mathcal{T}}$; source dataset $\mathcal{D}_{\mathcal{S}}$; initial solutions $\mathbf{x}_{\mathcal{T}}^{(1)}, \dots, \mathbf{x}_{\mathcal{T}}^{(n_{\mathcal{T}})}$

Output: Best solution found

Step 1 - Evaluation of Initial Solutions:

for $i \leftarrow 1$ **to** $n_{\mathcal{T}}$ **do** $y_{\mathcal{T}}^{(i)} \leftarrow \text{Evaluate } \mathbf{x}_{\mathcal{T}}^{(i)}$

$\mathcal{D}_{\mathcal{T}} \leftarrow \{(\mathbf{x}_{\mathcal{T}}^{(i)}, y_{\mathcal{T}}^{(i)})\}_{i=1}^{n_{\mathcal{T}}}$

while $n_{\mathcal{T}} \leq \text{computational budget}$ **do**

Step 2 - Learn the *Mani-TGP* model:

$p(f|\mathcal{D}_{\mathcal{S}}, \mathcal{D}_{\mathcal{T}}) \leftarrow \text{fit } \textit{Mani-TGP} \text{ on } \mathcal{D}_{\mathcal{T}}, \mathcal{D}_{\mathcal{S}}$

Step 3 - Optimizing α for $\mathbf{x}_{\mathcal{T}}^{(*)}$:

$\mathbf{x}_{\mathcal{T}}^{(*)} = \arg \max_{\mathbf{x} \in \mathcal{X}_{\mathcal{T}}} \alpha(\mathbf{x}; p(f|\mathcal{D}_{\mathcal{S}}, \mathcal{D}_{\mathcal{T}}))$

$y_{\mathcal{T}}^{(*)} \leftarrow \text{Evaluate } \mathbf{x}_{\mathcal{T}}^{(*)}$

$\mathcal{D}_{\mathcal{T}} \leftarrow \{\mathcal{D}_{\mathcal{T}}, (\mathbf{x}_{\mathcal{T}}^{(*)}, y_{\mathcal{T}}^{(*)})\}$

$n_{\mathcal{T}} \leftarrow n_{\mathcal{T}} + 1$

end

return $\arg \min_{\mathbf{x}_{\mathcal{T}}^{(i)} \in \{\mathbf{x}_{\mathcal{T}}^{(1)}, \dots, \mathbf{x}_{\mathcal{T}}^{(n_{\mathcal{T}})}\}} y_{\mathcal{T}}^{(i)}(\mathbf{x}_{\mathcal{T}}^{(i)})$

function α induced from $p(f|\mathcal{D}_{\mathcal{S}}, \mathcal{D}_{\mathcal{T}})$. The dataset $\mathcal{D}_{\mathcal{T}}$ is then augmented with the newly evaluated solution.

One of the degrees of freedom in the design of a BO algorithm is the choice of the acquisition function, the role of which is to provide a trade-off between *exploration* and *exploitation* of the search space. While a plethora of acquisition functions have been proposed over the years, perhaps the most commonly used is the *expected improvement* (EI) over the best objective function value ($f_{\min}$) that has been found so far during the search [15]. For a minimization problem, the EI at some unsampled point $\mathbf{x}_{\mathcal{T}}^{(*)}$ is given by:

$$\alpha_{EI}(\mathbf{x}_{\mathcal{T}}^{(*)}) = \int_{-\infty}^{\infty} \max\{f_{\min} - f, 0\} p(f|\mathcal{D}_{\mathcal{S}}, \mathcal{D}_{\mathcal{T}}, \mathbf{x}_{\mathcal{T}}^{(*)}) df. \quad (5.2)$$

Another simple approach is the *upper confidence bound* (UCB) acquisition function [158]. For a minimization problem, the UCB at $\mathbf{x}_{\mathcal{T}}^{(*)}$ is given by:

$$\alpha_{UCB}(\mathbf{x}_{\mathcal{T}}^{(*)}) = -\tilde{\mu}(\mathbf{x}_{\mathcal{T}}^{(*)}) + \beta^{1/2}\tilde{\sigma}(\mathbf{x}_{\mathcal{T}}^{(*)}), \quad (5.3)$$

where β is an appropriately defined constant. In this chapter, both acquisition functions are used with *Mani*-TBO and the baseline BO algorithm throughout the experimental study.

5.4.1 Regret Bounds Analysis of *Mani*-TBO

From [158], the regret bounds for a classical BO approach applied to a *discretized* search space $\mathcal{X}$, with UCB acquisition function, has the form:

$$Pr\{R_n \leq \sqrt{C_1\beta_n\gamma_n n} \ \forall n \geq 1\} \geq 1 - \delta \quad (5.4)$$

where n is the number of rounds (samples evaluated) in BO, R_n is the cumulative regret, $\beta_n = 2 \log(|\mathcal{X}|n^2\pi^2/6\delta)$ for $\delta \in (0, 1)$ where $C_1 = 8/\log(1 + \sigma_n^{-2})$, and γ_n is the maximum information gain. In particular, if the instantaneous regret of BO at the n th evaluation is considered to be $r_n = f(x_n) - f(x^{opt})$ for a minimization problem, then the cumulative regret R_n is the sum of instantaneous regrets: $R_n = \sum_{i=1}^n r_i$. Clearly, a lower regret value indicates superior performance. Further, γ_n in Eq. 5.4 is formally given by:

$$\gamma_n = \max_{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)} \in \mathcal{X}} I([y^{(1)}, \dots, y^{(n)}]; \mathbf{f}) \quad (5.5)$$

where $\mathbf{f} = [f(\mathbf{x}^{(1)}), \dots, f(\mathbf{x}^{(n)})]$, and I is the mutual information between $\mathbf{f}$ and the observations $y^{(1)}, \dots, y^{(n)}$. Mathematically, the mutual information is expressed as,

$$I([y^{(1)}, \dots, y^{(n)}]; \mathbf{f}) = \frac{1}{2} \log |I + \sigma_n^{-2}K|. \quad (5.6)$$

In order to show that *Mani*-TBO (with the UCB acquisition function) leads to superior convergence characteristics in comparison to classical BO, it suffices

to show that the theoretical regret bound obtained as a consequence of transfer learning in Mani-TBO is tighter. To this end, the maximum information gain $\tilde{\gamma}_{n_{\mathcal{T}}}$ in a target task $\mathcal{T}$, given a source dataset $\mathcal{D}_{\mathcal{S}}$, is equivalently proven to be indeed bounded from above by $\gamma_{n_{\mathcal{T}}}$. Here, $\gamma_{n_{\mathcal{T}}}$ is analogous to γ_n in Eq. 5.4 and represents the maximum mutual information gain after $n_{\mathcal{T}}$ evaluation rounds in classical BO corresponding to the no transfer case. Thus, if $\tilde{\gamma}_{n_{\mathcal{T}}} \leq \gamma_{n_{\mathcal{T}}}$, then the resultant regret bound (convergence) of Mani-TBO is theoretically better.

Theorem 1. *Given the UCB acquisition function, $\tilde{\gamma}_{n_{\mathcal{T}}} \leq \gamma_{n_{\mathcal{T}}}$ if $\lambda \neq 0$, else $\tilde{\gamma}_{n_{\mathcal{T}}} = \gamma_{n_{\mathcal{T}}}$.*

Proof. The proof follows a similar style as Theorem 1 in [159]. The maximum information gain for Mani-TBO is:

$$\tilde{\gamma}_{n_{\mathcal{T}}} = \max_{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n_{\mathcal{T})}} \in \mathcal{X}_{\mathcal{T}}} \mathbb{I}([y_{\mathcal{T}}^{(1)}, \dots, y_{\mathcal{T}}^{(n_{\mathcal{T})}}]; \mathbf{f}_{\mathcal{T}} | \mathcal{D}_{\mathcal{S}}) \quad (5.7)$$

Based on $C_{\mathcal{T}}$ in Equation 3.22, the *conditional* mutual information takes the form,

$$\begin{aligned} & \mathbb{I}([y_{\mathcal{T}}^{(1)}, \dots, y_{\mathcal{T}}^{(n_{\mathcal{T})}}]; \mathbf{f}_{\mathcal{T}} | \mathcal{D}_{\mathcal{S}}) \\ &= \frac{1}{2} \log |I + \sigma_{\mathcal{T}}^{-2} (K_{\mathcal{T}\mathcal{T}} - K_{\mathcal{T}\mathcal{S}}(K_{\mathcal{S}\mathcal{S}} + \sigma_{\mathcal{S}}^2 I)^{-1} K_{\mathcal{S}\mathcal{T}})|. \end{aligned} \quad (5.8)$$

