A SEQUENTIAL DESIGN FOR APPROXIMATING THE PARETO FRONT USING THE EXPECTED PARETO IMPROVEMENT FUNCTION

DISSERTATION

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Dianne Carrol Bautista, M.S.

* * * * *

The Ohio State University

2009

Dissertation Committee:

Prof. Thomas J. Santner, Adviser
Prof. Peter F. Craigmile
Prof. William I. Notz

Approved by

________________________________________
Adviser
Graduate Program in Statistics
ABSTRACT

This thesis proposes a methodology for the simultaneous optimization of multiple goal functions via computer experiments.

Some technical challenges associated with the black box multiobjective problem (MOP) can be enumerated as follows: the presence of conflicting goals imply that more optimization effort is invested to find a good range of solutions that are simultaneously optimal against these competing criteria; the highly non-linear mapping between the inputs in the design space and the goal functions in objective space may complicate the solution process; and in common with global optimization, the run-time costs of simulation severely limit the number of evaluations that can be made.

In view of these, the aim is to compute efficiently and identify a set of good solutions that collectively provide an even coverage of the Pareto front, the set of optimal solutions for a given MOP. The members of the Pareto front comprise the set of compromise solutions from which a decision maker chooses a final design that resonates best with his or her preferences.

To reduce the computational overhead, we adopt a surrogate-guided optimization approach. The idea is to build fast approximations that can replace the long-running simulator during optimization while also being reasonably accurate at predicting the latter in the unevaluated feasible design points. This brings about a tremendous gain
in efficiency at the price of extra uncertainty due to the speculative nature of the search for optimal points. Consequently, two competing issues need to be balanced: the global exploratory search for improving surrogate accuracy and local exploitative search for converging rapidly to the optimal points. In a fully sequential optimization design, a key ingredient for achieving this balance is the criterion for selecting the next design point for costly-function evaluation.

Among the various surrogates considered so far, none has demonstrated a mechanism for balancing the tension between local exploitation and global exploration as automatically and as naturally as Gaussian processes have done, as illustrated by the Efficient Global Optimization algorithm for single-objective optimization. We therefore attempt to extend the $EI$ framework to solve the black box MOP.

The existing literature on Gaussian process-guided sequential designs for the MOP is scarce on multivariate emulators that effectively incorporate dependencies in the objective function vector. It is also scant on improvement criteria suitably defined for the MOP, that can decisively localize solutions in the vicinity of the Pareto front.

Our proposed $EmaX$ algorithm addresses this lack. We implement a multivariate Gaussian process emulator that guides the sequential search for optimal solutions by means of the expected Pareto improvement function. We considered two models of the covariance structure: a non-separable independence model and a separable dependence model which exemplifies a way of accounting for the covariances within the objective vector.

At each stage, the “current best” solutions are first identified. These solutions dominate other feasible solutions in the current experimental design, but do not dominate each other. Then a constrained non-linear program is solved to locate the
design point that presents the greatest potential Pareto improvement to the current non-dominated front.

Based on the maximin fitness function, the Pareto improvement is essentially a free upgrade offered by a prospective design point to at least one of the currently identified best designs, in at least one of the objectives. It bears an analogous interpretation to its usage in economics as a change or action in economic management which upgrades the condition of one or more members without worsening the circumstances of the other members. The idea is to progressively add increments of improvements until ideally, a state of Pareto equilibrium is reached where no more free upgrades are possible. At that point, trading-off in the performance criteria happens when moving from one Pareto solution to another.

We demonstrated the viability of the EmaX algorithm on five MOP’s with relatively low dimensionality and offering various degrees of difficulty in terms of the shape of the Pareto front. Three sequential algorithms were compared: the IGP-PI, IGP-EmaX, and CoH-EmaX. The IGP procedures use a surrogate for the outputs based on the independence model while CoH-EmaX is based on a dependence model. The EmaX criterion was contrasted with a contending improvement criterion called the probability of improvement or PI.

On the five MOP’s tested, the EmaX criterion generally performed better than the PI in terms of efficiently and evenly covering the Pareto front. The solutions obtained by the EmaX algorithm were generally more spread out along the Pareto front than the solutions obtained using the PI-directed sequential design which were clustered or biased in some regions of the Pareto front, even as the latter algorithm delivered bigger solution sets.
As regards the gain of modeling dependencies, the *EmaX* algorithm based on a Gaussian process with a separable dependence covariance structure fared better than the non-separable independence model in terms of closeness to the best approximated Pareto front, as measured by the binary epsilon factor and in terms of the hypervolume indicator, a measure of the size of the dominated region.

This endeavor has definitely inspired areas for future investigation within the scope of Gaussian process-assisted sequential optimization designs. The implementation of the *EmaX* algorithm leaves an ample room for improving its algorithmic efficiency and precision. Enhancements in the area of multivariate emulation, particularly in specifying covariance structures that offer alternative ways of accounting for the conflicting structure in the objective vector can be explored. Recently, a competing infill sampling criterion to the expected improvement, called the conditional minimizer entropy or the *CME* criterion has been proposed that merits investigation. Finally, Handl and Knowles (2007) advance novel applications of multiobjective optimization methods or Pareto set approaches to solve “conventional” problems that are worth testing.
Dedicated to Marco and Francesco Maria.
ACKNOWLEDGMENTS

My profound gratitude to Marco for the Love, logistics, and inspiration...I know I will never thank you enough; my family for their constant support and prayers.

I thank and commend my advisor, Prof. Thomas Santner for the gift of his person and exemplifying a kind of mentoring that has made a positive impact on me as a parent, a teacher, and statistics consultant. You have opened doors and given me opportunities to outdo myself. I will not forget the time in SAMSI, specially. Thank you, Prof. Santner!

I also thank my thesis committee members, Prof. Bill Notz and Prof. Peter Craigmile for their insights and feedback which have improved my thesis significantly; to Prof. Angela Dean for her encouragement, and Prof. Nagaraja for sharing time to help me with a derivation. My heartfelt thanks to all professors who have influenced my thinking, doing, and approach to statistics – Prof. Cressie, Prof. Critchlow, Prof. Goel, Prof. Verducci, Prof. Wolfe, Prof. Fligner, Prof. Lee, Prof. Hans, Prof. Lemeshow, Prof. Peruggia, and Prof. Berliner. Thank you Prof. Stasny for keeping us in top form.

I would not have made this without the help of the department’s staff. Thank you Patty Kathy, Lisa, Eric and Brian, and Paul. Thank you all for your assistance...and to LaDonna.
To my friends in the department whose company kept me grounded through the years – Kimberly, Rajib, Namhee, Gang, and Soma; my friends in the Newman Center – Anne-Marie, Valentine, Dinna, Jenny, Magdalena, Socorro, Joan, Fr. Vinny, Fr. Chuck, Fr. Larry and Fr. Dave. You are all special! You also know that you have a place to stay once in Singapore.

Above all, I thank GOD, Jesus, and the Holy Spirit, for everything...Their goodness on the persons, the things, both tangible and intangible, all that was given to me generously and abundantly. This thesis is my gift back to YOU.
VITA

February 4, 1972 .......................... Born - Philippines

1992 ......................................... B.S. Statistics, University of the Philippines

1996 ......................................... M.S. Statistics, University of the Philippines

1992-1998 ................................. Assistant Professor, School of Statistics, University of the Philippines

1999-2004 ................................. Assistant Professor, Mathematics and Natural Sciences Division, University of the Philippines in the Visayas, Cebu College

2004-2005 ................................. Graduate Research Associate, College of Human Ecology, The Ohio State University

2005-2006 ................................. Graduate Teaching Associate, Department of Statistics, The Ohio State University

2006-2007 ................................. Graduate Research Associate, Department of Statistics, The Ohio State University

2008-present .............................. Senior Biostatistician, Singapore Clinical Research Institute
PUBLICATIONS


FIELDS OF STUDY

Major Field: Statistics
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td>vi</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>vii</td>
</tr>
<tr>
<td>Vita</td>
<td>ix</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xiv</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xvi</td>
</tr>
<tr>
<td>Chapters:</td>
<td></td>
</tr>
<tr>
<td>1. Models for Computer Experiments with Multivariate Outputs</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Review of Computer Experiments</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Modeling Computer Experiments with Multiple Outputs</td>
<td>9</td>
</tr>
<tr>
<td>1.3.1 Introduction</td>
<td>9</td>
</tr>
<tr>
<td>1.3.2 Overview of Gaussian process emulation</td>
<td>10</td>
</tr>
<tr>
<td>1.3.3 Models for the Cross-Covariance</td>
<td>14</td>
</tr>
<tr>
<td>1.3.4 Predicting Output from Computer Experiments</td>
<td>28</td>
</tr>
<tr>
<td>1.4 Restricted Maximum Likelihood Estimation of $\psi$</td>
<td>37</td>
</tr>
<tr>
<td>1.5 The Multiobjective Problem (MOP)</td>
<td>42</td>
</tr>
<tr>
<td>1.5.1 Introduction</td>
<td>42</td>
</tr>
<tr>
<td>1.5.2 Pareto Optimality</td>
<td>43</td>
</tr>
<tr>
<td>1.5.3 Review of GP-guided Solution Approaches for the Black box MOP’s</td>
<td>51</td>
</tr>
</tbody>
</table>
2. The Expected Pareto Improvement and EmaX Algorithm for Approximating the Pareto Front ........................................... 74

2.1 The Pareto or Maximin Improvement Function .................. 76
2.2 The EmaX Algorithm .................................................. 82
2.3 The Expected Pareto Improvement when $m = 2$ ................ 85

3. Numerical Test Results ................................................... 99

3.1 Quality Assessment of the Pareto Front Approximation ........ 100
3.1.1 The Hypervolume Indicator ....................................... 105
3.1.2 The Binary Epsilon Indicator, $I_e$ .............................. 107
3.2 MOP Test Functions .................................................... 108
3.2.1 The $wsnl$ test functions ......................................... 110
3.2.2 The $knowles$ test functions ..................................... 112
3.2.3 The $obj4d2$ test function ....................................... 113
3.2.4 The $hedarg2$ test functions ..................................... 115
3.2.5 The $dltz1a$ test function ....................................... 118
3.3 Numerical Test Results .................................................. 119
3.3.1 Approximating the Pareto Front of the $wsnl$ MOP .......... 123
3.3.2 Approximating the Pareto Front of the $knowles$ MOP ....... 127
3.3.3 Approximating the Pareto Front of the $obj4d2$ MOP ....... 130
3.3.4 Approximating the Pareto Front of the $hedarg2$ MOP ...... 135
3.3.5 Approximating the Pareto Front of the $dltz1a$ MOP ....... 138
3.3.6 Summary of Results ................................................ 141

4. Summary and Conclusions ............................................... 143

4.1 Summary of Performance Assessments .............................. 146
4.2 Directions for Future Work ........................................... 149
4.2.1 Refinements to the EmaX Algorithm ............................ 150
4.2.2 Enhancing Multivariate Emulation ............................... 153
4.2.3 Alternative Sequential Sampling Criteria ..................... 154
4.2.4 Extension of the EmaX Algorithm to more general spaces .. 154
4.2.5 Novel Applications of Pareto set approaches ................. 155
4.2.6 Systematizing the evaluation and comparison of the perfor-

mance of the algorithms ................................................ 156

Appendices ................................................................. 157

Appendices:
LIST OF TABLES

Table                  Page

3.1 Dominance Relations on Objective Vectors and Approximation Sets . 103

3.2 The MOP-specific grid resolution used to determine the minimum and maximum of each \( Y_k(\cdot) \) and the “best” approximation to the true Pareto front, \( P_{YG}^G \) containing \( n(P_{YG}^G) \) approximate solutions. For the \( dltz1a \) test function, the best approximation was based on the analytical solution. . . . . . . 110

3.3 The number of solutions in \( P_Y^o, o \in \Omega \). . . . . . . . . . . . . . . . . . . 124

3.4 A comparison of the size of the dominated region. Note \( I_H(P_{YG}^G) = 0.9102 \) 125

3.5 The binary epsilon indicator. Column 2 compares \( P_Y^o, o \in \Omega \) to a common reference, \( P_Y^G \). Columns 3 - 5 shows the \( I_\epsilon(P_Y^o, P_Y^{o'}) \), \( o \neq o' \in \Omega \). . . . . . 126

3.6 The binary epsilon indicator. Column 2 compares \( P_Y^o, o \in \Omega \) to a common reference, \( P_Y^G \). Columns 3 - 5 shows the \( I_\epsilon(P_Y^o, P_Y^{o'}) \), \( o \neq o' \in \Omega \). . . . . . 127

3.7 The number of solutions in \( P_Y^o, o \in \Omega \). . . . . . . . . . . . . . . . . . . 128

3.8 A comparison of the size of the dominated region. Note \( I_H(P_{YG}^G) = 0.8591 \) 128

3.9 The binary epsilon indicator. Column 2 compares \( P_Y^o, o \in \Omega \) to a common reference, \( P_Y^G \). Columns 3 - 5 shows the \( I_\epsilon(P_Y^o, P_Y^{o'}) \), \( o \neq o' \in \Omega \). . . . . . 129

3.10 The binary epsilon indicator. Column 2 compares \( P_Y^o, o \in \Omega \) to a common reference, \( P_Y^G \). Columns 3 - 5 shows the \( I_\epsilon(P_Y^o, P_Y^{o'}) \), \( o \neq o' \in \Omega \). . . . . . 130

3.11 The number of solutions in \( P_Y^o, o \in \Omega \). . . . . . . . . . . . . . . . . . . 132

3.12 A comparison of the size of the dominated region. Note \( I_H(P_{YG}^G) = 0.4444 \). 133
3.13 The binary epsilon indicator. Column 2 compares $P_o^Y$, $o \in \Omega$ to a common reference, $P_0^G$. Columns 3 - 5 shows the $I_e(P_o^Y, P_0'^Y)$, $o \neq o' \in \Omega$.

3.14 The binary epsilon indicator. Column 2 compares $P_o^Y$, $o \in \Omega$ to a common reference, $P_0^G$. Columns 3 - 5 shows the $I_e(P_o^Y, P_0'^Y)$, $o \neq o' \in \Omega$.

3.15 The number of solutions in $P_o^Y$, $o \in \Omega$.

3.16 A comparison of the size of the dominated region. Note $I_H(P_j^G) = 0.1942$.

3.17 The binary epsilon indicator. Column 2 compares $P_o^Y$, $o \in \Omega$ to a common reference, $P_0^G$. Columns 3 - 5 shows the $I_e(P_o^Y, P_0'^Y)$, $o \neq o' \in \Omega$.

3.18 The binary epsilon indicator. Column 2 compares $P_o^Y$, $o \in \Omega$ to a common reference, $P_0^G$. Columns 3 - 5 shows the $I_e(P_o^Y, P_0'^Y)$, $o \neq o' \in \Omega$.

3.19 The number of solutions in $P_o^Y$, $o \in \Omega$.

3.20 A comparison of the size of the dominated region. Note $I_H(P_j^G) = 0.9995$.

3.21 The binary epsilon indicator. Column 2 compares $P_o^Y$, $o \in \Omega$ to a common reference, $P_0^G$. Columns 3 - 5 shows the $I_e(P_o^Y, P_0'^Y)$, $o \neq o' \in \Omega$.

3.22 The binary epsilon indicator. Column 2 compares $P_o^Y$, $o \in \Omega$ to a common reference, $P_0^G$. Columns 3 - 5 shows the $I_e(P_o^Y, P_0'^Y)$, $o \neq o' \in \Omega$.

3.23 Overall rank of the algorithms with respect to $I_H(P_j^G)$, $I_e(P_o^Y, P_0'^Y)$ for $n_0 = 5d$, $N_{max} = 20d$. A rank of 1 is the most preferable and 3, least preferable.

3.24 Overall rank of the algorithms with respect to $I_H(P_j^G)$, $I_e(P_o^Y, P_0'^Y)$ for $n_0 = 10d$, $N_{max} = 20d$. A rank of 1 is the most preferable and 3, least preferable.
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Pareto dominance illustrated on a biobjective problem with two variables</td>
</tr>
<tr>
<td>1.2</td>
<td>A geometric interpretation of the Henkenjohann-Kunert improvement function for $m = 2$ and $g = 1$: the shortest Euclidean distance of a potential design point $x_i \notin D_n$ to the curve $DI_{max} = 0.51$</td>
</tr>
<tr>
<td>1.3</td>
<td>The regions of integration for determining the probability of improvement when $m = 2$.</td>
</tr>
<tr>
<td>1.4</td>
<td>The region of improvement partitioned according to the number of non-inferior solutions in the current non-dominated front $\nu$ that can be dominated.</td>
</tr>
<tr>
<td>2.1</td>
<td>Partitioning of $\mathbb{R}_2$ by two non-dominated points.</td>
</tr>
<tr>
<td>3.1</td>
<td>Outcomes of three hypothetical algorithms for a bi-objective problem. The approximation sets are denoted as $A, B$ and $C$; the (discretized) Pareto Front $P$ consists of three objective vectors. Between $A, B$ and $C$, the following dominance relations hold: $A \gg C, A \succ C, B \succ C, A \succeq A, A \succeq B, A \succeq C, B \succeq C, B \succeq C, C \succeq C, A \succ B, A \succ C$, and $B \succ C$.</td>
</tr>
<tr>
<td>3.2</td>
<td>In the left and center panels, $A \succ B$, but the difference is more prominent in the center. In the right panel, $A$ and $B$ are indifferent by definition, but visual inspection suggests that $A$ may be better.</td>
</tr>
<tr>
<td>3.3</td>
<td>The shaded area indicates the size of the dominated zone of the approximation set $A = {y^1, y^2, y^3}$, measured by the hypervolume indicator, $I_H$. The reference point, $r$, bounds the calculation of the hypervolume.</td>
</tr>
<tr>
<td>3.4</td>
<td>The shaded volume indicates the size of the dominated zone of an approximation set with seven points in three dimensions.</td>
</tr>
</tbody>
</table>
3.5 Grid approximated-Pareto optimal set and Pareto front of the *wsnl* MOP
are indicated in bold-faced dots. ........................................... 112

3.6 The Pareto set and the Pareto front of the *knowles* MOP are both
disconnected. ................................................................. 113

3.7 The design space of the test function *obj4d2* with an irregularly shaped
Pareto set. ........................................................................ 114

3.8 Two-dimensional projections of the objective space and Pareto front
of test function *obj4d2* ...................................................... 116

3.9 Two-dimensional projection of design space and grid-approximated
Pareto optimal set for *hedary2* test functions. ......................... 117

3.10 Two-dimensional projections of the objective space and Pareto front
of the *hedary2* test function. ................................................ 118

3.11 Design Space and Pareto optimal set for the *dltz1a* test functions.
Design variable $x_1$ is inactive while $x_j$, $j = 2, 3, 4, 5, 6$ are active. Hence
the projection of the Pareto set on the plane $(x_1, x_j)$, $j = 2, 3, 4, 5, 6$
is a line, such as shown in the graph of $(x_1, x_2)$. The two-dimensional
projections of a pair of active design variables $(x_i, x_j)$, $i \neq j$, is a point,
like the graph of $x_5$ and $x_6$. ............................................. 120

3.12 Test function *dltz1a*: A linear Pareto front in objective space. .. 121

3.13 A comparison of *IGP-PI*, *IGP-EmaX*, *CoH-EmaX* algorithms for approxi-
mating the Pareto Front of the function *wsnl*. .......................... 123

3.14 A comparison of *IGP-PI*, *IGP-EmaX*, *CoH-EmaX* algorithms for approxi-
mating the Pareto Front of the function *knowles*. ..................... 127

3.15 Comparison in $y_1$ and $y_2$ of *IGP-PI*, *IGP-EmaX*, *CoH-EmaX* algorithms
for approximating the Pareto Front of the *obj4d2* test functions. .... 131

3.16 Comparison in $y_1$ and $y_4$ of *IGP-PI*, *IGP-EmaX*, *CoH-EmaX* algorithms
for approximating the Pareto Front of the *obj4d2* test functions. .... 132
3.17 Comparison in $y_2$ and $y_4$ of IGP-PI, IGP-EmaX, CoH-EmaX algorithms for approximating the Pareto Front of the obj4d2 test functions. . . . . . 133

3.18 Comparison in $y_1$ and $y_2$ of IGP-PI, IGP-EmaX, CoH-EmaX algorithms for approximating the Pareto Front of the hedarg2 test functions . . . . . 136

3.19 Comparison in $y_1$ and $y_2$ of IGP-PI, IGP-EmaX, CoH-EmaX algorithms for approximating the Pareto Front of the function dltz1a . . . . . . . . . 139
CHAPTER 1

MODELS FOR COMPUTER EXPERIMENTS WITH MULTIVARIATE OUTPUTS

1.1 Introduction

Due to the growth in computing power and speed, computer modeling and simulation of physical processes have become relevant in the study of many scientific and engineered systems. These activities continue to expand in domains that have used it for some time and new application areas are emerging. Scientists and engineers have relied on computer simulation to assess the risk of high consequence systems that cannot be tested physically such as the catastrophic failure of a nuclear power plant, a fire spreading through a high rise building, or a nuclear weapon caught in a transport accident.

Natural systems have also been the object of interest. Climate and weather models inform today’s debate on climate change; computer simulations of environmental impact have been made, for example, in the analysis of surface water quality, air toxic emissions, and hazardous waste management, particularly the underground storage of nuclear waste.
For engineered systems, safety and reliability concerns have motivated the development of computer experiments (CE) on existing and proposed systems operating at design, off-design, and failure-mode conditions in accident scenarios. Numerical simulations from computational methods in structural mechanics, heat transfer, fluid mechanics, shock physics are used as virtual prototypes to help produce acceptable designs for systems. Virtual tests are also being carried out to develop new hardware or entire systems to eliminate the causes of failure or at worst, moderate its costs to the environment, the public, an individual, or a company.

1.2 Review of Computer Experiments

The starting point of a computer model is a mathematical model relating the systems’ outputs to a set of inputs. In engineering, such mathematical models are expressed as systems of partial differential equations (PDE) and require computer programs to numerically solve them. Typical programs can contain several thousands of lines of code and can take anywhere from hours to days to evaluate even in the world’s fastest computers. The mathematical model, or the computer program that implements it, is often called a simulator.

The inputs and outputs of a simulator can have high dimensionality like hundreds of input and output quantities. Computer codes of this size and complexity can therefore be thought of as black boxes in that not much is understood about the code’s mapping from inputs to outputs. The mapping is also considered to be deterministic, meaning that after all inputs have been assigned values, the simulator produces a single value for each output it generates; a re-run at the same input generates duplicate outputs.
This is in stark contrast with physical experiments where a response corresponding to a set of treatments (the input variables) is measured along with unknown nuisance factors which can cause variation in the outcome of replicated runs. In particular, the measurement of treatment effects is blurred by *noise*—random or non-systematic effects and *bias*—the systematic effects of nuisance or uncontrolled factors. Noise can arise from discrepancies due to unavoidable inaccuracies in the measuring instruments or non-homogeneity in the experimental units. Bias may be caused by an unrecognized flaw in the design or philosophy of the experiment that offsets the measurements consistently.

The key differentiating concept here is *repeatability*. The response is *deterministic* in computer experiments but *stochastic* in physical experiments.

This contrast engenders a distinction in the *meaning* and *management* of uncertainty as well as in the sorts of errors produced. To manage uncertainties in comparing treatments, classical experimental design uses *randomization* to control for the bias due to *unrecognized* nuisance factors and *replication* and *blocking* to mitigate or model the effects of *recognized* factors. In computer experiments, noise and bias due to uncontrolled factors does not occur so that randomization, replication, and blocking of experimental units are not required. Space-filling experimental designs that allow greatest empirical investigation of the input space are more useful in computer experiments than classical fractional factorial and blocked designs.

Any model of reality is bound to have deficiencies in the form of errors and uncertainties which make predictions imperfect. In computer experiments, the *mathematical model* is the main source of uncertainty which is why the uncertainty is described as *epistemic*, the kind that derives from the modeler’s partial knowledge or incomplete
understanding of the physical system and its surrounding environment. For example, the modeler is not sure about the correct formulation of the mathematical model, or details about the formulation like the ‘right’ coefficients to plug in, or the correct boundary conditions to solve the relevant system of PDEs. On account of these reasons and of the simplifying assumptions made, the computer model’s representation of reality will inevitably be biased. The *simulation error* or the difference between reality and the computer model’s prediction is the *model bias*. Non-random errors also exist in the code’s *implementation* such as conversion from continuum partial differential equations to discrete numerical modeling, e.g., errors expected from the mesh resolution in a finite element solve, the computer’s finite precision arithmetic, and round-off and truncation errors. Since simulation and implementation errors are systematic, both can be reduced. Predictions can be sharpened by incorporating additional information into, or calibrating the computer model. Computations can be made more accurate by reconfiguring default settings or using more powerful computing machines.

In the last twenty years, there has been an upsurge in the engineering community’s ability to build finite element models to simulate the behavior of complex processes. Furthermore, the ability to rapidly adjust these simulation models to keep up with design changes has also increased.

The net result is that the use of *simulation-based optimization* to develop new engineered systems has increased. There are obstacles however, the very long running times and lack of gradient information in some areas have made it less convenient, despite the steady growth in computing power and speed. For instance, single evaluations of finite element analyses to predict a structure’s performance or computational
fluid dynamics models to visualize the flow over a body – both cases can take some
days up to a week to evaluate. Hence any optimization algorithm applied directly
on those codes will be slow. Even if it were possible to optimize the simulator di-
rectly, this painful strategy will exhaust the budget in no time, barring any important
follow-up studies. Missed deadlines are also likely to delay the whole design process.

The rationale behind the surrogate model approach is to construct fast mathemat-
ical approximations that can be used in lieu of the long-running simulator to facilitate
such ends as input space exploration, optimization, or reliability analysis. The un-
derlying assumption is that, once constructed, the surrogate will be many orders of
magnitude faster than the simulator while also being reasonably accurate at predict-
ing the latter in unsampled design points. Meeting these criteria, such a surrogate can
be relied upon to identify active inputs, visualize functional relationships, investigate
trade-offs, and possibly provide fresh insights in real time. The simulator can then be
called to verify the tentative conclusions drawn from interfacing with the surrogate.

For carrying out design optimization, the surrogate model is usually embedded
within a sequential design—a strategy for finding optimal points by performing ex-
periments successively in a direction of improvement. While the basic idea of the
surrogate approach is simple, there are numerous details that need to be sorted out
for its successful implementation. In the case of surrogate-guided optimization these
are: (1) the selection of a surrogate model, (2) the choice of design points on which
to train the surrogate, (3) identification of the best method to exploit the surrogate
to identify new and improved designs, and (4) the use of the surrogate to examine
the trade-offs among conflicting objectives.
The first is a question of design for CE’s. If no prior knowledge exists about the relationship between inputs and outputs, space-filling designs minimize both this uncertainty and the surrogate bias, by spreading points to gather information throughout the design space. Surrogate bias refers to the error of the surrogate in predicting the simulator. The prediction error at any point in input space usually varies proportionally to its distance from the closest sampled design point. Points near any sampled point will generally be predicted more accurately than those that are far away. This implies that uneven designs—those having poor coverage, can result in predictors that are inaccurate in sparsely sampled regions of input space.

As for the choice of surrogate, a special class of surrogates called emulators have the special advantage of enabling a probabilistic assessment of the uncertainty induced by the prediction process. An emulator is a statistical model of a deterministic function which provides, for any design point \( x \) in the input space, a predictive distribution for the simulator output at \( x \), say \( Y(x) \). The mean of this distribution for a given \( x \) is often regarded as the surrogate’s approximation to \( y(x) \) and the distribution about the mean is a measure of the uncertainty describing how close the surrogate will be to \( y(x) \). Furthermore, because the output is deterministic, it is reasonable to seek emulators that interpolate the data to acknowledge the fact that the simulator output is completely known at inputs where the simulator was run. For all other points, the distribution of \( Y(x) \) should indicate a mean value that represents a realistic interpolation or extrapolation of all previously acquired data and that the probability distribution around the prediction (mean) should plausibly describe the uncertainty about how \( Y(x) \) might interpolate or extrapolate.
The kriging predictor is a very popular emulator in the literature of CE’s (Santner, Williams, and Notz 2003). If the simulator is believed to behave smoothly over its domain, then the choice of a GP-based surrogate is well-grounded. Other surrogates have been used like neural networks, radial basis functions, or support vector machines (Wang and Shan 2007; Ponweiser, Wagner, and Vincze 2008) but none of these approximations explicitly account for the uncertainty in prediction in their implementation.

So far we have suggested implementing the surrogate model approach with a design having a space-filling property and a GP model that allows an assessment of the uncertainty in the prediction process, respectively. The last two details in the implementation – how to use the surrogate to identify new and improved designs and to examine the trade-offs among conflicting objectives, are addressed in this thesis.

The interest in this thesis is on simulation-based multiobjective optimization. In engineering design, multiple performance targets usually make it likely for some to be conflicting: minimize weight, cost, number of defects, limit a critical temperature, stress, vibration response, maximize reliability, throughput, reconfigurability, agility, or design robustness. Assuming that no prior preferences are given, conflicting aspirations generally imply that the solution will be non-unique. The task is then to identify the best trade-offs or compromises among the competing objectives.

This trend of simulation-based optimization has been energized by competition in a market constantly pressed by a need to decrease lead times and the cost of delivering products to prospective users. More efficient methods that reduce both the time taken to evaluate design concepts and the number of evaluations needed for optimization are continually being developed. State-of-the-art multiobjective optimization, as
evidenced by current commercial software (BOSS/Quattro, iSight, modeFRONTIER, OPTIMUS, and the freely available softwares DAKOTA from Sandia Laboratories, and PISA from the ETH Laboratory in Zurich) have mostly implemented population-based or nature-inspired algorithms such as evolutionary and genetic algorithms, particle swarm or ant colony optimization (Gobbi, Haque, Papalambros, and Mastinu 2005; Simpson, Toropov, Balabanov, and Viana 2008; Knowles, Thiele, and Zitzler 2006). Some previous studies have demonstrated the efficiency or utility of these algorithms in very high-dimensional problems. The main drawback in some is the cpu time required in the processing.

This thesis takes a different course by formulating a multiobjective extension of the Efficient Global Optimization algorithm (EGO) of Jones, Schonlau, and Welch (1998). The EGO algorithm is a Gaussian process-guided sequential design that finds the global optimum by locating at each stage, that design point showing the greatest potential of improving on the current best one. The task at hand poses two major challenges: one is developing a multivariate GP emulator that exploits the conflicting structure in the objectives to emulate the multi-output simulator, and two, defining a suitable improvement function for the multiobjective problem to efficiently guide the sequential approximation of the Pareto front—the trade-off curve or surface representing a set of optimal solutions. The current work expands the literature on surrogate-guided black box optimization approaches.

The thesis is organized as follows: Chapter 1 introduces the Gaussian process (GP) framework for analyzing computer experiments. In view of the multi-output setting, two GP models are considered – the independence model which ignores the cross-covariances among the objectives and a dependence model that attempts to
capture these effects. Chapter 1 also introduces the key concepts of dominance and Pareto optimality in multiobjective optimization. Chapter 2 contains the main contribution of this research namely, the expected maximin improvement or the EmaX update criterion embedded in a Gaussian process-guided sequential design. Its analytic expression is derived for the bi-objective case and a Monte Carlo approximation is proposed for the general case. Chapter 3 presents comparisons on the quality of the Pareto front approximation between the EmaX-updated sequential design and another GP-guided sequential design using a competing update criterion called the probability of improvement. In terms of the GP emulator, the efficiency gained by modeling cross-covariance effects is also investigated. The comparisons are made on five multiobjective test problems covering some variety with respect to the difficulties posed. Finally Chapter 4 concludes with a summary of results and directions for future work.

1.3 Modeling Computer Experiments with Multiple Outputs

1.3.1 Introduction

A major disincentive for performing optimization via computer experiments is the runtime cost of the simulator which limits the number of runs that can be made. Our approach seeks to replicate the behavior of the simulator to perform optimization much faster by using an approximate model.

In this section, we describe the theoretical and computational aspects of Gaussian process emulation.
1.3.2 Overview of Gaussian process emulation

We regard the simulator or computer code as a deterministic function $y(\cdot)$ that takes an input vector $x \in \mathcal{X} \subset \mathbb{R}^d, d \geq 1$ and returns an output

$$y(x) = (y_1(x), y_2(x), \ldots, y_m(x))^T, m > 1.$$ 

Let

$$D_n = \{x_1, \ldots, x_n\}, n \geq 1$$

be a given design at which the computer code is run to produce outputs

$$y(x_1), y(x_2), \ldots, y(x_n).$$

Consider the problem of predicting $y(x^*)$ for an unseen input vector $x^* \in \mathcal{X}$ on the basis of the data $\{(x_i, y(x_i)) : x_i \in D_n\}$.

Koehler and Owen (1996) discuss the Bayesian and frequentist approaches to computer experiments which are differentiated by the way randomness is introduced to measure how much a predicted value, say $\hat{y}(x_0)$, differs from the true value $y(x_0)$ for a new input site $x_0 \in \mathcal{X}$.

In the Bayesian formulation, $y(\cdot)$ is a realization of a random process. A prior distribution is placed on the space of all functions from $\mathcal{X} = [0, 1]^d$ to $\mathbb{R}_m$. This prior is then combined with information from training data $\{y(x_1), \ldots, y(x_n)\}$ to produce a posterior distribution used to predict $y(\cdot)$ at new input sites.

A Bayesian approach can be based on a spatial model adapted from Matheron’s kriging model which treats the bias or the systematic departure of the response surface from a linear model, as the realization of a stationary Gaussian process. The classical best linear unbiased predictor or BLUP is used to predict the simulator output at new input sites.
A fully Bayesian approach views the simulator $y(x), x \in X$ as an unknown function and consequently represents the uncertainty by a prior $m$-variate Gaussian process with mean vector $\mu(x)$ and positive definite covariance matrix $\Sigma_0$; the Gaussian process is usually chosen for convenience, as the posterior process given a vector of observed data on a set of input sites, is also a Gaussian process. Here, the posterior process is the object of interest since it is used for prediction. For this reason it is also referred to as the predictive process (Currin, Mitchell, Morris, and Ylvisaker 1991). The actual form of the predictor $\hat{y}(x_o)$ depends on the specification of a loss function which quantifies the loss incurred when $\hat{y}(x_o)$ is used to predict $y(x_o)$.

On the other hand, the frequentist approach introduces randomness via sampling techniques, taking values of $x_1, x_2, \ldots, x_n$ that are partially determined by pseudo-random number generators. The randomness in the $x_i$ is then propagated through to randomness in the $y(x_i)$. Its approach to prediction and computer experiments is based on numerical integration.

In this thesis, we will adopt a Bayesian perspective. We will combine prior information describing the mapping from $x \in X \subset \mathbb{R}^d$ to $y(x) \in \mathbb{R}^m$ with information from the data to predict the simulator at unevaluated input sites. A definition of an $m$-variate Gaussian process follows.

**Definition 1.3.1** Suppose that $X$ is a fixed subset of $\mathbb{R}^d, d \geq 1$ having positive $d$-dimensional volume. We say that $Y(x) = (Y_1(x), Y_2(x), \ldots, Y_m(x))^T$ is an $m$-variate Gaussian process provided that for any $n \geq 1$ and any choice of $x_1, x_2, \ldots, x_n$ in $X$, the vector $(Y^T(x_1), Y^T(x_2), \ldots, Y^T(x_n))^T$ has a multivariate normal distribution.

The GP is a rich class of functions and we assume that $y(\cdot)$ can be expressed as a draw from a mixture of GP’s. Using GP emulators also allows tractable derivations of
the predictive distribution on account of well-established results on multivariate normality. For our purpose, we consider the class of second-order stationary or stationary GP’s.

**Definition 1.3.2** An $m$-variate Gaussian process $Y(x), x \in X$ is defined to be second-order stationary if $E(Y(x)) = \mu \in \mathbb{R}_m$ for all $x \in X$, $E(Y(x) - \mu)(Y(x') - \mu)^T = \text{Cov}(Y(x), Y(x')) = K(x - x')$ for any given pair of inputs $x, x' \in X$. If $x = x'$, then $K(0) = \text{Var}(Y(x))$ which we denote by $\Sigma_0$ for all $x \in X$.

At any untried input vector $x_0 \in X$, we can think of the predictive distribution of $Y_0 = Y(x_0)$ as capturing information about $Y_0$ provided in the data \{\(Y(x_1) = y(x_1), Y(x_2) = y(x_2), \ldots, Y(x_n) = y(x_n)\}\} given the designs $x_i \in D_n$. We organize the data as an $n \times m$ matrix $Y_{n,m}$ as follows:

\[
Y_{n,m} = \begin{pmatrix}
Y_1(x_1) & Y_2(x_1) & \ldots & Y_m(x_1) \\
Y_1(x_2) & Y_2(x_2) & \ldots & Y_m(x_2) \\
\vdots & \vdots & \ddots & \vdots \\
Y_1(x_n) & Y_2(x_n) & \ldots & Y_m(x_n)
\end{pmatrix} = \begin{pmatrix}
Y^T(x_1) \\
Y^T(x_2) \\
\vdots \\
Y^T(x_n)
\end{pmatrix}
\]

(1.1)

and let $Y_{n,m} = (Y^T(x_1), Y^T(x_2), \ldots, Y^T(x_n))^T \in \mathbb{R}_{mn}$ be the data vector. If $Y_0$ and \{\(Y(x_i) : x_i \in D_n\)\} are dependent quantities, then the conditional distribution $[Y_0|Y_{n,m}]$ is the logical choice for the predictive distribution. Accordingly, we first derive the joint distribution $[Y_0, Y_{n,m}]$ based on a two-stage hierarchical model.

