EFFICIENT MULTI-OBJECTIVE SURROGATE OPTIMIZATION OF COMPUTATIONALLY EXPENSIVE MODELS WITH APPLICATION TO WATERSHED MODEL CALIBRATION

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Taimoor Akhtar

August 2015

© 2015 Taimoor Akhtar
ALL RIGHTS RESERVED

EFFICIENT MULTI-OBJECTIVE SURROGATE OPTIMIZATION OF COMPUTATIONALLY EXPENSIVE MODELS WITH APPLICATION TO WATERSHED MODEL CALIBRATION

Taimoor Akhtar, Ph.D.

Cornell University 2015

This thesis introduces efficient algorithms for multi-objective optimization of computationally expensive simulation optimization problems. Implementation of efficient algorithms and their advantage of use for calibration of complex and deterministic watershed simulation models is also analyzed.

GOMORS, a novel parallel multi-objective optimization algorithm involving surrogate modeling via Radial Basis Function approximation, is introduced in Chapter 2. GOMORS is an iterative search algorithm where a multi-objective search utilizing evolution, local search, multi method search and non-dominated sorting is done on the surrogate function to select numerous points for simultaneous expensive evaluations in each algorithm iteration. A novel procedure, "multi-rule selection", is introduced that simultaneously selects evaluation points (which can be computed in parallel) within an algorithm iteration through different metrics. Results are compared against ParEGO and the widely used NSGA-II on numerous test problems including a hypothetical groundwater PDE problem. The results indicate that GOMORS outperforms ParEGO and NSGA-II within a budget of 400 function evaluations. The superiority of performance of GOMORS is more evident for problems involving a large number of decision variables (15-25 decision variables).

The second contribution (Chapter 3) to the thesis is a comparative analysis of

algorithms for multi-objective calibration of complex watershed models. Since complex watershed models can be computationally expensive, we analyze and compare performance of various algorithms within a limited evaluation budget of 1000 evaluations. The primary aim of the analysis is to assess effectiveness of algorithms in identifying "meaningful trade-offs" for multi-objective watershed model calibration problems within a limited evaluation budget. A new metric, referred as the Distributed Cardinality index, is introduced for quantifying the relative effectiveness of different algorithms in identifying "meaningful trade-offs". Our results indicate that GOMORS (the algorithm introduced in Chapter 2), outperforms various other algorithms, including ParEGO and AMALGAM, in computing good and meaningful trade-off solutions, within a limited simulation evaluation budget.

The third and final contribution (see Chapter 4) to the thesis is MOPLS, a Multi-Objective Parallel Local Stochastic Search algorithm for efficient optimization of computationally expensive problems. MOPLS is an iterative algorithm which incorporates simultaneous local candidate search on response surface models within a synchronous parallel framework to select numerous evaluation points in each iteration. MOPLS was applied to various test problems and multi-objective watershed calibration problems with 4, 8 and 16 synchronous parallel processes and results were compared against GOMORS, ParEGO and AMALGAM. The results indicate that within a limited evaluation budget, MOPLS outperforms ParEGO and AMALGAM for computationally expensive watershed calibration problems, when comparison is made in function evaluations. When parallel speedup is taken into consideration and comparison is made in wall clock time, the results indicate that overall performance of MOPLS is better than GOMORS, ParEGO and AMALGAM.

BIOGRAPHICAL SKETCH

Taimoor Akhtar was born on December 30, 1983 in Multan, Pakistan. After finishing his high school education in Multan, he moved to Lahore, Pakistan in 2000 to pursue undergraduate education at the Lahore University of Management Sciences in Computer Science. After finishing his undergraduate education, he worked for the United Bank as a Management Trainee and an Inventory Control Manager in Shahsons Pakistan (Pvt) Ltd from 2004 to 2007.

Taimoor joined Cornell University as an MEng Student in Operations Research and Information Engineering in August, 2007 after being awarded the Fulbright Scholarship. After finishing his Masters, he enrolled in the Ph.D. program in the Civil and Environmental Engineering Department at Cornell as a Fulbright scholar, under the supervision of Prof. Christine Shoemaker.