Next, based on Theorem 1 in [1], it is known that $\tilde{K} = \begin{pmatrix} K_{\mathcal{S}\mathcal{S}} & K_{\mathcal{S}\mathcal{T}} \\ K_{\mathcal{T}\mathcal{S}} & K_{\mathcal{T}\mathcal{T}} \end{pmatrix}$ is positive semi-definite (PSD) for $|\lambda| \leq 1$. Therefore, from the *Schur complement theorem*, it follows that the matrix $K_{\mathcal{T}\mathcal{T}} - K_{\mathcal{T}\mathcal{S}}(K_{\mathcal{S}\mathcal{S}} + \sigma_{\mathcal{S}}^2 I)^{-1} K_{\mathcal{S}\mathcal{T}}$ appearing in Eq. 5.8 is guaranteed to be PSD. In addition, $K_{\mathcal{T}\mathcal{S}}(K_{\mathcal{S}\mathcal{S}} + \sigma_{\mathcal{S}}^2 I)^{-1} K_{\mathcal{S}\mathcal{T}}$ is naturally PSD given a valid covariance kernel $k(\cdot, \cdot)$. Notice from Eqs. 3.18 and 3.19 that $K_{\mathcal{T}\mathcal{S}} = K_{\mathcal{S}\mathcal{T}} = 0$ if $\lambda = 0$.

Given the above, the *Minkowski determinant inequality* is invoked, which

states that for PSD matrices A and B , $|A + B| \geq |A| + |B| \geq |A|$. Accordingly, setting $A = I + \sigma_{\mathcal{T}}^{-2}(K_{\mathcal{T}\mathcal{T}} - K_{\mathcal{T}\mathcal{S}}(K_{\mathcal{S}\mathcal{S}} + \sigma_{\mathcal{S}}^2 I)^{-1}K_{\mathcal{S}\mathcal{T}})$ and $B = \sigma_{\mathcal{T}}^{-2}K_{\mathcal{T}\mathcal{S}}(K_{\mathcal{S}\mathcal{S}} + \sigma_{\mathcal{S}}^2 I)K_{\mathcal{S}\mathcal{T}}$, it immediately follows that,

$$\begin{aligned} & \mathbb{I}([y_{\mathcal{T}}^{(1)}, \dots, y_{\mathcal{T}}^{(n_{\mathcal{T}})}]; \mathbf{f}_{\mathcal{T}} | \mathcal{D}_{\mathcal{S}}) \\ & \leq \mathbb{I}([y_{\mathcal{T}}^{(1)}, \dots, y_{\mathcal{T}}^{(n_{\mathcal{T}})}]; \mathbf{f}_{\mathcal{T}}) \text{ if } \lambda \neq 0 \end{aligned} \tag{5.9}$$

and,

$$\begin{aligned} & \mathbb{I}([y_{\mathcal{T}}^{(1)}, \dots, y_{\mathcal{T}}^{(n_{\mathcal{T}})}]; \mathbf{f}_{\mathcal{T}} | \mathcal{D}_{\mathcal{S}}) \\ & = \mathbb{I}([y_{\mathcal{T}}^{(1)}, \dots, y_{\mathcal{T}}^{(n_{\mathcal{T}})}]; \mathbf{f}_{\mathcal{T}}) \text{ if } \lambda = 0, \end{aligned} \tag{5.10}$$

for any set $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n_{\mathcal{T}})} \in \mathcal{X}_{\mathcal{T}}$. The statement of the theorem is directly implied by the above. ■

The theorem suggests that if there exists any source-target similarity, as reflected by a non-zero value of the learned parameter λ , then *Mani*-TBO is able to exploit the relationship to find the optimal solutions more quickly, on average. However, if no source-target similarity exists, then, in principle, the proposed approach should be no worse than classical BO (assuming the UCB acquisition function). It should be noted that the theorem is agnostic to the form the feature transformation M . Specifically, these theoretical regret bounds apply to both learned, and random feature transformations M .

5.5 Evaluating the *Mani*-TBO Algorithm

In this section, the efficacy of the proposed *Mani*-TBO algorithm is showcased against a classical baseline BO approach on a series of synthetic examples; the multi-modal *Griewank*, *Rastrigin*, and *Six-Hump Camel* functions, as well as the

unimodal *Sphere* function [135].

5.5.1 Experimental Details of *Mani*-TBO

For these experiments, the baseline GP is obtained from the *gpml* package [136]. In the current implementation of *Mani*-TGP, the scarcely available data in typical BO applications is catered to by evolving (small) single-layer neural networks with rectified linear activation function for the source feature transformations. Additionally, random and learned feature transformations M are considered, with the LOO log predictive probability (LOO-LPP - Eq. 3.24) criteria used for training model hyperparameters.

In these experiments, the EI and UCB acquisitions are both considered as the acquisition function, with β in Equation 5.3 defined as per [158], as well as learned and random feature transformations M . In all experiments, target optimization starts with a dataset containing 5 training points that are generated by LHS. The search of the acquisition function (i.e. Eq. 5.2 or 5.3) for both *Mani*-TBO and traditional BO is conducted using a genetic algorithm (GA) with the following evolutionary operators: the simulated binary crossover (SBX) [102], the polynomial mutation [103], and the $(\mu + \lambda)$ -ES [104] selection strategy. In all numerical experiments shown in this chapter, $\mu = \lambda$. Details of the parameter settings are shown in Table 5.1.

Not limiting to Bayesian optimization, the efficacy of *Mani*-TBO is highlighted against *random search* which has seen empirical success for hyperparameter tuning problems under strict computational budgets [160, 161]. In order to ensure that the obtained results are not misleading under the influence of randomization, the performances of all optimizers are averaged across 20 independent runs (under a limited computational budget).

Table 5.1: Parameter settings for the GA used to optimize the acquisition function for both *Mani*-TBO and traditional BO.

Parameter	Setting
Number of GA iterations	500
Size of GA Population	20
Mutation Probability	$1/d_{\mathcal{T}}$
SBX and Mutation Index	10
Crossover Probability	0.9

5.5.2 Experimental Results of *Mani*-TBO on Synthetic Functions

In the first set of examples, the target optimization tasks are the 5D Griewank, 5D Rastrigin, and 2D Six-Hump Camel functions (see Table 5.2 for the characteristics of the functions). As *related* source tasks, 50 data points are drawn from distinct evaluations of the target functions, but with 2 additional randomly inserted features. These details are summarized in Table 5.3. As before, the source feature transformation mechanism of the *Mani*-TGP model is found to induce high correlation between source and target tasks, improving the performance of *Mani*-TBO compared to traditional BO (as well as random search) - as shown in Figures 5.1a-c and Table 5.4. To elaborate, the min function value achieved by BO-EI at 30 function evaluations (FEs) is accomplished by *Mani*-TBO-EI at roughly 23 FEs for the 5D Griewank function and 20 FEs for the 5D Rastrigin function. This equates to a savings of 23% and 30%, respectively.

Table 5.2: The characteristics of the synthetic functions used in the experimental evaluation of *Mani*-TBO.

Functions	Dataset Characteristics	Global Minimum
<i>Griewank</i>	Consists of many widespread, regularly distributed local minima.	$f(x^{(opt)}) = 0, x^{(opt)} = (0, \dots, 0)$.
<i>Rastrigin</i>	Consists of many widespread, regularly distributed local minima.	$f(x^{(opt)}) = 0, x^{(opt)} = (0, \dots, 0)$.
<i>Six-Hump Camel</i>	Consists of 4 local minima and 2 global minima.	$f(x^{(opt)}) = -1.0316$, $x^{(opt)} = (0.0898, -0.7126)$ and $(-0.0898, 0.7126)$.
<i>Sphere</i>	Continuous function that is convex and unimodal.	$f(x^{(opt)}) = 0, x^{(opt)} = (0, \dots, 0)$.

Table 5.3: The configuration of the synthetic test functions used in this numerical study. Note that the *random features* correspond to the number of randomly inserted features that do not affect the value of the objective function.