In the first stage, we suppose that $Y(x)$ follows an $m$-variate Gaussian process (GP) conditional on some parameter vector $(\mu, \Sigma_0, \psi)$ denoting respectively the mean, variance, and correlation parameters of the GP. In symbols,

\[
Y(x) \sim \text{GP}(\mu, K(\cdot; \Sigma_0, \psi))
\]

(1.2)
In the second stage we assume that a parametrization of \((\mu, \Sigma_0)\) follows a non-informative distribution, that is,

\[
\pi(\mu, \Sigma_0) \propto |\Sigma_0|^{-(m+1)/2}
\]

where \(\pi(D)\) denotes the distribution of a random quantity \(D\) and \(|A|\) denotes the determinant of a positive definite matrix \(A\). We treat \(\psi\) as known purely for computational convenience. The GP specification implies a willingness to assume that \(y(\cdot)\) varies continuously over \(\mathcal{X}\) and that for any given \(x \in \mathcal{X}\), the uncertainty about \(y(x)\) can be described by a multivariate normal distribution having mean and variance functions which depend upon \((\mu, \Sigma_0, \psi)\). A constructive approach is taken to form multivariate GP models with a valid cross-covariance structure—an important consideration when using GP emulators. The non-informative second-stage distribution reflects the idea that only weak information is available about \((\mu, \Sigma_0)\). Given this particular two-stage model, we show that the conditional distribution \([Y_0|Y_{n,m}, \psi]\) behaves as a multivariate \(t\) process which we then take as our emulator. In reality however, \(\psi\) is unknown, so we replace it by some estimator \(\hat{\psi}\), e.g. maximum likelihood, restricted maximum likelihood, or posterior mode. Plugging in \(\hat{\psi}\) tends to be optimistic, meaning that it causes an understatement of the prediction uncertainty by disregarding the variability between \(\hat{\psi}\) and its true unknown value. However, “plug-in” prediction is justifiable in situations where varying model parameters over reasonable ranges results in marginal changes in the sizes of the associated prediction variances ((Diggle and Ribeiro 2007)).
1.3.3 Models for the Cross-Covariance

In the first stage of the model we postulate that

\[ Y(x) = B^T f(x) + W(x) \]  

(1.4)

where \( Y(x) = (Y_1(x), Y_2(x), \ldots, Y_m(x))^T \in \mathbb{R}_m, m > 1 \) is a vector of computer experiment outputs, \( f(x) = (f_1(x), f_2(x), \ldots, f_p(x))^T \in \mathbb{R}_p, p \geq 1 \) is a vector of known regression functions common to all the \( Y_k \)'s, \( B = (\beta_1, \ldots, \beta_m) \in \mathbb{R}_{p,m} \) is a matrix of unknown regression coefficients with each \( \beta_k = (\beta_{k,1}, \beta_{k,2}, \ldots, \beta_{k,p})^T \in \mathbb{R}_p \), and \( W(x) = (W_1(x), \ldots, W_m(x))^T \in \mathbb{R}_m \) is a zero mean stationary Gaussian process, the key ingredient through which we capture the dependence among the components \( Y_1(x), Y_2(x), \ldots, Y_m(x) \). We write this \( m \)-variate process

\[ W(x) \sim GP(0, K(\cdot)) \]

where \( K(x - x') \) is an \( m \times m \) matrix whose \((i, j)^{th}\) element is \( \text{Cov}(W_i(x), W_j(x')) \).

Therefore, from Equation [1.4], it follows that

\[ Y(x) \sim GP(B^T f(x), K(\cdot)). \]  

(1.5)

For any arbitrary collection of \( n \) input sites \( x_1, x_2, \ldots, x_n \) we can also organize the multivariate outcomes as an \( n \times m \) matrix \( W_{n,m} \) analogous to \( Y_{n,m} \) in Equation (1.1).

We can then write the \( mn \times 1 \) latent vector as \( W_{n,m} = (W^T(x_1), \ldots, W^T(x_n))^T \).

Then

\[ W_{n,m} \sim MVN(0, \Sigma W_{n,m}) \]
where $\Sigma W_{n,m} \in \mathbb{R}_{mn,mn}$ can be partitioned into $n^2$ blocks of size $m \times m$, with the $(u, v)^{th}$ block equal to the cross-covariance matrix $K(x_u - x_v)$ for $1 \leq u, v \leq n$

$$\Sigma W_{n,m} = \begin{bmatrix}
K(0) & K(x_1, x_2) & \ldots & K(x_1 - x_n) \\
K(x_2 - x_1) & K(0) & \ldots & K(x_2 - x_n) \\
\vdots & \vdots & \ddots & \vdots \\
K(x_n - x_1) & K(x_n - x_2) & \ldots & K(0)
\end{bmatrix} \quad (1.6)$$

Then it follows that

$$\text{vec} \ Y_{n,m}^T \sim \text{MVN}( (I_n \otimes B^T) F(n), \Sigma W_{n,m} ) \quad (1.7)$$

where $F(n) = (f^T(x_1), \ldots, f^T(x_n))^T$, $I_n$ is the $n \times n$ identity matrix, and $\otimes$ is the Kronecker product operator. If we denote the variance of the data vector $Y_{n,m}$ by $\Sigma Y_{n,m}$ then

$$\Sigma Y_{n,m} = \Sigma W_{n,m} \quad (1.8)$$

To implement the first stage, we face the challenge of choosing a valid matrix-valued cross-covariance function, a more demanding task than choosing a real-valued function in the case of univariate GP’s.

**Definition 1.3.3** An $n \times n$ real matrix $A$ is called positive definite if

$$z^T A z > 0$$

for all non-zero real vectors $z \in \mathbb{R}_n$. The determinant of a positive definite matrix is always positive so a positive definite matrix is always non-singular.

We require that for any arbitrary number $n$ and choice of input sites, $x_1, \ldots, x_n$, the resulting $\Sigma W_{n,m}$ be symmetric and positive definite. By itself, $K(\cdot)$ need not be symmetric nor positive definite but must satisfy

$$K(x - x') = K^T(x' - x)$$
so that $\Sigma W_{n,m}$ is symmetric. At $x = x'$, $K(x - x')$ must become symmetric and positive definite because it is the covariance matrix of the latent components $W_1, \ldots, W_k$ specific to input vector $x$.

To ensure positive-definiteness, we follow a constructive approach which has been used earlier in the environmental sciences (Ver Hoef and Barry 1998; Wackernagel 2003; Finley, Banerjee, Ek, and McRoberts 2008) as well as in the computer modeling literature (Kennedy and O’Hagan 2000; Williams, Lehman, Santner, and Notz 2002). One example is to set

$$W(x) = AZ(x)$$  \hspace{1cm} (1.9)

with $Z(x) = (Z_1(x), \ldots, Z_m(x))^T \in \mathbb{R}_m$ where $(Z_1(\cdot), \ldots, Z_m(\cdot))$ are mutually independent, zero-centered, unit variance, stationary Gaussian processes and $Z_k(\cdot)$ has a correlation function $\rho_k(\cdot; \vartheta_k)$ with correlation parameter vector $\vartheta_k$. The matrix $A$ is selected as an $m \times m$ non-singular matrix to make sure that $\Sigma Y_{n,m}$ is positive definite. Since

$$\text{Cov}(Z_k(x), Z_j(x')) = 0$$

for $k \neq j$ for all $x$ and $x'$ are, an immediate consequence is that

$$K_Z(x - x'; \vartheta) = \begin{pmatrix}
\text{Cov}(Z_1(x), Z_1(x')) & \cdots & \text{Cov}(Z_1(x), Z_m(x')) \\
\text{Cov}(Z_2(x), Z_1(x')) & \cdots & \text{Cov}(Z_2(x), Z_m(x')) \\
\vdots & \ddots & \vdots \\
\text{Cov}(Z_m(x), Z_1(x')) & \cdots & \text{Cov}(Z_m(x), Z_m(x'))
\end{pmatrix} = \text{diag}\{\rho_1(x - x'; \vartheta_1), \ldots, \rho_m(x - x'; \vartheta_m)\}$$  \hspace{1cm} (1.10)

The cross-covariance matrix $K_Z(\cdot; \vartheta)$ with

$$\vartheta = (\vartheta_1, \ldots, \vartheta_m)$$

is positive definite since the diagonals are the values of a set of real-valued positive-definite functions $\rho_1(\cdot), \ldots, \rho_m(\cdot)$. More precisely, since the $Z_k$’s have unit variance,
\( K_Z(\cdot; \vartheta) \) is, in fact, a cross-correlation matrix. Defining \( Z_{n,m} \in \mathbb{R}_{n,m} \) analogously as in \( Y_{n,m} \) in Equation (1.1), and letting
\[
Z_{n,m} = (Z^T(x_1), \ldots, Z^T(x_n))^T \in \mathbb{R}_{mn},
\]
it follows that
\[
Z_{n,m} \sim MVN(0, \Sigma_{Z_{n,m}}) \tag{1.11}
\]
where
\[
\Sigma_{Z_{n,m}} = \begin{pmatrix}
K_Z(0; \vartheta) & K_Z(x_1 - x_2; \vartheta) & \cdots & K_Z(x_1 - x_n; \vartheta) \\
K_Z(x_2 - x_1; \vartheta) & K_Z(0; \vartheta) & \cdots & K_Z(x_2 - x_n; \vartheta) \\
\vdots & \vdots & \ddots & \vdots \\
K_Z(x_n - x_1; \vartheta) & K_Z(x_n - x_2; \vartheta) & \cdots & K_Z(0; \vartheta)
\end{pmatrix} \tag{1.12}
\]
In Equation (1.12), the block diagonals of \( \Sigma_{Z_{n,m}} \) are \( m \times m \) identity matrices since these equal
\[
K_Z(0, \vartheta) = \text{diag}\{\rho_1(0; \vartheta_1), \ldots, \rho_m(0; \vartheta_m)\}
\]
where \( \rho_k(0; \vartheta_k) = 1 \) for any valid correlation function. The \((u, v)^{th}\) off-diagonal block is the \( m \times m \) diagonal matrix
\[
K_Z(x_u - x_v; \vartheta) = 0_{m,m}
\]
with \( x_u - x_v \neq 0 \in \mathbb{R}_d \). Hence \( \Sigma_{Z_{n,m}} \) is an identity matrix of order \( mn \times mn \).

The implications of defining the latent process \( W(x) \) by (1.9) are as follows:

1. Recalling that \( K(x_u - x_v; \vartheta) = \text{Cov}(W(x_u), W(x_v)) \in \mathbb{R}_{m,m} \) where \( \vartheta \) is the vector of correlation parameters for the process \( Z(x) \). Since \( W(x) = AZ(x), x \in \mathcal{X} \) we have that for any \( x_u, x_v \),
\[
K(x_u - x_v; \vartheta) = \text{Cov}(W(x_u), W(x_v)) = AK_Z(x_u - x_v; \vartheta)A^T \tag{1.13}
\]
2. By (1), when \( x_u = x_v \), then \( K(0; \vartheta) = AA^T \). This implies that
\[
A = K^{\frac{1}{2}}(0, \vartheta) \tag{1.14}
\]
Since \( K(0, \vartheta) \in \mathbb{R}_{++}^{m,m} \) represents the variance-covariance matrix of the model outputs \( Y_1, \ldots, Y_m \) for the same input, we can identify the non-singular matrix transform \( A \) as the square root or Cholesky factor of \( K(0, \vartheta) \). The one-to-one correspondence between a positive definite matrix and its Cholesky factor is known (Harville 1997), so the matrix \( A \) is well-defined. Without loss of generality, we take \( A \) as a lower triangular matrix. Hence \( A \) indirectly models the within-input covariance of the components of the \( W(x) \) process.

3. The \( mn \times mn \) cross-covariance matrix of \( Y_{n,m} \) can be expressed in terms of \( A \) and \( \Sigma_{Z_{n,m}} \) as follows:
\[
\Sigma_{Y_{n,m}} = \begin{pmatrix}
K(0; \vartheta) & \ldots & K(x_1 - x_n; \vartheta) \\
K(x_2 - x_1; \vartheta) & \ldots & K(x_2 - x_n; \vartheta) \\
\vdots & \ddots & \vdots \\
K(x_n - x_1; \vartheta) & \ldots & K(0; \vartheta)
\end{pmatrix} = \begin{pmatrix}
AKZ(0; \vartheta)A^T & \ldots & AKZ(x_1 - x_n; \vartheta)A^T \\
AKZ(x_2 - x_1; \vartheta)A^T & \ldots & AKZ(x_2 - x_n; \vartheta)A^T \\
\vdots & \ddots & \vdots \\
AKZ(x_n - x_1; \vartheta)A^T & \ldots & AKZ(0; \vartheta)A^T
\end{pmatrix} = (I_n \otimes A) \begin{pmatrix}
I_m & \Upsilon_{1,2}(\vartheta) & \ldots & \Upsilon_{1,n}(\vartheta) \\
\Upsilon_{1,2}(\vartheta) & I_m & \ldots & \Upsilon_{2,n}(\vartheta) \\
\vdots & \vdots & \ddots & \vdots \\
\Upsilon_{n,1}(\vartheta) & \Upsilon_{n,2}(\vartheta) & \ldots & I_m
\end{pmatrix} \times (I_n \otimes A^T) = (I_n \otimes A) \Sigma_{Z_{n,m}} (I_n \otimes A^T) \tag{1.15}
\]
where
\[
\Upsilon_{u,v}(\vartheta) = diag\{\rho_1(x_u - x_v; \vartheta_1), \ldots, \rho_m(x_u - x_v; \vartheta_m)\} \in \mathbb{R}_{++}^{m,m}
\]
The constructive approach here generalizes some of the known cross-covariance structures proposed for GP models of multiple output computer experiments. We now show how particular specifications of the matrix $A$ and $K_Z(\cdot; \vartheta)$ give rise to these known models. We develop the idea that $A$ accounts for the covariances due to the model outputs while $K_Z(\cdot; \vartheta)$ models the covariances due to the model inputs or the spatial effects which may depend upon the $Y_k$’s.

If we specify $A$ to have diagonal structure, the outcome is a model assuming independence among the model outputs $Y_k$. This model is a special case of the more general dependence model which simply provides that $A \in \mathbb{R}_{m,m}^+$. Formally, we describe the independence model by letting

$$A = \text{diag}(\sigma_1, \ldots, \sigma_m) \text{ where } \sigma_k > 0 \ \forall \ k$$

$$\Rightarrow AA^T = \text{diag}(\sigma_1^2, \ldots, \sigma_m^2)$$

In the second line of Equation (1.16), we recognize the left hand side as the quantity $K(0, \vartheta)$, the within-input covariance matrix of $Y(x)$. The diagonal structure for $A$ ignores any dependencies among the components of $Y(x)$, $x \in X$ and reduces multivariate emulation into building $m$ GP emulators, one for each $Y_k$, using what has now become a standard recipe first introduced by Sacks, Welch, Mitchell, and Wynn (1989). Such is the model adopted by Keane (2006), Forrester, Sobester, and Keane (2008), and Knowles (2006) who all proposed sequential designs for solving a multiobjective optimization problem using computer experiments.

If we assume a common correlation structure for the $Z(\cdot)$ process, that is a common correlation structure for all the $Z_k$’s, the end result is a separable cross-covariance
model for the \( Y(\cdot) \) process. The separable case is a subset of the more general non-
separable model which allows different correlation functions for each \( Z_k \). In symbols,
a common correlation structure is written as

\[
\rho_k(\cdot; \vartheta_k) = \rho(\cdot; \vartheta^*) \quad \text{with} \quad \vartheta_k = \vartheta^* \quad \text{for all} \quad k
\]  

(1.17)

**Definition 1.3.4**  
A separable cross-covariance structure holds for the \( m \)-variate Gaussian process \( Y(x) \), \( x \in \mathcal{X} \), if for any \( n \geq 1 \), and any collection \( \{x_1, \ldots, x_n\} \) of input sites, the \( mn \times mn \) cross-covariance matrix of \( Y_{n,m} \) can be expressed as the Kronecker
product of matrices \( R(\vartheta^*) \in \mathbb{R}^{n,n}_+ \) and \( AA^T \in \mathbb{R}^{m,m}_+ \), that is,

\[
\Sigma_{Y_{n,m}} = R(\vartheta^*) \otimes AA^T
\]  

(1.18)

where

\[
R(\vartheta^*) = \begin{pmatrix}
1 & \rho(x_1 - x_2; \vartheta^*) & \ldots & \rho(x_1 - x_n; \vartheta^*) \\
\rho(x_1 - x_2; \vartheta^*) & 1 & \ldots & \rho(x_2 - x_n; \vartheta^*) \\
\vdots & \vdots & \ddots & \vdots \\
\rho(x_n - x_1; \vartheta^*) & \rho(x_n - x_2; \vartheta^*) & \ldots & 1
\end{pmatrix}
\]

To see how a common correlation structure induces separability in the cross-covariance
of \( Y_{n,m} \), observe that

\[
K_{Z}(x_u - x_v; \vartheta) = I_m \otimes \rho(x_u - x_v; \vartheta^*)
\]

\[
\Rightarrow \Sigma_{Z_{n,m}} = R(\vartheta^*) \otimes I_m
\]  

(1.19)

Substituting the last equality of Equation (1.19) into the expression for the cross-
covariance of \( Y_{n,m} \) in Equation (1.15) we have

\[
\Sigma_{Y_{n,m}} = (I_n \otimes A) \Sigma_{Z_{n,m}} (I_n \otimes A^T)
\]

\[
= (I_n \otimes A) (R(\vartheta^*) \otimes I_m) (I_n \otimes A^T)
\]

\[
= R(\vartheta^*) \otimes AA^T
\]  

(1.20)
Note that the term $AA^T$ denotes the within-input covariance matrix. The last equality follows from the distributive law particularly

$$(C \otimes D)(E \otimes G) = CE \otimes DG$$

governing Kronecker products, provided the matrices are compatible with multiplication. The separable cross-covariance model is used by Conti, Gosling, Oakley, and O’Hagan (2007) in their work on dynamic emulation and Rougier (2008) in his construction of a multivariate emulator called the Outer Product Emulator.

What has been demonstrated so far is that, specific forms for $A$ and $K_{Z(\cdot ; \vartheta)}$ lead to currently proposed cross-covariance structures. To complete the presentation here, we list four possible models and give more detailed discussion in the next paragraphs regarding the model implication for $Y_{n,m}$.

- **Model I:**
  $$A = diag\{\sigma_1, \ldots, \sigma_m\} \text{ and } K_{Z(\cdot ; \vartheta)} = \rho(\cdot ; \vartheta^*)$$

- **Model II:**
  $$A \in \mathbb{R}_{m,m}^+ \text{ and } K_{Z(\cdot ; \vartheta)} = \rho(\cdot ; \vartheta^*)$$

- **Model III:**
  $$A = diag\{\sigma_k, \ldots, \sigma_m\} \text{ and } K_{Z(\cdot ; \vartheta)} = diag(\rho_1(\cdot ; \vartheta_1), \ldots, \rho_m(\cdot ; \vartheta_m))$$

- **Model IV:**
  $$A \in \mathbb{R}_{m,m}^+ \text{ and } K_{Z(\cdot ; \vartheta)} = diag(\rho_1(\cdot ; \vartheta_1), \ldots, \rho_m(\cdot ; \vartheta_m))$$

Briefly, Model I is a *separable independence* model, and its non-separable extension is Model III. Model II is a *separable dependence* model and its non-separable extension is
Model IV. In this thesis, only GP emulators with Models II and III cross-covariance structures were implemented. These are discussed in later subsections where we highlight the important implications. Since only the non-separable independence model is considered, we label the emulator with Model II structure as the independent Gaussian processes model or IGP for short. As for the GP emulator with Model III structure, we name it CoH after Conti, Gosling, Oakley, and O’Hagan (2007) whose work we apply here. Another model used for computer experiments with multiple responses is the Autoregressive Model used by Kennedy and O’Hagan (2000) and Williams, Lehman, Santner, and Notz (2002). This model exemplifies the Model IV structure which we also discuss to underline the variety of covariance of structures allowed by the constructive approach.

Model I: A Separable Independence Model

The separable independence model is the least complex among the four models for the cross-covariance. It is obtained by imposing special structures on the within-input covariance matrix and on the correlation structure of the elementary processes \( Z_k \). Particularly, we assume independence among the model outputs \( Y_k(\cdot) \) and a common correlation structure for the \( Z_k(\cdot) \). These are summarized as follows:

\[
A = \text{diag} \{ \sigma_1, \ldots, \sigma_m \} \quad (1.21)
\]

\[
\rho_k(\cdot; \vartheta_k) = \rho(\cdot; \vartheta^*) \quad \forall k
\]

where \( \sigma_k > 0 \) for all \( k \) to ensure positive-definiteness. Here onwards, we use the notation \( \Lambda \equiv \text{diag} \{ \sigma_1, \ldots, \sigma_m \} \) to differentiate this special case from the general \( A \) notation. In the second line of Equation (1.21), \( \vartheta_k = \vartheta^* \) for all \( k \), that is, \( \vartheta^* \) is the common correlation parameter vector. These special structures imply that given any
two scalar outputs $Y_j(\cdot)$ and $Y_k(\cdot)$ for all $x, x' \in \mathcal{X}$

\[
\text{Cov}(Y_j(x), Y_k(x')) = \begin{cases} 
0, & \text{if } j \neq k \\
\sigma_j^2 \rho(x - x'; \vartheta^*), & \text{if } j = k 
\end{cases} 
\]  
(1.22)

or in terms of vector outputs,

\[
\text{Cov}(Y(x), Y(x')) = \begin{cases} 
\Lambda \Lambda^T, & x = x' \\
\rho(x - x'; \vartheta^*) \Lambda \Lambda^T & x \neq x'
\end{cases} 
\]  
(1.23)

which leads to the following expression for the cross-covariance of $Y_{n,m}$:

\[
\Sigma_{Y_{n,m}} = \begin{pmatrix}
\Lambda \Lambda^T & \rho_{1,2} \Lambda \Lambda^T & \cdots & \rho_{1,n} \Lambda \Lambda^T \\
\rho_{2,1} \Lambda \Lambda^T & \Lambda \Lambda^T & \cdots & \rho_{2,n} \Lambda \Lambda^T \\
\vdots & \vdots & \ddots & \vdots \\
\rho_{n,1} \Lambda \Lambda^T & \rho_{n,2} \Lambda \Lambda^T & \cdots & \Lambda \Lambda^T
\end{pmatrix} = R(\vartheta^*) \otimes \Lambda \Lambda^T 
\]  
(1.24)

where

\[
R(\vartheta^*) = \begin{pmatrix}
1 & \rho_{1,2} & \cdots & \rho_{1,n} \\
\rho_{2,1} & 1 & \cdots & \rho_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
\rho_{n,1} & \rho_{n,2} & \cdots & 1
\end{pmatrix} \in \mathbb{R}^+_{n,n},
\]

and $\rho_{u,v} = \rho(x_u - x_v; \vartheta^*)$ for $u, v = 1, \ldots, n$. The cross-covariance matrix in Equation (1.24) is positive definite by construction and symmetric for as long as $\rho(\cdot; \vartheta^*)$ is an even function of its argument, that is $\rho(x_u - x_v) = \rho(x_v - x_u)$.

**Model II: A Non-Separable Independence Model**

Another way to model multivariate computer output

\[
Y(x) = (Y_1(x), \ldots, Y_m(x))^T
\]

is to emulate each component separately which means building $m$ single-output GP emulators, one for each $Y_k(\cdot)$. This is equivalent to specifying a non-separable independence model for the cross-covariance which assumes a diagonal structure for $A$
and allows each elementary process $Z_k(\cdot)$ to have its own correlation structure. In symbols,

$$A = \Lambda = \text{diag}\{\sigma_1, \ldots, \sigma_m\}$$

$$K_Z(\cdot; \vartheta) = \text{diag}\{\rho_1(\cdot; \vartheta_1), \ldots, \rho_m(\cdot; \vartheta_m)\}$$

$$\in \mathbb{R}^{+}_{m,m}$$  \hspace{1cm} (1.25)

where $\sigma_k > 0$ for all $k$. For any $1 \leq j, k \leq m$ and for any pair of inputs, $x, x' \in \mathcal{X}$, this cross-covariance model provides that

$$\text{Cov}(Y_j(x), Y_k(x')) = \begin{cases} 0, & j \neq k \\ \sigma_j^2 \rho_j(x - x'; \vartheta_j), & j = k \end{cases}$$  \hspace{1cm} (1.26)

In terms of the vector outputs $Y(x)$ and $Y(x')$,

$$\text{Cov}(Y(x), Y(x')) = \begin{cases} \Lambda \Lambda^T, & x = x' \\ \Lambda \left[\oplus_{k=1}^m \rho_k(x - x'; \vartheta_k)\right] \Lambda^T, & x \neq x' \end{cases}$$  \hspace{1cm} (1.27)

where

$$[\oplus_{k=1}^m \rho_k(x - x'; \vartheta_k)] = \text{diag}\{\rho_1(x - x'; \vartheta_1), \ldots, \rho_m(x - x'; \vartheta_m)\}$$

Therefore, the covariance matrix of $Y_{n,m}$ is given by

$$\Sigma_{Y_{n,m}} = \begin{pmatrix} \Lambda \Lambda^T & \Lambda \Delta_{1,2} \Lambda^T & \ldots & \Lambda \Delta_{1,n} \Lambda^T \\ \Lambda \Delta_{2,1} \Lambda^T & \Lambda \Lambda^T & \ldots & \Lambda \Delta_{2,n} \Lambda^T \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda \Delta_{n,1} \Lambda^T & \Lambda \Delta_{n,2} \Lambda^T & \ldots & \Lambda \Lambda^T \end{pmatrix}$$  \hspace{1cm} (1.28)

where $\Delta_{u,v} = (\oplus_{k=1}^m \rho_k(x_u - x_v'; \vartheta_k)) \in \mathbb{R}_{m,m}$.

We now compare and contrast the two independence models. The idea of emulating the model outputs independently when the components are believed to be related in some way poses potential modeling inefficiencies. Despite this, the independence model continues to be used because of its simplicity and tractability which are its attractive features.
If the outputs \((Y_1(x), Y_2(x), \ldots, Y_m(x))\) represent different quantities with disparate measurement scales, the assumption of common correlation structure may cause inadequacies in the emulator. In some cases, Model II is appropriate following pre-processing of the values by means of a normalization. Alternatively, a non-separable version can be pursued since it accommodates output-specific scale and/or roughness parameters. However, when the model outputs are all measurements of a single quantity, for example, temperatures across time or space or stresses at different points of a body, then a common correlation structure is a rational choice with the added benefit of having fewer parameters to estimate. The separable independence model imposes a more stringent structure than its non-separable extension.

**Model III: A Separable Dependence Model**

One of the simple ways to produce a valid cross-covariance function for \(Y(x), x \in \mathcal{X}\) is to let \(\rho(\cdot; \vartheta^*)\) be a valid correlation function of some stationary univariate GP process and \(AA^T \in \mathbb{R}_{m,m}\) be a positive definite matrix. Then set

\[
\text{Cov}(Y(x), Y(x')) = \rho(x - x'; \vartheta^*) \times AA^T
\]

where \(AA^T\) is the within input covariance matrix associated with \(Y(x)\), for \(x \in \mathcal{X}\), and \(\rho(\cdot; \vartheta^*)\) attenuates the correlation as the separation between \(x\) and \(x'\) increases. This summarizes the gist of the separable dependence model. In terms of the constructive approach, we attain this cross-covariance structure by specifying a common correlation structure for the elementary processes \(Z_k(\cdot)\), and taking a more general form for \(A \in \mathbb{R}_{m,m}^+\). As a result, the covariance between any pair of model outputs, \(Y_j(x)\) and \(Y_k(x')\) for any \(x, x' \in \mathcal{X}\) is given by

\[
\text{Cov}(Y_j(x), Y_k(x')) = \rho(x - x'; \vartheta^*) \times \sigma_{jk}
\]
where $\sigma_{jk} = \sigma_{kj}$ is the covariance between $Y_j(\cdot)$ and $Y_k(\cdot)$ found as the $(j,k)\text{th}$ entry of $AA^T$.

It follows from Equations (1.29) and (1.30) that the cross-covariance matrix of $Y_{n,m}$ has the form:

$$
\Sigma Y_{n,m} = R(\vartheta^\ast) \otimes AA^T.
$$

(1.31)

where $R(\vartheta^\ast)$ is defined below Equation (1.24).

One important feature of Equation (1.31) is that the cross-covariance factors into a part that could be ascribed purely to the spatial association $R(\vartheta^\ast)$ and a part attributable to the model outputs $(AA^T)$. This separability, which is key to tractable statistical modeling, also helps in finding the determinant and inverse of the $mn \times mn$ cross-covariance $\Sigma Y_{n,m}$ which are:

$$
|\Sigma Y_{n,m}| = |R(\vartheta^\ast)|^m |AA^T|^n
$$

$$
\Sigma^{-1} Y_{n,m} = R^{-1}(\vartheta^\ast) \otimes (AA^T)^{-1}.
$$

(1.32)

Thus the inversion of $\Sigma Y_{n,m}$ deals with two matrices of more manageable dimensions, $m \times m$ and $n \times n$, instead of one that is $mn \times mn$. The overall reduction in complexity however comes with a price–a highly restrictive, possibly unsatisfying structure for the cross-covariance. As Rougier (2007) pointed out, Equation (1.31) implies

$$
\frac{\text{Cov} (Y_j(x), Y_k(x'))}{\text{Cov} (Y_j(x^*), Y_k(x^{**}))} = \frac{\rho(x - x')}{\rho(x^* - x^{**})}
$$

(1.33)

which depends only in the model inputs $\{x, x', x^*, x^{**}\}$ for any selected pair of $\{Y_j, Y_k\}$. When $x = x'$ and $x^* = x^{**}$, the ratio in Equation (1.33) would be a constant for any pair of model outputs.
Dropping the restriction of common correlation structure from the separable dependence model, the non-separable dependence model is obtained. By far this is the most complex among all the cross-covariance models considered. This is a generalization of the autoregressive model (AR) used by Kennedy and O’Hagan (2000) and Williams, Lehman, Santner, and Notz (2002). We show here how the cross-covariance induced by the AR specification falls into this class of models.

The AR model of order $m$ provides

$$Y_j(x) = \beta_j^T f_j(x) + A_j Z(x)$$

(1.34)

where $\beta_j \in \mathbb{R}_p$ is the $j^{th}$ column of $B$, $f_j \in \mathbb{R}_p$ is a vector of known regression functions common to all the $Y_j$’s, $A_j$ is the $j^{th}$ row of $A$, the Cholesky factor of the within-input covariance matrix, and $Z(x)$ is an $m$-dimensional GP with mutually independent components having

- $E(Z_j(x)) = 0$,
- $Var(Z_j(x)) = 1$, and
- $Cov(Z_j(x), Z_j(x')) = \rho_j(x - x'; \vartheta^*)$.

The assumption of a shared set of regressors can be relaxed, and in most AR cases used in the literature, the order is usually less than $m$. From Equation (1.34) it follows that for any pair of outputs $1 \leq j, k \leq m$ and any pair of inputs $x, x' \in \mathcal{X}$,

$$Cov(Y_j(x), Y_k(x')) = \sum_{l=1}^{m} a_{jl} a_{kl} \rho_l(x - x'; \vartheta_l)$$

(1.35)
where \( a_{ji} \) is the \( l^{th} \) entry in \( A_j = (a_{j1}, \ldots, a_{jm}) \) which is the \( j^{th} \) row of \( A \). It follows that

\[
\text{Cov}(Y(x), Y(x')) = A \left[ \bigoplus_{k=1}^{m} \rho_k(x - x'; \theta_k) \right] A^T. \tag{1.36}
\]

The cross-covariance matrix \( \Sigma_{Y_{n,m}} \) is therefore equal to

\[
\Sigma_{Y_{n,m}} = \begin{pmatrix}
AA^T & A\Delta_{1,2}A^T & \cdots & A\Delta_{1,n}A^T \\
A\Delta_{2,1}A^T & AA^T & \cdots & A\Delta_{2,n}A^T \\
\vdots & \vdots & \ddots & \vdots \\
A\Delta_{n,1}A^T & A\Delta_{n,2}A^T & \cdots & AA^T
\end{pmatrix} \tag{1.37}
\]

where \( \Delta_{u,v} \) is as defined in Equation (1.28).

### 1.3.4 Predicting Output from Computer Experiments

In this section we discuss techniques for predicting the output of the simulator \( y(x) = (y_1(x), y_2(x), \ldots, y_m(x))^T \) based on training data. By prediction we mean the problem of presenting a point guess of the realization of a random quantity (Santner, Williams, and Notz 2003). Our proposed sequential optimization design for the black box multiobjective problem uses the prediction methods discussed here.

The following discussion describes optimal predictors under the classical or kriging model and under a fully Bayesian approach.

#### The Kriging Model

Suppose \( \mathcal{X} = [0, 1]^d \subset \mathbb{R}_d \) is the design space, and let \( x \in \mathcal{X} \) be a scaled \( d \)-dimensional vector of input values. The kriging approach views the deterministic function \( y(\cdot) \) as a realization of a stochastic process \( Y(x), x \in \mathcal{X} \) and uses a two component model for \( Y(x) \): a linear term that models the drift or trend in the response and a stationary Gaussian process that models the systematic departure of the response from the linear model (Sacks, Welch, Mitchell, and Wynn 1989). The
model is written as:

\[ Y(x) = F(x)\beta + W(x) \]  \hfill (1.38)

where

\[
F(x) = \begin{pmatrix}
    f_T^1(x) & 0_{1,p_2} & \ldots & 0_{1,p_m} \\
    0_{1,p_1} & f_T^2(x) & \ldots & 0_{1,p_m} \\
    \vdots & \vdots & \ddots & \vdots \\
    0_{1,p_1} & 0_{1,p_2} & \ldots & f_T^m(x)
\end{pmatrix} \in \mathbb{R}_{m,p}
\]

\[
f_k(x) = \begin{pmatrix}
    f_{k1}(x) \\
    f_{k2}(x) \\
    \vdots \\
    f_{kp_k}(x)
\end{pmatrix} \in \mathbb{R}_{p_k}
\]

\[
\beta = \begin{pmatrix}
    \beta_1 \\
    \beta_2 \\
    \vdots \\
    \beta_m
\end{pmatrix} \in \mathbb{R}_p, \quad \beta_k = \begin{pmatrix}
    \beta_{k1} \\
    \beta_{k2} \\
    \vdots \\
    \beta_{kp_k}
\end{pmatrix} \in \mathbb{R}_{p_k}.
\]

The full column rank \( m \times p \) matrix \( F(\cdot) \) contains known regression functions for each of the \( Y_k(\cdot) \), reflected in each row by the \( p_k \)-vector \( f^T_k(\cdot) \) for \( 1 \leq k \leq m \), and \( p = \sum_{k=1}^m p_k \) is the total number of regressors; \( \beta \) is a \( p \times 1 \) vector of unknown regression parameters consisting of the \( p_k \times 1 \) vectors \( \beta_k \) of regression coefficients of \( Y_k(\cdot) \), \( W(x) \) is an \( m \)-variate stationary Gaussian process having the following properties: \( E(W(x)) = 0_m, Var(W(x)) = \Sigma_0 \in \mathbb{R}_{m,m}^+ \) for all \( x \in \mathcal{X} \), and for a given pair \( x_u, x_v \in \mathcal{X} \), \( Cov(W(x_u), W(x_v)) = K(x_u - x_v; \Sigma_0, \psi) \), and \( \psi \) is a vector of correlation parameters. The dimension of \( \psi \) depends on the assumed cross-covariance structure, for example, imposing separability results in a \( \psi \) vector having a lower dimension compared to the \( \psi \)-vector in the more general non-separable case.

The kriging model in (1.38) implies that for any \( x \in \mathcal{X} \), the response \( Y(x) \) is \( m \)-variate multivariate normal with mean \( F(x)\beta \) and variance \( K(0; \Sigma_0, \psi) = \Sigma_0 \).

More importantly, the bias or systematic departure of the response surface from the
linear model is treated as a realization of an $m$-variate stationary Gaussian process, $W(\cdot)$.