This document is dedicated to the loved ones I lost, including my grandfathers, who were and will always be a source of inspiration. I also dedicate this work to my parents, sister, brother, and Rasha, my lovely wife to be.

ACKNOWLEDGEMENTS

Firstly, I would like to thank Professor Christine Shoemaker, for her wonderful support throughout my Ph.D. program. The fact that she is a wonderful advisor is well known and well documented. She also goes an extra mile to support her students, and has not only been an academic advisor for me, but also a wonderful motivator, mentor and counselor. Peter Frazier has been a great motivator, and has provided wonderful support as a committee member . I sincerely thank him for the wonderful ideas he provided for my research and his motivation for my work. Peter Diamessis's enthusiasm and support have been extremely refreshing, and I thank him for being an integral member of my supervising committee. I am especially in debt to Oliver Gao, who acted as a committee member for the final stretch of my Ph.D. program, and went out of his way to assist me in finishing up my work at Cornell.

No words can describe the support my family has given to me. I love you all and salute you all for always being there for me. Baba, Ammi Sherry, Amna Waqas bhai, Rasha, Anum, Mahnoor, Ayan; you guys are awesome, and I cant thank you enough for being absolutely wonderful. This could not have been possible without your love and support.

My friends have been my family away from home and I thank them all for sometimes being the motivation I needed, and sometimes being the distraction I wanted. Ghazi, Adnan, Zulfi, AT, Bajwa, Kashif, Gul, Ammar, Nabeel, Uzair, Rahul, Basit, Freeha, Tushar, Ijlal bhai and Ali; Thank you all. A special mention of my musical friends is absolutely necessary. Sarim, Manan, Mahin, Panda and Kritarth, you guys have been amazing. My cosins and best of friends, have provided wonderful support despite being thousands of miles away. Saad, Arsal, Sunny Umair, Awais, Zubair, Junaid and Zain, thank you.

I would especially like to acknowledge the support of all of my colleagues, who have also been great friends. My office mates Sue Nee, Ying and Josh; the comedian down the hall, Antoine; John, Aman, Mint, Julie; working with you guys has been an absolute privilege. Finally, my greatest acknowledgement goes to the Environmental and Water Resource Systems (EWRS) Group: Christine Shoemaker, Jery Stedinger and Pete Loucks. Learning from all of you has been an honor, and I hope to make all of you proud in future. Thankyou EWRS, and thank you, Cornell.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Computationally expensive simulation models exist in high numbers. Some examples of such models are watershed simulations, groundwater model simulations, crashworthiness of cars, etc. Computationally expensive models usually act like black box functions, where the relationship between the model inputs and outputs is hard to define. These models aim to mimic natural phenomena, where the relationship between the varying inputs and outputs is highly complex. Due to the complexity of these models, computing the output of each simulation can be time-consuming. Models which take considerable time in evaluation of one scenario are commonly referred to computationally expensive simulations, or computationally expensive black box functions, since the relationship between inputs and outputs are hard to define.

Simulation models are developed to understand relationships between certain parameters and model outputs. Based on the understanding of these relationships, analysis and forecasting exercises can be carried out, in order to make informed and sophisticated decisions. Many situations exist where the aim is to deduce parameters / model inputs / decisions which lead to optimized values of certain objectives e.g., costs, benefits etc. In order to deduce optimized decisions / parameters, simulation models can be coupled with sophisticated optimization algorithms.

Simulation optimization problems can be divided into 2 broad categories: 1) calibration optimization and 2) decision optimization. For example, decision optimization involves the deduction of optimal decisions e.g., pumping rates for groundwater contamination cleanup (the objectives could be cost minimization,

cleanup maximization, etc.). Calibration optimization involves the deduction of values for simulation model parameters (which are unknown and hard to compute e.g., hydraulic conductivity in groundwater) that ensure that model outputs are as close to observed data as possible.