Functions	Source		Target	
	Original Dimension	Random Dimensions	Original Dimension	Random Dimensions
<i>Griewank</i>	5	2	5	0
<i>Rastrigin</i>	5	2	5	0
<i>Six-Hump Camel</i>	2	2	2	0
<i>Sphere</i>	7	0	5	0

Figure 5.1: (a)-(d) Optimization search convergence trend comparison between *Mani*-TBO, traditional BO (from scratch) - using both UCB and EI acquisition functions, and random search for the 5D Griewank, 5D Rastrigin, 2D Six-Hump Camel and 5D Sphere functions, respectively. Results are averaged across 20 independent runs.

In another example (Figure 5.1d and Table 5.4), a 5D sphere function serves as the target optimization task (see Table 5.2 for the characteristics of the function). Source data is generated from a full 7D sphere function (i.e., there are no ran-

dom features). However, even in this relatively challenging scenario, *Mani*-TBO continues to show superior convergence characteristics as it leverages whatever transferable knowledge is available. Specifically, to achieve a min function value of 2500, *Mani*-TBO-EI requires 17 FEs relative to 25 FEs for BO-EI, which represents a savings of 32%.

Its interesting to note also that, in all cases, the performance of both the learned and random feature transformation variants of *Mani*-TBO are roughly equivalent. During the earlier stages of the search process, the random feature transformation performs better, with the exception of the 2D *Six-Hump Camel* where the convergence characteristics of *Mani*-TBO-EI are superior to *Mani*-TBO-EI-RT. At the latter stages of the optimization process, however, learning the transformation produces superior results. Specifically, at 25 FEs, learning outperforms random *M* in 5/8 cases. One possible explanation for this observation is the fact that the random feature transformation variant of *Mani*-TBO optimizes fewer hyperparameters relative to its learned feature transformation counterpart. As fewer data points are available at the earlier stages of the search process, the former, with fewer hyperparameters, sees a larger convergence rate. However, in the later stages, with a larger population of solutions, the learned variant is able to achieve superior results. Based on these results and observations, for the computationally expensive real world case studies that follow, the learned feature transformation variant of *Mani*-TBO will be used.

Table 5.4: Comparing the mean performance and standard deviation of *Mani*-TBO and BO using the Expected Improvement (EI) and Upper Confidence Bound (UCB) acquisition functions, with learned and random (RT) source feature transformations. Statistically significant superior results relative to BO for the particular acquisition function evaluated using a Wilcoxon signed rank test with a 95% confidence interval are presented in bold.

	<i>Griewank</i>				<i>Rastrigin</i>			
	15 FEs	20 FEs	25 FEs	30 FEs	15 FEs	20 FEs	25 FEs	30 FEs
<i>Mani</i> -TBO - EI	79.06 ± 15.40	17.27 ± 27.46	1.87 ± 0.60	1.18 ± 0.20	2568.08 ± 670.79	569.06 ± 1114.04	64.97 ± 23.21	41.79 ± 12.44
<i>Mani</i> -TBO - EI - RT	76.58 ± 19.67	4.48 ± 5.72	1.88 ± 0.61	1.10 ± 0.26	2208.80 ± 686.14	129.02 ± 86.81	77.64 ± 26.13	42.05 ± 17.37
BO - EI	82.11 ± 15.89	76.18 ± 16.53	42.76 ± 42.08	42.09 ± 42.74	2655.82 ± 610.25	2066.39 ± 1173.61	1043.09 ± 1254.95	649.07 ± 1183.83
<i>Mani</i> -TBO - UCB	75.54 ± 17.87	66.83 ± 27.66	6.31 ± 5.44	3.66 ± 1.51	2519.99 ± 628.80	2280.80 ± 820.73	158.36 ± 70.91	82.05 ± 42.95
<i>Mani</i> -TBO - UCB - RT	69.70 ± 17.09	66.25 ± 21.07	7.55 ± 4.89	3.31 ± 2.29	2122.14 ± 582.97	1951.55 ± 716.54	269.36 ± 125.59	83.03 ± 52.11
BO - UCB	82.11 ± 15.89	82.11 ± 15.89	78.20 ± 13.48	41.66 ± 42.83	2655.82 ± 610.25	2655.82 ± 610.25	2532.52 ± 827.00	1773.58 ± 1327.69
<i>Sphere</i>								
<i>Six-Hump Camel</i>								
<i>Mani</i> -TBO - EI	10 FEs	15 FEs	20 FEs	25 FEs	15 FEs	20 FEs	25 FEs	30 FEs
<i>Mani</i> -TBO - EI - RT	-0.16 ± 0.56	-0.42 ± 0.44	-0.43 ± 0.45	-0.49 ± 0.34	2660.74 ± 493.96	279.98 ± 546.55	27.06 ± 18.49	6.91 ± 4.50
BO - EI	-0.34 ± 0.44	-0.38 ± 0.43	-0.54 ± 0.30	-0.55 ± 0.29	2026.80 ± 995.02	241.28 ± 567.10	25.04 ± 12.42	8.11 ± 6.57
<i>Mani</i> -TBO - UCB	0.28 ± 1.09	-0.06 ± 0.85	-0.41 ± 0.31	-0.45 ± 0.26	2815.69 ± 330.53	2778.78 ± 301.69	2495.06 ± 901.50	2494.34 ± 903.58
<i>Mani</i> -TBO - UCB - RT	-0.02 ± 0.86	-0.48 ± 0.34	-0.57 ± 0.32	-0.64 ± 0.29	2632.22 ± 529.44	2513.73 ± 494.12	206.05 ± 166.18	82.21 ± 52.65
BO - UCB	0.03 ± 0.90	-0.33 ± 0.34	-0.40 ± 0.24	-0.45 ± 0.25	2560.64 ± 499.65	2385.50 ± 725.16	151.35 ± 109.49	50.16 ± 41.24
	0.28 ± 1.09	0.28 ± 1.09	0.01 ± 0.90	-0.41 ± 0.36	2815.69 ± 330.53	2815.69 ± 330.53	2815.69 ± 330.53	41.67 ± 13.75

Figure 5.2: Random samples from the USPS (**Left** - Source) and MNIST (**Right** - Target) datasets with identical scaling.

5.6 Real World Examples

5.6.1 Accelerated AutoML using Heterogeneous Knowledge Transfer

Determining the optimal set of hyperparameters for machine learning models in a principled manner can be a time consuming affair as it requires a rigorous cross-validation procedure. Yet, an appropriate selection of these hyperparameters can often be the difference between state-of-the-art performance and a useless model. This task becomes especially daunting when dealing with large datasets and high-dimensional feature spaces. As the evaluation of a given set of hyperparameters entails the complete training of the underlying model, in such cases, the task can be considerably expensive.

Table 5.5: Upper and lower bounds of hyperparameters in the source and target neural networks of the accelerated AutoML case study.

Features	Source	Target	Lower Bound	Upper Bound
Learning Rate	✓	✓	10^{-5}	0.5
Momentum	✓	✓	0	1
L2 Regularization	✓	✓	0	0.1
Batch Size	✓	✓	32	256
Nodes in Hidden Layers	✓	×	10	200

In this experiment, the efficacy of *Mani*-TBO with regard to harnessing hid-

den correlations between the hyperparameter configurations of different neural network architectures is demonstrated. In particular, the target task is considered to be the hyperparameter tuning exercise of a convolutional neural network (CNN), with one convolutional layer followed by two densely connected layers, applied to the MNIST dataset (28 x 28 pixel images). 50 source data points are taken from hyperparameter configurations of the CNN applied to the USPS dataset (16 x 16 pixel images). The network utilizes dropout for regularization, the rectifier linear unit activation function, and weight updates via stochastic gradient descent with momentum. Note that as the USPS dataset is (1/6)th the size of MNIST, it is much cheaper to evaluate.

The network hyperparameters considered in the source are the learning rate, momentum coefficient, regularization coefficient, batch size, dropout probability, and the number of hidden units in the fully connected layers. On the other hand, in the larger target optimization problem, one of the source hyperparameters (namely, the number of hidden units in the fully connected layers) is assumed fixed to 100 (as a means of limiting the complexity of the learning algorithm), thereby constituting a heterogeneous transfer setting. The bounds of the hyperparameters considered for both datasets are summarized in Table 5.5.

While both MNIST and USPS cater to handwritten digit recognition, the two learning tasks are seemingly disparate as it is difficult to *a priori* ascertain the relation between neural networks designed for two related but distinct problems. A close inspection of the two datasets, as presented in Figure 5.2, reveals several obvious differences, e.g. image resolution, slanted numbers, padding, and thicker lines. Consequently, the direct utilization of a model trained on one dataset for another is unlikely to succeed [162]. Nevertheless, observations from Figure 5.4a show that the *Mani*-TBO approach is capable of utilizing the underlying similarities across the two datasets to accelerate the search for good hyperparameter configurations as opposed to the other baseline algorithms. For example, to achieve a min prediction error of 0.014, BO - EI requires an average of 30 FEs,

while *Mani*-TBO - EI requires 20. This equates to approximately 10 FEs saved (or $\sim 33\%$ savings).