Let our data consist of simulator outputs

$$\{Y(x_1) = y(x_1), Y(x_2) = y(x_2), \ldots, Y(x_n) = y(x_n)\}$$

evaluated on a set of input sites $D_n = \{x_1, \ldots, x_n\} \subset \mathcal{X}$. Recall from Subsection 1.3.3 our notation for the data matrix:

$$Y_{n,m} = \begin{pmatrix}
Y_1(x_1) & Y_2(x_1) & \cdots & Y_m(x_1) \\
Y_1(x_2) & Y_2(x_2) & \cdots & Y_m(x_2) \\
\vdots & \vdots & \ddots & \vdots \\
Y_1(x_n) & Y_2(x_n) & \cdots & Y_m(x_n)
\end{pmatrix}$$

We now tackle the goal of predicting $y(x_o)$ at a new input site $x_o$ given the information from $Y_{n,m}$. Let $Y_k \equiv (Y_k(x_1), \ldots, Y_k(x_n))^T \in \mathbb{R}_n$, and $W_k \equiv (W_k(x_1), \ldots, W_k(x_n))^T \in \mathbb{R}_n$, $1 \leq k \leq m$. One way to write the data in terms of Equation (1.38) is as follows:

$$\begin{pmatrix}
Y_1 \\
Y_2 \\
\vdots \\
Y_m
\end{pmatrix} = \begin{pmatrix}
F_1 & 0_{n,p_1} & \cdots & 0_{n,p_m} \\
0_{n,p_1} & F_2 & \cdots & 0_{n,p_m} \\
\vdots & \vdots & \ddots & \vdots \\
0_{n,p_1} & 0_{n,p_2} & \cdots & F_m
\end{pmatrix}\begin{pmatrix}
\beta_1 \\
\beta_2 \\
\vdots \\
\beta_m
\end{pmatrix} + \begin{pmatrix}
W_1 \\
W_2 \\
\vdots \\
W_m
\end{pmatrix}$$

The $n \times p_k$ matrix $F_k$ contains the known regression functions of $Y_k$ evaluated on $D_n$. Since $W(\cdot)$ is a zero-mean process, it follows that

$$E(W_{mn}) = 0_{mn} \text{ and } E(W(x_o)) = 0_m.$$
We write \( \Sigma_{w_{mn}} \equiv Var(W_{mn}) \) and \( C \equiv cov(W_{mn}, W(x_o)) \), where

\[
\Sigma_{w_{mn}} = \begin{pmatrix}
\Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1m} \\
\Sigma_{21} & \Sigma_{22} & \cdots & \Sigma_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
\Sigma_{m1} & \Sigma_{m2} & \cdots & \Sigma_{mm}
\end{pmatrix} \in \mathbb{R}_{mn,mn}
\]

\[
\Sigma_{jk} = \begin{pmatrix}
\text{Cov}(W_j(x_1), W_k(x_1)) & \text{Cov}(W_j(x_1), W_k(x_2)) & \cdots & \text{Cov}(W_j(x_1), W_k(x_n)) \\
\text{Cov}(W_j(x_2), W_k(x_1)) & \text{Cov}(W_j(x_2), W_k(x_2)) & \cdots & \text{Cov}(W_j(x_2), W_k(x_n)) \\
\vdots & \vdots & \ddots & \vdots \\
\text{Cov}(W_j(x_n), W_k(x_1)) & \text{Cov}(W_j(x_n), W_k(x_2)) & \cdots & \text{Cov}(W_j(x_n), W_k(x_n))
\end{pmatrix} \in \mathbb{R}_{n,n}
\]

\[
C = \begin{pmatrix}
c_{11} & c_{12} & \cdots & c_{1m} \\
c_{2m} & c_{22} & \cdots & c_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
c_{m1} & c_{m2} & \cdots & c_{mm}
\end{pmatrix} \in \mathbb{R}_{mn,m}
\]

\[
c_{jk} = \begin{pmatrix}
\text{Cov}(W_j(x_1), W_k(x_o)) \\
\text{Cov}(W_j(x_2), W_k(x_o)) \\
\vdots \\
\text{Cov}(W_j(x_n), W_k(x_o))
\end{pmatrix} \in \mathbb{R}_n, \quad c_j = \begin{pmatrix}
c_{1j} \\
c_{2j} \\
\vdots \\
c_{mj}
\end{pmatrix} \in \mathbb{R}_m
\]

A common optimality criterion for predicting in the univariate instance is the mean squared error or MSE. Consider any \( m \times 1 \) vector function of the data \( Y_{n,m} \),

\[
\hat{Y}_0 \equiv \hat{Y}(x_0; Y_{n,m})
\]

to be a predictor of the \( m \times 1 \) vector \( Y(x_o) \). We extend the MSE criterion to the multivariate case by defining the mean squared prediction error matrix of \( \hat{Y}_0 \):

\[
MSPE(\hat{Y}_0) = E \left[ (\hat{Y}_0 - Y(x_0)) (\hat{Y}_0 - Y(x_0))^T \right] \in \mathbb{R}_{m,m}^+
\]

The expectation in Equation (1.40) is taken over the joint distribution of \( Y(x_o) \) and \( \text{vec} Y_{n,m} \), which in the present case is an \( m(n+1) \)-variate multivariate normal distribution. The diagonals of the \( MSPE(\hat{Y}_0) \) matrix equal the univariate MSE’s of \( \hat{Y}_k(x_o) \), for \( 1 \leq k \leq m \). We define \( \hat{Y}^*(x_0) \) to be the best MSPE predictor of \( Y(x_o) \)
if the matrix difference \([MSPE(\hat{Y}_0) - MSPE(\hat{Y}^*(x_0))]) is non-negative definite for every \(\hat{Y}_0\). We now generalize the fundamental theorem of prediction stated in Santner, Williams, and Notz (2003) for the univariate case to the present multivariate setting.

**Theorem 1.3.5** Suppose that \((Y(x_0), Y_{n,m})\) has a joint distribution for which the conditional expectation of \(Y(x_0)\) given \(Y_{n,m}\) exists. Then \(\hat{Y}(x_0, Y_{n,m}) = E(Y(x_0)|\text{vec}Y_{n,m})\) is an optimal predictor of \(Y(x_0)\).

**Proof** For notational convenience, let \(\hat{Y}_0 \equiv \hat{Y}(x_0, Y_{n,m}), Y_0 \equiv Y(x_0)\). Then

\[
MSPE(\hat{Y}_0) = E \left[ (\hat{Y}_0 - Y_0)^T (\hat{Y}_0 - Y_0) \right] = E \left[ (\hat{Y}_0 - E(Y_0|Y_{n,m}) + E(Y_0|Y_{n,m}) - Y_0)^T \right]
\]

\[
= E \left[ (\hat{Y}_0 - E(Y_0|Y_{n,m})) (\hat{Y}_0 - E(Y_0|Y_{n,m}))^T \right]
\]

\[
+ E \left[ (E(Y_0|Y_{n,m}) - Y_0) (E(Y_0|Y_{n,m}) - Y_0)^T \right]
\]

\[
+ E \left[ (\hat{Y}_0 - E(Y_0|Y_{n,m})) (E(Y_0|Y_{n,m}) - Y_0)^T \right]
\]

\[
+ E \left[ (E(Y_0|Y_{n,m}) - Y_0) (\hat{Y}_0 - E(Y_0|Y_{n,m}))^T \right] \tag{1.41}
\]

By the law of iterated expectations, the last line of the third equation in (1.41) is equal to

\[
E \left\{ E \left[ (E(Y_0|Y_{n,m}) - Y_0) (\hat{Y}_0 - E(Y_0|Y_{n,m}))^T \right] \mid Y_{n,m} \right\} = E \left\{ E \left[ (E(Y_0|Y_{n,m}) - Y_0) \mid Y_{n,m} \right] (\hat{Y}_0 - E(Y_0|Y_{n,m})) \right\} = 0_{m,1} \times (\hat{Y}_0 - E(Y_0|Y_{n,m})) = 0_{m,1}
\]
Applying the same result on expectations to the penultimate line of (1.41) gives:

\[
MSPE(\hat{Y}_0) = E \left[ (\hat{Y}_0 - E(Y_0|Y_{n,m})) (\hat{Y}_0 - E(Y_0|Y_{n,m}))^T \right] + MSPE(\hat{Y}_0^*)
\]

where \(\hat{Y}_0^* \equiv E(Y_0|Y_{n,m})\). Finally,

\[
\left\{ MSPE(\hat{Y}_0) - MSPE(\hat{Y}_0^*) \right\} = E \left[ (\hat{Y}_0 - E(Y_0|Y_{n,m})) (\hat{Y}_0 - E(Y_0|Y_{n,m}))^T \right]
\]

(1.42)

The right hand side is the expectation of a non-negative definite matrix which is non-negative definite for every \(\hat{Y}_0 \neq \hat{Y}_0^*\).

The fundamental theorem of prediction states that the conditional mean of \(Y_0\) given the data is the best MSPE predictor. The result holds even for non-normal multivariate joint distributions of \((Y_0, Y_{n,m})\). The only requirement is that the conditional mean exists.

In general, the best MSPE predictor \(\hat{Y}_0^*\) may be hard to derive except in special cases such as when \(Y(\cdot)\) is a Gaussian process and the regression parameters \(\beta\) are known. For this reason, we restrict our consideration of predictors to the class of linear unbiased predictors, as in Sacks, Welch, Mitchell, and Wynn (1989). A linear predictor \(B^T \text{vec } Y_{n,m}\) is an unbiased predictor of \(Y(x_0)\) if

\[
E(B^T \text{vec } Y_{n,m}) = E(Y(x_0))
\]

(1.43)

In view of the model (1.38), this implies that

\[
E(B^T \text{vec } Y_{n,m}) = F\beta
\]

(1.44)

If Equation (1.44) holds for every \(\beta \in \mathbb{R}_p\) then \(\hat{Y}_B \equiv B^T \text{vec } Y_{n,m}\) is defined to be uniformly unbiased for \(Y_0\). Analytically, uniform unbiasedness removes the need to
know the values of $\beta$. The only requirement is that the mean of $\hat{Y}_0$ equals the mean of $Y_0$ under the kriging model (1.38). Uniform unbiasedness also implies that

$$B^T \mathcal{F} = F(x_0) \quad (1.45)$$

and

$$MSPE(\hat{Y}_B) = B^T \Sigma_{w_{nn}} B - B^T C - C^T B + \Sigma_0. \quad (1.46)$$

The result in (1.45) results from substituting (1.38) and (1.39) into (1.43), while the second implication can be shown algebraically, using the same strategy for proving Theorem 1.3.5 and applying the definition of unbiasedness.

An expression for the best linear unbiased predictor or BLUP can be derived under model (1.38). We start by defining the BLUP of $Y_0$.

**Definition 1.3.6** A predictor $\hat{Y}_A \equiv A^T \text{vec} Y_{n,m}$ is the best linear unbiased predictor or BLUP of $Y(x_0)$ if for any other linear predictor $\hat{Y}_B$ of $Y(x_0)$ the matrix difference

$$[MSPE(\hat{Y}_B) - MSPE(\hat{Y}_A)]$$

is non-negative definite for every $B$ such that $\hat{Y}_B$ is unbiased.

To determine the matrix $A$ that meets the requirements of a BLUP, we minimize the criterion $g(x_0)$ in (1.47) with respect to $A$ and the $p \times m$ matrix of Lagrange multipliers $\Lambda$. The two derivatives are then set to 0 with the proper dimensions:

$$g(x_0) = MSPE(\hat{Y}_A) + 2\Lambda^T[\mathcal{F}^T A^T - F(x_0)] \quad (1.47)$$
The expression for $MSPE(\hat{Y}_A)$ in (1.47) that is minimized is given in (1.46). The derivatives are given in Equations (1.48)

\[
\frac{\partial g(x_0)}{\partial A} = 2\Sigma W A - 2C + 2F\Lambda \\
\frac{\partial g(x_0)}{\partial \Lambda} = 2(F^T A - F^T(x_0))
\]

Therefore the matrix $A$ that makes the MSPE matrix difference in Definition 1.3.6 non-negative for all $B$ can be solved from the system of equations given by

\[
\begin{pmatrix}
0_{p,p} & F_{p,mm} \\
F_{mn,p} & \Sigma w_{mn}
\end{pmatrix}
\begin{pmatrix}
\Lambda_{p,m} \\
A_{mn,m}
\end{pmatrix} =
\begin{pmatrix}
F^T(x_0)_{p,m} \\
C_{mn,m}
\end{pmatrix}
\]

(1.49)

By using a result on the inverse of partitioned matrices in Appendix B of Santner, Williams, and Notz (2003), we have

\[
\left(\begin{array}{c|c}
0 & F^T \\
\hline
F & \Sigma w_{mn}
\end{array}\right)^{-1} = \left(\begin{array}{ccc}
-G & G \Sigma^{-1} W F^T \\
\Sigma^{-1} W F & \Sigma^{-1} W F G F^T \Sigma^{-1} W
\end{array}\right)
\]

(1.50)

\[
G = (F^T \Sigma^{-1} w_{mn} F)^{-1}
\]

\[
\Sigma W = \Sigma w_{mn}
\]

Solving the system of equations in (1.49) using the formula for the inverse in (1.50) gives an expression for $A$. Therefore the BLUP $\hat{Y}_A = A^T \text{vec} Y_{n,m}$ is given by:

\[
\hat{Y}_A = F(x_0)\hat{\beta}_{GLS} + C^T \Sigma^{-1}_W \left(\text{vec} Y_{n,m} - F \hat{\beta}_{GLS}\right)
\]

(1.51)

where

\[
\hat{\beta}_{GLS} = G F^T \Sigma^{-1}_W \text{vec} Y_{n,m}
\]

(1.52)

is the generalized least squares estimator of $\beta$ and $MSPE(\hat{Y}_A)$ equals

\[
MSPE(\hat{Y}_A) = \Sigma_0 - C^T \Sigma W C + \left(F(x_0) - C^T \Sigma^{-1}_W F \right) \left(F^T \Sigma^{-1}_W F \right)^{-1} \\
\times \left(F(x_0) - C^T \Sigma^{-1}_W F \right)^T
\]

(1.53)
A Bayesian Model

In the Bayesian approach to computer experiments, the deterministic function \( y(\cdot) \) is treated as an unknown function. The two-component model for the random function \( Y(\cdot) \) in (1.38) is thought as representing prior uncertainty in \( y(\cdot) \). The prior model is updated using a preliminary training sample \( \{ Y(x_1) = y(x_1), \ldots, Y(x_n) = y(x_n) \} \) of \( n \) selected evaluations of the simulator to give the posterior distribution of \( y(\cdot) \). The posterior distribution is officially regarded as the predictive distribution or the emulator of \( y(\cdot) \) at unseen input sites in \( \mathcal{X} \). Assuming the Gaussian process prior specification in (1.38), the posterior is also a Gaussian process conditional on the regression and covariance function hyperparameters, namely \( \beta, \Sigma_0, \text{ and } \psi \). Conditioning upon the preliminary sample constrains the realization from the Gaussian posterior process to pass through or interpolate the observed data points, inducing a posterior distribution for the hyperparameters (Conti, Gosling, Oakley, and O’Hagan 2007).

Bayesian inference regards the hyperparameters \( (\beta, \Sigma_0, \psi) \) as random variables, in contrast with the kriging model. As such, it does not formally differentiate between parameter estimation and prediction. This provides a context for accounting for parameter uncertainty in predictive inference.

The prior model for \( y(x) \) is

\[
Y(x) = F(x)\beta + W(x)
\]

(1.54)

where \( W(\cdot) \) is a zero-mean vector, second-order stationary \( m \)-variate Gaussian with covariance function \( K(\cdot; \psi) \); \( \text{Var}(W(x)) = K(0_d; \psi) \) which we denote by \( \Sigma_0 \). A prior model is specified for the regression parameters \( \beta \) and covariance function \( \Sigma_0 \).
The correlation parameter vector $\psi$ will be treated as known. We capitalize on the results in Williams (2000) and Conti, Gosling, Oakley, and O’Hagan (2007), providing the posterior or predictive distribution of $Y(\cdot)$ given the data $Y_{n,m}$ as a multivariate Student’s $t$ process under the model specified in (1.54) with a non-informative prior on $(\beta, \Sigma_0)$. The details of the prior are given in Equations (1.55)-(1.57). We take the mean and variance of the predictive distribution as the predictor $\hat{Y}(\cdot)$ and measure of prediction uncertainty respectively.

$$
\pi(\beta, \Sigma_0) = \pi(\beta) \pi(\Sigma_0) \quad (1.55)
$$

$$
\pi(\beta) \propto 1 \quad (1.56)
$$

$$
\pi(\Sigma_0) \propto |\Sigma_0|^{-\frac{m+1}{2}} \quad (1.57)
$$

Since the parameter $\psi$ is not known, we substitute it with an estimate $\hat{\psi}$ using restricted maximum likelihood or a Bayesian posterior mode. Determining the posterior mode requires the specification of a prior model for $\psi$.

### 1.4 Restricted Maximum Likelihood Estimation of $\psi$

To obtain estimates of the correlation parameter vector $\psi$, we resort to restricted maximum likelihood estimation (REML), also called residual or modified maximum likelihood. REML is a method of estimating the parameters in dispersion matrices, which includes but is not limited to the estimation of variances, covariances, components of variance, and functions of variances and covariances such as ratios and sums (Speed 2006). Being based on likelihoods, it requires distributional assumptions on $Y(x)$ and is not specific to any particular design matrix, as are analysis of variance estimators. Hence REML estimation can be applied to a wide variety of models.
For ease of presentation, let \( \widetilde{Y} \equiv \text{vec} Y_{n,m} \). Under our model (1.39), the \( mn \times 1 \) vector of responses \( \widetilde{Y} \) has expected value \( F \beta \) and variance \( V(\psi) \equiv \Sigma W_{mn}(\psi) \), where \( F \) is an \( mn \times p \) matrix of known quantities, \( \beta \) is a \( p \times 1 \) vector of unknown parameters, \( \psi \) is a vector of correlation parameters, and \( V(\psi) \) is an \( mn \times mn \) symmetric and positive definite matrix. The REML estimator \( \hat{\psi}_{REML} \) of \( \psi \) is the value of \( \psi \) that maximizes the restricted likelihood given by:

\[
L_{REML}(\psi) = \text{Constant} \times |V(\psi)|^{-\frac{1}{2}} |F^T V^{-1}(\psi) F|^{-\frac{1}{2}} \times \exp \left\{ -\frac{1}{2} \left( \widetilde{Y} - F \hat{\beta}_{GLS}(\psi) \right)^T V^{-1}(\psi) \left( \widetilde{Y} - F \hat{\beta}_{GLS}(\psi) \right) \right\} 
\]

where

\[
\hat{\beta}_{GLS}(\psi) = (F^T V(\psi) F)^{-1} F V^{-1}(\psi) \widetilde{Y}
\]

is the generalized least squares estimator of \( \beta \). Equivalently, \( \hat{\psi}_{REML} \) can be found by maximizing the restricted log likelihood function denoted by \( l_R(\psi) \) and given by

\[
-2l_R(\psi) = \ln (\det(V(\psi))) + \ln (\det(F^T V(\psi)^{-1} F)) + \widetilde{Y}^T V(\psi)^{-1} [I_{mn} - Q(\psi)] \widetilde{Y}
\]

where \( \det(A) \) denotes the determinant of a square matrix \( A \), \( I_{mn} \) is the \( mn \times mn \) identity matrix, and \( Q(\psi) = F (F^T V(\psi)^{-1} F)^{-1} F V^{-1}(\psi) \) is the projector onto the space of \( F \). For a given \( \psi \), the maximum likelihood estimate of \( \beta \) is \( \hat{\beta}_{GLS}(\psi) \) stated below Equation (1.58). The log likelihood of \( \widetilde{Y} \) evaluated at \( \hat{\beta}_{GLS}(\psi) \), denoted by \( l(\hat{\beta}_{GLS}(\psi), \psi) \) is

\[
-2l(\hat{\beta}_{GLS}(\psi), \psi) = \ln (\det(V(\psi))) + \widetilde{Y}^T V(\psi)^{-1} [I_{mn} - Q(\psi)] \widetilde{Y}
\]

We notice that Equation (1.60) differs from Equation (1.59) by the term

\[
\ln (\det(F^T V^{-1}(\psi) F))
\]
which effectively adjusts for the degrees of freedom lost in estimating the parameter $\beta$. By doing so, REML estimators achieve *unbiasedness* for estimating variance and covariance parameters, which is the standard argument for favoring REML over maximum likelihood. Here onwards, we omit writing $\psi$ when no confusion can result.

To illustrate, we present an example that is due to Speed (2006). Suppose $V = \sigma^2 I_n$, where $\sigma^2$ is a variance for $Y = (Y_1, \ldots, Y_n)$ where the $Y_i, 1 \leq i \leq n$ are uncorrelated responses under this model. Then restricted log likelihood $-2l_R$ under this model is given by Equation (1.59) with

$$\ln (\det(V)) = n \ln \sigma^2$$
$$Y^T V^{-1} (I_n - Q) Y = \sigma^{-2} \text{RSS}$$
$$\ln (\det(F^T V^{-1} F)) = -p \ln \sigma^2 + \ln (\det(F^T F))$$

where $RSS = Y^T (I_n - F(F^T F)^{-1} F^T) Y$ is the regression residual sum of squares. We have

$$-2l_R(\sigma^2) = (n - p) \ln \sigma^2 + \ln (\det(F^T F)) + \sigma^{-2} RSS.$$  

Taking the derivative of $-2l_R(\sigma^2)$ with respect to $\sigma^2$ and equating to 0 leads to the unbiased estimate $\hat{\sigma}^2_{REML} = (n - p)^{-1} RSS$ instead of the biased maximum likelihood estimate $\hat{\sigma}^2_{ML} = n^{-1} RSS$. The estimator $\hat{\sigma}^2_{ML}$ is a *downward* biased estimator with bias factor $\frac{n-p}{n}$.

The idea behind REML estimation is to work with a likelihood of a linear function of the responses that do not depend on the regression parameter $\beta$. Consider a statistic $Z \equiv A^T \tilde{Y}$ where $A$ is a $mn \times (mn - p)$ matrix of full column rank such that $A^T F = 0_{mn-p}$. Thus the columns of $A$ form a basis for the linear subspace that is orthogonal to the linear subspace spanned by the columns of $F$. Standard
computations show that $Z \sim N_p(0_{mn-p}, A^T V A)$. The REML likelihood is then the density of $Z$ and its likelihood is given by

$$L_{REML}(\psi) = \text{Constant} \times |A^T V(\psi) A| \exp \left\{ -\frac{1}{2} \tilde{Y}^T A(A^T V(\psi) A)^{-1} A^T \tilde{Y} \right\}$$  \hspace{1cm} (1.61)$$

Comparing Equation (1.61) to Equation (1.58), we deduce that any effect of the choice of $A$ on $L_{REML}(\psi)$ enters through the “Constant” which implies that $\hat{\psi}_{REML}$ is invariant to the choice of $A$. LaMotte (2007) directly derived Equation (1.58) from Equation (1.61) by proving the propositions stated in Equations (1.62) and (1.63).

$$\tilde{Y}^T A(A^T V^{-1} A) A^T \tilde{Y} = \tilde{Y}^T \left[ V^{-1} - V^{-1} \mathcal{F} (\mathcal{F} V^{-1} \mathcal{F})^{-1} \mathcal{F}^T V^{-1} \right] \tilde{Y}$$  \hspace{1cm} (1.62)$$

$$= \tilde{Y}^T V^{-1} [I_{mn} - Q] \tilde{Y}$$

$$= \left( \tilde{Y} - \mathcal{F} \hat{\beta}_{GLS} \right)^T V^{-1} \left( \tilde{Y} - \mathcal{F} \hat{\beta}_{GLS} \right)$$

$$\det(A^T V A) = \text{Constant} \times \det(V) \det(\mathcal{F}^T V^{-1} \mathcal{F})$$  \hspace{1cm} (1.63)$$

Assuming independence among the components of $Y(x) = (Y_1(x), \ldots, Y_m(x))$, the REML likelihood is a product of the individual REML likelihoods corresponding to each $n \times 1$ vector $Y_k = (Y_k(x_1), \ldots, Y_k(x_n))^T$. The $mn \times mn$ covariance matrix $V$ has a block diagonal structure where each block diagonal equals the $n \times n$ covariance matrix $V_k$ of $Y_k(x_i)$, $1 \leq i \leq n$ and $1 \leq k \leq m$. From linear algebra, we have

$$\det \left( \begin{array}{ccc}
V_1 & 0_{n,n} & \cdots & 0_{n,n} \\
0_{n,n} & V_2 & \cdots & 0_{n,n} \\
\vdots & \vdots & \ddots & \vdots \\
0_{n,n} & 0_{n,n} & \cdots & V_m
\end{array} \right) = \prod_{k=1}^m \det(V_k)$$  \hspace{1cm} (1.64)$$

$$\left( \begin{array}{ccc}
V_1 & 0_{n,n} & \cdots & 0_{n,n} \\
0_{n,n} & V_2 & \cdots & 0_{n,n} \\
\vdots & \vdots & \ddots & \vdots \\
0_{n,n} & 0_{n,n} & \cdots & V_m
\end{array} \right)^{-1} = \left( \begin{array}{ccc}
V_1^{-1} & 0_{n,n} & \cdots & 0_{n,n} \\
0_{n,n} & V_2^{-1} & \cdots & 0_{n,n} \\
\vdots & \vdots & \ddots & \vdots \\
0_{n,n} & 0_{n,n} & \cdots & V_m^{-1}
\end{array} \right)$$  \hspace{1cm} (1.65)$$
We utilize results (1.64) and (1.65) to write the REML log likelihood in the case of independence as follows:

\[-2l_r(\psi) = \ln \left( \prod_{k=1}^{m} \det(V_k) \right) + \ln \left( \prod_{k=1}^{m} \det \left( F_k^T V_k^{-1} F_k \right) \right) \]

\[+ \ln \prod_{k=1}^{m} \left[ Y_k^T \left( I_n - F_k \left( F_k^T V_k^{-1} F_k \right)^{-1} \right) F_k^T V_k \right] \]

\[= \sum_{k=1}^{m} \left( \ln \left( \prod_{k=1}^{m} \det(V_k) \right) + \ln \left( \prod_{k=1}^{m} \det \left( F_k^T V_k^{-1} F_k \right) \right) \right) \]

\[+ \ln \left( \prod_{k=1}^{m} \det \left( Y_k^T V_k^{-1} [I_n - F_k \left( F_k^T V_k^{-1} F_k \right)^{-1} F_k^T V_k] Y_k \right) \right) \]

\[= \sum_{k=1}^{m} (-2l_R(\psi_k)) \]

where \( F_k \) is a \( n \times p_k \) matrix of known quantities and \( \psi_k \) is the correlation parameter vector associated with the response \( Y_k(\cdot) \).

In the case that separability in the cross-covariance structure holds, that is, \( V = \Gamma \otimes \Sigma_0 \) where \( \Gamma \) is \( n \times n \) and \( \Sigma_0 \) is \( m \times m \), McCullagh (2008) writes the REML log likelihood as

\[-2l_R(\psi) = -m \ln \left( \det(\Gamma^{-1} [I_n - Q^*]) \right) + n \ln \det(\Sigma_0) + \text{tr} \left( Y_{n,m} \Gamma^{-1} Q^* Y_{n,m} \Sigma_0^{-1} \right) \]

where \( Q^* = I_n - F \left( F^T \Gamma^{-1} F \right)^{-1} F^T \Gamma^{-1} \) and \( F_1 = \ldots = F_m = F \) is an \( n \times p \) matrix of full column rank containing the set of regressors shared by all \( Y_k \), \( 1 \leq k \leq m \).

Harville (1974) rendered a Bayesian interpretation of REML estimators by pointing out that using \( A^T \hat{Y} \) to make inferences about \( \psi \) is equivalent to ignoring prior information about \( \beta \) and using all the data. He also demonstrated that by integrating out \( \beta \) in the log likelihood with respect to a flat prior, the REML log likelihood can be viewed as a marginal likelihood. Another interpretation suggested by Smyth and Verbyla (1996) is based on the observation that the normal likelihood generated by
\[ \hat{\mathbf{Y}} | \hat{\beta}_{GLS}(\psi) \] coincides with the REML likelihood apart from terms not involving \( \psi \). These interpretations of the log REML likelihood both as a \textit{marginal} and \textit{conditional} likelihood for \( \psi \) follows from the independence of \( \hat{\beta}_{GLS} \) and all error contrasts, that is any matrix \( \mathbf{A} \) for which \( \mathbf{A}^T \hat{\mathbf{Y}} = \mathbf{0} \), under the assumption of normality (Speed 2006).

1.5 The Multiobjective Problem (MOP)

1.5.1 Introduction

The scalar concept of optimality is clear but does not extend when multiple objectives are simultaneously evaluated. When objectives compete, there is not one feasible solution that concurrently optimizes all the criteria. Thus instead of a unique and ideal solution, a \textit{set of compromise solutions} dominating all the others serves the role of an optimum. Accordingly, in a \textit{multiobjective optimization problem}, or MOP, the goal is to find these compromise solutions which represent the best trade-offs among the conflicting desiderata.

The scenario considered here involves \textit{minimizing} \( m \) objectives

\[
\mathbf{y}(\mathbf{x}) = (y_1(\mathbf{x}), y_2(\mathbf{x}), \ldots, y_m(\mathbf{x}))
\]

over the space of all designs \( \mathbf{x} \in \mathcal{X} \). We assume that all \( m \) objectives are equally important meaning that no additional information on the preferences of the decision-maker is available.

In 1872, Vilfredo Pareto gave a characterization of dominating solutions which has provided a basis for defining optimality in the multiobjective setting. Briefly, a feasible solution \( \mathbf{x}^* \) dominates another feasible solution \( \mathbf{x} \) if \( \mathbf{x}^* \) performs at least as well as \( \mathbf{x} \) for every \( y_j(\cdot) \) and outperforms \( \mathbf{x} \) in at least one \( y_j(\cdot) \). Since all \( y_j(\cdot) \), \( j = 1, 2, \ldots, m \) are to be minimized, this means that \( \mathbf{x}^* \) dominates \( \mathbf{x} \) if \( y_j(\mathbf{x}^*) \leq \)
$y_j(x)$ for all $j = 1, 2, \ldots, m$ and there exists at least one $j = 1, 2, \ldots, m$ such that $y_j(x^*) < y_j(x)$.

Aside from Pareto dominance, other notions of dominance also exist such as Geoffrion’s dominance and lexicographic dominance (Steuer 1986). We adopt Pareto dominance because it is the most widely used optimality criterion for solving the MOP.

We confine the discussion to black box MOP’s where explicit expressions of the objectives as a function of $x$ are absent. Instead, a computer code evaluates the values of the vector of objectives at a given design point $x$. Here onwards, we assume that the computer code has been previously validated so it suffices for the purpose of optimization.

Throughout, we represent the design configuration as a vector $x \in \mathcal{X}$, containing control or manufacturing variables that are purely continuous and box-constrained. In the previous section, $\mathcal{X}$ was called the input space. When speaking from an optimization perspective, $\mathcal{X}$ will be referred to as the design, or search space to provide a proper context for the problem.

This section is subdivided into two parts. In Subsection 1.5.2 we formalize the black box multiobjective optimization problem and define Pareto optimality. We conclude, in Subsection 1.5.3, with a review of the literature on black box multiobjective optimization.

1.5.2 Pareto Optimality

A multiobjective optimization is a vector optimization problem. It is the problem of finding a set of design vectors which optimize a set of objective functions. Stated
formally,

\[
\min_{x \in \mathcal{X}} y(x)
\]

(1.69)

with

\[
\mathcal{X} = \{ x = (x_1, \ldots, x_d) : a_i \leq x_i \leq b_i, \ i = 1, \ldots, d \}.
\]

where \( y(x) = (y_1(x), \ldots, y_m(x))^T \), \( m \) is the number of objective functions, \( d \) is the number of design variables, \( a_i, b_i \in \mathbb{R} \) are lower and upper bounds respectively of \( x_i \) for \( i = 1, \ldots, d \), and \( \mathcal{X} \) is the design space or search space. The image of \( \mathcal{X} \) under the computer code is called the objective function space or objective space,

\[
\mathcal{Y} = \{ y(x) = (y_1(x), y_2(x), \ldots, y_m(x))^T : x \in \mathcal{X} \}.
\]

The features of the objective space are unknown a priori. We state the MOP as a minimization process; when the maximum is desired, we minimize its negative.

The expression minimize \( y(x) \) is not clear because \( y(x) \) does not have a natural ordering when \( y(x) \) is vector-valued. We now develop a notion of optimality for an MOP by defining Pareto dominance.

**Definition 1.5.1** Let \( x \) and \( x^* \) be design vectors in \( \mathcal{X} \). Define

- \( x \preceq x^* \) (**\( x^* \) weakly dominates \( x \)**) if and only if \( y_j(x^*) \leq y_j(x) \) for all \( j \in \{1, 2, \ldots, m\} \)

- \( x \prec x^* \) (**\( x^* \) dominates \( x \)**) if and only if \( x \preceq x^* \) and \( y_j(x^*) < y_j(x) \) for at least one \( j \in \{1, 2, \ldots, m\} \)

- \( x \sim x^* \) (**\( x^* \) is indifferent to \( x \)**) if and only if \( x^* \) does not dominate \( x \) and \( x \) does not dominate \( x^* \).
We illustrate these ideas in Figure 1.1 which shows an MOP with \( d = 2 \) and \( m = 2 \). The rectangular design space \( \mathcal{X} \) on the left is projected on the objective space \( \mathcal{Y} \), the set of points delimited by the closed curve on the right side of the figure. Relative to the fixed \( \mathbf{x} \in \mathcal{X} \), the figure shows three zones in \( \mathcal{Y} \) namely, the dominated zone, the dominance zone, and the indifference zone (Audet, Savard, and Zghal 2007).

The dominated zone is the quadrant located “northeast” of \( \mathbf{y}(\mathbf{x}) \) in \( \mathcal{Y} \). All design configurations \( \mathbf{x}^* \in \mathcal{X} \) that map to the dominated zone have objective values \( y_j(\mathbf{x}^*) > y_j(\mathbf{x}) \), \( j = 1, 2 \). Hence, these design configurations are inferior to \( \mathbf{x} \). The dominance zone is the quadrant situated “southwest” of \( \mathbf{y}(\mathbf{x}) \) in \( \mathcal{Y} \). Design configurations \( \mathbf{x}^* \in \mathcal{X} \) that map to the dominance zone have objective values \( y_j(\mathbf{x}^*) < y_j(\mathbf{x}) \), \( j = 1, 2 \). Since these design vectors improve on \( \mathbf{x} \) in both objectives, they dominate \( \mathbf{x} \) and are therefore better than \( \mathbf{x} \). Lastly, the two quadrants in the “northwest” and “southeast” of \( \mathbf{y}(\mathbf{x}) \) in \( \mathcal{Y} \) jointly comprise the indifference zone. These are objective vectors associated with design configurations \( \mathbf{x}' \in \mathcal{X} \) that improve on \( \mathbf{x} \) in one aspect at the expense of the other objective. These designs neither dominate nor are they dominated by \( \mathbf{x} \).

In Figure 1.1 we deduce the following order relations among points \( \mathbf{x}, \mathbf{x}_A, \mathbf{x}_B, \) and \( \mathbf{x}_C \): \( \mathbf{x}_A \prec \mathbf{x} \prec \mathbf{x}_C, \mathbf{x}_A \prec \mathbf{x}_C, \mathbf{x} \sim \mathbf{x}_B, \mathbf{x}_A \sim \mathbf{x}_B, \mathbf{x}_B \sim \mathbf{x}_C \). We now define Pareto optimality.

**Definition 1.5.2** A point \( \mathbf{x}^* \in \mathcal{X} \) is defined as Pareto optimal if and only if there is no \( \mathbf{x} \in \mathcal{X} \) such that \( \mathbf{x}^* \prec \mathbf{x} \). Equivalently, for any \( \mathbf{x} \in \mathcal{X} \) and \( i \in \{1, 2, \ldots, m\} \) such that

\[
y_j(\mathbf{x}) \leq y_j(\mathbf{x}^*) \quad j = 1, 2, \ldots, m \text{ and } j \neq i,
\]
then $y_i(x) > y_i(x^*)$. If $x^*$ is Pareto optimal then $y(x^*)$ is called Pareto efficient. Denote the Pareto set by $P_X \subset X$ and its image the Pareto front, $P_Y \subset Y$.

The definition basically says that to improve on a Pareto optimal point somewhere, one would have to degrade or worsen its performance somewhere else. Thus the Pareto set contains all design configurations that are sufficiently optimized that no free upgrades are possible; its elements must be mutually indifferent to each other. Accordingly, the rational choice for a final design comes from the Pareto set.

The terms non-dominated or non-inferior are synonymous with Pareto optimal. For simplicity, we call the collection of all Pareto optimal designs as the Pareto set and its corresponding image, the Pareto front. The Pareto set is a subset of the design space, $X$ while the Pareto front is a subset of the objective space, $Y$. The shape of the latter describes the nature of the trade-offs between the different objectives.
We introduce two important points in $\mathcal{Y}$ – the ideal or utopia point and the nadir point. The ideal point $y^* \in \mathbb{R}_m$ is defined as the vector whose components are the individual minima of each objective function:

$$y^* = \left( \min_{x \in \mathcal{X}} y_1(x), \min_{x \in \mathcal{X}} y_2(x), \ldots, \min_{x \in \mathcal{X}} y_m(x) \right).$$

The nadir point $y_* \in \mathbb{R}_m$ is defined as the vector whose components are the individual maxima of each objective function:

$$y^* = \left( \max_{x \in \mathcal{X}} y_1(x), \max_{x \in \mathcal{X}} y_2(x), \ldots, \max_{x \in \mathcal{X}} y_m(x) \right).$$

Neither the ideal nor the nadir point need be attainable. Usually, there does not exist an $x \in \mathcal{X}$ such that $y(x) = y^*$ or $y(x) = y_*$. These hypothetical points serve as reference points in some solution approaches to the black box MOP, particularly those that seek solutions $x \in \mathcal{X}$ that minimize a real-valued function of $(y(x) - y^*)$ or maximize a real-valued function of $(y_* - y(x))$. In Chapter 3, the nadir point is used as a reference point for calculating a quality indicator of the Pareto front approximation called the hypervolume indicator.

We now demonstrate a procedure for identifying non-dominated design configurations. Consider an MOP with $m = 2$ objectives to be minimized, $y_1(\cdot)$ and $y_2(\cdot)$ and designs $x_1, x_2 \in \mathcal{X}$ whose images are distinct in the objective space, that is,

$$(y_1(x_1), y_2(x_1)) \neq (y_1(x_2), y_2(x_2)).$$

Since the two designs are distinct, $x_1$ will be dominated by $x_2$ if

$$y_1(x_1) \geq y_1(x_2) \text{ and } y_2(x_1) \geq y_2(x_2).$$

This is equivalent to

$$\min((y_1(x_1) - y_1(x_2)), (y_2(x_1) - y_2(x_2))) \geq 0.$$
In general, given a set of \( n \) designs \( D_n = \{ x_1, x_2, \ldots , x_n \} \subset X \), the \( u \)th design \( x_u \) is dominated if:

\[
\max_{x_v \in D_n, x_v \neq x_u} \min ( (y_1(x_u) - y_1(x_v)), (y_2(x_u) - y_2(x_v))) \geq 0 \tag{1.70}
\]

The expression on the left hand side of (1.70) is called the maximin fitness of design vector \( x_u \in D_n \) for any number \( m \) of objectives to be minimized (Balling 2000).