Within the watershed modeling paradigm, multiple criteria can be considered important for the calibration process. Hence, the watershed model calibration problem can be formulated as a multi-objective optimization problem [3] [2]. A multi-objective optimization problem can be solved by obtaining decision maker's preference, aggregating criteria into a single criterion, and solving a single-objective optimization problem to obtain a calibration. However, a predefined user preference might lead to a significant loss of information regarding trade-offs between conflicting objectives [3] [4]. A multi-objective optimization problem can also be solved by employing a *posteriori* [1] optimization strategy, where objectives are considered separately during the optimization process, and the output of the optimization process is a set of trade-off solutions, also called Pareto solutions. The availability of trade-offs provides added information to decision makers.

This thesis tackles simulation optimization problems for calibration and decision analysis within a multi-objective optimization framework. The simulation models under discussion are computationally expensive. Hence, the aim of this research is to 1) develop efficient algorithms for multi-objective optimization of computationally expensive simulation optimization problems (discussed in Chapters 2 and 4), and 2) apply efficient multi-objective optimization to assess and analyze the effectiveness of the developed algorithms for calibration of computationally expensive watershed models (Chapter 3).

To achieve the desired efficiency, iterative surrogate approximations of the objective functions are utilized and the algorithms are designed to be efficient when computation is done in parallel. The algorithms developed are new because of the way they use a combination of previously evaluated points and the surrogate approximations to select multiple points for simultaneous expensive evaluations in the next algorithm iteration.

1.1 References

[1] Jared L. Cohon and David H. Marks. A review and evaluation of multi-objective programing techniques. *Water Resources Research*, 11(2):208–220, 1975.

[2] Andreas Efstratiadis and Demetris Koutsoyiannis. One decade of multi-objective calibration approaches in hydrological modelling: a review. *Hydrological Sciences Journal*, 55(1):58–78, February 2010.

[3] Hoshin Vijai Gupta, Soroosh Sorooshian, and Patrice Ogou Yapo. Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information. *Water Resources Research*, 34(4):751, 1998.

[4] P Yapo, H Gupta, and S Sorooshian. Multi-objective global optimization for hydrologic models. *Journal of Hydrology*, 204(1-4):83–97, January 1998.

CHAPTER 2

MULTI OBJECTIVE OPTIMIZATION OF COMPUTATIONALLY EXPENSIVE MULTI-MODAL FUNCTIONS WITH RBF SURROGATES AND MULTI-RULE SELECTION[1]

2.1 Introduction

Multi-objective optimization (MO) approaches involve a large number of function evaluations, which make it difficult to use MO in simulation - optimization problems where the optimization is multi-objective and the nonlinear simulation is computationally expensive and has multiple local minima (multi modal). Many applied engineering optimization problems involve multiple objectives and the computational cost of evaluating each objective is high (e.g minutes to days per objective function evaluation by simulation) [14] [9]. We focus on special algorithms that aim to produce reasonably good results within a limited number of expensive objective function evaluations.

Many authors (for instance, Deb et al. [6] and Zhang et al. [53]) have successfully employed evolutionary strategies for solving multi-objective optimization problems. Even with the improvement over traditional methods, these algorithms require, typically, many objective function evaluations which can be infeasible for computationally expensive problems. Added challenges to multi-objective optimization of expensive functions arise with increase in dimensionality of the decision variables and objectives.

The use of iterative response surface modeling or function approximation

[1]This Chapter has been published as a Journal Paper in [1].