5.6.2 A Case Study in Global Black-box Optimization in Materials Engineering augmented by Heterogeneous Knowledge Transfer

The engineering design application domain is generally characterized by incremental enhancements over time to the products under development. As such, design exercises are bound to share much in common, giving rise to the potential of knowledge transfer. Further, high-fidelity physics simulations, which are essential elements of engineering analysis, are typically computationally expensive, consuming several minutes, hours, or even days for a single evaluation; often forming major bottlenecks in the design cycle. Therefore, there is much to be gained through savings in the number of function evaluations required to achieve near-optimal designs.

Figure 5.3: Workflow illustration of a generic *Liquid Composite Moulding* process for fiber-reinforced polymer composite parts [2].

To this end, the efficacy of *Mani*-TBO is demonstrated on a real-world *simulation-based* engineering design optimization problem. In particular, herein, two distinct members of a family of rigid tool *liquid composite moulding* (LCM) processes, i.e.

Table 5.6: Upper and lower bounds of design variables of the I/C-LCM and RTM processes.

Features	Source	Target	Lower Bound	Upper Bound
$V_{closure}$	✓	✓	1 mm/min	10mm/min
P_{inj}	✓	✓	1 MPa	10 MPa
T_{mold}	✓	✓	293 K	348 K
T_{resin}	✓	✓	293 K	348 K
$V_{closure}^{final}$	✓	×	1 mm/min	10mm/min
H_{inj}	✓	×	0.8 cm	1 cm

Figure 5.4: Optimization search convergence trend comparison between *ManiTBO*, BO and random search for: **(a)** automatic hyperparameter tuning (5 target and 6 source dimensions), and **(b)** materials engineering (4 target and 6 source dimensions) problems.

the *resin transfer moulding* (RTM), and the *injection/compression liquid composite moulding* (I/C-LCM) is considered. These processes, illustrated in Figure 5.3, are popular methodologies for the manufacture of high-quality composite parts in the aero and auto industries.

In this case study, the manufacture of a particular composite material (glass-fibre + epoxy) part using the two distinct processes are considered. The target task is the single phase RTM process that involves 4 design variables, namely, (a) speed of mold closure $V_{closure}$, (b) resin injection pressure P_{inj} , (c) preheated mold temperature T_{mould} , and (d) preheated resin temperature T_{resin} . For *ManiTBO*, source data is drawn from a previous optimization exercise on a related I/C-LCM process. Unlike RTM, the I/C-LCM process has 6 design variables, four of which are overlapping with RTM, while the other two are additional.

The new features included in the case of I/C-LCM are (a) mold cavity thickness during resin injection H_{inj} , and (b) velocity of final mold closure $V_{closure}^{final}$. The design variables for the two tasks and their bounds are summarized in Table 5.6. For a more detailed discussion about the two processes, the interested reader is referred to [86]. In this section, the objective function is a weighted aggregation of the *process time* (which determines the throughput) and the *peak internal force* (which determines the set up and running cost of peripheral equipment).

Although RTM and I/C-LCM have different feature dimensionalities, they belong to the same family of rigid tool LCM processes. Thus, one intuitively expects there to exist some form of underlying synergy between them. This intuition explains the observation in Figure 5.4b, where, under the proposed source feature transformation, there is a resulting impetus to the convergence rate of *Mani*-TBO. For example, for achieving an objective function value of 0.04, *Mani*-TBO with UCB takes an average of 10 function evaluations, whereas traditional BO (which starts the search from scratch) consumes 17 function evaluations (indicating a savings of approximately 41%).

5.7 Summary

In this chapter, in fulfilment of Research Objectives 1, and 4, inspired by a human-like approach of transferring knowledge across problems, a generalization of *transfer Bayesian optimization* is proposed to address problems that may originate from heterogeneous domains. To the best of my knowledge, this represents the first work that develops a Bayesian optimization framework that is augmented by knowledge transfer capabilities across optimization problems with non-homogeneous search spaces. In particular, a *manifold transfer Gaussian process* (*Mani*-TGP) model is developed that learns a new representation for source features, such that they are better aligned with the target task of interest. As an alternative strategy, a random feature representation can be used for *Mani*-

TGP in place of a learned one in order to reduce the complexity of the learning process at the cost of diminished expressive capabilities. Further, the extent of transfer between problems during optimization is automatically modulated by the *Mani*-TGP model which explicitly learns how similar the problems are. Tighter regret bound results for *Mani*-TBO compared to the classical BO case are derived thereby providing theoretical justifications for the algorithm.

An exploratory empirical analysis of the proposed *Mani*-TGP model verifies its capabilities of learning the identity matrix as the feature transformation when source and target tasks are identical (at least for low-dimensional problems), and a λ value close to 1. In addition, the predictive performance of the *Mani*-TGP model is shown to be superior to traditional GP on a dataset from the aircraft engine design industry. The experiments examining the capabilities of *Mani*-TBO on both synthetic test functions and practical case studies verify the effectiveness of incorporating the *Mani*-TGP model within the framework of Bayesian optimization, saving as much as 41% of function evaluations, even in this relatively challenging heterogeneous transfer case.

Chapter 6

Conclusion

The overarching objective of this dissertation, entitled “*Transfer Optimization for Complex Engineering Design*”, is to enhance the performance of surrogate-assisted optimization algorithms through the utilization of knowledge from related but distinct tasks using transfer learning approaches. Focused specifically on the application domain of complex engineering design, this goal was successfully achieved through the development of 3 novel adaptive transfer learning methodologies for regression, and two frameworks for transfer surrogate-assisted optimization that showed superior convergence characteristics relative to their traditional counterparts.

6.1 Conclusions

- In partial fulfilment of Research Objectives 1 through 4 (see Section 1.2), a case study within the application area of aircraft engine design is presented in Chapter 3 where the capabilities of transfer learning are demonstrated. In particular, three use cases are described that represent scenarios design engineers commonly face in practice. The first use case corresponds to a scenario where the baseline parameters (which describe the configuration of the engine) is identical between the source and target datasets but an

additional target input parameter is considered. In such cases, a simple naïve-transfer Gaussian process model (NT-GP - where the two datasets are simply concatenated), can lead to superior predictive performance relative to traditional GP. However, when the baseline parameters differ between source and target, this simplistic methodology leads to negative transfer. Consequently, adaptive transfer learning methodologies that are capable of automatically accounting for the similarities between source and target are proposed and evaluated. The methodologies considered in this study are the adaptive transfer GP [1], and three novel methodologies:

- **Metaheuristic Instance Selection for Transfer:** A model-independent supervised transfer learning algorithm that selects the most relevant subset of source data, using metaheuristic search algorithms, in an adaptive fashion in order to maximize target predictive accuracy.
- **Transfer-Stacking Gaussian Process - Multi-problem Surrogate:** A model-based supervised transfer learning algorithm that transfers multiple base-models adaptively using transfer stacking in order to improve predictive performance.
- **Manifold Transfer Gaussian Process:** A supervised transfer learning algorithm that facilitates the utilization of data from a heterogeneous source domain using a source feature transformation that is learned jointly with model parameters.

In use case 2, where the baseline parameters differ but inputs are identical i.e. homogeneous transfer, utilizing these adaptive methodologies leads to, generally, improved predictive performance relative to both traditional GP and NT-GP. Similarly, in use case 3, where both the baseline parameters and the input dimensionalities differ i.e. heterogeneous transfer, *Mani*-TGP shows superior performance relative to NT-GP and traditional GP. To further evaluate the performance of these algorithms, real world data from Rolls-Royce

corresponding to three use cases are evaluated, leading to similar results. These promising outcomes demonstrate the capabilities of transfer learning in adaptively leveraging upon knowledge from source datasets in order to achieve improved accuracy in the predictive models.

- In fulfilment of Research Objectives 1 through 3 (see Section 1.2), a novel transfer evolutionary optimization framework for computationally expensive multiobjective optimization problems based on the key idea of multi-problem surrogates (TEMO-MPS) is presented in Chapter 4. In many complex engineering design application domains, the complexity of the task necessitates the division and simplification of tasks. Consequently, at any one time, there is generally a host of project groups working in parallel on a variety of related design problems. As such, there exists a tremendous amount of information streaming in constantly that can be leveraged in order to improve optimization performance. Accordingly, TEMO-MPS adaptively exploits knowledge embedded in surrogate models from distinct (but possibly related) design exercises as they become available to augment the optimization of a new target problem of interest.