Suppose the values of the objectives for the designs in \( D_n \) are scaled between zero and one as follows:

\[
y^s_k(x_u) = \frac{y_k(x_u) - y_{\text{min},k}}{y_{\text{max},k} - y_{\text{min},k}}
\]

\[
y_{\text{min},k} = \min_{x_i \in D_n} y_k(x_i)
\]

\[
y_{\text{max},k} = \max_{x_i \in D_n} y_k(x_i)
\]

for \( k = 1, \ldots, m \). If \( y_k(x_i) = c \) for all \( x_i \in D_n \), we set \( y^s_k(x_i) = 1 \).

For any \( x_u \in D_n \) define the scaled maximin fitness function to be

\[
g(x_u) = 1 - \max_{x_v \in D_n, x_v \neq x_u} \min ( (y^s_1(x_u) - y^s_1(x_v)), (y^s_2(x_u) - y^s_2(x_v))) \tag{1.72}
\]

In calculating \( g(x_u) \), the minimum is taken over all the objectives and the maximum is taken over the \( n - 1 \) design vectors \( x_v \neq x_u \) that are in \( D_n \).

**Proposition 1.5.3** Let \( x_u \in D_n \). Then \( 0 \leq g(x_u) \leq 2 \).

**Proof** For any pair \( x_i \in D_n \) and \( x_j \in D_n \setminus \{x_i\} \),

\[
-1 \leq (y^s_k(x_i) - y^s_k(x_j)) \leq +1, \ k \in \{1, \ldots, m\}
\]

It follows that

\[
-1 \leq \min_k (y^s_k(x) - y^s_k(x_i)) \leq +1
\]
Therefore,
\[
x_i \max_{x_j \in \mathcal{D}_n, x_j \neq x_i} \left( \min_{1 \leq k \leq m} [y_k(x) - y_k(x_i)] \right) \in [-1, 1]
\]
Hence
\[
g(x_i) = 1 - \max_{x_j \in \mathcal{D}_n, x_j \neq x_i} \left( \min_{1 \leq k \leq m} [y_k(x) - y_k(x_i)] \right)
\]
\[
\in [0, 2]
\]

**Proposition 1.5.4** A design vector \( x_i \in \mathcal{D}_n \) is non-dominated by any other \( x_j \in \mathcal{D}_n \setminus \{x_i\} \) provided \( 1 < g(x_i) \leq 2 \).

**Proof** Suppose \( 1 < g(x_i) \leq 2 \). Then
\[
-1 \leq \max_{x_j \in \mathcal{D}_n, x_j \neq x_i} \min_{k=1, \ldots, m} (y_k(x_i) - y_k(x_j)) < 0.
\]
Thus for all \( x_j \in \mathcal{D}_n \setminus \{x_i\} \),
\[
-1 \leq \min_{k=1, \ldots, m} (y_k(x_i) - y_k(x_j)) < 0
\]
Therefore, for every \( x_j \) there exists a \( k \in \{1, 2, \ldots, m\} \) such that
\[
y_k(x_i) < y_k(x_j)
\]
Hence by definition, \( x_i \) is non-dominated.

**Proposition 1.5.5** A design vector \( x_i \in \mathcal{D}_n \) is weakly dominated or dominated if
\( 0 \leq g(x_i) \leq 1 \).

**Proof** We are given that \( 0 \leq g(x_i) \leq 1 \). It follows that
\[
0 \leq \max_{x_j \in \mathcal{D}_n, x_j \neq x_i} \left( \min_{k=1, \ldots, m} [y_k(x_i) - y_k(x_j)] \right) \leq 1.
\]
This implies that there exists $x_j \in D_n$ such that $0 \leq \min_{k=1,\ldots,m}[y_k^i(x_i) - y_k^j(x_j)] \leq 1$. Then $y_k^i(x_i) \geq y_k^j(x_j)$ for all $k = 1, \ldots, m$. If strict inequality holds for some $k$, then $x_i$ is strictly dominated by $x_j$, else if equality holds for all $k = 1, \ldots, m$, then $x_i$ is weakly dominated by $x_j$ and vice versa.

Thus if $0 \leq g(x_i) \leq 1$ then $x_i$ is dominated by some current input $x_j \in D_n$. It is an inferior solution. On the other hand, if $1 < g(x_i) \leq 2$ then $x_i$ is not dominated by any contemporary design $x_j \in D_n$. It is therefore a currently efficient or non-dominated solution.

Maximin fitness values can change when new design configurations are added to $D_n$. If a new design configuration $x^* \notin D_n \subset X$ improves on any existing non-inferior design in at least one of objectives, it will alter the composition of the current set of “best” designs. The addition of a new design configuration that is inferior relative to $D_n$ will not change the composition of the current best set.

The simplicity of Equation (1.72) makes the process of identifying the non-dominated designs in a given set of designs $D_n$ easy and efficient. Three nested loops are constructed. The outermost loop is over designs $x_i \in D_n$, the middle loop is over designs $x_j \neq x_i$ in $D_n$, and the innermost loop is over the $k = 1, \ldots, m$ objectives.

Recall that the task at hand is to identify the Pareto set, $P_X$, in $X$, or equivalently, to approximate the Pareto Front, $P_Y$, its image in $Y$. In this thesis, we develop a sequential optimization design whereby we add points one-at-a-time to an initial space-filling design based on an improvement function tailored to the black box MOP.

In a nutshell, our proposed algorithm proceeds by identifying, at each stage $t$, the set of non-dominated solutions, say $P^t_X$, and taking its map $P^t_Y$ as the current approximate to the true Pareto Front. A new design vector $x^*$ is then chosen to
maximize the improvement $g(x)$ on $\mathcal{P}_x$. In a biobjective landscape, the image of the new point, $y(x^*)$, should map to the southwest of the current frontier $\mathcal{P}_y$. Ideally we would maximize the improvement however, we use a computationally inexpensive surrogate in lieu of the actual computer code. This increases efficiency but also introduces uncertainty in evaluating the true improvement of a potential design $x_{new}$. Thus in the end, we choose the next point $x_{new}^*$ so that it maximizes the expected improvement over $\mathcal{P}_x$. At each pass the current set of best designs is identified by means of the scaled maximin fitness function $g(x)$ defined in Equation (1.72). In our sequential algorithm, we deliberately seek out new design configurations that improve on the current non-dominated set using the scaled version of the maximin fitness to guide the progressive identification of the Pareto set.

1.5.3 Review of GP-guided Solution Approaches for the Black box MOP’s

We now review solution approaches to the black box MOP, particularly those that have been developed with Gaussian process emulators. The black box MOP poses unique challenges, the most salient of which are the following:

- Several competing desiderata are aspired at once, barring the existence of a unique solution that uniformly satisfies all the criteria. Thus, more optimization effort is invested to find a good range of design configurations that are optimal in some sense compared to the effort required to produce a single design that is optimal against a single goal.
• Achieving a wide and even coverage of the set of efficient points in the objective space could be complicated by a highly nonlinear mapping between the inputs in the design space and the goal functions in objective space.

• In common with single-objective optimization, the use of expensive computer simulations to evaluate the performance of potential design configurations restricts the number of expensive evaluations that can be made.

Here the goal would be to find at least a set of well-performing solutions while keeping the costs of optimization as low as possible by using a small number of simulation runs.

Surrogate-guided optimization is a popular technique for optimizing costly black box functions. It is a speculative search procedure that returns a set of putatively better designs which are verified by the expensive simulator as long as the computational budget allows. To be viable, surrogate-guided approaches must achieve model accuracy within a few number of evaluations, particularly in the neighborhood of potential optima.

Surrogates can take the form of less expensive solvers based on simplified physical models or data-driven approximations obtained by evaluating the simulator at selected design sites and interpolating or smoothing the function values. Wang and Shan (2007) review data-driven surrogates that have made inroads in engineering design applications such as polynomial regression models, splines, radial basis functions, neural networks, support vector machines, Gaussian processes and hybrid models.

While the use of surrogates significantly reduces the number of expensive simulations, it also introduces a new trade-off between exploiting potentially optimal regions and improving global model accuracy by exploring uncertain areas. In the quest for
rapid convergence to optima, early and aggressive exploitation risks missing an entire region of high quality designs. On the other hand, being on the conservative side means more sampling effort and therefore a reduction in efficiency. A key ingredient to achieving a decent compromise between these competing targets is the criterion for selecting the next design point or points for costly-function evaluation.

The expected improvement criterion of the Efficient Global Optimization or EGO algorithm (Schonlau 1997) for single objective optimization proposes a way of capturing the intended balance between local exploitation and global exploration. Suppose $x_{min}^n$ is the current best design and $y_{min}^n = y(x_{min}^n)$ is the current best feasible value of the function $y(\cdot)$ observed after $n$ evaluations. The improvement of an unseen design site $x' \in \mathcal{X}$ over $x_{min}^n$ is indicated by $I(x') = \max(y_{min}^n - Y(x'), 0)$, where the unobserved simulator output

$$
Y(x') \sim N(\hat{y}(x'), s^2(x'))
$$

$$
\hat{y}(x') = E(Y(x')|y(x_1), \ldots, y(x_n))
$$

$$
s^2(x') = Var(Y(x')|y(x_1), \ldots, y(x_n)).
$$

Consequently, the expected improvement at $x'$ over $x_{min}^n$ has a closed form expression given by

$$
E(I(x')) = (y_{min}^n - \hat{y}(x'))\Phi \left( \frac{y_{min}^n - \hat{y}(x')}{s(x')} \right) + s(x')\phi \left( \frac{y_{min}^n - \hat{y}(x')}{s(x')} \right)
$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the cumulative distribution function and density function of the normal distribution respectively. The term $y_{min}^n - \hat{y}(x)$ in Equation (1.74) represents the predicted improvement at $x$ and drives the local search while the term $s(x')$ describes the uncertainty about the prediction $\hat{y}(x')$ and weighs in the global
search. In this way, the expected improvement or \( EI \) criterion automatically balances the tension between exploitation and exploration.

The GP-guided algorithms reviewed here reflect the general consensus that the MOP can be approached in two ways, according to how user preferences are incorporated in the formulation of the optimization problem. \textit{Scalarization approaches} aggregate the multiple objectives into a single objective or a series of single objective functions and solve the resulting scalar optimization problem. This requires the formation of an overarching function containing the contributions of the \( m \) objectives \( y_1(\cdot), \ldots, y_m(\cdot) \), which in turn calls for an a priori assignment of weights \( w_1, \ldots, w_m \) for each element of \( y(\cdot) \). \textit{Pareto set approaches} on the other hand keep the \( m \) objectives separate throughout the optimization process and do not require a priori expressions of decision-maker preferences or priorities. Solutions are typically located based on the idea of dominance which distinguishes between inferior and non-inferior solutions. Both classes of methods seek to provide the decision-maker with a set of optimal design configurations to choose from. This acknowledges the fact that in practice, design alternatives are chosen on the strength of objectives which may not be explicit in \( y(\cdot) \) possibly because these are difficult to quantify (de Weck 2004).

The EGO algorithm is frequently cited in the literature of black box MOP’s typically dominated by population-based search algorithms. A number of recent publications in the field advance Pareto set approaches that either incorporate elements or propose multivariate extensions of the \( EI \) algorithm. Obayashi (2006) proposed \textit{Multi-EGO}, a multiobjective genetic algorithm (MOGA) applied to the set of \( E(I_k(x)) \), \( k = 1, \ldots, m \), the expected improvement function associated with each component of \( y(x) \). Keane (2006) and Forrester, Sobester, and Keane (2008) tackled
the MOP by using a kriging-assisted genetic algorithm that implements statistical improvement criteria similar to the EI. Zhou, Ong, Nair, Keane, and Lum (2007) developed an optimization framework for evolutionary algorithms enhanced by kriging and radial basis functions. Ponweiser, Wagner, and Vincze (2008) cite references who reviewed various surrogate modeling techniques and their manner of integration into evolutionary algorithms.

In the area of sequential optimization designs, the EI framework is acknowledged in Henkenjohann and Kunert (2007) and Wenzel and Kunert (2008) who both resolved the MOP via a scalarization approach using the desirability index of Harrington (1965)—a weighted geometric mean of desirability functions. Henkenjohann and Kunert developed a version of an improvement function based on the desirability index.

Another scalarization approach was put forward by Knowles (2006) called ParEGO, a concatenation of “Pareto” and “EGO”. The algorithm converts the $m$ objective functions into a single ‘cost’ (objective) function using the augmented Tschebycheff norm and approximates the Pareto efficient set by solving a series of single-objective optimizations using the EGO framework.

A different scalarization strategy is obtained by recasting the MOP as a constrained single-objective problem. This was demonstrated in the sequential algorithm of Williams, Lehman, Santner, and Notz (2002) who solved a biobjective problem by choosing one function to minimize, say $y_1(x)$ subject to $y_2(x) \leq \epsilon_q$. In non-linear multiobjective optimization, this is known as the $\epsilon$-constraint or $\epsilon$-attainment method. The Pareto set can be generated by solving the constrained minimization problem for different values of $\epsilon$ in the constraint function.
We now discuss GP-guided sequential approaches that embrace the EI principle. Though Keane’s (2005) algorithm employs a population-based search, we review their proposed improvement criteria which we later integrate in a sequential optimization design for comparison with our proposed method.

There are other non-GP based approaches that do not account for the uncertainty in the surrogate. The Pareto Set Pursuing (PSP) algorithm of Wang and Shan (2005) used polynomial regression models and a sampling guidance function to direct the search of solutions towards the Pareto front. The Psuedo-Response Surface algorithm of Mullur and Messac (2008) utilized radial basis functions and the Normal Constraint method, which calls for solving a series of single-objective optimization problems to approximate the Pareto front. There are also EA algorithms that use GP surrogates but do not balance between exploitation and exploration (Voutchkov and Keane 2006). These algorithms will not be further discussed here.

Scalarization approaches

A widespread approach to the black box MOP is the conversion of the problem to a scalar-valued parameter dependent optimization problem. A few common examples are:

- **Weighted Sum Approach.** The MOP is transformed and solved as a convex combination of the objectives:

\[
\min_{\mathbf{x} \in \mathcal{X}} \sum_{k=1}^{m} w_k y_k(\mathbf{x})
\]  

(1.75)

where \( w_k \geq 0 \) for \( k = 1, \ldots, m \) are weights such that \( \sum_{k=1}^{m} w_k = 1 \). Here, the weight vector \( (w_1, \ldots, w_m) \) is the optimization parameter.
- **ε-Constraint method.** Only one of the \( m \) objectives is minimized and all the other objectives are transformed into constraints by introducing upper bounds.

\[
\min_{x \in \mathcal{X}} y_k(x) \tag{1.76}
\]

such that

\[
y_j(x) \leq \epsilon_j, \ j \in \{1, \ldots, m\} \setminus \{k\}
\]

In this formulation, the parameter is the set of upper bounds \( \epsilon_j \in \{1, 2, \ldots, m\} \setminus \{k\} \) for a \( k \in \{1, 2, \ldots, m\} \).

- **Approximation to a Reference Point.** The method involves finding a feasible solution \( x' \in \mathcal{X} \) such that \( y(x') \) is close to \( r \in \mathbb{R}^m \), where \( r \) is a reference point, usually the ideal or utopia point defined earlier in Subsection 1.5.2.

\[
\min_{x \in \mathcal{X}} \left( \sum_{k=1}^{m} |y_k(x) - r_k|^q \right)^{\frac{1}{q}} \tag{1.77}
\]

where \( 1 \leq q \leq \infty \). The parameter in this instance is the point \( r = (r_1, \ldots, r_m) \).

In specifying the parameters, the decision-maker’s preferences or priorities are factored into the problem. When these are unavailable, one can construct a Pareto set of optimal solutions by working through a series of single-objective optimizations by varying the parameter settings.

The scalarization approach has the major advantage of allowing the application of well-developed search methodologies for single-objective optimization. However, for a given MOP, it is in no way apparent what scalarizing function to use and how to vary its parameter values in order to access all parts of the Pareto front (Keane 2006). Two relevant sequential algorithms in this category are Knowles (2006) and Henkenjohann and Kunert (2007).
The ParEGO algorithm builds up the Pareto set for a given MOP by solving a series of single-objective optimization problems. In his implementation, Knowles scalarized the MOP via the augmented Tchebycheff norm given by

\[ z_{\lambda}(x) = \max_{k=1,...,m} [\lambda_k y_k(x)] + \eta \sum_{k=1}^{m} \lambda_k y_k(x) \]  \hspace{1cm} (1.78)

where \( \eta \) is a small positive value (set to 0.05 by Knowles), \( \lambda = (\lambda_1, \ldots, \lambda_m) \) is a vector of weights specifying the contribution of each \( y_k(\cdot) \) in the aggregate. A set of evenly distributed weight vectors, \( \Lambda_s \) is defined for the purpose. More precisely,

\[ \Lambda_s = \left\{ \lambda = (\lambda_1, \ldots, \lambda_m) \mid \sum_{k=1}^{m} \lambda_k = 1 \text{ and } \lambda_k \in \left\{ 0, \frac{1}{s}, \frac{2}{s}, \ldots, 1 \right\} \right\} \]  \hspace{1cm} (1.79)

The number \( s \geq 1 \) controls the cardinality of \( \Lambda_s \). Let \( |\Lambda_s| \) denote the number of weight vectors. Then

\[ |\Lambda_s| = \binom{s+m-1}{m-1}. \]

To illustrate, suppose \( m = 2 \) and \( s = 1 \), then \( |\Lambda_2| = 2 \) and \( \Lambda_2 = \{(0, 1, (1, 0)\} \). If \( s = 3 \), then \( |\Lambda_3| = 4 \) and \( \Lambda_3 = \{(0, 1), (1, 0), (2, 0), (0, 0)\} \).

At each iteration, a vector \( \lambda \) is drawn at random without replacement from \( \Lambda \). Evidently, the quantity \( s \) sets an upper bound to the number of single-objective optimizations performed and ultimately, to the number of solutions found. Consequently, the inputs to the ParEGO algorithm consist of \( y(\cdot), d, m, s \). The following pseudo-code outlines how the algorithm proceeds:

1. Initialize by generating a space-filling design of \( n \) points, \( D_n = (x_1, \ldots, x_n) \).
   Evaluate the design vectors in \( D_n \) to get \( y(x_i), i = 1, \ldots, n \). Set the iteration index, \( t=1 \).
2. While the stopping criterion is not met, draw a weight vector $\lambda_t$ without replacement from $\Lambda$;

3. Use $\lambda_t$ to weigh the components of $y(x)$ in computing the aggregate function $z_{\lambda_t}(x)$.

4. With the data pairs $(x_i, z(x_i))$, $i = 1, \ldots, n$, estimate the correlation function parameters. Denote the estimate by, $\hat{\theta} = (\theta_1, \ldots, \theta_d)$.

5. Given $\hat{\theta}$, find a design $x^* \in \mathcal{X}$ that maximizes the expected improvement function given in Equation (1.74).

6. Update the current design by augmenting $x^*$ to $D_n$. Evaluate $y(x^*)$ and obtain $z_{\lambda_t}(x^*)$. Update the training data by augmenting $(x^*, z_{\lambda_t})$ to the previous design.

7. Set $t = t + 1$ and go to Step 2.

Knowles constructs a GP emulator for $z_{\lambda}(\cdot)$, therefore the improvement function is in terms of $z_{\lambda}(\cdot)$. He used a separable Gaussian correlation function and estimated the parameters by maximum likelihood. To maximize the expected improvement criterion, he coded an evolutionary algorithm that worked on predicted values of $z_{\lambda}(x')$ provided by the GP emulator.

We claim that the choice of the augmented Tchebycheff norm raises a question on the validity of its emulation by a Gaussian process. Assuming that $y(\cdot)$ is an $m$-variate Gaussian process, then $Ly(x)$ is also an $m$-variate Gaussian process where $L = diag(\lambda_1, \ldots, \lambda_m)$ and $\lambda = (\lambda_1, \ldots, \lambda_m) \in \Lambda^s$. The result follows from the fact that a linear transformation of a Gaussian process is also a Gaussian process. Hence
for a given $x \in \mathcal{X}$, $Ly(x)$ is a multivariate normal random vector. However, the distribution of the maximum of a multivariate normal vector is no longer normal but takes a more complicated distribution (Arellano-Valle and Genton 2008). Thus it is conceptually problematic to emulate the dominant term of Knowles’ scalarizing function, $\max_{k=1,\ldots,m} \lambda_k y_k(x)$, as a Gaussian process.

Deb (2001) however provides a compelling justification for its choice: this scalarization guarantees that every Pareto optimal solution corresponds to an optimal solution to the resulting single objective problem for any convex or non-convex MOP. In our view this is a stronger argument in its favor. Indeed, in the numerical experiments involving nine MOP’s, Knowles (2006) showed that ParEGO fared acceptably well and generally outperformed a standard genetic algorithm.

Finally, the fact that the weights are varied stochastically at each iteration implies that the resulting finite set approximation of the Pareto front is also random. Suppose $T$ weight vectors are drawn from $\Lambda$ such that $T |\Lambda| << 1$. Then the variation in the resulting approximation sets needs to be accounted for in assessing the quality of approximation. Accordingly, Knowles uses criteria appropriate for stochastic multi-objective optimizers such as those based on so-called attainment functions (Fonseca, da Fonseca, and Paquete 2005). This stochastic feature makes the results of ParEGO incomparable with our proposed algorithm. For this reason, we don’t include the ParEGO algorithm in our comparisons.

**Henkenjohann and Kunert (2007)** The main contribution is a novel definition of an improvement function for the MOP based on the concept of desirability. The desirability function was introduced by Harrington (1965) and refined later by Derringer
and Suich (1980) and Derringer (1994). To this day, desirability functions continue to be used in the simultaneous optimization of several responses among practitioners of designed experiments and response surface methodology and also in the quality engineering community.

The desirability function maps the value of an objective function to the unit-free interval $[0, 1]$, so that a larger value indicates a better quality and therefore greater desirability with respect to this particular target. Information on the decision maker’s preferences including technical considerations is required to specify the form and parameters of the desirability function. One-sided desirability functions are utilized when the relationship between an objective and its corresponding utility or desirability is monotonically increasing or decreasing. Thus for “larger-is-better” or “smaller-is-better” responses, Harrington proposed a desirability function in the form of a Gompertz curve:

$$d(y(x)) = \exp(-\exp(- (\beta_0 + \beta_1 y(x))))$$  \hspace{1cm} (1.80)

where $d(\cdot)$ is the desirability function, $y(x)$ is an objective function value evaluated at a feasible design point $x$, and $\beta_0, \beta_1 \in \mathbb{R}$ are the parameters of the $d(\cdot)$. For responses where deviations from a nominal value in both directions result in a loss of quality, a two-sided desirability function could be specified as follows:

$$d(y(x)) = \exp\left(-\left|\frac{y(x) - \frac{USL + LSL}{2}}{\frac{USL - LSL}{2}}\right|^\nu\right)$$  \hspace{1cm} (1.81)

where $0 < \nu < \infty$, $LSL$ and $USL$ are the lower and upper specification limits.

To compare the overall merit, a desirability index is computed as a weighted geometric mean of the individual desirability functions, that is,

$$DI(x) = \prod_{k=1}^m d_k(y_k(x))^{w_k}$$  \hspace{1cm} (1.82)
where $DI(x)$ is the desirability index corresponding to $x$, $d_k(\cdot)$ is the desirability function associated with the $k^{th}$ objective $y_k(x)$, and $w_k \geq 0$ are weights such that $\sum_{k=1}^{m} w_k = 1$. If all objectives are equally important, we have the special case where $w_k = \frac{1}{m}$ for all $k = 1, \ldots, m$ and the desirability index reduces to the geometric mean of the $m$ desirability functions.

Henkenjohann and Kunert’s approach to the black box MOP follows the expected improvement framework of the EGO algorithm: It starts with a space-filling design of points in design space, constructs a multivariate GP emulator, and samples new design points sequentially according to an expected improvement criterion. The expectation of the novel improvement function is derived assuming one-sided desirability functions for all the $m$ objectives. They applied their algorithm to optimize a metal spinning process which is an MOP with $m = 3$ and $d = 3$.

Assuming that all desirability functions have been fully specified, let

$$d(x_1), d(x_1), \ldots, d(x_n)$$

be the set of desirability vectors determined using Equation (1.80) for each $k = 1, \ldots, m$ and let

$$DI(x_1), DI(x_2), \ldots, DI(x_n)$$

be the corresponding desirability indices obtained using Equation (1.82) for each $x_i$ in the current experimental design. The vector of desirability functions maps the objective vectors in $\mathcal{Y}$ to the desirability space denoted by the unit hypercube $[0, 1]^m$. Define $DI_{max} = \max_{i=1,\ldots,n} DI(x_i)$, the highest attained desirability value among all the design points sampled so far. The quantity $DI_{max}$ plays the same role as $y_{min}$ in the EGO algorithm. Now define $\Delta_{max}$ as the set of all points $\delta$ in the desirability
space whose desirability index equals $DI_{max}$, that is

$$\Delta_{max} = \{ \delta = (\delta_1, \ldots, \delta_m) \in [0, 1]^m : \prod_{k=1}^{m} \delta^{w_k}_k = DI_{max} \}.$$  

This set is guaranteed to be non-empty as it contains at least one element, namely the design vector $x_i$ that provided $DI_{max}$. For an MOP with $m = 2$, the graph of $\Delta_{max}$ is shown in Figure 1.5.3.

---

**Figure 1.2:** A geometric interpretation of the Henkenjohann-Kunert improvement function for $m = 2$ and $g = 1$: the shortest Euclidean distance of a potential design point $x_i \notin D_n$ to the curve $DI_{max} = 0.51$
The improvement function at an unseen point \( x' \in \mathbf{X} \) proposed by Henkenjohann and Kunert takes the form:

\[
I_{HK}^g(x') = \begin{cases} 
\min_{\delta \in \Delta_{max}} \| \delta - D(x') \|^g, & \text{if } DI(x') > DI_{max} \\
0, & \text{if } DI(x') \leq DI_{max}
\end{cases}
\]

(1.83)

where \( \| \cdot \| \) is the Euclidean norm, \( g > 0 \) is a parameter tuning the local-global search, \( D(x') \) is the desirability vector associated with design \( x' \), and \( DI(x') \) is the desirability index of the design \( x' \).

Since it is true that \( DI(x) \leq DI_{max} \) for any feasible \( x \) in the current design, it follows that \( I_{HK}^g(x) = 0 \) by Equation (1.83).

Whenever \( D(x') > DI_{max} \), the Henkenjohann-Kunert or HK improvement function is the shortest distance from the point \( D(x') \) to the curve (or hypersurface) \( \Delta_{max} \) in the \( m \)-dimensional desirability space. Given \( g \) and the coordinates of \( D(x') \), the shortest distance can be found by first locating the point, say \( \delta^* \in \Delta_{max} \) that lies closest to \( D(x') \). This can be done by implementing a mathematical program that finds the constrained minimum of a scalar-valued function in several variables starting at some initial value. Once \( \delta^* \) is found, the HK improvement at \( x' \) is given by \( \| \delta^* - D(x') \| \).

On account of \( y(x') \) being unobserved at this stage, the quantities \( D(x') \) and \( DI(x') \) are treated as random, and therefore also \( I_{HK}^g(x') \) is random. The expected HK-improvement at a given feasible point \( x \in \mathbf{X} \) is

\[
E[I_{HK}^g(x')|y_1^{(n)}, \ldots, y_m^{(n)}] = \int_{z \in [0,1]^m: DI(z) > DI_{max}} \| \delta^* - z \| f_{D(x')}|y_1^{(n)}, \ldots, y_m^{(n)}(z)dz
\]

(1.84)

The determination of the expected improvement capitalizes on the multivariate normality of \( y(x') \) given the data and treating the mean, variance, and correlation
parameters of the GP process as known quantities. If independence among the
responses is supposed, the distribution of $D(x')$ can be written down using the results
of Trautmann and Weihs (2006) who derived the density of $d_k(\cdot)$ for one or two-sided
specifications. Henkenjohann and Kunert worked out the density of $D(x')$ assuming
correlated responses and a one-sided specification for all $d_k(\cdot), k = 1, \ldots, m$. In their
implementation, the expected improvement at a given design point was determined
numerically, though details were not provided.

There is a vagueness in the specification of the GP model as well as in the method
of estimation of the relevant correlation parameters. We inferred the model by means
of a reconstruction based on the stated facts in the paper. Apparently,

$$Y(x) = F(x)\beta + W(x)$$

(1.85)

where

$$f_k(x) = (f_{k1}(x), \ldots, f_{kp_k}(x))^T \in \mathbb{R}_{p_k}$$

$$F(x) = \begin{pmatrix} f_1^T(x) & 0 & \cdots & 0 \\ 0 & f_2^T(x) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & f_m^T(x) \end{pmatrix} \in \mathbb{R}_{m,p}$$

(1.86)

$$\beta_k = (\beta_{k1}, \ldots, \beta_{kp_k})^T \in \mathbb{R}_{p_k}$$

$$\beta = (\beta_1^T, \ldots, \beta_m^T)^T \in \mathbb{R}_p$$

(1.87)

$$W(x) \sim GP(0, K(\cdot))$$

(1.88)

Each $y_k(\cdot)$ had its own set of regressors – either a constant or a constant plus first-
order effects. The $W(\cdot)$ is a zero mean, stationary, $m$-variate Gaussian process
such that $K(\cdot)$ defines a non-separable dependence model for the cross-covariance
of $y(x_1), \ldots, y(x_n)$ for any arbitrary collection of $x_1, \ldots, x_n$ and for any $n \geq 1$. 

65
They report that each $y_k(\cdot)$ was fit separately as in the case of an IGP model (using the SAS Proc MIXED module) yielding estimates of the process variance and spatial effects represented by the scale parameters of a separable Gaussian correlation function. They also state that dependence or correlations among the responses $y_1(x), \ldots, y_m(x)$ were determined using “previous runs”. The resulting GP emulator of $y(x)$ is multivariate normal given data from the current design and plug-in estimates of GP parameters. We could not verify the form of the variance-covariance matrix used to obtain the mean $\hat{y}(x)$ nor verify whether such variance-covariance reproduces the covariance matrix at $x'$, provided as

$$\text{Cov}(y_k(x), y_j(x)) = \rho_{j,k} \hat{\sigma}_k \hat{\sigma}_j$$

where $\hat{\sigma}_k^2$ and $\hat{\sigma}_j^2$ are the estimates of the process variance of $y_k(\cdot)$ and $y_j(\cdot)$ respectively, obtained from the IGP fits while the $\rho_{jk}$ are entries in the correlation matrix apparently determined outside the current design.

Trautmann and Weihs (2004) asserted that a solution found by maximizing the desirability index, $DI(x)$ is necessarily Pareto optimal. The statement is true because of the monotone or dominance-preserving property of the desirability index meaning that for any $x_1, x_2 \in X$ such that $x_1$ weakly dominates $x_2$, or $x_2 \preceq x_2$, we have that $DI(x_2) \leq DI(x_1)$. To show that the desirability index indeed satisfies this property:

$$x_2 \preceq x_1 \implies y_k(x_1) \leq y_k(x_2) \quad \forall k$$

$$\implies d_k(y_k(x_1)) \geq d_k(y_k(x_2)) \quad \forall k$$

$$\implies m \prod_{k=1}^m d_k(y_k(x_1))^{w_k} \geq m \prod_{k=1}^m d_k(y_k(x_2))^{w_k}$$

$$\implies DI(x_1) \geq DI(x_2)$$
To show that it is Pareto optimal: since $DI(x)$ is a (weighted) geometric mean, maximizing the latter is equivalent to minimizing the negative of the weighted sum of the log-transformed desirabilities, that is

$$\max_{x \in \mathcal{X}} DI(x) \Leftrightarrow \max_{x \in \mathcal{X}} \sum_{k=1}^{m} w_k \log d_k(y_k(x))$$

$$\Rightarrow \min_{x \in \mathcal{X}} -\sum_{k=1}^{m} w_k \log d_k(y_k(x)) \quad (1.89)$$

The logarithmic transformation, being one-to-one, is order-preserving so minimizing a positive function say $h(x)$ is equivalent to minimizing $\log h(x)$. The solution to the weighted sum minimization problem in (1.89), if it exists, is always Pareto optimal for the particular weight settings $w = (w_1, \ldots, w_m)$ chosen (Deb 2001). Thus a solution that maximizes the desirability index is Pareto optimal.

Currently, Wenzel and Kunert (2008) are developing a procedure for calculating the expected HK improvement under the condition of correlated responses with a two-sided specification for some $d_k$, $k = 1, \ldots, m$.

**Pareto Set Approach**

Exploring the potential solutions to an MOP without knowing the preferences a priori or assuming a specific utility function to meaningfully combine the $y_k(\cdot)$’s amounts to identifying all the non-inferior solutions. Thus Pareto points can be viewed as an abstraction of specific utility or cost functions (Geilen, Basten, Theelen, and Otten 2007). The solutions that define the trade-off curve or surface represent exactly those design configurations that can be potentially optimal under cost or utility functions that depend monotonically on the different goal functions.
The algorithm developed by Keane implements a kriging-assisted non-dominated sorting genetic algorithm (Deb, Pratap, and Agarwal 2002) that uses statistical criteria to evolve a generation of solutions towards the Pareto front. The statistical criteria can be considered multivariate extensions of EGO’s improvement function. Closed form expressions of these criteria were derived for \( m = 2 \), assuming independent Gaussian processes for the components of \( y(\cdot) \). Unlike the formulation of Knowles (2006) and Henkenjohann and Kunert (2007), Keane keeps the multiple objectives separate throughout the optimization avoiding the need for a scalarizing function to combine the \( m \) objectives.

The property that the Pareto points may be ordered when \( m = 2 \), and not when \( m > 2 \) is exploited in the derivation of closed form expressions. To begin, consider minimizing \( y(x) = (y_1(x), y_2(x)) \), \( x \in \mathcal{X} \). Let \( \mathcal{P}_X^n \) be the current set of non-dominated configurations identified from the current design \( \mathcal{D}_n = (x_1, \ldots, x_n) \) and let \( \mathcal{P}_Y^n \) be the set of objective vectors associated with the designs in \( \mathcal{P}_X^n \). Suppose there are \( q_n \geq 1 \) members in \( \mathcal{P}_X^n \). Let

\[
x_{[1]}, x_{[2]}, \ldots, x_{[q_n]}
\]

be such that

\[
y_1(x_{[1]}) \leq y_1(x_{[2]}) \leq \ldots \leq y_1(x_{[q_n]}).
\]

Since the \( q_n \) design points are all non-dominated, it must be true that

\[
y_2(x_{[1]}) \geq y_2(x_{[2]}) \geq \ldots \geq y_2(x_{[q_n]}).
\]

Sorting the members of \( \mathcal{P}_X^n \) with respect to \( y_1(\cdot) \) in ascending order automatically sorts the \( y_2(\cdot) \) values in descending order.
Consider the event that a new design \( \mathbf{x}' \) will improve on a single member of \( \mathcal{P}_n \), say \( (y_1(\mathbf{x}[1]), y_2(\mathbf{x}[1])) \). Since \( m = 2 \), this may come about in three ways – either \( \mathbf{x}' \) improves on \( \mathbf{x}[1] \) in \( y_1(\cdot) \), \( \mathbf{x}' \) improves on \( \mathbf{x}[1] \) in \( y_2(\cdot) \), or (3) \( \mathbf{x}' \) improves \( \mathbf{x}[1] \) on both \( y_1(\cdot) \) and \( y_2(\cdot) \). Define

\[
i_K(\mathbf{x}') = \begin{cases} 
1, & \text{if } [y_1(\mathbf{x}') < y_1(\mathbf{x}[1])] \text{ or } [y_2(\mathbf{x}') < y_2(\mathbf{x}[1])] \\
0, & \text{else}
\end{cases}
\] (1.90)

The function \( i_K(\mathbf{x}') \) therefore indicates the event that a feasible new point \( \mathbf{x}' \) improves on a current non-dominated design \( \mathbf{x}[1] \). Since \( \mathbf{y}(\mathbf{x}') \) is unobserved we represent it as a random quantity \( \mathbf{Y}(\mathbf{x}') = (Y_1(\mathbf{x}'), Y_2(\mathbf{x}')) \) and consequently treat the improvement at \( \mathbf{x}' \) as a random variable \( I_K(\mathbf{x}') \) defined as

\[
I_K(\mathbf{x}') = \begin{cases} 
1, & \text{if } [Y_1(\mathbf{x}') < y_1(\mathbf{x}[1])] \text{ or } [Y_2(\mathbf{x}') < y_2(\mathbf{x}[1])] \\
0, & \text{else}
\end{cases}
\] (1.91)

Then the expected improvement at \( \mathbf{x}' \notin \mathcal{D}_n \) can be expressed as follows:

\[
E[I_K(\mathbf{x}')|\text{data}] = \int_{\mathbb{R}^2} I_K(\mathbf{x}') \phi(\mathbf{y}(\mathbf{x}')) d\mathbf{y}'
\]

\[
= P \left( Y_1(\mathbf{x}') \leq y_1(\mathbf{x}[1]) \cup Y_2(\mathbf{x}') \leq y_2(\mathbf{x}[1]) \right)
\]

\[
= \Phi \left( \frac{y_1(\mathbf{x}[1]) - \hat{y}_1(\mathbf{x}')}{{s}_1(\mathbf{x}')} \right) + \Phi \left( \frac{y_2(\mathbf{x}[1]) - \hat{y}_2(\mathbf{x}')}{{s}_2(\mathbf{x}')} \right) - P \left( Y_1(\mathbf{x}') \leq y_1(\mathbf{x}[1]) \cap Y_2(\mathbf{x}') \leq y_2(\mathbf{x}[1]) \right)
\] (1.92)

where \( \phi(\cdot) \) is the bivariate normal distribution, the predictive distribution of \( \mathbf{Y}(\mathbf{x}') \) given the data. The mean and variance parameters of the predictive distribution are
respectively,

\[ \hat{y}(x') = (\hat{y}_1(x'), \hat{y}_2(x')) \]

\[ \Sigma(x') = \begin{pmatrix} s_1^2(x') & \rho s_1(x') s_2(x') \\ \rho s_1(x') s_2(x') & s_2^2(x') \end{pmatrix} \]

Assuming independence between \( Y_1(\cdot) \) and \( Y_2(\cdot) \), the expectation in Equation (1.92) has the convenient form:

\[
E[I_K(x') | data] = \Phi \left( \frac{y_1(x[1]) - \hat{y}_1(x')}{s_1(x')} \right) + \Phi \left( \frac{y_1(x[1]) - \hat{y}_1(x')}{s_2(x')} \right)
\]

Next, consider the event that \( x' \) improves on the set \( P^n_X \). This can happen in one of two ways: (1) \( x' \) dominates at least one of the \( x_u \in P^n_X, \ u = 1, \ldots, q_n \) or (2) \( x' \) does not dominate any \( x_u \) nor does any \( x_u \) dominate \( x' \).