techniques inside an optimization algorithm can be highly beneficial in reducing time for computing objectives for multi-objective optimization of such problems. Since the aim of efficient multi-objective optimization is to identify good solutions within a limited number of expensive function evaluations, approximating techniques can be incorporated into the optimization process to reduce computational costs. Gutmann [17] introduced the idea of using radial basis functions (RBF) [4] for addressing single objective optimization of computationally expensive problems. Jin et. al. [20] appears to be the first journal paper to combine a non quadratic response surface with a single objective evolutionary algorithm by using neural net approximations. Regis and Shoemaker [38] were the first to use Radial Basis Functions (not a neural net) to improve the efficiency of an evolutionary algorithm with limited numbers of evaluations. Later they introduced a non-evolutionary algorithm Stochastic-RBF [37] , which is a very effective radial basis function-based method for single objective optimization of expensive global optimization problems. These methods have been extended to include parallelism [41]; high-dimensional problems [43]; constraints [39]; local optimization [50] [51]; integer problems [33] [32] and other extensions [40] [42]. Kriging-based methods have also been explored for addressing single objective optimization problems [18] [21] [12]. Jones et al. [21] introduced Efficient Global Optimization (EGO), which is an algorithm for single objective global optimization within a limited budget of evaluations.

Response surface methods have also become popular for multi-objective optimization problems, with kriging-surrogate techniques being the most popular. Various authors have used kriging-based methods by extending the EGO framework for multi-objective optimization of computationally expensive problems. For instance, Knowles [23] combined EGO with Tchebycheff weighting to con-

vert an MO problem into numerous single objective problems. Tsang et. al. [54] and Emmerich et. al. [11] combined EGO with evolutionary search assistance. Ponweiser et. al. [34] and Beume et. al. [3] explored the idea of maximizing expected improvement in hypervolume. Authors have also explored the use of other function approximation techniques inside an optimization algorithm, including Radial Basis Functions (RBFs) [52] [15] [26] [46] [22], Support Vector Machines [27] [44] and Artificial Neural Networks [8]. Evolutionary algorithms are the dominant optimization algorithms used in these methods. Some papers also highlight the effectiveness of local search in improving performance of response surface based methods [25] [46] [22].

Various authors have indicated that RBFs could be more effective than other approximation methods in multi-objective optimization of computationally expensive problems with more than 15 decision variables [10] [44] [31]. Various authors have employed RBFs for surrogate-assisted multi-objective optimization of expensive functions with focus on problems with more than 15 decision variables. For instance, Karakasis and Giannakoglou [22] employ RBFs within an MOEA framework and use an inexact pre-evaluation phase (IPE) to select a subset of solutions for expensive evaluations in each generation of an MOEA. They also recommend the use of locally fitted RBFs for surrogate approximation. Georgopoulou and Giannakoglou [15] build upon [22] by employing gradient based refinements of promising solutions highlighted by RBF approximation during each MOEA generation. Santana et. al. [46] divide the heuristic search into two phases where the first phase employs a global surrogate-assited search within an MOEA framework, and rough set theory is used in the second phase for local refinements.

This paper focuses on the use of RBFs and evolutionary algorithms for multi-objective optimization of computationally expensive problems, where the number of function evaluations are limited relative to the problem dimension (e.g. to a few hundred evaluations for the example problems tested here). An added purpose of the investigation is to be able to solve MO problems where the number of decision variables varies between 15 and 25.

To this effect we propose a new algorithm, GOMORS, that combines radial basis function approximation with multi-objective evolutionary optimization, within the general iterative framework of surrogate-assisted heuristic search algorithms. Our approach is different from prevalent RBF based MO algorithms that use evolutionary algorithms [52] [15] [26] [46] [22]. Most RBF based evolutionary algorithms employ surrogates in an inexact pre-evaluation phase (IPE) in order to inexpensively evaluate child populations after every MOEA generation. By contrast our proposed methodology employs evolutionary optimization within each iteration of the algorithm framework to identify numerous potential points for expensive evaluations. Hence, multiple MOEA generations evolve via surrogated-assisted search in each algorithm iteration in GOMORS.

The novelty of the optimization approach is in the amalgamation of various evaluation point selection rules in order to ensure that various value functions are incorporated in selecting some points for expensive evaluations from the potential points identified by surrogated-assisted evolutionary search during each algorithm iteration. The combination of multiple selection rules targets a balanced selection between exploration, exploitation and front diversification. The selection strategy incorporates the power of local search and also ensures that the algorithm can be used in a parallel setting to further improve its efficiency.