To explore the capabilities of the proposed framework, experiments were conducted on a series of synthetic benchmark functions, a complex composite materials manufacturing case study and a case study on automatic hyperparameter tuning of machine learning models. Results show that TEMO-MPS significantly outperformed ParEGO, a popular surrogate-assisted multi-objective optimization algorithm, saving as much as 70% of function evaluations through relevant knowledge exploitation. Careful analysis indicates that the capabilities of TEMO-MPS is affected by the similarity between source and target tasks. Specifically, the adaptive nature of the MPS component is theoretically shown to play a key role in regulating the contributions of the source models - exploiting relationships when sources are highly

correlated, and minimizing negative impact when sources are unrelated.

- Finally, in fulfilment of Research Objectives 1 and 4 (see Section 1.2), a novel transfer Bayesian optimization framework (*Mani*-TBO) that can facilitate transfer across heterogeneous problems is presented in Chapter 5. In practical scenarios, design analyses are in constant state of flux, rapidly changing various components of the problem in order to better understand the phenomena under study. To elaborate, these problems can have similar, but not necessarily the same, search space - leading to differences in dimensionality. The capability to transfer knowledge across such domains would be a tremendous boon when dealing computationally expensive problems. To address this issue, the proposed framework leverages the *Manifold Transfer Gaussian Process* (*Mani*-TGP) model which learns a new representation for source features in order to better align the source to the target task of interest. In particular, the instantiation of the framework considered induces a new latent representation through an l -layer neural network. Further, the extent of which to transfer is automatically modulated by the *Mani*-TGP model which explicitly learns how similar the problems are. Further, tighter regret bound results relative to classical Bayesian optimization are derived, thereby providing theoretical justification for the algorithm.

To explore the capabilities of the proposed framework, experiments are carried out on a series of synthetic benchmark functions, a complex composite manufacturing case study and a hyperparameter tuning exercise of a machine learning model. Analysis of the predictive capabilities of confirm that, as seen in the use case 3 of the aircraft engine design case study, *Mani*-TGP demonstrates improved predictive accuracy relative to GP after a certain minimal amount of target data points are available. Further, optimization results show superior convergence characteristics as compared to classical Bayesian optimization when considering identical acquisition functions, as

well as to CMAES and Random Search - saving as much as 41% of function evaluations.

6.2 Open Problems

In this dissertation, the groundwork for transfer optimization, in which knowledge from distinct (but possibly related) sources is leveraged in order to improve convergence performance, has been laid. In what follows, several potential future directions for further research within the domain of knowledge transfer for optimization is proposed.

- While promising, in the present study, the *Mani*-TBO framework is restricted to the utilization of data from only a single source. In practical scenarios, where sources may be plentiful, the proper selection of the aforementioned source may become a challenging task unto itself. Consequently, it would be a tremendous boon to be able to adaptively transfer knowledge from multiple sources in order to achieve further improved convergence characteristics.
- The *Mani*-TGP model utilizes a fully-connected l -layer neural network for its feature transformation in order to align the source features to the target so as to be more conducive for transfer. While demonstrably effective in the evaluated test scenarios, other forms of neural architecture could facilitate transfer from other sources. For example, a convolutional neural network architecture could be beneficial when the source data has strong local spatial correlation (e.g. images, sequences). Certainly, the possibility of transferring knowledge across different types of data in order to better model the phenomena under investigation can be of tremendous benefit.
- In the present study, the instantiation of the TS-GP module in the TEMO-MPS framework incorporates a linear meta-regressor to adaptively combine

the numerous source models with the target model in order to improve predictive performance. While effective, further investigation could be carried out on the possibility of utilizing non-linear meta-regressors in order to provide greater flexibility in the manner in which the various models are combined. Such a methodology could potentially better leverage knowledge from the various source models, thereby further improving upon the convergence characteristics of TEMO-MPS.

References

- [1] Bin Cao, Sinno Jialin Pan, Yu Zhang, Dit-Yan Yeung, and Qiang Yang. Adaptive transfer learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 407–712. AAAI Press, Atlanta, Georgia, July 11–15 2010.
- [2] Yew Soon Ong and Abhishek Gupta. Evolutionary multitasking: A computer science view of cognitive multitasking. *Cognitive Computation*, 8(2):125–142, March 2016.
- [3] Jorge Nocedal and Stephen J Wright. Conjugate gradient methods. *Numerical optimization*, pages 101–134, 2006.
- [4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [5] David B Fogel. An introduction to simulated evolutionary optimization. *IEEE transactions on neural networks*, 5(1):3–14, 1994.
- [6] Yaochu Jin and Jürgen Branke. Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on evolutionary computation*, 9(3):303–317, 2005.
- [7] John David Anderson and J Wendt. *Computational fluid dynamics*, volume 206. Springer, 1995.

- [8] Wenbin Song and Andy J Keane. Surrogate-based aerodynamic shape optimization of a civil aircraft engine nacelle. *AIAA Journal*, 45(10):2565–2574, 2007.
- [9] Yew S Ong, Prasanth B Nair, and Andrew J Keane. Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA Journal*, 41(4):687–696, April 2003.
- [10] Yew Soon Ong, Prasanth B Nair, and Kai Yew Lum. Max-min surrogate-assisted evolutionary algorithm for robust design. *IEEE Transactions on Evolutionary Computation*, 10(4):392–404, 2006.
- [11] Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on evolutionary computation*, 6(5):481–494, 2002.
- [12] Chaoli Sun, Jinliang Ding, Jianchao Zeng, and Yaochu Jin. A fitness approximation assisted competitive swarm optimizer for large scale expensive optimization problems. *Memetic Computing*, pages 1–12, 2016.
- [13] Zongzhao Zhou, Yew Soon Ong, Prasanth B Nair, Andy J Keane, and Kai Yew Lum. Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(1):66–76, 2007.
- [14] Dudy Lim, Yaochu Jin, Yew Soon Ong, and Bernhard Sendhoff. Generalizing surrogate-assisted evolutionary computation. *IEEE Trans. Evol. Comput.*, 14(3):329–355, December 2010.
- [15] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

- [16] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, October 2010.
- [17] Sushil J Louis and John McDonnell. Learning with case-injected genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8(4):316–328, 2004.
- [18] Kevin Swersky, Jasper Snoek, and Ryan P Adams. Multi-task bayesian optimization. In *Advances in Neural Information Processing Systems 26*, pages 2004–2012. Curran Associates, Inc., Lake Tahoe, NV, December 5–10 2013.
- [19] Matthias Feurer, Jost Tobias Springenberg, and Frank Hutter. Initializing bayesian hyperparameter optimization via meta-learning. In *AAAI*, pages 1128–1135, 2015.
- [20] Alan Tan, Ramon Sagarna, Abhishek Gupta, Yew-Soon Ong, and Chi Keong Goh. Knowledge transfer through machine learning in aircraft design. *IEEE Computational Intelligence Magazine*, 12(4):48–60, 2017.
- [21] Alan Tan, Yew-Soon Ong, Abhishek Gupta, and Chi-Keong Goh. Multi-problem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems. *IEEE Transactions on Evolutionary Computation*, 23(1):15–28, 2017.
- [22] Gobinda G Chowdhury. Natural language processing. *Annual Review of Information Science and Technology*, 37(1):51–89, January 2003.
- [23] Nuria M Oliver, Barbara Rosario, and Alex P Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):831–843, August 2000.

- [24] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, Graz, Austria, September 7–13 2006.
- [25] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749, April 2005.
- [26] Jaime G Carbonell, Ryszard S Michalski, and Tom M Mitchell. An overview of machine learning. In *Machine learning*, pages 3–23. Elsevier, 1983.
- [27] Jack D Mattingly, William H Heiser, and David T Pratt. *Aircraft Engine Design*. AIAA Education Series, Reston, VA, 2nd edition, 2002.
- [28] Egbert Torenbeek. *Advanced Aircraft Design: Conceptual Design, Technology and Optimization of Subsonic Civil Airplanes*. John Wiley & Sons, Hoboken, New Jersey, 2013.
- [29] Joaquim RRA Martins and Andrew B Lambe. Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51(9):2049–2075, July 2013.
- [30] P Fuglsang and H Aagaard Madsen. Optimization method for wind turbine rotors. *Journal of Wind Engineering and Industrial Aerodynamics*, 80(1):191–206, March 1999.
- [31] R Choudhary, A Malkawi, and PY Papalambros. Analytic target cascading in simulation-based building design. *Automation in construction*, 14(4):551–568, November 2005.
- [32] Ilan Kroo, Steve Altus, Robert Braun, Peter Gage, and Ian Sobieski. Multidisciplinary optimization methods for aircraft preliminary design. In *5th Symposium on Multidisciplinary Analysis and Optimization*, pages 697–709. AIAA, Panama City Beach, FL, September 7–9 1994.