The probability of improvement can be obtained by integrating over the shaded area in Figure 1.3. The integration is facilitated by considering the rectangles that comprise the shaded area which gives

\[
E[I_K(x') | data] = \int_{-\infty}^{y_1(x[1])} \int_{-\infty}^{\infty} \phi(Y_1(x'), Y_2(x')) dY_1(x') dY_2(x') \quad (1.94)
\]

\[ + \sum_{u=1}^{q_n-1} \int_{y_1(x_u)}^{y_1(x_{u+1})} \int_{-\infty}^{y_2(x_u)} \phi(Y_1(x'), Y_2(x')) dY_1(x') dY_2(x') \]

\[ + \int_{y_1(x[q_n])}^{\infty} \int_{-\infty}^{\infty} \phi(Y_1(x'), Y_2(x')) dY_1(x') dY_2(x') \]
Figure 1.3: The regions of integration for determining the probability of improvement when $m = 2$.

By the multivariate normality of $Y(x')$ and assuming independence between $Y_1(\cdot)$ and $Y_2(\cdot)$, this integral evaluates to

$$E[I_K(x')|data] = \Phi\left(\frac{y_1(x_1) - \hat{y}_1(x')}{s_1(x')}\right)$$

$$+ \sum_{u=1}^{q_n-1} \left\{ \Phi\left(\frac{y_1(x_{[u+1]}) - \hat{y}_1(x')}{s_1(x')}\right) - \Phi\left(\frac{y_1(x_{[u]}) - \hat{y}_1(x')}{s_1(x')}\right) \right\} \Phi\left(\frac{y_2(x_{[u]}) - \hat{y}_2(x')}{s_2(x')}\right)$$

$$+ \left\{ 1 - \Phi\left(\frac{y_1(x_{[q_n]}) - \hat{y}_1(x')}{s_1(x')}\right) \right\} \Phi\left(\frac{y_2(x_{[q_n]}) - \hat{y}_2(x')}{s_2(x')}\right)$$

(1.95)
The resulting quantity is the Keane’s \textit{probability of improvement} or \textit{PI}. The probability of improvement criterion works irrespective of the scaling differences of the \( y_k(\cdot) \). When used as a search goal however, the \textit{PI} criterion does not necessarily promote a thorough exploration of the Pareto front since it is not biased by the amount of improvement achieved (Keane 2006; Emmerich, Giannakoglou, and Naujoks 2006). Keane overcomes this by maximizing instead the first moment of the \textit{PI} around the current Pareto frontier, which according to Keane, is the multiobjective equivalent of the \textit{EI} criterion of EGO.

An \textit{enhanced PI} criterion was proposed by Hawe and Sykulski (2007) who partitioned the region of improvement as shown in Figure 1.5.3. The partition is generated by projecting the components of \( \mathbf{y}(\mathbf{x}_{[u]}) = (y_1(\mathbf{x}_{[u]}), y_2(\mathbf{x}_{[u]})), \ u = 1, \ldots, q_n \) on the corresponding \( y_1(\cdot) \) and \( y_2(\cdot) \) coordinates. Hawe and Sykulski illustrate the natural levels of improvement, \( \nu = 0, 1, \ldots, q_n \), where the \( \nu^{th} \) level of improvement produces a solution that dominates exactly \( \nu \) of the current non-dominated points \( \mathbf{x}_{[u]} \in \mathcal{P}^n_\mathcal{X} \). The number of disjoint sets in the partition when \( m = 2 \) is \( \frac{(q_n+1)(q_n+2)}{2} \). Using this partitioning, it is then possible to set the expected improvement equal to the probability that a new design point \( \mathbf{x}' \in \mathcal{X} \) dominates at least \( \nu \) existing Pareto-optimal solutions. The method can be extended to \( m > 2 \).
Figure 1.4: The region of improvement partitioned according to the number of non-inferior solutions in the current non-dominated front $\nu$ that can be dominated.
CHAPTER 2

THE EXPECTED PARETO IMPROVEMENT AND EMAX ALGORITHM FOR APPROXIMATING THE PARETO FRONT

In this chapter we propose a sequential experimental design strategy to find a set of Pareto solutions that are optimal for a given black-box MOP. The expected Pareto also called the expected maximin improvement or EmaX algorithm relies on a Gaussian process emulator to guide the sequential choice of design configurations to localize the efficient points that define the Pareto boundary in objective space. The use of stochastic model surrogates make it possible to infer the location of these optimum points before resorting to the expensive simulation code.

Given a design vector \( x \in X \), we assume that the simulator returns multiple outputs \( y_1(x), y_2(x), \ldots, y_m(x) \) with \( m > 1 \). In our application, each \( y_j(\cdot) \) is an object of direct interest though in other applications, functions of the \( y_j \)'s like integrals or linear combinations may be more relevant. Particularly, we regard \( y(\cdot) = \{y_1(\cdot), \ldots, y_m(\cdot)\} \) as a set of targets to be minimized in a multiobjective optimization problem.

To approximate the Pareto front, the EmaX algorithm utilizes a sequential design strategy where experiments are performed successively in a direction of improvement. At the onset, a space-filling design \( D_n = \{x_1, \ldots, x_n\} \) is exploited to gain information
about $y(\cdot)$ throughout the design space $\mathcal{X}$. Thereafter, each stage capitalizes on the data $Y_{n,m}$ accumulated through the previous stages to identify a single design vector, $x^*$ having the greatest potential of improving upon the set of non-inferior solutions found so far, denoted by $\mathcal{P}^n \subset \mathcal{D}_n$. This most promising design configuration, $x^*$, is determined by maximizing a novel expected multivariate improvement criterion which we call the expected Pareto or expected maximin improvement function denoted by $I_M(\cdot)$. This quantitative criterion derives from the definition of Pareto dominance itself. The Pareto or maximin improvement at a potential input $x^*$ can be thought of as a distance metric between $y(x^*) \in \mathbb{R}_m$ and the current non-dominated front, $\mathcal{P}^n$, a point set in the $m$-dimensional objective space. The algorithm terminates after a fixed number of inputs have been augmented to the initial experimental design, or after a set of convergence criteria is met. The Ema$X$ algorithm follows the structure of expected improvement (EI) algorithms which can be outlined as follows:

1. Choose an initial space-filling design $\mathcal{D}_n = \{x_1, \ldots, x_n\}$.
2. Evaluate the simulator at each $x_i \in \mathcal{D}_n$ to generate the outputs $y(x_1), \ldots, y(x_n)$.
3. Estimate the correlation parameters of the Gaussian process model, $\hat{\theta}$.
4. Check the termination criterion.
5. Find the input $x_{n+1} \in \mathcal{X}$ that maximizes the expected improvement.
6. Evaluate the simulator at $x_{n+1}$ to obtain $y(x_{n+1})$.
7. Set $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{x_{n+1}\}$, increment $n = n + 1$ and go to Step 3.

We describe the Pareto improvement function in Section 2 and the Ema$X$ algorithm in Section 2.2. We then derive a closed form expression of the expected Pareto improvement function at arbitrary $x$ in Section 2.3. The derivation is made assuming a Gaussian process model for $y(\cdot)$ with known variance and cross-correlation parameters.
2.1 The *Pareto* or *Maximin* Improvement Function

In neoclassical economics, a “Pareto improvement” is an action that impairs no one and benefits at least one person. The theory proposes that Pareto improvements will continue adding to the economy until a state of Pareto equilibrium is reached where no more Pareto improvements can be made. A Pareto improvement is also seen as a change in economic management that upgrades the condition of one or more members of the community without worsening the circumstances of anyone.

The concept of a Pareto improvement is appropriate in a design optimization scenario where competing desiderata are simultaneously aspired. Any change in the design that improves the performance in one aspect without degrading the performance in others constitutes a Pareto improvement. Intuitively, efficiency can be attained via increments of Pareto improvements until we arrive at a Pareto equilibrium where all aspects are sufficiently optimized that no “free” upgrades are possible. At such point, any improvement made in one objective necessarily causes a deterioration in another objective.

In many-objective optimization, Pareto dominance is used to establish a preference relation among candidate solutions which ultimately leads to a characterization of optimality. Any non-dominated solution is Pareto optimal for a given MOP. In the last chapter, we used the *maximin fitness function* to determine the non-dominated solutions in a given set of design configurations. In this chapter we show a novel way to exploit the *maximin fitness function* to find potential non-dominated solutions.

Balling (2000) introduced and developed the *maximin fitness function* for optimization using a genetic algorithm (GA)—an adaptive heuristic search algorithm that works by generating designs iteratively using a population of artificially-producing
test solutions. A fitness function basically measures a solution’s potential for reproductive success, or optimality in short, so that those that are deemed fit or optimal are allowed to breed and mix to produce the next generation of solutions. The evolutionary process continues until a fixed number of generations is reached or when the current generation attains a pre-determined fitness threshold. Later, Balling (2003) applied the *maximin fitness function* to solve a constrained bi-objective optimization problem on land use and transportation planning for high-growth cities and metropolitan regions. The decision space consisted of 12 design inputs with continuous and qualitative attributes. One distinct feature of the *maximin fitness function* is that its formulation derives directly from the definition of Pareto dominance. To motivate our multivariate improvement function, we briefly discuss the *maximin fitness function*.

Consider an MOP with $m$ targets to be minimized $y_1, y_2, \ldots, y_m$, with the $y_j$’s properly standardized where necessary to facilitate comparisons on a common reference scale. For a pair of design vectors $x_u, x_v \in X$, recall that $x_u$ weakly dominates $x_v$ if

\[
y_j(x_v) \geq y_j(x_u) \quad \text{for all } j = 1, 2, \ldots, m
\]

\[
\Rightarrow \quad \min_{j=1,\ldots,m} (y_j(x_v) - y_j(x_u)) \geq 0 \quad (2.1)
\]

and $x_u$ dominates $x_v$ if in addition to (2.1), the following also holds:

\[
y_j(x_v) > y_j(x_u) \quad \text{for at least one } j = 1, 2, \ldots, m \quad (2.2)
\]

The only case where (2.1) is satisfied but not (2.2) is when

\[
y_j(x_v) = y_j(x_u) \quad \text{for all } j = 1, 2, \ldots, m \quad (2.3)
\]
Any two designs which satisfy Equation (2.3) are called duplicate designs even if \( x_v \neq x_u \). Any two designs which do not satisfy Equation (2.3) are called distinct designs. If all design vectors in \( D_n \) are distinct then weak domination implies domination. From here onwards we assume that the \( y_j, 1 \leq j \leq m \) have been properly standardized.

**Definition 2.1.1** Suppose \( D_n = \{x_1, \ldots, x_n\} \) is a given set of design vectors for which \( y(x_i) = (y_1(x_i), \ldots, y_m(x_i))^T \) is known. The maximin fitness function at design vector \( x_v \), is defined as

\[
\varphi(x_v) = \max_{x_i \in D_n^v} \min_{j=1,\ldots,m} (y_j(x_v) - y_j(x_i))
\]  

(2.4)

where \( D_n^v \) is the set \( D_n \) with \( x_v \) removed.

In a genetic algorithm, the maximin fitness is minimized to drive the evolution of solutions towards the Pareto optimal front.

Computing the maximin fitness is simple—the minimum is first taken over all the \( m \) objectives, followed by the maximum over all \( n-1 \) design vectors in \( D_n^v \). Therefore to obtain the fitness values for all design vectors in \( D_n \), three nested loops are needed. The inner loop index \( j \) ranges from 1 to \( m \), the middle loop indices \( i \neq v \), and the outer loop index \( v \) is from 1 to \( n \). In total, there are \( m(n-1)n \) comparisons. The maximin fitness “ranks” the design vectors within \( D_n \) in this way:

1. If \( \varphi(x_v) > 0 \) then design vector \( x_v \) is dominated by at least one member of \( D_n \).
   It is an inferior solution.

2. If \( \varphi(x_v) < 0 \) then \( x_v \) is currently non-dominated. With respect to \( D_n \), it is a non-inferior solution. This status may however change with the addition of new design vectors.
3. If $\varphi(x_v) = 0$ then $x_v$ is weakly dominated. In this case, it is either dominated or a duplicate non-dominated design.

It is evident from Equation (2.4) that $\varphi(\cdot)$ incorporates Pareto dominance in its formulation. Balling (2003) noted that for dominated designs $\varphi(x_v)$ is a metric of distance to the non-dominated front. He also observed based on numerical experiments with $m = 2$, that $\varphi(\cdot)$ penalizes clustering of non-dominated designs; moreover, $\varphi(\cdot)$ rewards designs at the middle of convex fronts and that are non-dominated and at the extremes of fronts that are concave and non-dominated.

To improve on the current best set, it suffices to consider only $P_n^x$ (or $P_n^y$), rather than the entire $D_n$ since there is no point in upgrading designs that are known to be dominated.

We end our description of maximin fitness with some comments. The apparent built-in penalty on clustering or crowding of non-dominated designs is a welcome feature, except in cases when the Pareto optimal front itself is known to be concentrated or unevenly dense in certain regions of the objective space. Barring these cases, the result is encouraging since it suggests that the use of the maximin fitness function can generate an approximation that represents the Pareto front well by an even coverage of points. An evenly-covered Pareto front does not however guarantee an evenly-covered set of Pareto optima in the design space. The mapping from design to objective space influences the extent to which evenness of representation in the one space translates to the evenness of representation in the other space. The reported “bias” for solutions at the middle of convex non-dominated fronts or at the extremes of concave ones, is a plus since these regions characterize the trade-off dynamic—where a positive increment in one objective is accompanied by a severe deterioration in another (Branke, Deb, Dierolf, and Osswald 2004). The knee or the extremes of concave
regions in the Pareto front can be considered the “compromise” region where a decision maker practically chooses his or her final solution (Branke, Deb, Dierolf, and Osswald 2004; Parreiras and Vasconcelos 2008).

In a GA, the role of the maximin fitness function is to direct the evolution of solutions towards the Pareto optimal front one generation at a time. In our sequential design strategy, we set Pareto dominance as a direction for improvement and use the maximin fitness function to guide the identification of prospective non-dominating solutions, similar to the role of an improvement function in an expected improvement algorithm. This implies that any design vector \( x^* \notin \mathcal{D}_n \) that is not dominated by any member of \( \mathcal{P}_n^\mathcal{X} \) or better still, one that dominates every member of \( \mathcal{P}_n^\mathcal{X} \) ‘improves’ the performance of all designs in \( \mathcal{D}_n \). The amount of improvement at \( x^* \) will be quantified by its maximin fitness value taken with respect to \( \mathcal{D}_n \). Here onwards, this value will be known as the Pareto improvement at \( x^* \). Formally, we define the \textit{Pareto improvement} at a potential design vector \( x \in \mathcal{X} \) as follows:

**Definition 2.1.2** Let \( \mathcal{D}_n = \{x_1, x_2, \ldots, x_n\} \) be the current experimental design with corresponding simulator outputs \( \{y(x_i) \in \mathbb{R}_m : x_i \in \mathcal{D}_n\} \). Let \( \mathcal{P}_n^\mathcal{X} = \{x_i \in \mathcal{D}_n : \varphi(x_i) < 0\} \subset \mathcal{D}_n \) be the current set of non-dominated designs. The Pareto improvement function at a prospective design vector \( x \in \mathcal{X} \), or \( I_M(x) \) is defined by

\[
I_M(x) = \max_{x_i \in \mathcal{P}_n^\mathcal{X}} \min_{j=1,\ldots,m} (y_j(x) - y_j(x_i)) \tag{2.5}
\]

We state some salient features of the \textit{Pareto improvement function} \( I_M(\cdot) \).

1. Given \( \mathcal{D}_n \) and the set of simulator outcomes \( \{y(x_i) : x_i \in \mathcal{D}_n\} \), \( I_M(x) \) is a random quantity that depends solely on the random quantity \( Y(x) \), on account of \( x \) being unevaluated. Thus the conditional distribution, \( [I_M(x)|\mathcal{D}_n, Y_{n,m}] \) also depends on the distribution \( [Y(x)|Y_{n,m}] \).
2. According to Equation [2.5], the improvement of \( \mathbf{x} \) over an existing design vector \( \mathbf{x}_i \) equals the best upgrade value or the most negative difference \( Y_j(\mathbf{x}) - Y_j(\mathbf{x}_i) \) among all the \( m \) objectives. The inner min operator thus induces a pairwise comparison between \( \mathbf{x} \) and \( \mathbf{x}_i \), a point-to-point comparison so to speak. On the other hand, the improvement of \( \mathbf{x} \) over the set of designs \( \mathcal{P}_X^n \) equals the the worst among the best pairwise gains. Hence the outer max operator is a the point-to-set comparison.

3. The quantity \( I_M(\mathbf{x}) \) measures the Pareto improvement of \( \mathbf{x} \) over \( \mathcal{P}_X^n \). The improvement equals the maximin fitness of \( \mathbf{x} \) relative to the design vectors in \( \mathcal{P}_X^n \). Consequently, if \( I_M(\mathbf{x}) > 0 \), then \( \mathbf{x} \) is a dominated solution. Therefore \( \mathbf{x} \) is inferior to the solutions in \( \mathcal{P}_X^n \). If \( I_M(\mathbf{x}) = 0 \), then \( \mathbf{x} \) is a weakly dominated point. It is either a dominated design or a duplicate non-dominated design. Either way, \( \mathbf{x} \) does not decisively improve \( \mathcal{P}_X^n \). If \( I_M(\mathbf{x}) < 0 \), then \( \mathbf{x} \) is a non-dominated solution which decisively improves the best designs in \( \mathcal{P}_X^n \).

4. If \( \mathbf{x}^* \) is indifferent to every member of \( \mathcal{P}_X \), that is,

\[
\mathbf{x}^* \sim \mathbf{x}_i, \quad \forall \mathbf{x}_i \in \mathcal{P}_X^n
\]

then \( \mathbf{x}^* \) augments \( \mathcal{P}_X^n \). If \( \mathbf{x}^* \) dominates any solution in \( \mathcal{P}_X^n \), or

\[
\mathbf{x}^* \succ \mathbf{x}_i, \quad \text{for at least one } \mathbf{x}_i \in \mathcal{P}_X^n
\]

then \( \mathbf{x}^* \) knocks off the dominated design from \( \mathcal{P}_X^n \) and replaces it. Both events cause a refinement in the shape of \( \mathcal{P}_X^n \) in objective space. Thus identifying such designs in \( \mathcal{X} \) is the key to sharpening our approximation of the Pareto front.
5. In view of the preceding remark, maximizing the Pareto improvement constrains the search for design configurations $x^*$ whose objective vectors, $y(x^*)$ live in the non-dominated region or dominance zone of $P^n_X$. Intuitively, we expect this improvement criterion to eventually drive the approximation towards the true Pareto front. Since an improvement occurs when $I_M(x) < 0$, the term maximizing the Pareto improvement actually means minimizing $I_M(x)$.

As (1) explains, $I_M(x)$ is random; therefore we assess the amount of improvement possible at $x \notin D_n$ by the expected Pareto improvement where the expectation is taken with respect to the conditional distribution of $y(x)$ given the data $Y_{n,m}$ and “known” GP parameters.

Starting with an initial space-filling design, the EmaX algorithm updates the current experimental design $D_n$ by choosing the next point $x_{n+1}$ satisfying

$$x_{n+1} = \arg\max_{x \in X} E[-I_M(x)|Y_{n,m}, \psi]$$ (2.6)

which is our multivariate improvement function. Equation (2.6) is implemented as a mathematical program and its solution, $x_{n+1}$ is augmented to $D_n$. In the next section we derive an analytic expression of the expected Pareto improvement at a potential design vector $x$ when $m = 2$ assuming a GP model for $y(x)$ with $K(\cdot; \Sigma_0, \psi)$ completely specified.

### 2.2 The EmaX Algorithm

The EmaX algorithm is a GP-guided expected improvement algorithm that approximates the Pareto Front of a black box MOP. It is a fully sequential optimization design that draws on Pareto dominance to define a direction of improvement in a problem with multiple objectives and sets the maximin fitness as its improvement
criterion. At each stage, a new design configuration is chosen so that it maximizes
the expected Pareto improvement defined in Equation (2.6). The \textit{EmaX} algorithm
is outlined as follows:

1. Choose an initial design with a space-filling property, such as a maximin Latin
   hypercube sample $\mathcal{D}_0 = \{x_1, \ldots, x_{n_0}\}$. Set $n = n_0$.

2. Identify $\mathcal{P}_x^n$, the current non-dominated set of design vectors in $\mathcal{D}_n$ using the
   maximin fitness function $\varphi(x_i), x_i \in \mathcal{D}_n$.

3. Estimate the covariance parameter vector $\psi$ via maximum likelihood, restricted
   maximum likelihood, or by maximizing the posterior density $[\psi | \text{data}]$ in a
   Bayesian modeling approach. Let $\hat{\psi}$ denote this estimate which will be plugged
   into the GP-emulator of $y(\cdot)$.

4. Choose $x_{n+1}$ as the maximizer of the \textit{expected Pareto improvement} given the
   existing data, that is,

   $\begin{equation}
   x_{n+1} = \arg \max_{x \in \mathcal{X}} E\{ -I_M(x) \mid \text{data}, \hat{\psi} \} \tag{2.7}
   \end{equation}$

   where $E\{ \cdot \mid \text{data}, \hat{\psi} \}$ denotes the expectation taken with respect to the con-
   ditional distribution of $y(x)$ given the data and the plug-in estimate $\psi$. This
   conditional distribution is the GP emulator.

5. Check if the stopping criterion is met. If not, set $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{x_{n+1}\}$. Submit
   the augmented point to the simulator to obtain $y(x_{n+1})$. Increment $n$ to $n + 1,$
   and go to Step 2. If the stopping criterion is met, say at $n = N$, then the
   final non-dominated set $\mathcal{P}_y^N$ is our approximation to the Pareto front, and its
   pre-image $\mathcal{P}_x^N$, our approximation to the Pareto set in $\mathcal{X}$.
We now give the details of our implementation of the *EmaX* algorithm. We chose a maximin Latin hypercube design (McKay, Beckman, and Conover 1979; Johnson, Moore, and Ylvisaker 1990) for our initial design. The space-filling property is achieved by configuring *n* points in *d*-dimensional $\mathbf{X}$ so that no two points share a coordinate and that the smallest distance between all pairs of points is maximized. The maximin LHD is called as a built-in function (*lhsdesign.m*) in *Matlab*.

The *EmaX* algorithm was envisioned to terminate after a certain number of design configurations have been added, particularly two times the initial sample size. Two possible choices for the size of the initial design were explored – $5d$ and $10d$. The smaller size attempts to simulate a very tight computational budget where the maximum number of runs is capped at $10d$. The bigger sample size implements a finding by Loepky, J., and Welch (2008) whose simulation study found support for the viability of the informal rule of $10d$ as an initial size for computer experiments.

A user-contributed *Matlab* function file, *paretoset.m* due to Cao (2007) was used to identify the set of non-dominated design vectors in a given experimental design. The file was obtained online from Matlab Central, an open source for Matlab and Simulink users.

In the GP emulator, a constant mean-vector was assumed and for the correlation type, a separable structure was chosen from either the cubic or Gaussian families. The family which led to more numerically stable estimates for a given test function was chosen. Numerical stability was monitored by the frequency of ill-conditioning warnings output by *Matlab*.

The estimation of the correlation parameter vector was made via restricted maximum likelihood (REML) in the case of the (non-separable) IGP models, and by maximization of the posterior density in the Conti and O’Hagan separable dependence model. The *Matlab* version (Han and Santner 2005) of the PerK program (Williams
2001) was used to find the REML plug-in estimates while a Matlab code was written to find the posterior mode in the Bayesian model using the built-in function fmincon.m from Matlab’s Optimization toolbox. The function fmincon.m finds the minimum of a constrained nonlinear multivariable function. The same fmincon.m was used to maximize two improvement criteria – the probability of improvement and the Pareto improvement.

2.3 The Expected Pareto Improvement when \( m = 2 \)

In this section we derive the analytic formula of the expected Pareto improvement function when \( m = 2 \), assuming a completely specified Gaussian process model \( Y(x), x \in \mathcal{X} \) for the simulator \( y(\cdot) \).

The Pareto improvement at an unevaluated design \( x_0 \in \mathcal{X} \) was defined in Equation (2.5) as

\[
I_M(x_0) = \max_{x_i \in P_n^\mathcal{X}} \min_{j=1,...,m} (Y_j(x_0) - Y_j(x_i))
\] (2.8)

where \( Y(x_0) = (Y_1(x_0), \ldots, Y_m(x_0)) \) and \( P_n^\mathcal{X} \) is the set of non-dominated solutions identified from a set of design vectors \( \mathcal{D}_n = \{x_1, \ldots, x_n\} \) by computing the maximin fitness function \( \varphi(x_i) \) from the standardized outputs of the simulator,

\[
Y(x_1) = y(x_1), \ldots, Y(x_n) = y(x_n)
\]

We first consider the simple problem of determining the expected \( I_M(x_0) \) when there are currently two points in the frontier or equivalently, \( P_n^\mathcal{X} = \{x_1, x_2\} \). Later, we extend this expression to the general case of two or more non-inferior designs.

Recall that we postulated a model for \( Y(x) \) as follows

\[
Y(x) = B^T f(x) + W(x)
\] (2.9)
where

• \( B = (\beta_1, \ldots, \beta_m) \in \mathbb{R}_{p,m} \) is a matrix of known regression coefficients with each \( \beta_j = (\beta_{j1}, \ldots, \beta_{jp}) \in \mathbb{R}_p \), \( p \geq 1 \)

• \( f(\cdot) = (f_1(\cdot), \ldots, f_p(\cdot)) \) vector of known regression functions common to all \( Y_j(\cdot), j = 1, \ldots, m \)

• \( W(x) \) is a stationary, zero mean vector, \( m \)-variate Gaussian process with covariance function \( K(\cdot; \Sigma_0, \psi) \)

Suppose that for an arbitrary collection of \( n + 1 \) design configurations,

\[ \{x_0, x_1, \ldots, x_n\} \subset \mathcal{X}, \quad n \geq 1 \]

we have

\[ (Y(x_0) = y(x_0), Y(x_1) = y(x_1), \ldots, Y(x_n = y(x_n))) \]

Define

\[
Y_{(n+1), m} = \begin{pmatrix}
Y^T(x_0) \\
Y^T(x_1) \\
\vdots \\
Y^T(x_n)
\end{pmatrix} = \begin{pmatrix}
Y_1(x_0) & Y_2(x_0) & \cdots & Y_m(x_0) \\
Y_1(x_1) & Y_2(x_1) & \cdots & Y_m(x_1) \\
\vdots & \vdots & \ddots & \vdots \\
Y_1(x_n) & Y_2(x_n) & \cdots & Y_m(x_n)
\end{pmatrix} \in \mathbb{R}_{n+1,m} \quad (2.10)
\]
It follows from our specification for $\mathbf{Y}(\mathbf{x})$, $\mathbf{x} \in \mathcal{X}$ in (2.9) that the joint distribution of $(\mathbf{Y}_0, \mathbf{Y}_{n,m})$ is multivariate normal, that is,

$$
\begin{pmatrix}
\mathbf{Y}(\mathbf{x}_0) \\
\mathbf{Y}(\mathbf{x}_1) \\
\vdots \\
\mathbf{Y}(\mathbf{x}_n)
\end{pmatrix}
\sim MVN_{m(n+1)} \left( (\mathbf{I}_{n+1} \otimes \mathbf{B}^T) \mathbf{F}_{n+1}, \mathbf{\Sigma}_{\mathbf{Y}_{n+1,m}} \right)
$$

(2.11)

where

$$
\mathbf{F}_{n+1} = (f^T(\mathbf{x}_1), \ldots, f^T(\mathbf{x}_n))^T \in \mathbb{R}_{np}
$$

$$
\mathbf{\Sigma}_{\mathbf{Y}_{n+1,m}} = 
\begin{pmatrix}
\mathbf{\Sigma}_0 & \mathbf{\kappa}_0^T \\
\mathbf{\kappa}_0 & \mathbf{\Sigma}_{\mathbf{Y}_{n,m}}
\end{pmatrix}
$$

(2.12)

$$
\mathbf{\Sigma}_{\mathbf{Y}_{n,m}} = 
\begin{pmatrix}
\mathbf{\Sigma}_0 & \mathbf{K}(\mathbf{x}_1 - \mathbf{x}_2) & \mathbf{K}(\mathbf{x}_1 - \mathbf{x}_n) \\
\mathbf{K}(\mathbf{x}_2 - \mathbf{x}_1) & \mathbf{\Sigma}_0 & \mathbf{K}(\mathbf{x}_2 - \mathbf{x}_n) \\
\vdots & \vdots & \vdots \\
\mathbf{K}(\mathbf{x}_n - \mathbf{x}_1) & \mathbf{K}(\mathbf{x}_n - \mathbf{x}_2) & \mathbf{\Sigma}_0
\end{pmatrix}
$$

(2.13)

$$
\mathbf{\kappa}_0^T = (\mathbf{K}(\mathbf{x}_0 - \mathbf{x}_1) \ldots \mathbf{K}(\mathbf{x}_0 - \mathbf{x}_n)) \in \mathbb{R}_{m,mn}
$$

(2.14)

Then it follows from the property of multivariate normality that the conditional distribution is also multivariate normal, that is,

$$
\mathbf{Y}_0|\mathbf{Y}_{n,m} \sim MVN \left( \hat{\mathbf{y}}(\mathbf{x}_0), \mathbf{\Sigma}_0 \mathbf{Y}_{n,m} \right)
$$

(2.15)
where
\[
\hat{y}(x_0) = B^T f(x_0) + \kappa_0^T \Sigma_{Y_{n,m}}^{-1} \left( \text{vec} Y_n^T - (I_n \otimes B^T F_n) \right) \tag{2.16}
\]
\[
\Sigma_{0|Y_{n,m}} = \Sigma_0 - \kappa_0^T \Sigma_{Y_{n,m}}^{-1} \kappa_0 \tag{2.17}
\]
\[
= \begin{pmatrix}
  s_2^2 & s_{12} \\
  s_{12} & s_2^2
\end{pmatrix}
\]

The expected Pareto improvement at a new design point \(x_0\) has the expression,
\[
E (I_M(x_0)|Y_{n,m}) = \int_{\mathbb{R}_n} I_M(x_0) \phi_m(y_0) dy_0 \tag{2.18}
\]

where \(\phi_m(\cdot)\) is the multivariate normal density with mean and variance parameters given in (2.16) and (2.17) respectively. In the expectation given in Equation (2.18), the simulator outputs \(Y(x_i) = y(x_i)\) for \(x_i \in D_n\) are constants with respect to the conditional distribution \([Y(x_0)|Y_{n,m}]\). Let \(y_i^j \equiv y_2(x_i)\) and re-write the Pareto improvement of \(x_0\) over an existing design \(x_i \in D_n\) as
\[
w_i = \min_{j=1,\ldots,m} (Y_j(x_0) - y_i^j) \tag{2.19}
\]

Therefore, when \(m = 2\) we have
\[
w_i = \min ( (Y_1(x_0) - y_1^i), (Y_2(x_0) - y_2^i) ) .
\]

Assuming that \(P^a_n = \{x_1, x_2\}\), the Pareto improvement of \(x_0\) over the set \(P^a_n\) equals
\[
I_M(x_0) = \max (w_1, w_2) . \tag{2.20}
\]
We derive the expected Pareto improvement by considering a partition of the biobjective space into regions $R_t$ with $t = 1, \ldots, T$, where we can conveniently determine the $w_1$ and $w_2$, and consequently $\max(w_1, w_2)$. Then

$$E \left( I_M(x_0) \mid Y_{n,m} \right) = \sum_{t=1}^{T} \int_{R_t} \max(w_1, w_2) \phi_m(y_0) dy_0$$

(2.21)

In the partition, the points $y(x_1) = (y_1^1, y_1^2)$ and $y(x_2) = (y_2^1, y_2^2)$ for $x_1, x_2 \in P_X$ determine the boundaries of the regions $R_t$ and also the number of regions $T > 0$. For greater clarity, let $y_1^i = a_i$ and $y_2^i = b_i$, so $y(x_1) = (a_1, b_1)$ and $y(x_2) = (a_2, b_2)$. Since these two points do not dominate each other, only one of the following cases must be true

$$a_1 \geq a_2 \quad \text{and} \quad b_1 \leq b_2 \quad \text{or}$$

$$a_1 \leq a_2 \quad \text{and} \quad b_1 \geq b_2$$

(2.22)

To determine $w_1$ and $w_2$, consider the parallel lines $L_i$ induced by the points $(a_i, b_i)$, $i = 1, 2$, which divide the support of $Y(x_0)$ into three mutually exclusive and exhaustive regions, $R_1, R_2$, and $R_3$. The region $R_1$ is bounded by $L_1$, $R_2$ is the region between the lines $L_1$ and $L_2$, while $R_3$ is the region above $L_2$. Figure 2.1 illustrates the idea. The lines $L_i$ have a common slope unity but individual intercepts equal to $a_i - b_i$ or $b_i - a_i$, depending upon whether we take the line $l_i$ or $L_i$. The line $l_i$ expresses $Y_1(x_0)$ as a function of $Y_2(x_0)$ whereas the line $L_i$ writes $Y_2(x_0)$ in terms of $Y_1(x_0)$. Either
way, the same partition of $\mathbb{R}_2$ is generated.

\[
Y_1(x_0) - a_i = Y_2(x_0) - b_i
\]

\[\Rightarrow l_i : Y_1(x_0) = Y_2(x_0) + a_i - b_i \text{ or } (2.23)
\]

\[
L_i : Y_2(x_0) = Y_1(x_0) + b_i - a_i
\]

Having defined the regions, we can now determine $w_1$ and $w_2$ as follows:

Figure 2.1: Partitioning of $\mathbb{R}_2$ by two non-dominated points
In the region $R_2$, the maximum between $(Y_1(x_0) - a_1)$ and $(Y_2(x_0) - b_2)$ is sought.

To obtain this, another line is constructed

$$\text{Case I } l_3 : \ Y_1(x_0) = Y_2(x_0) + a_1 - b_2$$

or

$$\text{Case II } L_3 : \ Y_2(x_0) = Y_1(x_0) + b_2 - a_1.$$  

On account of (2.22), this line must lie within $R_2$, that is,

$$a_2 - b_2 < a_1 - b_2 < a_1 - b_1$$

or

$$b_1 - a_1 < b_2 - a_1 < b_2 - a_2.$$  

Consequently, within the region $R_2$ two subregions are created: the first, $R_{21}$, is bounded by the lines $L_1$ and $L_3$, and the second, $R_{22}$ is bounded by the lines $L_3$ and $L_2$. Having done this, the maximum of $(w_1, w_2)$ can now be resolved.
The expected Pareto improvement at new input site \( x_0 \) can now be formulated as

\[
E[I_M(x_0)|Y_{n,m}] = \int_{R_1} (Y_2(x_0) - \min(b_1, b_2)) \phi(Y_0) dY_0 \\
+ \int_{R_2} (Y_1(x_0) - a_1) \phi(Y_0) dY_0 \\
+ \int_{R_2} (Y_2(x_0) - b_2) \phi(Y_0) dY_0 \\
+ \int_{R_3} (Y_1(x_0) - \min(a_1, a_2)) \phi(Y_0) dY_0
\]

(2.24)

where \( \phi(\cdot) \) is the bivariate normal density and \( Y_0 = (Y_1(x_0), Y_2(x_0)) \). The closed form of Equation (2.24) is found by applying a theorem due to Sharples and Pezzey (2007).

**Theorem 2.3.1** Suppose \( Y \sim MVN_m(0, \Sigma) \) such that \( |\Sigma| > 0 \). For \( \alpha \in \mathbb{R}_m \) with \( \| \alpha \| = 1 \) and \( \alpha \neq 0 \), and some \( c \in \mathbb{R} \), define

\[
R_{c}^{\alpha} = \{ Y \in \mathbb{R}_m : \alpha^T Y \geq c \}
\]

(2.25)

Let \( \gamma \in \mathbb{R}_m \) be a fixed arbitrary vector. Then

\[
\int_{R_{c}^{\alpha}} (\gamma^T Y) \phi_m dY = \frac{\alpha^T \Sigma \gamma}{\sqrt{2\pi \sigma^2}} \times \exp \left( \frac{-c^2}{2\sigma^2} \right)
\]

(2.26)

where \( \sigma^2 = \alpha^T \Sigma \alpha \).