2.2 Problem Description

Let $\mathcal{D}$ (domain space) be a unit hypercube and a subset of $\mathbb{R}^d$ and x be a variable in the domain space, i.e., $x \in [0,1]^d$. Let the number of objectives in the multi-objective problem equal k and let $f_i(x)$ be the i^{ith} objective. f_i is a function of x and $f_i : \mathcal{D} \mapsto \mathbb{R}$ for $1 \leq i \leq k$. The framework of the multi-objective optimization problem we aim to solve is as follows:

$$\begin{aligned} &\text{minimize } F(x) = [f_1(x), \ldots, f_k(x)]^T \\ &\text{subject to } x \in [0,1]^d \end{aligned} \tag{2.1}$$

Our goal for the multi-objective optimization problem is (within a limited number of objective function evaluations) to find a set of Pareto-optimal solutions $P^* = \{x_i^* \mid x_i^* \in \mathcal{D}, 1 \leq i \leq n\}$. P^* is defined via the following definitions:

Domination: A solution $x_1 \in \mathcal{D}$ dominates another solution $x_2 \in \mathcal{D}$ if and only if $f_i(x_1) \leq f_i(x_2)$ for all $1 \leq i \leq k$, and $f_i(x_1) < f_i(x_2)$ for at least one $i \in \{1, \ldots, k\}$.

Non-Domination: Given a set of solutions $S = \{x_i \mid x_i \in \mathcal{D}, 1 \leq i \leq n\}$, a solution $x^* \in S$ is non-dominated in S if there does not exist a solution $x_j \in S$ which dominates x^*.

Pareto Optimality: A candidate solution $x^* \in S$ which is non-dominated in S is called a globally Pareto-optimal solution if $S = \mathcal{D}$, i.e., S is the entire feasible domain space of the defined problem.

2.3 Algorithm Description

2.3.1 General Framework

The proposed optimization algorithm is called Gap Optimized Multi-objective Optimization using Response Surfaces (GOMORS). GOMORS employs the generic iterative setting of surrogate-assisted algorithms in which the response surface model used to approximate the costly objective functions is updated after each iteration. Multiple points are selected for expensive function evaluation in each iteration, evaluated in parallel, and subsequently used to update the response surface model. The algorithm framework is defined below:

Step 0 - Define Algorithm Inputs:

M - Maximum number of expensive function evaluations

r - The Gap radius parameter used in Step 2.3

t - Number of expensive evaluations to be performed after each algorithm iteration

Step 1 - Initial Evaluation Points Selection: Select an initial set of points $\{x_1, \ldots, x_m\}$, where $x_i \in \mathcal{D}$, for $1 \leq i \leq m$, using an experimental design. Evaluate the objectives $F = [f_1, \ldots, f_k]$ at the selected m points, via expensive simulations. Let $S_m = \{x_i \mid x_i \in \mathcal{D}, i = 1, \ldots, m\}$ denote the initial set of evaluated points. Let $P_m = \{x_i \in S_m \mid x_i$ is non-dominated in $S_m\}$ be the set of non-dominated points from S_m.

Step 2 - Iterative Improvement: Run algorithm iteratively until termination condition is satisfied:

While $m \leq M$

> **Step 2.1 - Fit/Update Response Surface Models:** Fit/update response surface models for each objective based on the set of already evaluated simulation points S_m. Let $\widehat{F}_m(x) = [\widehat{f}_{m,1}, \ldots, \widehat{f}_{m,k}]$ be the inexpensive approximate objective functions

obtained by fitting a response surface model on S_m.

Step 2.2 - Surrogate Assisted Global Search: Use a multi-objective evolutionary algorithm (MOEA) to solve the following optimization problem:

$$\text{minimize: } \widehat{F}_m(x) = [\widehat{f}_{m,1}(x), \ldots, \widehat{f}_{m,k}(x)]^T \tag{2.2}$$

Let $\widehat{P}_m^A = \{x_1, \ldots, x_{n_A}\}$ denote the solutions obtained by solving Equation 2.2, i.e, utilizing an MOEA for searching on the response surfaces.