- [33] Raphael T Haftka, Jaroslaw Sobieszczanski-Sobieski, and Sharon L Padula. On options for interdisciplinary analysis and design optimization. *Structural Optimization*, 4(2):65–74, January 1992.
- [34] Evin J Cramer, John E Dennis, Jr, Paul D Frank, Robert Michael Lewis, and Gregory R Shubin. Problem formulation for multidisciplinary optimization. *SIAM Journal on Optimization*, 4(4):754–776, July 1994.
- [35] Richard J Balling and Jaroslaw Sobieszczanski-Sobieski. Optimization of coupled systems: A critical overview of approaches. *AIAA Journal*, 34(1):6–17, January 1996.
- [36] Natalia M Alexandrov and M Yousuff Hussaini. *Multidisciplinary Design Optimization: State of The Art*, volume 80. SIAM, Philadelphia, PA, 1997.
- [37] Russell W Claus, Austin L Evans, and Gregory J Follen. Multidisciplinary propulsion simulation using NPSS. In *4th Symposium on Multidisciplinary Analysis and Optimization*, pages 191–196. AIAA, Cleveland, OH, September 21–23 1992.
- [38] G Medic, G Kalitzin, D You, M Herrmann, F Ham, E van der Weide, H Pitsch, and JJ Alonso. Integrated RANS/LES computations of turbulent flow through a turbofan jet engine annual research briefs. In *Annual Research Briefs 2006*, pages 275–285. Center for Turbulence Research, University of Stanford, Stanford, CA, 2006.
- [39] Robert C Martin. *Agile Software Development: Principles, Patterns, and Practices*. Prentice Hall, Upper Saddle River, NJ, 2002.
- [40] Angappa Gunasekaran. *Agile Manufacturing: The 21st Century Competitive Strategy*. Elsevier, Amsterdam, Netherlands, 2001.
- [41] M. N. Le, Y. S. Ong, Y. Jin, and B. Sendhoff. A unified framework for symbiosis of evolutionary mechanisms with application to water clusters

- potential model design. *IEEE Comput. Intell. Mag.*, 7(1):20–35, January 2012.
- [42] Nestor V Queipo, Raphael T Haftka, Wei Shyy, Tushar Goel, Rajkumar Vaidyanathan, and P Kevin Tucker. Surrogate-based analysis and optimization. *Progress in aerospace sciences*, 41(1):1–28, January 2005.
- [43] Timothy W Simpson, Vasilli Toropov, Vladimir Balabanov, and Felipe AC Viana. Design and analysis of computer experiments in multidisciplinary design optimization: A review of how far we have come or not. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 10–12. AIAA, Victoria, Canada, September 1–22 2008.
- [44] Alexander IJ Forrester and Andy J Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1):50–79, January 2009.
- [45] Alexander I J. Forrester, Andy J Keane, and Neil W Bressloff. Design and analysis of “noisy” computer experiments. *AIAA Journal*, 44(10):2331–2339, October 2006.
- [46] Wei Huang, Zhen-Guo Wang, Derek B Ingham, Lin Ma, and Mohamed Pourkashanian. Design exploration for a single expansion ramp nozzle (sern) using data mining. *Acta Astronautica*, 83:10–17, November 2013.
- [47] Wei Huang. Design exploration of three-dimensional transverse jet in a supersonic crossflow based on data mining and multi-objective design optimization approaches. *International Journal of Hydrogen Energy*, 39(8):3914–3925, March 2014.
- [48] David J J. Toal and Andy J Keane. Efficient multipoint aerodynamic design optimization via cokriging. *Journal of Aircraft*, 48(5):1685–1695, September 2011.

- [49] Nateri K Madavan, Man Mohan Rai, and Frank W Huber. Neural net-based redesign of transonic turbines for improved unsteady aerodynamic performance. Technical Report NASA/TM-1998-208754, National Aeronautics and Space Administration, Ames Research Center, Los Angeles, CA, November 1998.
- [50] Man Mohan Rai and Nateri K Madavan. Aerodynamic design using neural networks. *AIAA Journal*, 38(1):173–182, September 2000.
- [51] Cameron J Miller, Adam C Maser, Elena Garcia, and Dimitri N Mavris. Invent surrogate modeling and optimization of transient thermal responses. In *Proceedings of the 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. AIAA, Nashville, Tennessee, January 9–12 2012. doi: 10.2514/6.2012-1123.
- [52] Metin F Ozcan, Brian Kestner, Jimmy CM Tai, and Dimitri N Mavris. Pattern classification of a civilian turbofan’s state space for real-time surrogate modeling. In *49th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*. AIAA, Cleveland, OH, July 28–30 2013. doi: 10.2514/6.2013-4003.
- [53] Qasim Zeeshan, Dong Yunfeng, Saqlain Ghumman, Amer Farhan Rafique, and Ali Kamran. Support vector regression-driven multidisciplinary design optimization of multistage ground based interceptor. In *AIAA Modeling and Simulation Technologies Conference*. AIAA, Chicago, IL, August 10–13 2009. doi: 10.2514/6.2009-6238.
- [54] Oliver Bandte. *A Probabilistic Multi-criteria Decision Making Technique for Conceptual and Preliminary Aerospace Systems Design*. PhD thesis, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 2000.

- [55] Hong-Min Kim and Kwang-Yong Kim. Design optimization of rib-roughened channel to enhance turbulent heat transfer. *International Journal of Heat and Mass Transfer*, 47(23):5159–5168, November 2004.
- [56] Yang Yanhui, Liu Dong, He Ziyang, and Luo Zijian. Optimization of preform shapes by RSM and FEM to improve deformation homogeneity in aerospace forgings. *Chinese Journal of Aeronautics*, 23(2):260–267, April 2010.
- [57] Minh Nghia Le, Yew Soon Ong, Stefan Menzel, Chun-Wei Seah, and Bernhard Sendhoff. Multi co-objective evolutionary optimization: Cross surrogate augmentation for computationally expensive problems. In *2012 IEEE Congress on Evolutionary Computation*. IEEE, Brisbane, Australia, June 10–15 2012. doi: 10.1109/CEC.2012.6252915.
- [58] Nilay Papila, Wei Shyy, Lisa Griffin, and Daniel J Dorney. Shape optimization of supersonic turbines using global approximation methods. *Journal of Propulsion and Power*, 18(3):509–518, May 2002.
- [59] Zhangjun Huang, Chengen Wang, Jian Chen, and Hong Tian. Optimal design of aeroengine turbine disc based on kriging surrogate models. *Computers & Structures*, 89(1):27–37, January 2011.
- [60] RS Sellar, SM Batill, and JE Renaud. Response surface-based, concurrent subspace optimization for multidisciplinary system design. In *34th Aerospace Sciences Meeting and Exhibit*. AIAA, Reno, NV, January 15–18 1996. doi: 10.2514/6.1996-714.
- [61] AJ Keane. Wing optimization using design of experiment, response surface, and data fusion methods. *Journal of Aircraft*, 40(4):741–750, July 2003.
- [62] Daniel M Jaeggi, Geoffrey T Parks, William N Dawes, and P John Clarkson. Robust multi-fidelity aerodynamic design optimization using surrogate models. In *12th AIAA/ISSMO Multidisciplinary Analysis and Opti-*

- mization Conference. AIAA, Victoria, Canada, September 10–12 2008. doi: 10.2514/6.2008-6052.
- [63] Edmondo Minisci and Massimiliano Vasile. Robust design of a reentry unmanned space vehicle by multifidelity evolution control. *AIAA Journal*, 51(6):1284–1295, June 2013.
 - [64] Brendan Tracey, Karthik Duraisamy, and Juan J Alonso. A machine learning strategy to assist turbulence model development. In *53rd AIAA Aerospace Sciences Meeting*. AIAA, Kissimmee, FL, January 5–9 2015. doi: 10.2514/6.2015-1287.
 - [65] Karthik Duraisamy, Ze Jia Zhang, and Anand Pratap Singh. New approaches in turbulence and transition modeling using data-driven techniques. In *53rd AIAA Aerospace Sciences Meeting*. AIAA, Kissimmee, FL, January 5–9 2015. doi: 10.2514/6.2015-1284.
 - [66] Eric J Parish and Karthik Duraisamy. A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of Computational Physics*, 305:758–774, November 2016.
 - [67] Bojan Dolšak, Ivan Bratko, and Anton Jezernik. Application of machine learning in finite element computation. In *Machine Learning and Data Mining: Methods and Applications, chapter 5*. John Wiley & Sons, Hoboken, New Jersey, 1998.
 - [68] Bojan Dolšak. Finite element mesh design expert system. *Knowledge-based Systems*, 15(5):315–322, July 2002.
 - [69] Chang-Hoi Ahn, Sang-Soo Lee, Hyuek-Jae Lee, and Soo-Young Lee. A self-organizing neural network approach for automatic mesh generation. *IEEE Trans. Magn.*, 27(5):4201–4204, September 1991.