To evaluate the expected Pareto improvement, the integrals in Equation (2.24) are re-expressed to accommodate the forms in the theorem by an appropriate choice of \( \alpha \in \mathbb{R}_2 \) and \( c \in \mathbb{R} \). The vector \( Y(x_0) \) first needs to be translated to have zero mean.
Let

\[ Y_z = Y(x_0) - \hat{y}(x_0) \]

\[ = (Y_1(x_0) - \hat{y}_1(x_0), Y_2(x_0) - \hat{y}_2(x_0))^T \]

\[ = (Y_{z1}, Y_{z2}) \]

where \( \hat{y}(x_0) = (\hat{y}_1(x_0), \hat{y}_2(x_0))^T \) was given in (2.16).

We work out the first integral in Equation (2.24) by giving specific values for \( \alpha, c, \) and \( \gamma \). The first integral is equal to

\[
\int_{R_1} (Y_2(x_0) - \min(b_1, b_2)) \phi(Y_0) dY_0.
\]

The region \( R_1 \) is the set

\[ \{(Y_1, Y_2) \in \mathbb{R}_2 : Y_1 - Y_2 \geq a_1 - b_1 \}. \]

Consider the following:

\[
\alpha = \frac{1}{\sqrt{2}} \begin{pmatrix}
1 \\
1 \\
-1
\end{pmatrix}
\]

(2.27)

and

\[
c = \frac{1}{\sqrt{2}}(1 - 1) \begin{pmatrix}
a_1 - \hat{y}_1(x_0) \\
b_1 - \hat{y}_2(x_0)
\end{pmatrix}
\]

(2.28)

Since the integrand is \( Y_2(x_0) - \min(b_1, b_2) \), then \( \gamma = (0 \ 1)^T \). In terms of the translated variables \( Y_z = (Y_{z1}, Y_{z2}) \) the region \( R_1 \) can be expressed as

\[
R_1 \equiv R_c^\alpha = \{(Y_{z1}, Y_{z2}) \in \mathbb{R}_2 : \alpha^T Y_z \geq c. \}
\]

(2.29)
According to the theorem, the first integral in Equation (2.24) is

\[
\int_{R_1} (Y_2(x_0) - \min(b_1, b_2)) \phi(Y_0) dY_0 = \exp \left( - \frac{(a_1 - \hat{y}_1(x_0)) - [b_1 - \hat{y}_2(x_0)])^2}{s_1^2} \right) \\
\times \frac{s_{12} - s_1^2}{\sqrt{2\pi s_1^2}} \\
- \min(b_1, b_2) \times P(R_1) \tag{2.30}
\]

where the s terms are the entries of the 2 \times 2 conditional variance matrix \( \Sigma_0|Y_{n,m} \)

specified in (2.17) and \( s_1^2 = s_1^2 - 2s_{12} + s_2^2 \), and \( P(A) \) is notation for the probability

of the event \( A \).

The same exercise of finding appropriate values of \( \alpha \in \mathbb{R}_m, c \in \mathbb{R}, \) and \( \gamma \in \mathbb{R}_m \)

is applied to the rest of the integrals in Equation (2.24). This yields the closed form

expression for the expected Pareto improvement at design \( x_0 \):

\[
E[I_M(x_0)|Y_{n,m}] = \frac{s_{12} - s_2^2}{\sqrt{2\pi s_2^2}} \exp \left\{ -\frac{1}{2} \left( \frac{[a_1 - \hat{y}_1(x_0)] - [b_1 - \hat{y}_2(x_0)])^2}{s_1^2} \right) \right\} \\
- \min(b_1, b_2) P(R_1) \\
+ \frac{s_1^2 - s_{12}}{\sqrt{2\pi s_1^2}} \left( \exp \left\{ -\frac{1}{2} \left( \frac{[a_1 - \hat{y}_1(x_0)] - [b_2 - \hat{y}_2(x_0)])^2}{s_1^2} \right) \right\} \\
- \exp \left\{ -\frac{1}{2} \left( \frac{[a_1 - \hat{y}_1(x_0)] - [b_1 - \hat{y}_2(x_0)])^2}{s_2^2} \right) \right\} \right\} - a_1 P(R_{21}) \\
+ \frac{s_{12} - s_2^2}{\sqrt{2\pi s_2^2}} \left( \exp \left\{ -\frac{1}{2} \left( \frac{[a_2 - \hat{y}_1(x_0)] - [b_2 - \hat{y}_2(x_0)])^2}{s_2^2} \right) \right\} \\
- \exp \left\{ -\frac{1}{2} \left( \frac{[a_2 - \hat{y}_1(x_0)] - [b_2 - \hat{y}_2(x_0)])^2}{s_2^2} \right) \right\} \right\} - a_2 P(R_{22}) \\
+ \frac{s_{12} - s_1^2}{\sqrt{2\pi s_1^2}} \exp \left\{ -\frac{1}{2} \left( \frac{[b_2 - \hat{y}_2(x_0)] - [a_2 - \hat{y}_1(x_0)])^2}{s_2^2} \right) \right\} \\
- \min(a_1, a_2) P(R_{3})
\]

94
Expressions for \( P(R_t) \), \( t \in \{1, 21, 22, 3\} \) are available and are given in Equations (2.31)- (2.34):

\[
P(R_1) = P(Y(x_0) : Y_2(x_0) - b_1 < Y_1(x_0) - a_1) = P(Y_2(x_0) - Y_1(x_0) < b_1 - a_1) = \Phi \left( \frac{[b_1 - a_1] - [\bar{y}_2(x_0) - \bar{y}_1(x_0)]}{s^*_2} \right)
\]

\[
P(R_{21}) = P(Y(x_0) : Y_2(x_0) - b_1 < Y_1(x_0) - a_1 < Y_2(x_0) - b_2) = P(b_2 - a_1 < Y_2(x_0) - Y_1(x_0) < b_1 - a_1) = \Phi \left( \frac{[b_1 - a_1] - [\bar{y}_2(x_0) - \bar{y}_1(x_0)]}{s^*_2} \right) - \Phi \left( \frac{[b_2 - a_2] - [\bar{y}_2(x_0) - \bar{y}_1(x_0)]}{s^*_2} \right)
\]

\[
P(R_{22}) = P(Y(x_0) : Y_1(x_0) - a_1 < Y_2(x_0) - b_2 < Y_1(x_0) - a_2) = \Phi \left( \frac{[b_2 - a_2] - [\bar{y}_2(x_0) - \bar{y}_1(x_0)]}{s^*_2} \right) - \Phi \left( \frac{[b_1 - a_1] - [\bar{y}_2(x_0) - \bar{y}_1(x_0)]}{s^*_2} \right)
\]

\[
P(R_3) = P(Y(x_0) : Y_2(x_0) - b_2 > Y_1(x_0) - a_2) = P(Y_2(x_0) - Y_1(x_0) > b_2 - a_2) = 1 - \Phi \left( \frac{[b_2 - a_2] - [\bar{y}_2(x_0) - \bar{y}(x_0)]}{s^*_2} \right)
\]

where \( s^*_2 = s_2^2 - 2s_{12} + s_1^2 \), \( \Phi(\cdot) \) is the cumulative distribution function of a univariate standard normal distribution. We now provide an expression for the expected Pareto improvement, \( E(I_M(x_0)|Y_{n,m}) \) of a design point \( x_0 \notin D_n \) over the current non-dominated set of designs \( \mathcal{P}_{X}^{n} = \{x_1, \ldots, x_{n_{pf}}\} \) with \( 2 \leq n_{pf} < n \). Let the corresponding non-dominated front

\[
\mathcal{P}_{Y}^{n} = \{(a_1, b_1), \ldots, (a_{n_{pf}}, b_{n_{pf}})\}.
\]

Since the \((a_i, b_i), 1 \leq i \leq n_{pf}\) are mutually indifferent, then

\[
a_1 < a_2 < \ldots < a_{n_{pf}}
\]
implies that
\[ b_{n_{pf}} < b_{n_{pf}-1} < \ldots < b_1. \]

This yields
\[ b_1 - a_1 > b_2 - a_2 > \ldots > b_{n_{pf} - 1} - a_{n_{pf} - 1} > b_{n_{pf}} - a_{n_{pf}}. \] (2.35)

Define \( e_i = b_i - a_i \), and \( e_{[i]} \) be the \( i^{th} \) order statistics of the set \( \{e_1, \ldots, e_{n_{pf}}\} \) for \( 1 \leq i \leq n_{pf} \). From (2.35), we have the relation
\[ e_{[i]} = b_{n_{pf} - i + 1} - a_{n_{pf} - i + 1}. \] (2.36)

Let
\[ a_{[i]} \equiv e_{[i]}(a) \equiv a_{n_{pf} - i + 1} \] (2.37)
\[ b_{[i]} \equiv e_{[i]}(b) \equiv b_{n_{pf} - i + 1} \]

The current non-dominated front \( \mathcal{P}_Y^n \) can therefore be written as
\[ \mathcal{P}_Y^n = \{(a_{[1]}, b_{[1]}), \ldots, (a_{[n_{pf}]}, b_{[n_{pf}]})\}. \] (2.38)

For purposes of determining the
\[ w_i = \min \{ (Y_1(x_0) - a_i), (Y_2(x_0) - b_i) \}, \ 1 \leq i \leq n_{pf}, \]

we can regard the \( n_{pf} \) points of \( \mathcal{P}_Y^n \) as inducing a partition which consists of \( 2n_{pf} \) regions in the support of \( Y(x_0) = (Y_1(x_0), Y_2(x_0)) \). These regions are bounded by
the lines
\[ Y_2(x_0) = Y_1(x_0) + e_{[i]}, \quad 1 \leq i \leq n_{pf} \]
and
\[ Y_2(x_0) = Y_1(x_0) + e_{[i+1]}(b) - e_{[i]}(a), \quad 2 \leq i \leq n_{pf} - 1. \]
Suppose the regions are denoted by
\[ R_t, \quad t \in \{1, 21, 22, 31, 32, \ldots, n_{pf}1, n_{pf}2, n_{pf}, n_{pf} + 1\} \]
where
\[
\begin{align*}
R_1 & = \{ Y(x_0) : Y_2(x_0) - b[1] < Y_1(x_0) - a[1] \} \\
R_{q1} & = \{ Y(x_0) : Y_1(x_0) - a[q] < Y_2 - b[q] < Y_1(x_0) - a[q+1] \} \\
R_{q2} & = \{ Y(x_0) : Y_2(x_0) - b[q] < Y_1 - a[q+1] < Y_2(x_0) - b[q+1] \} \\
R_{n_{pf}+1} & = \{ Y(x_0) : Y_2(x_0) - b[n_{pf}] > Y_1(x_0) - a[n_{pf}] \}
\end{align*}
\]
for \( 2 \leq q \leq n_{pf} - 1 \). Applying the same strategy used for deriving the expected improvement when \( n_{pf} = 2 \), we obtain a general expression for \( n_{pf} \geq 2 \) which is stated in Theorem 2.3.2.

**Theorem 2.3.2** Let \( D_n = \{ x_1, \ldots, x_n \} \) be a set of \( n \) design configurations and \( P^n_X \subset D_n \) the set of non-dominated designs. Let the image of \( P^n_X \) be given by Equation (2.38). Under model (2.9), the expected Pareto improvement of an unseen design...
configuration \( x_0 \notin D_n \) over the current non-dominated set \( P^n_X \) is given by:

\[
E(I_M(x_0)) = \int_{R^2} I_M(x_0) \phi_2(y_0) dy_0
\]

where
\[
s^2 = s^2_1 + s^2_2 - 2s_{12}
\]

and
\[
\iota(a, b) = \exp \left\{ -\frac{1}{2s^2} [(a - \hat{y}_1(x_0)) - (b - \hat{y}_2(x_0))]^2 \right\}
\]
CHAPTER 3

NUMERICAL TEST RESULTS

Due to the complexity of the solution set of black box MOP’s, the development of efficient numerical methods is of special interest. In this section we compare the performance of three Pareto optimizers – the \textit{IGP-PI}, \textit{IGP-EmaX}, and \textit{CoH-EmaX}. All three algorithms make use of Gaussian process emulators to approximate the Pareto front by sequentially sampling the design space according to an expected improvement criterion in the tradition of the EGO algorithm.

A Pareto optimal set of solutions are all non-dominated with respect to each other. The movement from one Pareto point to another always involves a sacrifice in one objective in exchange for a gain in another. A decision maker presented with the Pareto front or “trade-off curve” can choose a solution that resonates best with his or her preferences.

For continuous problems, the Pareto front is a curve or surface and cannot be built in a finite number of iterations. We thus have to live with a finite approximation of the curve that is “good” in some sense.

The underlying goal in most black box MOP algorithms is to compute efficiently, and provide the decision maker with, a small set of solutions that captures, as well as possible, the whole range of choices. Intuitively, a kind of minimal sufficiency characterizes our idea of a good approximation. We attempt to measure this construct...
using three indicators: the number of solutions identified, a measure of the dominated area, and a measure of the closeness to the best Pareto front approximation obtained by a grid approximation. The last two metrics indirectly measure the quality of coverage of the Pareto front and we give reasons for their choice in the section to follow.

In practice, no property of a black box function is available, so rigorous tests for convergence such as the Kuhn-Tucker conditions are not pursued.

3.1 Quality Assessment of the Pareto Front Approximation

A well-approximated Pareto front provides a decision-maker with trade-off information which represents the range of reasonable optimal choices in the design space; the points in the trade-off curve are precisely the optimal solutions for all possible utility functions that depend monotonically on the $y_k(\cdot)$’s.

The output of a multiobjective algorithm is a pair of finite sets. The first is a set of non-dominated design configurations,

$$
P_N^X = \{x_1, x_2, \ldots, x_N\} \subset X,
$$

with the property that every $x \in P_N^X$ satisfies $x \sim x_j$ for all $x_j \in P_N^X$ and $x \neq x_j$.

The second is the set

$$
P_N^Y = \{y(x) : x \in P_N^X\} \subset Y,
$$

the set of objective vectors associated with those design configurations in $P_N^X$.

The sets $P_N^X$ and $P_N^Y$ are approximations of the true Pareto set $P_X$ and its image the Pareto front $P_Y$, respectively and $N$ is the number of solutions in the $P_N^X$ (or $P_N^Y$).
In this section, we discuss two indicators for assessing the quality of the approximation of the Pareto front. For ease of presentation later, we introduce the notation \( y^i \equiv y(x_i) \in \mathcal{Y} \). In writing \( y^i \) we assume that there is a design vector \( x_i \in \mathcal{X} \) such that \( y^i = y(x_i) = (y_1(x_i), \ldots, y_m(x_i)) \). We now formalize the notion of an approximation set.

**Definition 3.1.1** A finite point set \( A \subset \mathcal{Y} \) is called an approximation set of the Pareto front \( \mathcal{P}_Y \) of a given MOP if every distinct pair of objective vectors \( y^i, y^j \in A, i \neq j \) satisfies \( y^i \sim y^j \), for \( i, j = 1, \ldots, n_A \) where \( n_A \) is the number of solutions in \( A \).

An approximation set is therefore a set of non-dominated solutions which are mutually indifferent. Bearing in mind the possibility of duplicate designs, that is, two distinct design configurations in \( \mathcal{P}^N_X \) producing identical objective vectors, the number of unique points in \( \mathcal{P}^N_Y \) will be less than or equal to number of unique points in \( \mathcal{P}^N_X \).

Here, we generally consider approximations to the Pareto front, that is the image of the Pareto optimal set in \( \mathcal{X} \) but the definition can be applied to the approximation of the Pareto optimal set as well. In doing so, we take the point of view of a decision maker who is asked to select his or her preferred solution based on a comparison of the function values in \( \mathcal{Y} \) rather than in \( \mathcal{X} \). However, there also exist problems which call for the approximation of the Pareto optimal set in \( \mathcal{X} \) and not of the Pareto front \( \mathcal{P}_Y \). Nevertheless, we concentrate on the quality of approximation of \( \mathcal{P}_Y \).

By definition, approximation sets do not need to contain the Pareto efficient points, that is the objective vectors on \( \mathcal{P}_Y \). Non-efficient points are often generated by heuristic methods such as the three algorithms that are compared here. In fact, the distance between the approximation points and \( \mathcal{P}_Y \) is proposed as a quality criterion.
Like the original MOP, comparing the quality of approximation sets from different multiobjective algorithms or optimizers is not straightforward. The many-sidedness of quality includes the extent of coverage of the Pareto front, the number of solutions or cardinality of the approximating set, and the diversity of the design configurations represented, just to name the most common criteria in the literature.

Recall in Subsection 1.5.2 that Pareto dominance was defined to establish an ordering among the designs $x \in \mathcal{X}$ with respect to $y(x) = (y_1(x), \ldots, y_m(x))$. Relations between $x_1$ and $x_2$ were then described in terms of strict and weak dominance and indifference providing, in effect, a characterization of optimality. Given the many-sidedness of quality, some recent works (Zitzler, Thiele, Laumanns, Fonseca, and da Fonseca 2003; Paquete 2005; Knowles, Thiele, and Zitzler 2006) have applied the same concept in evaluating approximation sets to resolve which algorithms were better or Pareto optimal.

Table 3.1 presents a way of broadening the notion of Pareto dominance from the objective vectors to approximation sets.

Figure 3.1 illustrates these ideas with three approximation sets $A$, $B$, and $C$. The true Pareto Front, labeled $P$, is three-point curve in the southwest portion of the graph. The following relations hold:

1. $A$ strictly dominates $C$. Hence, $A$ dominates $C$, $A$ weakly dominates $C$, and $A$ is better than $C$.

2. The sets $A$, $B$, and $C$ weakly dominate themselves, but the same set cannot strictly dominate itself, nor be better than itself.

3. $A$ weakly dominates $B$ but $B$ does not weakly dominate $A$.

Thus dominance is a binary relation. While it provides a coherent way to order approximation sets, it is limited in detecting degrees of better and differences between
indifferent or non-comparable sets. Figure 3.2 illustrates this point. In the left and middle panels, approximation set $A$ strictly dominates $B$ but more prominently in the latter panel. Thus $A$ is better. However, the statement ‘$A$ is better than $B$’ does not say how much better $A$ is compared to $B$ in the middle panel. The right panel shows two non-comparable approximation sets but to most decision-makers, $A$ is better than $B$. More precise comparisons are enabled by quality measures.

Generally, quality measures can either be unary or binary. A unary quality measure assigns a real number to an approximation set, say $I(A)$. Examples include average distance to the Pareto front, the diversity of the approximation, the number of solutions in the approximation set, or the hypervolume indicator.
Figure 3.1: Outcomes of three hypothetical algorithms for a bi-objective problem. The approximation sets are denoted as $A$, $B$ and $C$; the (discretized) Pareto Front $P$ consists of three objective vectors. Between $A$, $B$ and $C$, the following dominance relations hold: $A \gg C$, $A \succ C$, $B \succ C$, $A \succeq A$, $A \succeq B$, $A \succeq C$, $B \succeq B$, $B \succeq C$, $C \succeq C$, $A \triangleright B$, $A \triangleright C$, and $B \triangleright C$.

Figure 3.2: In the left and center panels, $A \triangleright B$, but the difference is more prominent in the center. In the right panel, $A$ and $B$ are indifferent by definition, but visual inspection suggests that $A$ may be better.
On the other hand, a binary quality measure assigns a real number to a pair of approximation sets, say $I(A, B)$. Examples are the binary epsilon indicator and the coverage indicator. Whereas unary measures are absolute, binary measures are relative.

A “good” quality indicator possesses the following properties: (1) it is order-preserving, that is if $A \succeq B$ then $I(A) \geq I(B)$ and (2) it is strictly monotonic or sensitive to Pareto dominance that is, if $A \succeq B$ and $B \not\succeq A$ then $I(A) > I(B)$. Here, we adopt Knowles’ (2006) recommendation of two Pareto compliant indicators—the unary hypervolume indicator, $I_H$ and the binary epsilon indicator, $I_\epsilon$, both of which represent state-of-the-art as far as quality indicators are concerned.

### 3.1.1 The Hypervolume Indicator

In calculating the hypervolume indicator we make use of normalized objective vectors, that is, we rescale $y^i \in \mathbb{R}_m$ to get $z^i \in \mathbb{Z} = (0, 1)^m$ as follows:

$$
z^i_j = \frac{y^i_j - y_{\min,j}}{y_{\max,j} - y_{\min,j}}, \ j = 1, \ldots, m \tag{3.1}
$$

where $y_{\min,j}$ and $y_{\max,j}$ are the minimum and maximum respectively, of the $j^{th}$ objective values of a given set. Thus the $m$-dimensional objective space is now mapped to the $m$-dimensional unit hypercube, $\mathbb{Z} = [0, 1]^m$. Every $x \in \mathcal{X}$ maps to a point $z(x) = (z_1(x), z_2(x), \ldots, z_m(x)) \in \mathbb{Z}$. It follows that the utopia or best objective vector equals $(0, \ldots, 0)$ and the nadir or worst objective vector equals $(1, \ldots, 1)$. The utopia and nadir points are both $m$-dimensional row vectors.

The hypervolume indicator (Zitzler 1999), denoted by $I_H(A)$, measures the size of the dominated zone of a given approximation set, $A \subset [0, 1]^m$. The bigger its value, the better the approximation.
Definition 3.1.2 The hypervolume indicator of an approximation set \( A \subset [0,1]^m \), denoted by \( I_H(A) \) with reference point \((1,\ldots,1)\in\mathbb{R}^m\) is defined as

\[
I_H(A) = \int_{[0,1]^m} \alpha_A(z) dz
\]

where

\[
\alpha_A(z) = \begin{cases} 
1, & \text{if } A \succeq \{z\} \\
0, & \text{else}
\end{cases}
\]

The indicator function \( \alpha_A(z) \) indicates the region in \( Z \) that is dominated by the approximation set \( A \). If we pick an objective vector \( z \) randomly from \( Z \), then \( I_H(A) \) gives the probability that \( z \) is dominated by the approximation set \( A \). The indicator \( I_H(A) \) is therefore the volume of the multidimensional region bounded by the approximation set through the indicator function \( \alpha_A(z) \) and the chosen reference point. In Figure 3.3 the dominated zone measured by \( I_H(A) \) is shown for a biobjective problem while Figure 3.4 shows the case for an MOP with \( m = 3 \).

Figure 3.3: The shaded area indicates the size of the dominated zone of the approximation set \( A = \{y^1, y^2, y^3\} \), measured by the hypervolume indicator, \( I_H \). The reference point, \( r \), bounds the calculation of the hypervolume.

Figure 3.4: The shaded volume indicates the size of the dominated zone of an approximation set with seven points in three dimensions.
Fleischer (2002) proved that the hypervolume is maximized if and only if the
approximation set contains a Pareto optimum. Its drawback is its sensitivity to the
relative scaling of the objectives, the choice of the reference point, and to the presence
or absence of extremal points in the front (Bradstreet, While, and Barone 2007). In
our calculations, we designated the nadir point as the reference, and calculated \( I_H \)
using the recursive dimension-sweep algorithm (Fonseca, Paquete, and Lopez-Ibañez
2006) implemented in a C code by the authors.

3.1.2 The Binary Epsilon Indicator, \( I_\epsilon \)

The binary epsilon indicator (Zitzler, Thiele, Laumanns, Fonseca, and da Fonseca
2003) of an approximation set \( A \subset [0,1]^m \) relative to \( B \subset [0,1]^m \) denoted by \( I_\epsilon(A,B) \)
is intuitively, the smallest factor \( \epsilon \) by which we scale the set \( A \) so that it covers or
weakly dominates the set \( B \). By convention, the second argument is treated as the
reference set.

**Definition 3.1.3** Suppose, without loss of generality, a minimization problem with
\( m \) objectives. Given \( \epsilon > 0 \) an objective vector \( z^1 = (z_1^1, z_2^1, \ldots, z_m^1) \in [0,1]^m \) is said
to \( \epsilon \)-dominate another objective vector \( z^2 = (z_1^2, z_2^2, \ldots, z_m^2) \in [0,1]^m \), if and only if
for all \( 1 \leq k \leq m \),

\[
z_k^1 + 1 \leq \epsilon \cdot (z_k^2 + 1).
\]  

We define the binary epsilon indicator of \((A,B)\) as

\[
I_\epsilon(A,B) = \inf \{ \epsilon > 0 : A \succeq \epsilon \cdot B \}
\]

for any two approximation sets \( A, B \subset [0,1]^m \)
Roughly, a vector $z^1 \epsilon$-dominates another vector $z^2$ if we can multiply each objective value in $(z^2 + 1)$ by a factor $\epsilon > 0$ so that $(z^1 + 1)$ weakly dominates $\epsilon \cdot (z^2 + 1)$. Thus $z^a \gg z^b$ implies that there exists $\frac{1}{2} \leq \epsilon < 1$ such that $z^a \epsilon$-dominates $z^b$. The $\epsilon$-indicator gives the factor $\epsilon$ by which an approximation set is worse than another with respect to all objectives. Precisely, $I_\epsilon(A, B)$ equals the minimum factor $\epsilon > 0$ such that any objective vector in $B$ is $\epsilon$-dominated by at least one objective vector in $A$. We calculate the binary epsilon indicator, $I_\epsilon(A, B)$, as follows:

$$I_\epsilon(A, B) = \max_{z^b \in B} \min_{z^a \in A} \max_{1 \leq j \leq m} \frac{z^a_j + 1}{z^b_j + 1}$$ (3.5)

In general, $I_\epsilon(A, B) \neq I_\epsilon(B, A)$. If the reference set $B$ equals the best known Pareto front, then the binary epsilon indicator measures how much worse the approximation set $A$ is relative to the ideal. In (3.5), $\frac{1}{2} \leq I_\epsilon(A, B) \leq 2$. A value of one means that approximation set $A$ is not dominated by known the best one. Small values indicate better performance.

3.2 MOP Test Functions

It is of interest to see how well the algorithms solve MOP’s with certain type of features. The performance of $IGP-Emax$, $IGP-PI$, and $CoH-Emax$ were compared on a set of five MOP’s. Denoting the dimensions of $\mathbf{Y}$ and $\mathbf{X}$ by $(m, d)$ respectively, we tested the three algorithms on problems with $(m, d) \in \{(2, 2), (4, 2), (3, 4), (2, 6)\}$. Feature-wise, the selection of the MOP’s includes problems with convex, linear, and disconnected Pareto fronts in $\mathbf{Y}$, and functions with disconnected Pareto sets in $\mathbf{X}$. The size of the correlations among the objective functions was also considered in the choice of the final test suite.
The description below includes the name of the test function, its source, and the dimensions of the objective and design spaces written as $(m, d) = (M, D)$, with $M$ indicating the number of objectives and $D$, the number of decision variables. The objective functions to be minimized and the corresponding search space are stated. Whenever available, the analytic solutions are given, otherwise, the best answer is found by grid approximation. Special features of the problem are also noted. Finally, the $m \times m$ Pearson sample correlation matrix of the objective functions for $m > 2$, or the sample Pearson correlation coefficient is reported for $m = 2$. Since the correlation matrix is symmetric, only the upper triangular is given. The sample correlations were calculated from a uniform draw of $n = 100d$, points from the design space.

The graphs of the objective and design spaces are based on a grid approximation. For these plots, a regular grid of points was first constructed in $\mathbf{X}$ and then its image was mapped to the objective space. The grid resolution is stated for each MOP. When $m \geq 3$ or $d \geq 3$, two-dimensional projections are provided. In all the graphs, the Pareto front and the Pareto set are indicated by bold-faced dots. The use of a finer grid will improve the representation of the true Pareto Front and Pareto set. All the graphs shown were produced in Matlab Version 7 using the function `plot`.

Table 3.2 gives details on the resolution used to construct the grid upon which we approximated the true Pareto front. The resolution is expressed as $n_{x^1} \times \ldots \times n_{x^d}$, where $n_{x^t}d$ is the number of points spaced equally between the $t^{th}$ lower and upper bounds of $x^t$, $1 \leq t \leq d$. In addition, the minimum and maximum values based on the grid are also reported. These extreme points were used to standardize the objective vectors in the calculation of the hypervolume indicator and the binary epsilon indicator.

The three algorithms were run under the assumption of a budget of $N_{max} = 20d$ evaluations, inclusive of the initial training data which has a size equal to $n_0 = 5d$ or
Table 3.2: The MOP-specific grid resolution used to determine the minimum and maximum of each \( Y_k(\cdot) \) and the “best” approximation to the true Pareto front, \( P_G(Y) \) containing \( n(P_G(Y)) \) approximate solutions. For the \( dltz1a \) test function, the best approximation was based on the analytical solution.

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
\text{MOP} & \text{Grid Resolution} & n(P_G(Y)) & j & y_{j,\text{min}} & y_{j,\text{max}} \\
\hline
\text{wsnl} & 300 \times 300 & 932 & 1 & 4.5 & 3515.1 \\
 & & & 2 & -559.7 & -35.3 \\
\hline
\text{knowles} & 300 \times 300 & 227 & 1 & 3.6 & 25.4 \\
 & & & 2 & 13.9 & 25.1 \\
\hline
\text{obj4d2} & 300 \times 300 & 13263 & 1 & 0.0 & 433.3 \\
 & & & 2 & -1210100.0 & 0.0 \\
 & & & 3 & 0.0 & 300.0 \\
 & & & 4 & -200.0 & 0.0 \\
\hline
\text{hedarg2} & 100^4 & 3886 & 1 & 0.0 & 1242.9 \\
 & & & 2 & -9999.2 & 0.8 \\
 & & & 3 & -20.0 & 10.0 \\
\hline
\text{dltz1a} & 10^6 & 1000^* & 1 & 0.0 & 563.0 \\
 & & & 2 & 0.0 & 563.0 \\
\hline
\end{array}
\]

\( n_0 = 10d \). This means that for an initial design with \( n_0 = 5d \) points, \( 15d \) new design points will be sampled and for an initial design with \( n_0 = 10d \), \( 10d \) new design points will be added.

### 3.2.1 The \textit{wsnl} test functions

This MOP has \((m, d) = (2, 2)\). The objectives are moderately competing with \( r = -0.51 \). Originally formulated as a single-objective constrained minimization problem by Williams, Lehman, Santner, and Notz (2002), it is recast and solved here as a box-constrained bi-objective problem. Its name follows the authors’ last names in the order listed.

We state the MOP after defining preliminary functions \( n_i(x_1, x_2) \) for \( i = 1, 2 \).
\[ n_1(x) = \left[ x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1^2}{\pi} - 6 \right] + 10 \left( 1 - \frac{1}{8\pi} \right) \cos(x_1) + 10 \]
\[ n_2(x) = -\sqrt{(10.5 - x_1)(x_1 + 5.5)(x_2 + 0.5)} - \frac{1}{30} (x_2 - \frac{5.1x_1^2}{4\pi^2} - 6)^2 - \frac{1}{3} (1 - \frac{1}{8\pi}) \cos(x_1) - \frac{1}{3} \]

where \( x = (x_1, x_2) \). Two additional preliminary functions are defined by

\[ o_1(a, b, c, d) = 2(2a)^2 + 4.5(2b)^{1.5} + 2b + 14ac + 2\sqrt{bd} \]
\[ o_2(a, b, c, d) = 1.2(2b)(2a)^{1.3} + 4.5(2b)^3 + 2.0(2b)^{0.6} + 3.5(4ac)^{1.7} + (4bd)^{0.7} \]

where \( (a, b, c, d) \) are constants each taking values from the set \{0.25, 0.50, 0.75\}. This gives a total of \( 3^4 = 81 \) possible \( (a, b, c, d) \) combinations.

In terms of functions \( n_j(x) \) and \( o_j(a, b, c, d) \), \( j = 1, 2 \) the MOP is given as:

\[
\min_{x \in X} \begin{pmatrix}
y_1(x) &= n_1(x_1, x_2) \times \frac{1}{81} \sum_{i=1}^{81} o_1(a_i, b_i, c_i, d_i) \\
y_2(x) &= n_2(x_1, x_2) \times \frac{1}{81} \sum_{i=1}^{81} o_2(a_i, b_i, c_i, d_i)
\end{pmatrix}
\]

where \( X = \{(x_1, x_2) : -5 \leq x_1 \leq 10, 0 \leq x_2 \leq 10\} \). The Pareto Front is mainly convex and the Pareto set is almost a connected set if not for two ‘stray’ points at the southeast region. The objective and design spaces are shown in Figure 3.5. To approximate the Pareto front, the design space was discretized into a \( 300 \times 300 \) grid and \( y(x) \) evaluated on each point in the grid.
3.2.2 The knowles test functions

Like the previous test function, the knowles MOP is a low-dimensional problem with \((m, d) = (2, 2)\). Based on the sample correlation, the objectives do not appear as conflicting as in the wsnl MOP with \(r = 0.48\).

Letting \(\mathbf{x} = (x_1, x_2)\), the MOP as stated in the source is as follows:

\[
\begin{align*}
\min_{\mathbf{x} \in \mathcal{X}} \quad & \begin{cases} 
  y_1(\mathbf{x}) = 20 - r(\mathbf{x}) \cos(\phi(\mathbf{x})) \\
  y_2(\mathbf{x}) = 20 - r(\mathbf{x}) \sin(\phi(\mathbf{x}))
\end{cases}
\end{align*}
\]

where

\[
r(\mathbf{x}) = 9 - \left[ 3 \sin \left( \frac{5}{2(x_1 + x_2)^2} \right) + 3 \sin(4(x_1 + x_2)) + 5 \sin(2(x_1 + x_2) + 2) \right],
\]

\[
\phi(\mathbf{x}) = \frac{\pi}{12(x_1 - x_2 + 3)} \quad \text{and} \quad \mathcal{X} = \{ (x_1, x_2) : 0 \leq x_1, x_2 \leq 3 \}.
\]

The design and objective spaces are shown in Figure 3.6 with the Pareto front and Pareto set. The Pareto Front is disconnected, with one set in the vicinity of \(y_1(\mathbf{x}) = 4\) and another set at
14 < y_1(x) < 20. The Pareto set is also disconnected, consisting of a linear subset and a curved segment in the neighborhood of x_1 = 0. The source (Knowles 2006) reports the analytic solution to be the set of all pairs (x_1, x_2) that sum to 4.4116 represented by the linear subset. The curved segment was established by 300 × 300 grid approximation.

(a) Design Space of the knowles MOP

(b) Objective Space the knowles MOP

Figure 3.6: The Pareto set and the Pareto front of the knowles MOP are both disconnected.

### 3.2.3 The obj4d2 test function

This is an MOP with dimensions (m, d) = (4, 2) specially constructed to produce a correlation matrix with moderate to strong negative and positive elements. In Huang and Hao (2006), y_1(x) is called the Martin-Gaddy function and y_3(x) is the so called B2 function. An online source, [http://www.geatbx.com/docu/fcnindex-$01$.html](http://www.geatbx.com/docu/fcnindex-$01$.html), dedicated to genetic algorithms lists y_2(x) as the Rosenbrock function and y_4(x) is called the de Jong function. The original design spaces were modified to
have a common search space. The (sample) correlation matrix of \( \{y_1, y_2, y_3, y_4\} \) is:

\[
\begin{pmatrix}
1.00 & -0.55 & 0.61 & -0.67 \\
1.00 & -0.45 & 0.68 & \\
1.00 & -0.95 & \\
1.00 \\
\end{pmatrix}
\]

Marginally, the pair \((y_1(x), y_3(x))\) on the first and third columns, respectively, conflict with \((y_2(x), y_4(x))\) on the second and fourth, respectively, in the sense of having negative correlations. Within each pair, the objectives are moderately and positively correlated. The MOP is given as:

Figure 3.7: The design space of the test function \( obj4d2 \) with an irregularly shaped Pareto set.
where $\mathcal{X} = \{(x_1, x_2) : -10 \leq x_1, x_2 \leq 10\}$. The design space with the Pareto set is shown in Figure 3.7. The two-dimensional projections of the objective space with the Pareto front are illustrated in Figure 3.8. The Pareto front was found via grid approximation using a 300 × 300 lattice on the design space.

### 3.2.4 The hedarg2 test functions

This is an MOP with $(m, d) = (3, 4)$. It was originally formulated as a single-objective optimization problem with two function constraints, taken from an online source [http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/go.htm](http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/go.htm). The $3 \times 3$ sample correlation matrix shows that $y_1(x)$ and $y_2(x)$ are weakly positively related and that both are moderately conflicting with $y_3(x)$.

$$
\begin{pmatrix}
0.27 & -0.51 \\
-0.78 & 1.00
\end{pmatrix}
$$

Letting $x = (x_1, x_2, x_3, x_4)$, the MOP statement is as follows:

$$
\begin{align*}
\min_{x \in \mathcal{X}} 
\begin{pmatrix}
y_1(x) &= (x_1 - x_2)^2 + \left(\frac{x_1 + x_2 - 10}{3}\right)^2 \\
y_2(x) &= 100(x_1 - x_2)^2 + (x_1 - 1)^2 \\
y_3(x) &= x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 4 \cos(4\pi x_2) + 0.7 \\
y_4(x) &= \sum_{i=1}^{2} x_i^2
\end{pmatrix}
\end{align*}
$$
Figure 3.8: Two-dimensional projections of the objective space and Pareto front of test function \textit{obj4d2}

where \( \mathbf{X} = \{(x_1, x_2, x_3, x_4) : 0 \leq x_i \leq 10 \ i = 1, 2, 3, 4 \} \). The two-dimensional projections of the four-dimensional Pareto set in design space are plotted in Figure 3.9.
Figure 3.9: Two-dimensional projection of design space and grid-approximated Pareto optimal set for \( \text{hedarg2} \) test functions.

The two-dimensional projections of the three-dimensional Pareto surface, obtained by means of a \( 100^4 \) grid approximation are shown in Figure 3.10.
3.2.5 The \textit{dltz1a} test function

The \textit{dltz1a} is an MOP with \((m, d) = (2, 6)\). The function originated from the work of Deb, Thiele, Laumanns, and Zitzler (2001) but was later modified by Knowles (2006) to reduce the ruggedness in the objective functions. The modified version is used here. The correlation coefficient between \(y_1(\cdot)\) and \(y_2(\cdot)\) is \(r = -0.56\), a moderately strong negative linear relationship.