Step 2.3a - Identify Least Crowded Solution:. Using the crowding distance calculation procedure proposed by Deb et al. [6], identify the least crowded element of the expensively evaluated non-dominated set, P_m. Let x^{crowd} be the least crowded element of P_m (see definition in Table 4.1).

Step 2.3b - Local Search:. Use a multi-objective evolutionary algorithm in a small neighborhood of x^{crowd} to solve the following optimization problem:

$$\begin{aligned} &\text{minimize: } \widehat{F}_m(x) = \{\widehat{f}_{m,1}(x), \ldots, \widehat{f}_{m,k}(x)\} \\ &\text{subject to: } (x^{crowd} - r) \leq x \leq (x^{crowd} + r) \end{aligned} \tag{2.3}$$

The problem solved in Equation 2.3 we will call the "*Gap Optimization Problem*". Let $\widehat{P}_m^B = \{x_1, \ldots, x_{n_B}\}$ denote the solutions obtained by solving Equation (2.3) ,i.e, utilizing an MOEA for local search on the response surfaces, within a radius, r, around the least crowded solution, x^{crowd}.

Step 2.4 - Select Points for Expensive Function Evaluations: Our evaluation points are then selected from the sets $\widehat{P}_m^A$ and $\widehat{P}_m^B$ (from Equations 2.2 and 2.3, and also called candidate points) based on the rules described in Section 2.3.4. Let $S_{curr} = x_1, \ldots, x_t$ be the set of t points selected for expensive evaluations in current algorithm iteration. (Discussed in detail in Section 2.3.4)

Step 2.5 - Do expensive function evaluations and update Non-dominated solution set: Evaluate costly functions, F, on the selected evaluation points, S_{curr}. Update S_m, i.e., add the new expensively evaluated points to the set of already

evaluated points. Consequently, $m = m + t$, $S_m = \{S_m\} \cup \{S_{curr}\}$. Update P_m, i.e., compute $P_m = \{x_i \in S_m \mid x_i \text{ is non-dominated in } S_m\}$.

End Loop

Step 3 - Return Best Approximated Front: Return $P_M = \{x_i \in S_M \mid x_i \text{ is non-dominated in } S_M\}$ as an approximation to the globally optimal solution set, i.e, $P^* = \{x_i^* \mid x_i^* \text{ is non-dominated in } \mathcal{D}\}$.

The algorithm initiates with selection of an initial set of points for costly function evaluations of the objective function set, F. Latin hypercube sampling, a space-filling experimental design scheme proposed by McKay et al. [28] is used for selection of the initial evaluation set. The iterative framework of GOMORS (Step 2) follows, and consists of three major segments: 1) Building a Response Surface Model for approximating expensive objectives (Step 2.1), 2) Applying an Evolutionary algorithm for solving multi-objective response surface problems (Steps 2.2 and 2.3) and 3) Selecting multiple points for expensive evaluations from the solutions of the multi-objective response surface problems, and evaluating them in parallel (Step 2.4). The algorithm terminates after M expensive evaluations and the output is a non-dominated set of solutions P_M, which is an approximation to the true Pareto solution of the problem i.e P^*. Sections 2.3.2-2.3.4 and Appendix B discuss details of the major steps of the algorithm.

2.3.2 Response Surface Modeling

The first major component of the iterative framework is the procedure used for approximating the costly functions, F, by the inexpensive surrogates, $\widehat{F}_m(x) =$