- [70] Larry Manevitz, Malik Yousef, and Dan Givoli. Finite-element mesh generation using self-organizing neural networks. *Microcomputers in Civil Engineering*, 12(4):233–250, July 1997.
- [71] Tan Wei Min Alan, Ramon Sagarna, Abhishek Gupta, and Yew Soon Ong. Coping with data scarcity in aircraft engine design. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA AVIATION Forum*, Denver, Colorado, June 5–9 2017. doi: 10.2514/6.2017-4434.
- [72] Edwin V Bonilla, Kian Ming Adam Chai, and Christopher KI Williams. Multi-task gaussian process prediction. In *Advances in Neural Information Processing Systems 20*, pages 153–160. Curran Associates, Inc., Vancouver, Canada, December 3–6 2008.
- [73] Zhong-Hua Han and Stefan Görtz. Hierarchical kriging model for variable-fidelity surrogate modeling. *AIAA Journal*, 50(9):1885–1896, September 2012.
- [74] José Eugenio Valenzuela del Río and Dimitri Mavris. Gaussian process surrogate model for leveraging similar trends across concepts. *AIAA Journal*, 53(4):1002–1015, April 2015.
- [75] Abdelhamid Bouchachia. Incremental learning. In *Encyclopedia of Data Warehousing and Mining, Second Edition*, pages 1006–1012. IGI Global, Hershey, PA, 2009.
- [76] Dani Yogatama and Gideon Mann. Efficient transfer learning method for automatic hyperparameter tuning. *17th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 33:1077–1085, April 22-25 2014.
- [77] Dudy Lim, Yew Soon Ong, Abhishek Gupta, Chi Keong Goh, and Partha Sarathi Dutta. Towards a new praxis in optinformatics targeting knowledge re-use in evolutionary computation: simultaneous problem learn-

- ing and optimization. *Evolutionary Intelligence*, 9(4):203–220, December 2016.
- [78] Liang Feng, Yew Soon Ong, Siwei Jiang, and Abhishek Gupta. Autoencoding evolutionary search with learning across heterogeneous problems. *IEEE Trans. Evol. Comput.*, March 2017. doi: 10.1109/TEVC.2017.2682274.
- [79] Liang Feng, Yew Soon Ong, Meng-Hiot Lim, and Ivor W Tsang. Memetic search with interdomain learning: A realization between CVRP and CARP. *IEEE Trans. Evol. Comput.*, 19(5):644–658, October 2015.
- [80] Muhammad Iqbal, Will N Browne, and Mengjie Zhang. Reusing building blocks of extracted knowledge to solve complex, large-scale boolean problems. *IEEE Transactions on Evolutionary Computation*, 18(4):465–480, September 2014.
- [81] Mark W Hauschild, Martin Pelikan, Kumara Sastry, and David E Goldberg. Using previous models to bias structural learning in the hierarchical boa. *Evolutionary Computation*, 20(1):135–160, February 2012.
- [82] Shumeet Baluja. Incorporating a priori knowledge in probabilistic-model based optimization. In Martin Pelikan, Kumara Sastry, and Erick CantuPaz, editors, *Scalable Optimization via Probabilistic Modeling*, pages 205–222. Springer, New York City, NY, 2006.
- [83] Yiteng Zhai, Yew Soon Ong, and Ivor W Tsang. The emerging “big dimensionality”. *IEEE Comput. Intell. Mag.*, 9(3):14–26, July 2014.
- [84] Barbara Rakitsch, Christoph Lippert, Karsten Borgwardt, and Oliver Stegle. It is all in the noise: Efficient multi-task gaussian process inference with structured residuals. In *Advances in Neural Information Processing Systems 26*, pages 1466–1474. Curran Associates, Inc., Lake Tahoe, NV, December 5–10 2013.

- [85] Abhishek Gupta, Yew Soon Ong, and Liang Feng. Multifactorial evolution: toward evolutionary multitasking. *IEEE Trans. Evol. Comput.*, 20(3):343–357, July 2016.
- [86] Abhishek Gupta, Yew Soon Ong, Liang Feng, and Kay Chen Tan. Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE Trans. on Cybern.*, 47(7):1652–1665, May 2017.
- [87] Songqing Shan and G. Gary Wang. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization*, 41(2):219–241, August 2010.
- [88] Y. X. Zhang and C. H. Yang. Recent developments in finite element analysis for laminated composite plates. *Composite Structures*, 88:147–157, March 2009.
- [89] Shiliang Sun. A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7-8):2031–2038, February 2013.
- [90] Sham M Kakade and Dean P Foster. Multi-view regression via canonical correlation analysis. In *Proceedings of the 20th Annual Conference on Learning Theory*, pages 82–96. Springer-Verlag, San Diego, CA, June 13–15 2007.
- [91] Yong Zhang, Meng Joo Er, Rui Zhao, and Mahardhika Pratama. Multiview convolutional neural networks for multidocument extractive summarization. *IEEE transactions on cybernetics*, 47(10):3230–3242, 2017.
- [92] Kamalika Chaudhuri, Sham M Kakade, Karen Livescu, and Karthik Sridharan. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 129–136. ACM, Montreal, Canada, June 14–18 2009.

- [93] Philip E Gill, Walter Murray, and Margaret H Wright. *Practical optimization*. Academic press, 1981.
- [94] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.
- [95] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.
- [96] Joshua Knowles and David Corne. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 1. IEEE, 1999.
- [97] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103, 2001.
- [98] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.
- [99] Johannes Bader and Eckart Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary computation*, 19(1):45–76, 2011.
- [100] Dan Simon. *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.
- [101] Lawrence Davis. *Handbook of genetic algorithms*. 1991.
- [102] Hans-Georg Beyer and Kalyanmoy Deb. Self-adaptive genetic algorithms with simulated binary crossover. Technical report, Universität Dortmund, 2001.

- [103] Kalyanmoy Deb and Mayank Goyal. A combined genetic adaptive search (geneas) for engineering design. *Computer Science and Informatics*, 26:30–45, 1996.
- [104] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies—a comprehensive introduction. *Natural computing*, 1(1):3–52, 2002.
- [105] Matthias Ehrgott. A discussion of scalarization techniques for multiple objective integer programming. *Annals of Operations Research*, 147(1):343–360, 2006.
- [106] Hisao Ishibuchi, Yuji Sakane, Noritaka Tsukamoto, and Yusuke Nojima. Adaptation of scalarizing functions in moea/d: An adaptive scalarizing function-based multiobjective evolutionary algorithm. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 438–452. Springer, 2009.
- [107] Jiang Siwei, Cai Zhihua, Zhang Jie, and Ong Yew Soon. Multiobjective optimization by decomposition with pareto-adaptive weight vectors. In *Natural Computation (ICNC), 2011 Seventh International Conference on*, volume 3, pages 1260–1264. IEEE, 2011.
- [108] Qingfu Zhang, Wudong Liu, Edward Tsang, and Botond Virginas. Expensive multiobjective optimization by moea/d with gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3):456–474, 2010.
- [109] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.
- [110] Carl Edward Rasmussen and Chris Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