With \(\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6)\), the MOP is stated as:
\[
\begin{align*}
\min_{x \in \mathcal{X}} & \quad \begin{pmatrix}
y_1(x) & = & \frac{1}{2}x_1(1 + g(x)) \\
y_2(x) & = & \frac{1}{2}(1 - x_1)(1 + g(x))
\end{pmatrix} \\
\text{where } g(x) & = 100 \left[5 + \sum_{i \in \{2, \ldots, 6\}} (x_i - 0.5)^2 - \cos(2\pi(x_i - 0.5))\right] \\
\mathcal{X} & = \{ (x_1, x_2, x_3, x_4, x_5, x_6) : 0 \leq x_i \leq 1, \ i = 1, \ldots, 6 \}.
\end{align*}
\]

The Pareto Front is a triangular plane and the relatively high-dimensional Pareto set is reported in Knowles (2006) as consisting of all solutions where all design variables equal 0.50 except \(x_1\) which can take any value in \([0, 1]\). The Pareto set is a 5-dimensional hyperplane in \([0, 1]^6\). Knowing this, not every possible two-dimensional projection on the design space but only the two distinct patterns of the projection are displayed on Figure 3.11. In the left panel, the projection of the Pareto set is a line on the plane formed by the inactive variable \(x_1\) and an active design variable \(x_j, j = 2, 3, 4, 5, 6\). In the right panel, the projection is a point on the plane formed by two active design variables.

On Figure 3.12, the two-dimensional Pareto front is plotted based on a \(5^6\) grid approximation.

### 3.3 Numerical Test Results

We present and compare the results of the point-set approximation to the Pareto front using three black box algorithms—IGP-PI, IGP-EmaX, and CoH-EmaX. The results of these algorithms are evaluated against a common reference—a grid-approximation of the Pareto front generated by evaluating \(y(\cdot) \in \mathbb{R}_m\) on a \(d\)-dimensional lattice of design points in \(\mathcal{X}\).
Let $\mathcal{P}_Y$ denote the true Pareto front, $\mathcal{P}_Y^G$ be the grid-approximated Pareto front (the same set used for graphing the Pareto front in Section 3.2) which will serve as a reference set, and $\mathcal{P}_o^Y$ be the approximation set of an optimizer $o \in \{iPI, iEmX, cEmX\}$ where $iPI$ stands for $IGP-PI$, $iEmX$ for $IGP-EmaX$, and $cEmX$ for $CoH-EmaX$.

Our comparisons first visualize the three approximation sets, $\mathcal{P}_Y^{iPI}, \mathcal{P}_Y^{iEmX}$, and $\mathcal{P}_Y^{cEmX}$ on the objective space which also displays $\mathcal{P}_Y^G$. In doing so, we aim to communicate an overall picture of the adequacy of the approximation where closeness to $\mathcal{P}_Y$ and other quality constructs such as minimal sufficiency, or even coverage can be discerned intuitively. In particular, in cases where the quantitative indicators give conflicting pronouncements as to which approximation set is best, we inform our judgment by examining the plot.

Then a quantitative assessment follows where we report the number of solutions identified, and the values of two Pareto compliant metrics, namely the unary hypervolume indicator $I_H(\mathcal{P}_o^Y)$ which measures the volume of $[0,1]^m$ that is dominated...
by the standardized objective vectors of $\mathcal{P}_y^o$ and the (multiplicative) binary epsilon indicator, $I((\mathcal{P}_y^{o_1}, \mathcal{P}_y^{o_2}))$ which measures the factor $(1 + \epsilon)$, $\epsilon > 0$ by which $\mathcal{P}_y^{o_1}$ is worse than $\mathcal{P}_y^{o_2}$.

On the interpretation of these quantities, we consider the following criteria:

- A diverse set of efficient solutions is good hence, we would like any of the optimizers $\mathcal{P}_y^o$, $o \in \Omega$ to identify as many solutions as necessary to represent the Pareto front well. Therefore one should assess the cardinality of $\mathcal{P}_y^o$ vis-a-vis the positions of the identified solutions relative to the Pareto front. This means that an approximation set with more solutions is not necessarily better than an approximation set with fewer. There must be efficiency in the way solutions are spread out or distributed on the Pareto front.
In the most ideal situation, when $P_o Y$ coincides with $P_Y$, the hypervolume measure, $I_H(P_o Y)$ attains its maximum value. By measuring the volume of $[0, 1]^n$ that $P_o Y$ dominates, the metric $I_H(P_o Y)$ indirectly measures the accuracy of the Pareto front approximation. Given Fleischer’s result (2002) showing that $I_H(\cdot)$ is maximized if and only if the approximation set contains a Pareto point, larger values of $I_H(\cdot)$ are therefore preferred. When the objective vectors are standardized so $0 \leq y_k(\cdot) \leq 1$, then $0 \leq I_H(\cdot) \leq 1$. There are, however, issues surrounding the use of the $I_H(\cdot)$ as an indicator. We will make these clear as these arise.

The binary epsilon indicator $I_H(A, B)$ comparing approximation set $A$ to a reference set $B$, informally measures the minimum common amount by which one needs to improve every objective $y_k(\cdot)$ of every objective vector $y^o \in A$ so that the resulting set is as good as $B$. If $I_\epsilon(P_o Y, P_G Y)$ is calculated for all $o \in \Omega$, then the algorithm producing the least value of the indicator is considered the best. In general, for any reference set $B$, $I_\epsilon(A, B) \neq I_\epsilon(B, A)$, that is, $I_\epsilon(\cdot, \cdot)$ is not symmetric. Typically, if $I_\epsilon(A, B) < I_\epsilon(B, A)$ then $A$ is judged better than $B$ because it means that $A$ needs to improve by a smaller amount in order to perform as well as $B$ than the equivalent amount $B$ needs to be as good as $A$. Aside from comparing each $P_o Y$ with $P_G Y$, we also do pairwise comparisons between distinct approximation sets $P_o Y \in \Omega$ and $P_G Y' \in \Omega$.

For each $P_o Y$, there are two sets of results: one when the initial size $n_0 = 5d$ and another when $n_0 = 10d$, where $d$ is the number of design variables. We expect the quality of approximation to be superior when the algorithm uses more data. However, some MOP’s show otherwise. In any case it is of interest to see the effect, if any, of
the initial size of the training data on the quality of approximation of the Pareto Front.

These stated, we now present the results in the next subsections organized by the MOP.

### 3.3.1 Approximating the Pareto Front of the \textit{wsnl} MOP

For \textit{wsnl}, the algorithms based on the expected Pareto improvement, or \textit{EmaX} criterion fared better than the algorithm based on the probability of improvement. Visually, Figure 3.13 suggests that $P_{\mathcal{Y}}^{\text{EmaX}}$ is most preferable, followed by $P_{\mathcal{Y}}^{i\text{EmX}}$, while the outcome of $P_{\mathcal{Y}}^{i\text{PI}}$ is least favorable. This preferential order holds in both $n_0 = 5d$ and $n_0 = 10d$. The plot also suggests that a larger initial set brought marginal benefit to the quality of approximation of the $P_{\mathcal{Y}}^{\text{EmaX}}$ algorithm, little or no benefit in the case of $P_{\mathcal{Y}}^{i\text{EmX}}$ and $P_{\mathcal{Y}}^{i\text{PI}}$. The quantitative indicators reinforce this impression. Table 3.3 reveals that \textit{CoH-EmaX} identified 30 and 21 approximate solutions which are well-distributed along the Pareto front, as seen in Figure 3.13. For
this reason, $\mathcal{P}_Y^{EmaX}$ presents the best approximation of the Pareto Front; $\mathcal{P}_Y^{iEmX}$ is the smallest approximation set and placed second to $\mathcal{P}_Y^{EmaX}$ in terms of efficiency of representation; $\mathcal{P}_Y^{iPI}$ presents the least efficient representation, failing to spread out 5 or 23 solutions on the Pareto front. Thus in terms of evenness of coverage, the $EmaX$-guided algorithms are more favorable than $IGP-PI$. The effect of the initial size of the training data is most visible in the $IGP-PI$ and $CoH-EmaX$ algorithms where differences in the cardinality of the approximation set are discernible. In $\mathcal{P}_Y^{iPI}$, the increase in the number of approximate solutions did not improve the evenness of coverage of the Pareto Front as additional approximate solutions only crowded or clustered around portions of the objective space near the Pareto Front, rendering a patchy appearance. In terms of hypervolume measure, $I_H(\mathcal{P}_Y^o)$, Table 3.3 reveals that the $P_{cEmX}$ covered the largest dominated area followed by $P_{iEmX}$; however, $P_{iPI}$ did not lag far behind, despite the appreciable difference in the quality of representation. The $IGP-EmaX$ algorithm performed particularly well compared to $IGP-PI$. Even with a smaller approximation set $IGP-EmaX$ still beat $IGP-PI$, $I_H(\mathcal{P}_Y^{iEmX}) > I_H(\mathcal{P}_Y^{iPI})$ and its representation of $\mathcal{P}_Y$ is also visibly better. The hypervolume measures of the three algorithms are acceptable considering that $I_H(\mathcal{P}_Y^G) = 0.9102$ based on 932 solutions. The training set with $n_0 = 10d$ delivered a slightly higher hypervolume measure than the set based on $n_0 = 5d$ in the case of the $EmaX$-guided algorithms.

<table>
<thead>
<tr>
<th>$n_0$</th>
<th>$\mathcal{P}_Y^{iPI}$</th>
<th>$\mathcal{P}_Y^{iEmX}$</th>
<th>$\mathcal{P}_Y^{cEmX}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5d</td>
<td>5</td>
<td>13</td>
<td>30</td>
</tr>
<tr>
<td>10d</td>
<td>23</td>
<td>12</td>
<td>21</td>
</tr>
</tbody>
</table>

Table 3.3: The number of solutions in $\mathcal{P}_Y^o$, $o \in \Omega$. 

$P_{cEmX}$ covered the largest dominated area followed by $P_{iEmX}$; however, $P_{iPI}$ did not lag far behind, despite the appreciable difference in the quality of representation. The $IGP-EmaX$ algorithm performed particularly well compared to $IGP-PI$. Even with a smaller approximation set $IGP-EmaX$ still beat $IGP-PI$, $I_H(\mathcal{P}_Y^{iEmX}) > I_H(\mathcal{P}_Y^{iPI})$ and its representation of $\mathcal{P}_Y$ is also visibly better. The hypervolume measures of the three algorithms are acceptable considering that $I_H(\mathcal{P}_Y^G) = 0.9102$ based on 932 solutions. The training set with $n_0 = 10d$ delivered a slightly higher hypervolume measure than the set based on $n_0 = 5d$ in the case of the $EmaX$-guided algorithms.
but not for the *IGP-PI* algorithm where a marginal decrease was observed. The

<table>
<thead>
<tr>
<th>$n_0$</th>
<th>$\mathcal{P}_Y^{iPI}$</th>
<th>$\mathcal{P}_Y^{iEmX}$</th>
<th>$\mathcal{P}_Y^{cEmX}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5d</td>
<td>0.8465</td>
<td>0.8639</td>
<td>0.8837</td>
</tr>
<tr>
<td>10d</td>
<td>0.8169</td>
<td>0.8904</td>
<td>0.9022</td>
</tr>
</tbody>
</table>

Table 3.4: A comparison of the size of the dominated region. Note $I_H(\mathcal{P}_Y^G) = 0.9102$

*IGP-PI* algorithm rated poorly in representing $\mathcal{P}_Y$, as evidenced by the density of approximate solutions in the certain neighborhoods of $\mathcal{P}_Y^G$. However the manifest difference in the quality of representation does not translate into an equally perceptible difference in the hypervolume measure. This can be explained as follows: With the objective space mapped to the $m$-dimensional hypercube, the hypervolume indicator effectively measures the volume of $[0, 1]^m$, rather than of $\mathcal{Y}$, that is dominated by an approximation set. Provided that some points of the Pareto front lie close to the origin (the ideal point in terms of $\mathcal{Y}$), any algorithm that identifies solutions in this region is guaranteed to have a decent hypervolume measure, regardless of whether its distribution is even or not along the Pareto front. In the present case, the almost convex Pareto front is situated close to the origin. The *IGP-PI* algorithm exploited solutions in the vicinity of $y_1(x) = 0$; it identified a cluster of solutions in the “knee” region—the portion of the Pareto front that is closest to the origin in the Euclidean norm and another cluster of solutions in the neighborhood of $y_2(x) = 0$.

When compared to the best approximation, $\mathcal{P}_Y^G$, the first column in Table 3.5 shows that the set $\mathcal{P}_Y^{cEmX}$ needs to improve by the least amount to be as good as $\mathcal{P}_Y^G$ while $\mathcal{P}_Y^{iPI}$ requires the largest amount. The epsilon factor associated with $\mathcal{P}_Y^{iEmX}$ is slightly lower than that of $\mathcal{P}_Y^{iPI}$. Between the *EmX*-assisted algorithms, there
appears to be a gain in modeling the dependence among the objectives as opposed to treating them as independent processes. Between the independence models, the EmaX improvement criterion is superior to the probability of improvement. The same conclusions can be drawn by looking at the binary epsilon indicators of the optimizers when $n_0 = 10d$. The preferential ranking among the three algorithms with respect to the epsilon indicator is retained, $\text{CoH-EmaX} > \text{IGP-EmaX} > \text{IGP-PI}$ using a slight abuse of notation. Tables 3.5 and 3.6 show that the bigger initial experimental design did not produce significant enhancements in the epsilon factors of the three algorithms. For $\text{IGP-PI}$, a slight degradation in the $I_e(P^g_\gamma, P^{\epsilon}_{\gamma'})$ for $n_0 = 10d$ compared to $n_0 = 5d$ was noted. Considering all factors, $\text{CoH-EmaX}$ performed exceptionally well. Its manner of representing the Pareto front is consistent with our idea of minimal sufficiency. The size of the dominated region nearly equals that of the best approximation $P^g_\gamma$, and $P^{cEmaX}_\gamma$ needs to improve by the smallest factor $(1 + \epsilon)$ to be as good as the best approximation, $P^g_\gamma$. 

| $I_e(A, B)$, $n_0 = 5d$ |
|-----------------|-----------------|-----------------|-----------------|-----------------|
| $A$             | $P^G_\gamma$    | $P^{iPI}_\gamma$ | $P^{iEmaX}_\gamma$ | $P^{cEmaX}_\gamma$ |
| IGP-PI          | 1.1045          | -                | 1.1028          | 1.0883          |
| IGP-EmaX        | 1.0811          | 1.0485           | -               | 1.0703          |
| CoH-EmaX        | 1.0359          | 1.0175           | 1.0281          | -               |

Table 3.5: The binary epsilon indicator. Column 2 compares $P^o_\gamma$, $o \in \Omega$ to a common reference, $P^G_\gamma$. Columns 3 - 5 shows the $I_e(P^o_\gamma, P^{o'}_\gamma)$, $o \neq o' \in \Omega$. 

126
\( I(A, B) \), \( n_0 = 10d \)

<table>
<thead>
<tr>
<th></th>
<th>( \mathcal{P}_G )</th>
<th>( \mathcal{P}_{P1} )</th>
<th>( \mathcal{P}_{EmaX} )</th>
<th>( \mathcal{P}_{EmX} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGP-PI</td>
<td>1.1112</td>
<td>-</td>
<td>1.1073</td>
<td>1.1112</td>
</tr>
<tr>
<td>IGP-EmaX</td>
<td>1.0401</td>
<td>1.0120</td>
<td>-</td>
<td>1.0323</td>
</tr>
<tr>
<td>CoH-EmaX</td>
<td>1.0163</td>
<td>1.0120</td>
<td>1.0136</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.6: The binary epsilon indicator. Column 2 compares \( \mathcal{P}_G \), \( o \in \Omega \) to a common reference, \( \mathcal{P}_G \). Columns 3 - 5 shows the \( I(A, B) \), \( o \neq o' \in \Omega \).

Figure 3.14: A comparison of IGP-PI, IGP-EmaX, CoH-EmaX algorithms for approximating the Pareto Front of the function \textit{knowles}.

### 3.3.2 Approximating the Pareto Front of the \textit{knowles} MOP

Here the Pareto front is disconnected, one region, say \( \mathcal{P}_1 \), is near the line \( y_1(x) = 0 \) and the other, say \( \mathcal{P}_2 \), is more distant around \( y_1(x) = 15 \). Generally, the three algorithms exploited \( \mathcal{P}_1 \) more aggressively as evidenced by the number of solutions localized in the latter, while practically no action is observed in \( \mathcal{P}_2 \), except perhaps with the CoH-EmaX algorithm based on \( n_0 = 5d \). Starting with a bigger initial experimental design causes a marginal improvement in the representation of \( \mathcal{P}_1 \) only.
The tables on the quality metrics indicate that the *IGP-PI* benefitted the most, *IGP-EmaX* posted negligible gains, while *CoH-EmaX* worsened. With a bigger initial design, *IGP-PI* and *IGP-EmaX* found more solutions while *CoH-EmaX* did not. The three algorithms also fail to locate solutions in $\mathcal{P}_y^2$ using $n_0 = 10d$. Looking at the

<table>
<thead>
<tr>
<th>$n_0$</th>
<th>$\mathcal{P}_y^{IPI}$</th>
<th>$\mathcal{P}_y^{IEmX}$</th>
<th>$\mathcal{P}_y^{CEmX}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5d</td>
<td>2</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>10d</td>
<td>5</td>
<td>15</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 3.7: The number of solutions in $\mathcal{P}_y^n$, $o \in \Omega$.

hypervolume measure in Table 3.8 and the binary epsilon indicators in Tables 3.9 and 3.10, we observe that the between-algorithm differences in approximation are marked when the initial design is small. With an enlarged design however, the differences became insignificant. Whereas the increase in the hypervolume measure was slight in *IGP-EmaX*, it was significant in *IGP-PI*, as the number of solutions found rose from two to five points as reported in Table 3.7. The *IGP-EmaX* algorithm did not materially improve in the hypervolume measure with a bigger initial training set. For *CoH-EmaX*, the loss of three approximate solutions in the $\mathcal{P}_y^2$ region resulted in

| $n_0$ | $I_H(\mathcal{P}_y^n)$ | $\mathcal{P}_y^{IPI}$ | $\mathcal{P}_y^{IEmX}$ | $\mathcal{P}_y^{CEmX}$ |
|-------|------------------------|------------------------|------------------------|
| 5d    | 0.6526                 | 0.7388                 | 0.8231                 |
| 10d   | 0.7506                 | 0.7480                 | 0.7488                 |

Table 3.8: A comparison of the size of the dominated region. Note $I_H(\mathcal{P}_y^G) = 0.8591$.
an appreciable decrease in the size of the dominated area. Furthermore, Figure 3.14 illustrates that distribution of approximate solutions did not span the entire range of $\mathcal{P}_Y^1$.

The best performer based on $I_H(\cdot)$ and $I_\epsilon(\cdot)$, CoH-EmaX did not improve with the addition of more points in the initial design; indeed, its performance deteriorated. Its hypervolume measure at $n_0 = 10d$ decreased considerably and its epsilon factor compared to $\mathcal{P}_Y^G$ increased. The drop in hypervolume measure and the rise in the epsilon factor can be explained by the loss of three approximate solutions in $\mathcal{P}_Y^2$ found using the smaller design. The contribution of these points to the total dominated area was quite substantial. The loss of the solutions in $\mathcal{P}_Y^2$ found by CoH-EmaX meant

<table>
<thead>
<tr>
<th></th>
<th>$I_\epsilon(A, B)$, $n_0 = 5d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>$\mathcal{P}_Y^G$</td>
</tr>
<tr>
<td>IGP-PI</td>
<td>1.3175</td>
</tr>
<tr>
<td>IGP-EmaX</td>
<td>1.2527</td>
</tr>
<tr>
<td>CoH-EmaX</td>
<td>1.0832</td>
</tr>
</tbody>
</table>

Table 3.9: The binary epsilon indicator. Column 2 compares $\mathcal{P}_Y^o$, $o \in \Omega$ to a common reference, $\mathcal{P}_Y^G$. Columns 3 - 5 shows the $I_\epsilon(\mathcal{P}_Y^o, \mathcal{P}_Y^{o'})$, $o \neq o' \in \Omega$.

that $\mathcal{P}_Y^{CEmX}$ needed to be improved more to be as good as $\mathcal{P}_Y^G$. The compromised performance of CoH-EmaX as a result of the enlarged initial design is likely due to the accumulation of numerical errors in the maximization of the log posterior to locate the posterior mode, $\hat{\theta}$, its subsequent propagation to the covariance matrices that affect the form of the emulator which in turn affects the samples drawn in the Monte Carlo estimate of the expected Pareto improvement. We note that the CoH-EmaX prematurely terminated after the addition of nine new points starting with $n_0 = 10d$. 

but reached the stopping criterion of $N_{max} = 20d$ with the smaller initial design.

We take the premature termination as a challenge to make the code more robust to ill-conditioning, apparently the source of numerical instability. Overall the knowles

\[
I_e(A, B), \ n_0 = 10d
\]

<table>
<thead>
<tr>
<th>( A )</th>
<th>( P_{g}^{A} )</th>
<th>( P_{g}^{IP} )</th>
<th>( P_{g}^{EmX} )</th>
<th>( P_{g}^{EmX} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGP-PI</td>
<td>1.2430</td>
<td>-</td>
<td>1.0073</td>
<td>1.0044</td>
</tr>
<tr>
<td>IGP-EmaX</td>
<td>1.2433</td>
<td>1.0049</td>
<td>-</td>
<td>1.0087</td>
</tr>
<tr>
<td>CoH-EmaX</td>
<td>1.2464</td>
<td>1.0140</td>
<td>1.0115</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.10: The binary epsilon indicator. Column 2 compares \( P_{o}^{g} \), \( o \in \Omega \) to a common reference, \( P_{g}^{A} \). Columns 3 - 5 shows the \( I_e(P_{o}^{g}, P_{o'}^{g}) \), \( o \neq o' \in \Omega \).

MOP was not adequately represented by any of the three algorithms since an entire sub-region \( P_{g}^{2} \) was missed. An increase in the size of the initial training data did not remedy the problem; perhaps prolonging the search could recover some solutions in \( P_{g}^{2} \), probably at the price of introducing numerical instabilities.

3.3.3 Approximating the Pareto Front of the \textit{obj4d2} MOP

With \( m = 4 \), all possible two-dimensional projections of the Pareto front are shown in Figures 3.15,3.16 and 3.17. Given these partial views, the actual quality of the approximation is inconclusive from the plots, meaning that the evenness of the distribution, or the presence or absence of clustering, cannot be fully ascertained.

Table 3.11 shows that \textit{IGP-PI} yielded the largest approximation set, followed by \textit{CoH-EmaX} . The \textit{IGP-EmaX} algorithm delivered the smallest approximation set. These observations apply to both \( n_0 = 5d \) and \( n_0 = 10d \). In terms of efficiency of representation, \textit{CoH-EmaX} located approximate solutions consistently within the
Figure 3.15: Comparison in $y_1$ and $y_2$ of IGP-PI, IGP-EmaX, CoH-EmaX algorithms for approximating the Pareto Front of the obj4d2 test functions.

Pareto efficient region in all the two-dimensional projections of the Pareto Front, a behavior not shared by the two algorithms based on the independence model. This case suggests a plausible benefit of incorporating the dependence structure among the objectives, though this hypothesis merits further investigation. The hypervolume measure associated with $\mathcal{P}_Y^{cEmX}$ is consistently the greatest for both sizes of the initial training data. The dominated volume of $\mathcal{P}_Y^{iPI}$ is the least among the three optimizers. The clustering of approximate solutions is once again observed for IGP-PI in this MOP. Table 3.12 gives these details. The less than 50% hypervolume measure of $\mathcal{P}_Y^G$ evidently raises some doubt on the adequacy of the grid’s resolution in approximating...
the true Pareto front. However, a closer inspection of the two-dimensional projections of the Pareto front, particularly $y_3(x) versus y_4(x)$ and perhaps $y_1(x) versus y_4(x)$ reveals an aspect of the Pareto front that apparently makes $I_H(\cdot)$ unreliable as an
Figure 3.17: Comparison in $y_2$ and $y_4$ of $IGP$-$PI$, $IGP$-$EmaX$, $CoH$-$EmaX$ algorithms for approximating the Pareto Front of the $obj4d2$ test functions.

<table>
<thead>
<tr>
<th>$n_0$</th>
<th>$\mathcal{P}_Y^{PI}$</th>
<th>$\mathcal{P}_Y^{EmaX}$</th>
<th>$\mathcal{P}_Y^{CoH}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5d</td>
<td>0.1196</td>
<td>0.2418</td>
<td>0.3517</td>
</tr>
<tr>
<td>10d</td>
<td>0.1992</td>
<td>0.2725</td>
<td>0.3857</td>
</tr>
</tbody>
</table>

Table 3.12: A comparison of the size of the dominated region. Note $I_H(\mathcal{P}_Y^{G}) = 0.4444$. 

indicator of the space dominated by an approximation set. Zitzler (2003), the proponent of $I_H(\cdot)$ acknowledges that the $I_H(\cdot)$ is biased towards convex inner portions of the objective space. A related discussion was initiated in Subsection 3.3.1. If the
Pareto front is situated far from the origin, then $I_H(\cdot)$ becomes a dubious figure of merit. This explains the low hypervolume measure of $\mathcal{P}_Y^G$. The effect of grid resolution was very minimal. A $100 \times 100$ resolution provided $I_H(\mathcal{P}_Y^G) = 0.4420$ while a $300 \times 300$ grid returned $I_H(\mathcal{P}_Y^G) = 0.4444$.

The binary epsilon factors indicated in Tables 3.13 and 3.14 show that $\mathcal{P}_Y^{cEmX}$ needs to improve the least and $\mathcal{P}_Y^{iPl}$ the most to be as good as $\mathcal{P}_Y^G$. Both tables also show that $I_e(\mathcal{P}_Y^{cEmX}, \mathcal{P}_Y^{iEmX}) < I_e(\mathcal{P}_Y^{iEmX}, \mathcal{P}_Y^{cEmX})$ leading to the conclusion that $\mathcal{P}_Y^{cEmX}$ is better. This difference is negligible when $n_0 = 5d$ but becomes more evident when $n_0 = 10d$. The amplification of the difference can be explained by the deterioration of the quality of approximation of $IGP-EmaX$ as the initial design is enlarged. The deterioration is also evident in the hypervolume measure. The

<table>
<thead>
<tr>
<th>$I_e(A, B)$, $n_0 = 5d$</th>
<th>$B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>$\mathcal{P}_Y^G$</td>
</tr>
<tr>
<td>IGP-PI</td>
<td>1.6467</td>
</tr>
<tr>
<td>IGP-EmaX</td>
<td>1.2111</td>
</tr>
<tr>
<td>CoH-EmaX</td>
<td>1.1883</td>
</tr>
</tbody>
</table>

Table 3.13: The binary epsilon indicator. Column 2 compares $\mathcal{P}_Y^o$, $o \in \Omega$ to a common reference, $\mathcal{P}_Y^G$. Columns 3 - 5 shows the $I_e(\mathcal{P}_Y^o, \mathcal{P}_Y^{o'})$, $o \neq o' \in \Omega$. 

enlargement of the initial design achieved modest improvements in the $I_H(\cdot)$ and $I_e(\cdot)$ metrics for the $CoH-EmaX$ algorithm, but a dramatic improvement in $IGP-PI$. It caused a slight improvement in the hypervolume measure but a negligible downgrade in the binary epsilon factor of the $IGP-EmaX$ algorithm.
\[ I_t(A, B), \ n_0 = 10d \]

<table>
<thead>
<tr>
<th>( A )</th>
<th>( \mathcal{P}_Y^G )</th>
<th>( \mathcal{P}_Y^{IP} )</th>
<th>( \mathcal{P}_Y^{EmX} )</th>
<th>( \mathcal{P}_Y^{cEmX} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGP-PI</td>
<td>1.4055</td>
<td>-</td>
<td>1.4055</td>
<td>1.4055</td>
</tr>
<tr>
<td>IGP-EmaX</td>
<td>1.2337</td>
<td>1.0292</td>
<td>-</td>
<td>1.2291</td>
</tr>
<tr>
<td>CoH-EmaX</td>
<td>1.0910</td>
<td>1.0292</td>
<td>1.0427</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.14: The binary epsilon indicator. Column 2 compares \( \mathcal{P}_Y^o, \ o \in \Omega \) to a common reference, \( \mathcal{P}_Y^G \). Columns 3 - 5 shows the \( I_t(\mathcal{P}_Y^o, \mathcal{P}_Y^{o'}) \), \( o \neq o' \in \Omega \).

### 3.3.4 Approximating the Pareto Front of the hedarg2 MOP

A prominent feature of the Pareto front as depicted in Figure 3.18 is its severe non-convexity. As a result, the hypervolume measure of \( \mathcal{P}_Y^G \) is small. The three algorithms are able to localize a good number of solutions. Table 3.15 presents the identified number of solutions, with IGP-PI and CoH-EmaX yielding larger approximation sets compared to IGP-EmaX. The effect of the initial size of the training data on the cardinality of the approximation set is evident in IGP-EmaX and IGP-PI but not in CoH-EmaX. In terms of the hypervolume measure, the size of the dominated regions of the independence models are more similar than the size of the dominated region of CoH-EmaX. Using the hypervolume measure of \( \mathcal{P}_Y^G \) as an upper bound,

<table>
<thead>
<tr>
<th>( n_0 )</th>
<th>( \mathcal{P}_Y^{IP} )</th>
<th>( \mathcal{P}_Y^{EmX} )</th>
<th>( \mathcal{P}_Y^{cEmX} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>5d</td>
<td>36</td>
<td>31</td>
<td>50</td>
</tr>
<tr>
<td>10d</td>
<td>42</td>
<td>39</td>
<td>51</td>
</tr>
</tbody>
</table>

Table 3.15: The number of solutions in \( \mathcal{P}_Y^o, \ o \in \Omega \).
Figure 3.18: Comparison in $y_1$ and $y_2$ of IGP-PI, IGP-EmaX, CoH-EmaX algorithms for approximating the Pareto Front of the $hedarg2$ test functions.

It can be concluded that CoH-EmaX performed well. The increase in the number of
training data at the onset did not improve the hypervolume measure as much. The
same can be said for the IGP-PI and IGP-EmaX algorithms.

<table>
<thead>
<tr>
<th>$I_H(\mathcal{P}_Y^{n_0})$</th>
<th>$\mathcal{P}_Y^{iPI}$</th>
<th>$\mathcal{P}_Y^{iEmX}$</th>
<th>$\mathcal{P}_Y^{eEmX}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5d</td>
<td>0.1094</td>
<td>0.1254</td>
<td>0.1634</td>
</tr>
<tr>
<td>10d</td>
<td>0.1264</td>
<td>0.1488</td>
<td>0.1765</td>
</tr>
</tbody>
</table>

Table 3.16: A comparison of the size of the dominated region. Note $I_H(\mathcal{P}_Y^{G}) = 0.1942$

The findings are different between $n_0 = 5d$ and $n_0 = 10d$ in the epsilon indicator. In the case of the smaller design shown in Table 3.17, the approximation set returned by IGP-EmaX scored the smallest epsilon factor whereas IGP-PI showed the largest. This means that $\mathcal{P}_Y^{iEmX}$ was worse off $\mathcal{P}_Y^{G}$ by the smallest factor and needed to improve the least to be as good as best Pareto front approximation; the epsilon factor associated with $\mathcal{P}_Y^{eEmX}$ was the median among the three, however its difference from $\mathcal{P}_Y^{iEmX}$ was considerable. The set $\mathcal{P}_Y^{iPI}$ was the worst compared to $\mathcal{P}_Y^{G}$ requiring the largest epsilon factor to weakly dominate $\mathcal{P}_Y^{G}$. In the pairwise comparisons, the IGP-EmaX beat both CoH-EmaX and IGP-PI and CoH-EmaX was better than IGP-PI.

Table 3.18 presents the results when $n_0 = 10d$ where noticeable changes in the epsilon indicators of the EmaX-guided algorithms are seen. With CoH-EmaX, the factor improved, while with IGP-EmaX it degraded. The result is that CoH-EmaX became marginally better than IGP-EmaX. The epsilon factor of IGP-PI improved but very slightly. When compared with $\mathcal{P}_Y^{G}$, the approximation sets produced by the EmaX-guided designs are more preferable than IGP-PI.
Table 3.17: The binary epsilon indicator. Column 2 compares $\mathcal{P}_Y^o$, $o \in \Omega$ to a common reference, $\mathcal{P}_Y^G$. Columns 3 - 5 shows the $I_e(\mathcal{P}_Y^o, \mathcal{P}_Y^{o'})$, $o \neq o' \in \Omega$.

<table>
<thead>
<tr>
<th></th>
<th>$\mathcal{P}_Y^G$</th>
<th>$\mathcal{P}_Y^{IP}$</th>
<th>$\mathcal{P}_Y^{EmX}$</th>
<th>$\mathcal{P}_Y^{EmX}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGP-PI</td>
<td>1.6590</td>
<td>-</td>
<td>1.6590</td>
<td>1.6590</td>
</tr>
<tr>
<td>IGP-EmaX</td>
<td>1.0643</td>
<td>1.0295</td>
<td>-</td>
<td>1.0475</td>
</tr>
<tr>
<td>CoH-EmaX</td>
<td>1.2157</td>
<td>1.2151</td>
<td>1.2147</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.18: The binary epsilon indicator. Column 2 compares $\mathcal{P}_Y^o$, $o \in \Omega$ to a common reference, $\mathcal{P}_Y^G$. Columns 3 - 5 shows the $I_e(\mathcal{P}_Y^o, \mathcal{P}_Y^{o'})$, $o \neq o' \in \Omega$.

<table>
<thead>
<tr>
<th></th>
<th>$\mathcal{P}_Y^G$</th>
<th>$\mathcal{P}_Y^{IP}$</th>
<th>$\mathcal{P}_Y^{EmX}$</th>
<th>$\mathcal{P}_Y^{EmX}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGP-PI</td>
<td>1.6250</td>
<td>-</td>
<td>1.6217</td>
<td>1.6250</td>
</tr>
<tr>
<td>IGP-EmaX</td>
<td>1.0758</td>
<td>1.0137</td>
<td>-</td>
<td>1.0750</td>
</tr>
<tr>
<td>CoH-EmaX</td>
<td>1.0546</td>
<td>1.0331</td>
<td>1.0028</td>
<td>-</td>
</tr>
</tbody>
</table>

3.3.5 Approximating the Pareto Front of the $dltz1a$ MOP

The Pareto front can be seen as the minute line segment with a negative slope near the origin. Figure 3.19 illustrates that the three algorithms were able to exploit this region even with a small design. The exploitation apparently improved with a bigger design. Algorithms based on the independence model identified significantly more solutions than the $CoH-EmaX$. Table 3.19 gives the details. The $IGP-PI$ algorithm generally returned the most number of solutions, followed closely by $IGP-EmaX$. The approximation set of $CoH-EmaX$ contains less than half of the number obtained by either $IGP-PI$ or $IGP-EmaX$ when the initial size is $n_0 = 10d$. In Table 3.19 we
Figure 3.19: Comparison in $y_1$ and $y_2$ of IGP-PI, IGP-EmaX, CoH-EmaX algorithms for approximating the Pareto Front of the function $dltz1a$

![Comparison of algorithms](image)

(a) $n_0 = 30$

(b) $n_0 = 60$

<table>
<thead>
<tr>
<th>$n_0$</th>
<th>$\mathcal{P}_Y^{IP}$</th>
<th>$\mathcal{P}_Y^{EmX}$</th>
<th>$\mathcal{P}_Y^{CoEmX}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5d</td>
<td>7</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>10d</td>
<td>10</td>
<td>12</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 3.19: The number of solutions in $\mathcal{P}_Y^o$, $o \in \Omega$.

see that even with a significantly smaller set, the CoH-EmaX algorithm proved most efficient in distributing the solutions along the Pareto front. It obtained the highest hypervolume measure, outperforming supposedly the best Pareto front approximation. Since the Pareto front’s location in objective space is close to the origin, the hypervolume measure is close to unity for $\mathcal{P}_Y^G$; and since the three algorithms exploited the region quite remarkably, the size of the dominated region is also close to unity. Considering the number of solutions, the CoH-EmaX exemplifies a minimally sufficient representation. The epsilon indicators in Table 3.21 reinforce the earlier results: all three algorithms lie close to the Pareto front in the sense of requiring only
<table>
<thead>
<tr>
<th>( n_0 )</th>
<th>( \mathcal{P}_y^{iPI} )</th>
<th>( \mathcal{P}_y^{iEmX} )</th>
<th>( \mathcal{P}_y^{cEmX} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>5d</td>
<td>0.9809</td>
<td>0.9803</td>
<td>0.9989</td>
</tr>
<tr>
<td>10d</td>
<td>0.9966</td>
<td>0.9682</td>
<td>0.9999</td>
</tr>
</tbody>
</table>

Table 3.20: A comparison of the size of the dominated region. Note \( I_H(\mathcal{P}_y^G) = 0.9995 \)

a small epsilon factor to be as good as \( \mathcal{P}_y^G \). The epsilon factor of \( CoH-EmaX \) when compared to the \( IGP-PI \) or \( IGP-EmaX \) is at its minimum possible value indicating that its approximation set is not worse off, meaning that as it is, it already weakly dominates \( \mathcal{P}_y^{iPI} \) and \( \mathcal{P}_y^{iEmX} \). The results in Table 3.22 show that an enlarged ini-

| \( I_\epsilon(A, B), \, n_0 = 5d \) |
|---|---|---|---|---|
| \( A \) | \( \mathcal{P}_y^G \) | \( \mathcal{P}_y^{iPI} \) | \( \mathcal{P}_y^{iEmX} \) | \( \mathcal{P}_y^{cEmX} \) |
| IGP-PI | 1.0236 | - | 1.0001 | 1.0236 |
| IGP-EmaX | 1.0236 | 1.0051 | - | 1.0236 |
| CoH-EmaX | 1.0106 | 1.0000 | 1.0000 | - |

Table 3.21: The binary epsilon indicator. Column 2 compares \( \mathcal{P}_y^0, \, o \in \Omega \) to a common reference, \( \mathcal{P}_y^G \). Columns 3 - 5 shows the \( I_\epsilon(\mathcal{P}_y^0, \mathcal{P}_y^{o'}) \), \( o \neq o' \in \Omega \).

tial design improved the outcome of \( CoH-EmaX \) and \( IGP-PI \) but compromised the approximation of \( IGP-EmaX \). The set \( \mathcal{P}_y^{cEmX} \) is practically as good as \( \mathcal{P}_y^G \), and it also weakly dominates \( \mathcal{P}_y^{iPI} \) and \( \mathcal{P}_y^{iEmX} \). A pairwise comparison between \( \mathcal{P}_y^{iPI} \) and \( \mathcal{P}_y^{iEmX} \) show that the probability of improvement criterion fared better than the expected Pareto improvement criterion.
\[ I_t(A, B), n_0 = 10d \]

<table>
<thead>
<tr>
<th></th>
<th>( P_{Y}^{G} )</th>
<th>( P_{Y}^{IP} )</th>
<th>( P_{Y}^{EmX} )</th>
<th>( P_{Y}^{EmX} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGP-PI</td>
<td>1.0062</td>
<td>-</td>
<td>1.0060</td>
<td>1.0062</td>
</tr>
<tr>
<td>IGP-EmaX</td>
<td>1.1072</td>
<td>1.1004</td>
<td>-</td>
<td>1.1071</td>
</tr>
<tr>
<td>CoH-EmaX</td>
<td>1.0008</td>
<td><strong>1.0000</strong></td>
<td><strong>1.0000</strong></td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.22: The binary epsilon indicator. Column 2 compares \( P_{Y}^{o} \), \( o \in \Omega \) to a common reference, \( P_{Y}^{G} \). Columns 3 - 5 shows the \( I_{t}(P_{Y}^{o}, P_{Y}^{o'}) \), \( o \neq o' \in \Omega \).