- [111] Abhishek Gupta, Yew-Soon Ong, and Liang Feng. Insights on transfer optimization: Because experience is the best teacher. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):51–64, 2018.
- [112] Barış Koçer and Ahmet Arslan. Genetic transfer learning. *Expert Systems with Applications*, 37(10):6997–7002, 2010.
- [113] Amiram Moshaiov and Amir Tal. Family bootstrapping: A genetic transfer learning approach for onsetting the evolution for a set of related robotic tasks. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 2801–2808. IEEE, 2014.
- [114] Muhammad Iqbal, Bing Xue, Harith Al-Sahaf, and Mengjie Zhang. Cross-domain reuse of extracted knowledge in genetic programming for image classification. *IEEE Transactions on Evolutionary Computation*, 2017.
- [115] Liang Feng, Yew-Soon Ong, Ah-Hwee Tan, and Ivor W Tsang. Memes as building blocks: a case study on evolutionary optimization+ transfer learning for routing problems. *Memetic Computing*, 7(3):159–180, 2015.
- [116] Rémi Bardenet, Mátyás Brendel, Balázs Kégl, and Michele Sebag. Collaborative hyperparameter tuning. In *International Conference on Machine Learning*, pages 199–207, 2013.
- [117] Matthias Poloczek, Jialei Wang, and Peter Frazier. Multi-information source optimization. In *Advances in Neural Information Processing Systems*, pages 4291–4301, 2017.
- [118] Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, and Frank Hutter. Bayesian optimization with robust bayesian neural networks. In *Advances in Neural Information Processing Systems*, pages 4134–4142, 2016.
- [119] Matthias Feurer, Benjamin Letham, and Eytan Bakshy. Scalable meta-learning for bayesian optimization. *arXiv preprint arXiv:1802.02219*, 2018.

- [120] Martin Wistuba, Nicolas Schilling, and Lars Schmidt-Thieme. Scalable gaussian process-based transfer surrogates for hyperparameter optimization. *Machine Learning*, 107(1):43–78, 2018.
- [121] Valerio Perrone, Rodolphe Jenatton, Matthias Seeger, and Cedric Archambeau. Multiple adaptive bayesian linear regression for scalable bayesian optimization with warm start. *arXiv preprint arXiv:1712.02902*, 2017.
- [122] Yolanda Mack. *CFD-Based Surrogate Modeling of Liquid Rocket Engine Components via Design Space Refinement and Sensitivity Assessment*. PhD thesis, Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, 2007.
- [123] Selen Uguroglu and Jaime Carbonell. Feature selection for transfer learning. In *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part III*, pages 430–442. Springer-Verlag, Athens, Greece, September 5-9 2011.
- [124] D. Kale and Y. Liu. Accelerating active learning with transfer learning. In *2013 IEEE 13th International Conference on Data Mining*, pages 1085–1090, Dallas, TX, December 7–10 2013.
- [125] NASA Glen Research Center (2014, Aug 19). Enginesim v1.8a [Online]. Available: <http://www.grc.nasa.gov/WWW/K-12/airplane/ngnsim.html>.
- [126] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach. Third Edition*. Pearson, London, England, 2009.
- [127] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of Research on Machine Learning Applications*, pages 242–264. IGI Global, Hershey, PA, 2009.

- [128] T. F. Gonzalez. *Handbook of Approximation Algorithms and Metaheuristics*. Chapman & Hall/CRC, Boca Raton, FL, 2007.
- [129] P. Larrañaga and J. A. Lozano. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Boston, MA, 2002.
- [130] Michael P Perrone and Leon N Cooper. When networks disagree: Ensemble methods for hybrid neural networks. Technical report, BROWN UNIV PROVIDENCE RI INST FOR BRAIN AND NEURAL SYSTEMS, 1992.
- [131] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.
- [132] Roberto Calandra, Jan Peters, Carl Edward Rasmussen, and Marc Peter Deisenroth. Manifold gaussian processes for regression. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 3338–3345. IEEE, 2016.
- [133] Yoh-Han Pao, Gwang-Hoon Park, and Dejan J Sobajic. Learning and generalization characteristics of the random vector functional-link net. *Neuro-computing*, 6(2):163–180, 1994.
- [134] Mitchell A Potter and Kenneth A De Jong. A cooperative coevolutionary approach to function optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 249–257. Springer, 1994.
- [135] Xiaodong Li, Andries Engelbrecht, and Michael G Eptropakis. Benchmark functions for cec’2013 special session and competition on niching methods for multimodal function optimization. *RMIT University, Evolutionary Computation and Machine Learning Group, Australia, Tech. Rep*, 2013.

- [136] Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (gpml) toolbox. *Journal of Machine Learning Research*, 11(Nov):3011–3015, 2010.
- [137] Hisao Ishibuchi and Tadahiko Murata. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 28(3):392–403, 1998.
- [138] Carlos A Coello Coello, David A Van Veldhuizen, and Gary B Lamont. *Evolutionary algorithms for solving multi-objective problems*, volume 242. Springer, 2002.
- [139] Lam T Bui, Jing Liu, Axel Bender, Michael Barlow, Slawomir Wesolkowski, and Hussein A Abbass. Dmea: a direction-based multiobjective evolutionary algorithm. *Memetic Computing*, 3(4):271–285, 2011.
- [140] Long Nguyen, Lam T Bui, and Hussein A Abbass. Dmea-ii: the direction-based multi-objective evolutionary algorithm-ii. *Soft Computing*, 18(11):2119–2134, 2014.
- [141] Joshua Knowles. Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.
- [142] Tinkle Chugh, Yaochu Jin, Kaisa Miettinen, Jussi Hakanen, and Karthik Sindhya. A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2016.
- [143] Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, and Markus Vincze. Multiobjective optimization on a limited budget of evaluations using model-assisted $\mathcal{S}$ -metric selection. In *International Conference on Parallel Problem Solving from Nature*, pages 784–794. Springer, 2008.

- [144] Taimoor Akhtar and Christine A Shoemaker. Multi objective optimization of computationally expensive multi-modal functions with rbf surrogates and multi-rule selection. *Journal of Global Optimization*, 64(1):17–32, 2016.
- [145] Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011.
- [146] Yaochu Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft computing*, 9(1):3–12, 2005.
- [147] Thomas G Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer, 2000.
- [148] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. *Scalable test problems for evolutionary multiobjective optimization*. Springer, 2005.
- [149] Anne Auger, Johannes Bader, Dima Brockhoff, and Eckart Zitzler. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theoretical Computer Science*, 425:75–103, 2012.
- [150] Siwei Jiang, Yew-Soon Ong, Jie Zhang, and Liang Feng. Consistencies and contradictions of performance metrics in multiobjective optimization. *IEEE Transactions on Cybernetics*, 44(12):2391–2404, 2014.
- [151] Yew Soon Ong, Zongzhao Zhou, and Dudy Lim. Curse and blessing of uncertainty in evolutionary algorithm using approximation. In *2006 IEEE International Conference on Evolutionary Computation*, pages 2928–2935. IEEE, 2006.
- [152] William Andrew Walbran. *Experimental validation of local and global force simulations for rigid tool liquid composite moulding processes*. PhD thesis, ResearchSpace at Auckland, 2011.

- [153] Abhishek Gupta. *Numerical modelling and optimization of non-isothermal, rigid tool liquid composite moulding processes*. PhD thesis, ResearchSpace at Auckland, 2013.
- [154] Sherry Hsu, Matthias Ehrgott, and Piaras Kelly. Optimisation of mould filling parameters of the compression resin transfer moulding process. In *45th annual conference of the operations research society of New Zealand (ORSNZ)*, 2010.
- [155] Pengfei Wei, Ramon Sagarna, Yiping Ke, Yew-Soon Ong, and Chi-Keong Goh. Source-target similarity modelings for multi-source transfer gaussian process regression. In *International Conference on Machine Learning*, pages 3722–3731, 2017.
- [156] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems*, pages 2962–2970, 2015.
- [157] Hector Mendoza, Aaron Klein, Matthias Feurer, Jost Tobias Springenberg, and Frank Hutter. Towards automatically-tuned neural networks. In *Workshop on Automatic Machine Learning*, pages 58–65, 2016.
- [158] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, pages 1015–1022, 07 2010.
- [159] Alistair Shilton, Sunil Gupta, Santu Rana, and Svetha Venkatesh. Regret bounds for transfer learning in bayesian optimisation. In *Artificial Intelligence and Statistics*, pages 307–315, 2017.
- [160] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

- [161] Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, Nando De Freitas, et al. Bayesian optimization in high dimensions via random embeddings. In *IJCAI*, pages 1778–1784, 2013.
- [162] Alexander K Seewald. Digits-a dataset for handwritten digit recognition. *Austrian Research Institut for Artificial Intelligence Technical Report, Vienna (Austria)*, 2005.