### 3.3.6 Summary of Results

In Table we summarize the preferential order based on the quality of the approximation as provided by the plots and the quantitative indicators, namely the hypervolume measure and the binary epsilon factor. The most preferred algorithm is given a rank of one, and the least preferred is assigned a three; when we are indifferent towards the objects compared, we average the ranks similar to the procedure done in rank-based non-parametric tests such as the Wilcoxon Rank-Sum test. Solving

<table>
<thead>
<tr>
<th>MOP</th>
<th>IGP-PI</th>
<th>IGP-EmaX</th>
<th>CoH-EmaX</th>
</tr>
</thead>
<tbody>
<tr>
<td>wsnl</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>knowles</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>obj4d2</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>hedarg2</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>dltz1a</td>
<td>2.5</td>
<td>2.5</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.23: Overall rank of the algorithms with respect to \( I_{H}(P_{Y}^{o}) \), \( I_{t}(P_{Y}^{o}, P_{Y}^{G}) \) for \( n_0 = 5d \), \( N_{max} = 20d \). A rank of 1 is the most preferable and 3, least preferable.

the black box MOP under a tight computational budget is viable using GP-guided sequential designs in conjunction with a multivariate improvement criterion such as...
Keane’s probability of improvement or the expected Pareto improvement proposed here. On the MOP’s tested, the \textit{CoH-EmaX} algorithm generated approximation sets that were superior to \textit{IGP-EmaX} or \textit{IGP-PI} except on the \textit{knowles} MOP, where none of the algorithms clearly came out the winner. The \textit{IGP-PI} algorithm is the least preferred because of its inability to spread out its search to identify “representative” solutions along the Pareto front; it is fraught with the clustering problem. Though

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
MOP & IGP-PI & IGP-EmaX & CoH-EmaX \\
\hline
wsnl & 3 & 2 & 1 \\
knowles & 3 & 2 & 1 \\
obj4d2 & 3 & 2 & 1 \\
heargy2 & 3 & 2 & 1 \\
dltz1a & 2 & 3 & 1 \\
\hline
\end{tabular}
\caption{Overall rank of the algorithms with respect to $I_H(P^y_q)$, $I_e(P^o, P^G)$ for $n_0 = 10d$, $N_{max} = 20d$. A rank of 1 is the most preferable and 3, least preferable.}
\end{table}

The \textit{CoH-EmaX} turned to be most preferable with respect to $I_H(\cdot)$ and $I_e(\cdot, \cdot)$, the comparisons are not on an equal footing due to the difference in the estimation procedure of the correlation parameters. Thus the performance is confounded with the estimation method. An improvement in the performance of the \textit{IGP-EmaX} algorithm is possible if the posterior mode of $\psi$ is used for the plug-in predictions. As for enhancing the performance of the \textit{IGP-PI} algorithm, the more crucial factor is the modification of the probability of improvement in such a way as to pressure it to spread its exploration of points in the Pareto efficient region.
CHAPTER 4

SUMMARY AND CONCLUSIONS

Underlying almost every design endeavor, in engineering and elsewhere, is a multiobjective problem engendered by the fundamental tension between considerations of performance, cost, schedule, and risks. Whatever the context, taking a multiobjective perspective helps to avoid finding suboptimal design configurations.

In this thesis, we have addressed the black box multiobjective problem formulated as the simultaneous minimization of \( m > 1 \) objective functions, \( y_k : \mathcal{X} \mapsto \mathbb{R} \), \( 1 \leq k \leq m \) where the vector-valued function \( y(x) = (y_1(x), \ldots, y_m(x))^T \) is an output of a complex, numerically intensive computer model evaluated at a given feasible design point \( x \in \mathcal{X} \). The image of the design space \( \mathcal{X} \) under the computer model is called the objective space, \( \mathcal{Y} = \{ y(x) \in \mathbb{R}_m : x \in \mathcal{X} \} \).

In particular, we considered black box MOP’s having hyperrectangular design spaces, \( \mathcal{X} = \prod_{i=1}^d [a_i, b_i] \subset \mathbb{R}_d \) with real numbers \( a_i < b_i \) for \( i = 1, \ldots, d \). Moreover, we assumed that the design vector \( x = (x_1, x_2, \ldots, x_d) \) contained only fixed control or manufacturing variables that could be measured on a continuous scale. As regards the objective vector, \( y(x) \) we assumed that each \( y_k(\cdot) \) was a continuous function of \( x \in \mathcal{X} \).

In solving the black box MOP we assumed no additional information about the decision-maker’s preferences, priorities, or aspiration levels and proceeded to identify a
set of compromise solutions that were simultaneously optimal against the competing targets aspired at once. The members of the compromise set represent the best trade-offs among the objectives and are optimal in the sense of being non-dominated according to the Pareto characterization. The totality of all non-dominated solutions of a given MOP is the Pareto set, a subset of the design space and its image under the computer code is what is called the Pareto front, a subset of the objective space. Within the Pareto front, trade-off is inevitable when moving from one solution to another: a finite decrement in one objective (an improvement) is only possible at the expense of an increment (degradation) in another objective. Hence within the compromise set, no “free” upgrades are possible. Accordingly, the rational choice of final design configuration comes from the Pareto set.

The process of finding a compromise set is equivalent to approximating the Pareto front. Granting that \( y_k(x) \), \( 1 \leq k \leq m \) are all continuous in \( x \in \mathcal{X} \), the Pareto front (or Pareto set) may be infinite. This said, we endeavored to obtain a “good” finite point set approximation of the Pareto front in the most efficient manner, that is using a certain number of expensive computer runs reflective of a tight computational budget. A finite point set of objective vectors is considered a good approximation if it lies as close as possible to the true Pareto front and spreads its members as evenly as possible throughout the Pareto front providing in effect, a representative coverage of the range of efficient solutions. We summed up the goal of achieving a good Pareto front approximation while restraining the number of costly function evaluations as constructing a minimally sufficient representation of the Pareto front.

To this end, we resorted to a surrogate-assisted optimization approach and developed the \( \text{EmaX} \) algorithm. The algorithm uses a multivariate Gaussian process emulator to guide a sequential optimization design to approximate the Pareto front by progressively refining the current non-dominated front, until the total number
of costly function evaluations, including the initial training data, reaches a pre-
specified number. In common with the EGO algorithm, the initial experimental
design is augmented by promising feasible design points that are determined by max-
imizing an expected improvement function. At each stage, the non-inferior designs
\( \mathbf{x}_i^{**} \in \mathcal{P}_{\mathcal{X}}^n \subset \mathcal{D}_n \) are identified using the maximin fitness function. The set of objective
vectors \( \mathcal{P}_n^\mathcal{Y} = \{ \mathbf{y}(\mathbf{x}_i^{**} : \mathbf{x}_i^{**} \in \mathcal{P}_{\mathcal{X}}^n) \} \) associated with these non-inferior designs define the
current non-dominated front, a boundary that partitions \( \mathcal{Y} \) into the set of objective
vectors that are dominated by the members of \( \mathcal{P}_n^\mathcal{Y} \) region and the set of objective
vectors that dominate or are indifferent to the members of \( \mathcal{P}_n^\mathcal{Y} \). The second group of
objective vectors comprise the improvement region of the set \( \mathcal{P}_n^\mathcal{Y} \).

The proposed expected improvement function was devised to find design points
\( \mathbf{x}^* \in \mathcal{X} \) such that \( \mathbf{y}(\mathbf{x}^*) \) belongs to the improvement region; more precisely, by
maximizing the expected Pareto improvement, we sought a new design configuration
\( \mathbf{x} \in \mathcal{X} \) that presented the greatest potential upgrade to the standing set of optimal
solutions. The expected Pareto improvement can also be interpreted as a distance
metric of the new design point’s objective vector to the current set of non-dominated
experimental designs in objective space. Once found, the design point with the largest
Pareto improvement is verified by the expensive simulator and the new \( \mathbf{x} \) is added to
the current design.

We derived a closed form expression of the expected Pareto improvement in the
case of a biobjective optimization problem. On our implementation, for arbitrary
number of objective functions, Monte Carlo averaging was used to obtain the expected
Pareto improvement.

We implemented two multivariate emulators according to whether the dependence
among the \( y_k(\cdot) \), \( k = 1, \ldots, m \) was accounted for—the independence or IGP models
ignored the dependence structure, and a dependence model which exemplified one
way of capturing the dependence among the objectives. Both models were shown to be special cases of a particular constructive approach that expressed the objective vector process $Y(\cdot)$ as a linear transformation of a latent Gaussian process. Our two-stage specification resulted in a multivariate Student’s $t$ process emulator. In the IGP models, $y_k(\cdot)$-specific correlation parameters were estimated via restricted maximum likelihood whereas in the CoH dependence model the posterior density of the common correlation parameter vector given the data was maximized to obtain an estimate of the posterior mode. These correlation parameter vectors were plugged into the emulator, causing an understatement in the prediction variance.

We demonstrated the viability of the $EmaX$ algorithm on five MOP’s with relatively low dimensionality in either the design and/or objective spaces and of various degrees of difficulty in terms of the shape of the Pareto front. The performance of the expected Pareto improvement was compared with a contending improvement function called the probability of improvement or $PI$. We also compared the merits of modeling the dependencies among the objectives by contrasting the approximation sets obtained using the independence and dependence models implemented with the $EmaX$ improvement criterion.

4.1 Summary of Performance Assessments

We assessed the performance of the algorithms, $IGP-PI$, $IGP-EmaX$, and $CoH-EmaX$ based on the quality of approximation of the Pareto Front. These algorithms were also individually contrasted with what we called the best Pareto front approximation generated by a fine grid approximation (as fine as allowed by our computational resources). In our appraisals, the approximation sets were examined visually by plotting the obtained solutions on the objective space and quantitatively by using
two figures of merit: the hypervolume indicator and the binary epsilon indicator. The number of identified solutions was also noted.

The plots provided us with a visual impression of the extent to which a minimally sufficient representation was realized by each algorithm. Recognizing the inherent multiobjectivity in making quality comparisons, only Pareto compliant indicators, that is, those indicators sensitive to the Pareto dominance relations between approximation sets, were used for the quantitative comparison, namely the hypervolume indicator and binary epsilon indicator. These measures partly indicate minimal sufficiency in the representation. The hypervolume indicator gives the size or volume of the dominated region and is maximized by the inclusion of Pareto points. Larger values indicate better performance. The binary epsilon indicator measures the amount, or the epsilon factor by which an approximation set needs to be improved to be as good as a given reference set, which we fixed equal to the grid-approximated Pareto front. The smaller the epsilon factor, the better the approximation.

The algorithm \textit{IGP-EmaX} generally produced approximation sets showing a more even coverage of the Pareto front compared to the \textit{IGP-PI} which yielded a rather patchy representation due to the clustering of solutions in certain areas of the objective space. As Keane (2006) noted, the PI does not necessarily promote a thorough exploration of the solution space because the potential design points are not, we quote, “biased by the degree of improvement achieved”.

Not surprisingly, the hypervolume measure attained by \textit{IGP-EmaX} was greater than \textit{IGP-PI}, though the disparity does not reflect the appreciable difference in coverage seen in the plots. One exception was in the case of the \textit{dltz1a} test function where the \textit{IGP-PI} outperformed \textit{IGP-EmaX}, though both algorithms fared acceptably well in this MOP. This means that overall, the \textit{IGP-EmaX} dominated a larger portion of the solution space than did \textit{IGP-PI}. 

147
In terms of the binary epsilon indicator, the IGP-EmaX did undeniably better in the following test functions: wsnl, obj4d2, and hedarg2. The performance was more or less equal in knowles. The IGP-PI algorithm beat IGP-EmaX in dltz1a. These statements are based on the values of $I_\epsilon(P_o, P_G)$. This meant that the algorithm that had a better (smaller) epsilon factor needed to improve by a smaller measure in order to weakly dominate or be as good as the reference set.

The effect of enlarging the initial experimental design from 5d to 10d, on the hypervolume indicator and binary epsilon indicator appeared to be specific to the MOP and also to the algorithm. Starting with more training data did not necessarily increase the hypervolume measure or reduce the epsilon factor. Worse for some MOP’s, the algorithms’ hypervolume and binary epsilon indicators actually degraded. A plausible explanation could be the ill-conditioning in at least one of the $m$ covariance matrices of size $n \times n$ since the IGP models were fitted separately. With a bigger initial design, an algorithm could detect the efficient solutions early on the sequential stage. Since the stopping criterion is based on exhaustion of the sampling effort rather than on performance, after a certain time, succeeding iterations will tend to locate neighboring efficient points which in turn encourages ill-conditioning when the covariance matrices are inverted.

To compare the merit of accounting for the possible dependence structure in the objective vector, we compared the plots and the performance metrics of the IGP-EmaX and CoH-EmaX algorithms. Visually, the approximation sets delivered solutions that were comparably spread out, although the solutions from CoH-EmaX appeared closer to the Pareto front than those from IGP-EmaX. This observation is reinforced by the hypervolume indicator showing that the CoH-EmaX consistently covered a larger dominated portion of the objective space than the IGP-EmaX. The binary epsilon indicator also supported the visual impression that the
dependence-model guided the \textit{EmaX} algorithm to a more accurate representation of the Pareto front than the independence-model competitor. When compared to the grid-approximated Pareto front, the approximation set generated by \textit{CoH-EmaX} yielded a smaller $I_\epsilon(P^o, P^G)$ more often than \textit{IGP-EmaX}. The exceptions were in \textit{knowles} with an initial size of 10d where the $I_\epsilon(P^o, P^G)$ were marginally different, and in \textit{hedarg2} with 5d, where \textit{IGP-EmaX} significantly outperformed \textit{CoH-EmaX}.

As regards the effect of starting the algorithm with more initial observations on the hypervolume and binary epsilon indicators, the conclusion was different across the MOP’s considered. In \textit{knowles}, for instance, the degradation on both indicators from 5d to 10d was considerable in \textit{CoH-EmaX}. In the case of \textit{IGP-EmaX}, starting with more initial data did not practically change both indicators—the hypervolume measure slightly improved and the binary epsilon indicator slightly worsened. With \textit{wsnl, obj4d2}, and \textit{dltz1a} the metrics either improved appreciably or degraded very slightly in \textit{CoH-EmaX}. A different picture is seen in IGP, with MOP’s \textit{obj4d2} and \textit{dltz1a} where a worsening of both metrics occurred as a result of increasing the initial training data.

The \textit{EmaX} algorithms showed reasonably acceptable approximation sets considering a relatively tight budget of 20d runs inclusive of the initial training data, except in the \textit{knowles} MOP where the Pareto front was disconnected—only one part of the true front was well represented.

\section*{4.2 Directions for Future Work}

The current research has inspired a number of ideas for future investigation within the scope of GP-guided sequential designs. Five main themes will be elaborated below:
• Refinements to the current EmaX algorithm to increase algorithmic efficiency and precision

• Enhancements to multivariate emulation in particular, developing or exploring other dependence models that can account for the dependence structure of the objective vector

• Alternatives to improvement criteria to direct the progressive sampling of solutions towards the Pareto front

• Extension of the EmaX algorithm to accommodate more general design spaces, for example those that allow non-rectangular feasible spaces, include discrete inputs, or environmental or noisy inputs

• Consideration of novel applications of the black box MOP such as using the obtained approximation set to screen active design variables as a preliminary or complimentary analysis to more formal test

• Systematizing the evaluation and comparison of the performance of the algorithms by formulating the task as an experimental design problem to broaden and make stronger inferential statements regarding the performance of the EmaX algorithm.

4.2.1 Refinements to the EmaX Algorithm

There is ample room for improving the implementation of the EmaX algorithm to increase its efficiency and the precision of the resulting approximation set. One route to enhanced efficiency involves the use of an adaptive update process of the GP parameters. Presently, the EmaX algorithm re-fits the GP model at every stage of the
sequential design. In this regard, the kriging update strategy proposed by Gano, Renaud, Martin, and Simpson (2006) in variable-fidelity optimization can be assimilated into the IGP models. They propose a strategy called trust-region metamodel update management strategy or TR-MUMS which updates the model parameter estimates only when they produce a poor approximation which indicated by a quantity called the trust region ratio (TR) value. Consider a “TR-like” quantity:

\[
\rho^n_k = \frac{y_k(x_o) - y_k(x^*_n)}{\hat{y}_k(x_o) - \hat{y}_k(x^*_n+1)}
\] (4.1)

where \(x_o \in P^n\) is an initial value for the maximization of the expected Pareto improvement given \(D_n\), \(x^*_n+1\) is the obtained maximizer of the expected Pareto improvement, \(y_k(x_o)\) is the \(k^{th}\) objective function associated with \(x_o\), \(\hat{y}_k(x^*_n+1)\) is the predicted value of the \(k^{th}\) objective function of \(x^*_n+1\) computed during optimization, and \(y_k(x^*_n+1)\) is the \(k^{th}\) objective function after evaluating the simulator at \(x^*_n+1\). The quantity \(\rho^n_k\) is the ratio of the actual change in \(y_k(x)\) to the predicted change using the GP emulator. The ratio is calculated after \(y(x^*_n+1)\) is verified by the expensive code and indicates whether updates are necessary at the next sequential stage. If \(\rho^n_k \approx 1\), then no update is needed, however if the ratio is significantly less than unity then a refit of the GP model is necessary. A cut-off value can be specified for this purpose.

Another opportunity to gain efficiency is to employ an accurate GP emulator early on. In the numerical experiments conducted by Liu (2003), Integrated Mean Square-Gradient Enhanced kriging approach (IMSE-GEK) showed a substantial reduction in the prediction error when compared to kriging without using gradient information. Defining the error as,

\[
e_i = y_k(x_i) - \hat{y}_k(x_i)
\]
three error summaries were considered—the maximum prediction error, the mean prediction error, and the mean of the root MSE over a set input sites outside the training data. Interestingly, a reduction was observed in all these measures. The reason for preferring IMSE-GEK to MLE was not emphasized, and it may be worthwhile to determine the benefit of plug-in IMSE estimates in computing the empirical best linear unbiased estimate (EBLUP). Liu utilized the trapezoidal rule to numerically obtain the IMSE. The method of improving prediction accuracy by kriging with information on derivatives of the $y_k$ was presented earlier by citeNmorris93 and described in Santner, Williams, and Notz (2003, Forrester, Sobester, and Keane (2008).

Another refinement that increases efficiency and accuracy of the resulting approximation set can be made in the determination of the expected Pareto improvement. Currently, a Monte Carlo estimate based on 5000 draws from the emulator or predictive distribution is used, without resorting to any variance-reduction techniques.

The current implementation can also benefit from sophisticated numerical techniques or strategies to make the computations more numerically stable, for instance in the inversion of the $mn \times mn$ covariance matrices and the constrained optimization of the likelihood function to obtain the posterior mode EBLUP in the case of the Conti, Gosling, Oakley, and O’Hagan (2007) separable dependence model.

When the covariance matrix is badly-conditioned, the condition number is large and inaccuracies in the estimation occur. More precisely, consider the estimation of the correlation parameters in the IGP models were separate univariate GP models were fit for each $y_k(\cdot)$, $k = 1, \ldots, m$. Then $k^{th}$ prediction variance $s_k^2(x_o)$ at the design point $x_o$ is given by

$$s_k^2(x_o) = \hat{\sigma}_k^2 \left[ 1 - \left( \begin{array}{c} 1 \\ r_k(x_o) \end{array} \right)^T \left( \begin{array}{cc} 0 & 1^T \\ 1 & R_k \end{array} \right)^{-1} \left( \begin{array}{c} 1 \\ r_k(x_o) \end{array} \right) \right]$$
When the \( n \times n \) correlation matrix \( R \) is ill-conditioned, then its condition number, \( \text{cond}(R) \) is large, which implies that

\[
\sigma^2_k = \frac{1}{n-1}(y_n - \hat{\beta})^T R^{-1} (y_n - \hat{\beta})
\]

will also be large because it depends on \( R \). Thus small round-off errors made in \( r_k(x_o) \) can be amplified by a large \( \text{cond}(R) \), and the inflated value of \( \sigma^2_k \) also amplifies the round-off error. Thus numerical errors will unduly inflate \( s_k^2(x_o) \) and \( \sigma^2_k \) with adverse implications on the predictive distribution from where the Monte Carlo sample is drawn for computing the expected Pareto improvement. To mitigate the adverse effects of ill-conditioning, Liu (2003) introduced a constraint on the condition number of the covariance matrix, \( \text{cond}(R) < 1E6 \) in the non-linear optimization. The same strategy can be applied for obtaining the REML or posterior mode EBLUP’s.

4.2.2 Enhancing Multivariate Emulation

Two issues suggest themselves in this area, namely, the application of novel covariance models and the use of alternative improvement functions to guide the sequential optimization design.

Capturing the dependence structure in the objective vector is a study on its own. As a start, other GP covariance models can be formulated to accommodate the dependence, for instance a non-separable dependence structure, or another model that falls outside the class covered by the constructive approach discussed in Section 1.3.

A different, but not unrelated question, is whether the MOP can be reduced by means of a principal component analysis so that the dependence structure can be expressed as a meaningful linear combination or a contrast. A corollary idea would
then be to use these combinations to reduce the dimensionality of the MOP in a sensible way.

### 4.2.3 Alternative Sequential Sampling Criteria

On the subject of alternatives to improvement criteria, interesting work has been initiated on a competing sampling criterion called the *conditional minimizer entropy* or *CME* by Villemonteix, Vasquez, Sidorkiewicz, and Walter (2007) for global optimization. In a nutshell, the CME chooses design points that minimize the uncertainty in the location of minimizers. A probability distribution of minimizers is first approximated using conditional simulations of the GP given the available evaluations. The entropy of the conditional distribution of global minimizers is then taken as the uncertainty measure. The next design point is then identified by minimizing the expected entropy.

The performance of the enhanced probability of improvement proposed by Hawe and Sykulski (2007) can also be investigated in subsequent comparisons with *EmaX*, rather than Keane’s probability of improvement which showed a weakness in spreading out the solutions along the Pareto front. The enhanced probability

### 4.2.4 Extension of the *EmaX* Algorithm to more general spaces

Real-world multiobjective problems contend with various constraints, deal possibly with discrete or qualitative design variables and may include environmental variables, that is, those that are subject to random variation. Extending the *EmaX* algorithm to accommodate these characteristics makes it more practical and realistic. To this end, we can capitalize on efforts spent previously on the case of univariate GP’s.
We mention some starting points. With regard to handling non-rectangular spaces, the work on constrained global optimization by Sasena, Papalambros, and Goovaerts (2002). On the treatment of the GP covariance when qualitative inputs are present, Han, Santner, Notz, and Bartel () provide some leads. On handling environmental inputs for robust optimization, (Lehman, Santner, and Notz 2004) proposed a GP-guided sequential design for finding control variable settings $x^*_c$ at which a response $y_k(x^*_c)$ was insensitive to random variations in $x_e$, a set of environmental variables. On a different front, Stinstra and Hertog (2008) considered the problem of optimizing a response $y_k(x)$ such that the solution was robust against the inherent bias in the simulation model, the surrogate, as well as errors in the implementation of the code.

4.2.5 Novel Applications of Pareto set approaches

Once an approximation of the Pareto set is obtained, simple plots of the individual objectives versus each design variable can help identify design inputs that are active or consequential in producing well-performing solutions, as well as those that do not apparently have any effect.

In Handl and Knowles (2007), five modes of problem solving using multiobjective optimization (MOO) are discussed. Aside from the standard multiobjective optimization problem, they identified other problems that could be solved using a multiobjective approach. We present two.

One such problem utilizes the MOO as a tool to offset a measurement bias affecting an objective function. We describe the setting assuming one primary objective to be optimized. Let

$$y(x) = y^*(x) + m(g(x)), \ x \in \mathcal{X}$$
where $y^*(x)$ is an unknown and unbiased measure of the primary objective, $m(g(x))$ is a bias term with $m$ being an unknown monotone function of an observable function $g(x)$, and $y(x)$ is an observable but biased sum of $y^*(x)$ and $m(g(x))$. Suppose the minimum of $y^*(x) = y(x) - m(g(x))$ is sought. With $m$ being unknown, the single objective formulation cannot be solved. By reformulating the problem as a bi-objective problem however, a solution can be found:

$$\min_{x \in \mathcal{X}} (y(x), -g(x)).$$

Another mode is given as multiobjectivization where the MOO is used as a means to obtain an “improved search guidance” in what is originally a scalar optimization problem. An optimization problem can be rendered difficult by the properties of its search landscape: presence of excessive local optima or flat regions, those offering little or no objective function gradient. In the first scenario, the primary objective can be decomposed into several different functions in a bid to separate out the conflicting aspects of the problem, thereby reducing the number of local optima encountered by the search algorithm. In the second scenario, some “helper objectives” can be used in addition to the primary objective that might guide the search in the flat regions. They cite references for concrete examples.

### 4.2.6 Systematizing the evaluation and comparison of the performance of the algorithms

Zhao (2007) investigated the performance of different stochastic multiobjective optimizers on a judiciously chosen test set, known as the WFG suite which allows problem features to be varied systematically. In doing so, knowledge of the features that present difficulties for certain algorithms is gained. The test suite also permits
the experimenter to combine certain problem features to see how the algorithm’s performance changes. The point is that when testing in a controlled experiment, more definite statements can be made about the algorithm’s strengths and weaknesses.

Improving the integrity of the experiment also calls for upgrading the performance metrics, specially the hypervolume indicator, which was shown to be deficient in indicating the actual size of the dominated region when the objective space was non-convex or irregularly-shaped.

The comparison would also be greatly benefit from advances made with respect to visualization in higher dimensions.
APPENDIX A

LIST OF NOTATION
### A.1 Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>Autoregressive</td>
</tr>
<tr>
<td>BLUP</td>
<td>Best linear unbiased predictor</td>
</tr>
<tr>
<td>CE</td>
<td>Computer experiments</td>
</tr>
<tr>
<td>CME</td>
<td>Conditional Minimizer Entropy</td>
</tr>
<tr>
<td>DI</td>
<td>Desirability index</td>
</tr>
<tr>
<td>DLTZ</td>
<td>Deb, Laumanns, Thiele, and Zitzler</td>
</tr>
<tr>
<td>EA</td>
<td>Evolutionary Algorithm</td>
</tr>
<tr>
<td>EBLUP</td>
<td>Empirical best linear unbiased predictor</td>
</tr>
<tr>
<td>EGO</td>
<td>Expected Global Optimization</td>
</tr>
<tr>
<td>EI</td>
<td>Expected improvement</td>
</tr>
<tr>
<td>EmaX</td>
<td>Expected maximin</td>
</tr>
<tr>
<td>CoH</td>
<td>Conti and O’Hagan</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic algorithm</td>
</tr>
<tr>
<td>GP</td>
<td>Gaussian Process</td>
</tr>
<tr>
<td>IGP</td>
<td>Independent Gaussian process</td>
</tr>
<tr>
<td>LHD</td>
<td>Latin Hypercube Design</td>
</tr>
<tr>
<td>LSL</td>
<td>Lower Specification Limit</td>
</tr>
<tr>
<td>ML</td>
<td>Maximum likelihood</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Squared Error</td>
</tr>
<tr>
<td>MSPE</td>
<td>Mean Squared Prediction Error</td>
</tr>
</tbody>
</table>
MVN — Multivariate Normal distribution
MOP — Multiobjective Optimization Problem
PI — Probability of Improvement
PSP — Pareto Set Pursuing method
REML — Restricted maximum likelihood
RSS — Residual sum of squares
USL — Upper Specification Limit
WSNL — Williams, Santner, Notz, and Lehman
A.2 Symbols

\(d\) — number of design variables
\(m\) — number of objectives
\(n\) — size of the training data
\(n_{pf}\) — size of the current non-dominated set
\(p\) — total number of regression functions
\(p_k\) — number of regression functions of the \(k^{th}\) response
\(\mathcal{X}\) — \(d\)-dimensional design space
\(\mathcal{Y}\) — \(m\)-dimensional objective space
\(\mathcal{Z}\) — standardized objective space, \([0, 1]^m\)
\(\mathbf{x}\) — \(d \times 1\) vector of input variables
\(\mathbf{y}\) — \(m \times 1\) vector of simulator outputs
\(\mathbb{R}_d\) — \(d\)-dimensional Euclidean space
\(\mathbb{R}_{n,m}\) — \(n \times m\) matrix of reals
\(\mathcal{D}_n\) — current experimental design with \(n\) design points
\(\mathbb{R}_{m,m}^+\) — \(m \times m\) positive definite matrix of reals
\(\mathbf{I}_n\) — \(n \times n\) identity matrix
\(\mathcal{P}_\mathcal{Y}\) — true Pareto Front, subset of \(\mathcal{Y}\)
\(\mathcal{P}_\mathcal{X}\) — true Pareto set of design vectors, subset of \(\mathcal{X}\)
\(\mathcal{P}^n_\mathcal{Y}\) — current non-dominated front
\(\mathcal{P}^n_\mathcal{X}\) — current non-dominated set of design vectors
\(Y(x_0)\) — \(m \times 1\) response vector at unseen design point \(x_0\)

\(Y(x)\) — \(m \times 1\) vector of simulator outputs at design point \(x \in \mathcal{X}\)

\(Y_k\) — \(n \times 1\) vector of the responses on \(Y_k(x_i), \ 1 \leq i \leq n\)

\(Y(\cdot)\) — \(m\)-dimensional Gaussian process

\(W(\cdot)\) — \(m\)-dimensional zero mean Gaussian process

\(W_k\) — \(n \times 1\) vector of the latent process variables \(W_k(x_i), \ 1 \leq i \leq n\)

\(Z(\cdot)\) — \(m\)-dimensional zero mean, unit variance Gaussian process

with mutually independent components

\(K(\cdot)\) — covariance function of \(W(\cdot)\)

\(K_Z(\cdot)\) — covariance function of \(Z(\cdot)\)

\(\beta_k\) — \(p \times 1\) regression parameter vector associated with \(Y_k(\cdot)\)

\(B\) — \(p \times m\) matrix of unknown regression coefficients

\(f_k(\cdot)\) — \(p_k \times 1\) vector of known regression functions associated with \(Y_k(\cdot)\)

\(f(\cdot)\) — \(p \times 1\) vector of shared regression functions

\(F_n\) — \(np \times 1\) vector of regression functions obtained by stacking

\[f(x_i), \ 1 \leq i \leq n\]

\(\mathcal{F}\) — \(mn \times p\) matrix of known regression functions

\(Y_{n,m}\) — \(n \times m\) data matrix of outputs

\(W_{n,m}\) — \(n \times m\) matrix of \(W(x_i), \ 1 \leq i \leq n\)

\(Z_{n,m}\) — \(n \times m\) matrix of \(Z(x_i), \ 1 \leq i \leq n\)

vec \(Y_{n,m}^T\) — \(mn \times 1\) vector of outputs obtained by stacking

\[Y(x_i), \ 1 \leq i \leq n\]
\[ \tilde{Y} \] — vec \( Y_{n,m} \)

vec \( W^{T}_{n,m} \) — \( mn \times 1 \) vector of outputs obtained by stacking
\[ W(x_i), \ 1 \leq i \leq n \]

vec \( Z^{T}_{n,m} \) — \( mn \times 1 \) vector of outputs obtained by stacking
\[ Z(x_i), \ 1 \leq i \leq n \]

vec \( Y_{n,m} \) — \( mn \times 1 \) vector of outputs obtained by stacking
\[ Y_k, \ 1 \leq k \leq m \]

\( \mu \) — \( m \times 1 \) mean vector of the process \( Y(\cdot) \)

\( \beta \) — \( m \times 1 \) vector of regression parameters assuming a
constant mean function for each \( Y_k \)

\( \Sigma_0 \) — \( m \times m \) variance matrix of \( W(x) \)

\( \kappa_0 \) — \( mn \times m \) covariance matrix between \( Y(x_0) \) and \( Y(x_i), \ 1 \leq i \leq n \)

\( \psi \) — correlation parameter vector of \( W(\cdot) \)

\( \theta \) — \( md \times 1 \) correlation parameter vector of the process \( Z(\cdot) \)

\( \theta_k \) — \( d \times 1 \) correlation parameter vector associated with \( Z_k(\cdot) \)

\( \theta^* \) — \( d \times 1 \) correlation parameter vector assuming a
common correlation structure for the \( Z_k(\cdot) \)

\( \sigma^2_k \) — variance of \( Y_k(\cdot) \)

\( \rho_k(\cdot; \theta_k) \) — correlation function of \( Z_k(\cdot) \)

\( \Upsilon_{u,v}(\theta) \) — covariance between \( Z(x_u) \) and \( Z(x_v) \)

\( \Sigma W \) — \( mn \times mn \) variance matrix of vec \( W^{T}_{n,m} \)

\( \Sigma Z \) — \( mn \times mn \) variance matrix of vec \( Z^{T}_{n,m} \)

\( R(\theta^*) \) — \( n \times n \) correlation matrix in a separable cross-covariance structure
\( \Lambda \) — \( m \times m \) diagonal matrix with \( \sigma_k \) in the diagonals in a non-separable independence model

\( \Delta_{u,v} \) — \( m \times m \) matrix denoting the correlation between \( Z(x_u) \) and \( Z(x_v) \)

\( V(\psi) \) — \( \Sigma_{W_{n,m}}(\psi) \)

\( \Gamma \) — \( R(\theta^*) \) in a separable cross-covariance model

\( Q \) — \( F(FTV(\psi)^{-1}F) - 1 \)

\( l_R \) — restricted log-likelihood

\( L_{REML} \) — restricted maximum likelihood

\( \hat{Y}_0 \) — predictor of \( Y(x_0) \) given the data

\( \hat{\beta}_{GLS} \) — generalized least squares estimator of \( \beta \)

\( \hat{\Sigma}_{GLS} \) — generalized least squares estimator of \( \Sigma_0 \)

\( \hat{y}(x_0) \) — conditional mean of \( Y(x_0) \) given \( Y_{n,m} \)

\( \Sigma_{0|Y_{n,m}} \) — conditional variance of \( Y(x_0) \) given \( Y_{n,m} \)

\( I_M(x_0) \) — Pareto improvement at unseen design point \( x_0 \)

\( \varphi(x_i) \) — maximin fitness function evaluated at \( x_i \in D_n \)

\( \phi_m \) — \( m \)-dimensional multivariate normal density

\( \Phi(\cdot) \) — \( m \)-dimensional cumulative multivariate normal distribution

\( y^* \) — utopia point

\( y_* \) — nadir point
\( I_H(A) \) — hypervolume indicator of an approximation set \( A \)

\( I_\epsilon(A, B) \) — binary epsilon indicator of an approximation set \( A \)

with respect to a reference set \( B \)

\( \mathcal{P}_Y^G \) — grid-approximated Pareto front

\( \mathcal{P}_Y^{iPI} \) — approximation set produced by the \( IGP-PI \) algorithm

\( \mathcal{P}_Y^{iEmX} \) — approximation set produced by the \( IGP-EmaX \) algorithm

\( \mathcal{P}_Y^{cEmX} \) — approximation set produced by the \( CoH-EmaX \) algorithm

\( z_j(x) \) — standardized \( y_k(x) \)

\( \det(A), |A| \) — determinant of a square matrix \( A \)

\( 0_{n,p} \) — \( n \times p \) matrix of zeros

\( \pi(\beta) \) — the distribution of the random variable \( \beta \)
Bibliography