Table 2.1: Definitions of Sets and Variables

Item	Description
$F(x) = [f_1(x), \ldots, f_k(x)]$	Expensive objectives for the multiple-objective optimization problem
$P^* = \{x_1^*, \ldots, x_n^*\}$	Set of Pareto-optimal solutions given the objectives F
$S_m = \{x_1, \ldots, x_m\}$	Set of points which have been evaluated via costly simulation until iteration m of the algorithm
$\widehat{F}_m(x) = [\widehat{f}_{m,1}, \ldots, \widehat{f}_{m,k}]$	$\widehat{f}_{m,i}$ is the inexpensive response surface approximation of the i^{th} objective, f_i, generated using the points in S_m
$P_m = \{x \in S_m\}$	The *non-dominated* solutions in S_m based on expensive evaluations
$x^{crowd} = \arg\max_{x \in P_m}[d(x)]$	The least crowded element in P_m, according to the crowding definition, $d(.)$, where $d(x_j)$ indicates the function by Deb [7] to compute crowding distance for x_j given $(x_j, F(x_j))$ for all $x_j \in P_m$
$\widehat{P}_m^A = \{x_1, \ldots, x_{n_A}\}$	Candidate points obtained by *Surrogate-Assisted Global Search* on the approximate objective function set $\widehat{F_m}$ (see Equation 2.2)
$\widehat{P}_m^B = \{x_1, \ldots, x_{n_B}\}$	Candidate points obtained by *Gap Optimization* of the approximate objectives, $\widehat{F_m}$ (see Equation 2.3)
$H_V(P)$	Hypervolume [2] of the objective space dominated by the objective vectors corresponding to the set P. The hyperspace is such that a point $y \in \mathbb{R}^k$ in the hyperspace is bounded by a reference vector b (See Figure 2.1).

$[\widehat{f_{m,1}}, \ldots, \widehat{f_{m,k}}]$. A response surface model based on the evaluated points is generated in Step 2.1 of the algorithm and subsequently updated after each iteration. For example artificial neural networks [36], Support Vector Machines (SVM) [48], kriging [45], and radial basis functions (RBFs) [4][35] could be employed to generate the response surface model.

While kriging has been used in MO [23] [54] [34], the number of parameters to be estimated for kriging meta-models increases quickly with an increase in the number of decision variables [19] [18], and hence, re-evaluating kriging surrogates in each iteration of a kriging based algorithm may itself become a computationally expensive step. Various authors [43] [31] including Manriquez et. al [10] have reported that kriging-based surrogate optimization is not effective for high dimensional problems (approximately defined as problems with more than 15 decision variables). In [10] they demonstrate the relative effectiveness of RBF approximation in tackling such high dimensional problems. The GOMORS algorithm proposed in this paper hence makes use of RBFs as the surrogate model for approximating the costly functions, although other surrogate models could be used in the GOMORS strategy.

2.3.3 Evolutionary Optimization

The surrogate optimization problems defined in Steps 2.2 (*Surrogate-assisted global search*) and 2.3 (*Gap Optimization*) aim to find near optimal fronts given that the expensive functions, $F(x)$, are replaced by the corresponding response surface model approximations, i.e., $\widehat{F_m}(x)$ (see Table 4.1). In Steps 2.2 and 2.3, two different multi-objective optimization problems are solved on the surrogate surface.

Steps 2.2 and 2.3 solve the MO problems depicted in Equations 2.2 and 2.3

respectively. Since the objective functions of these problems are derived from the surrogate model, solving the MO problems is a relatively inexpensive step. However, the MO problems of Equations 2 and 3 are not trivial and could potentially have non-linear and multi-modal objectives. Hence, we employ evolutionary algorithms for MO optimization of surrogates in Steps 2.2 and 2.3.

Three algorithm were considered as potential alternatives for optimization of surrogates in Steps 2.2 and 2.3, namely, NSGA-II, [6], MOEA/D [53] and AMALGAM [49]. NSGA-II handles the evolutionary search optimization process by ranking and archiving parent and child populations according to a non-domination sorting. MOEA/D [53] uses aggregate functions, and simultaneously solves many single-objective Tchebycheff decompositions of multi-objective problems in an evolutionary generation. AMALGAM [49] is a multi-method evolutionary algorithm, which incorporates search mechanics of various algorithms. Extensive computer experiments on test problems were performed on GOMORS with either of NSGA-II, MOEA/D and AMALGAM as embedded evolutionary schemes (see Section B.3 of Appendix B for a detailed discussion on the experiments) and AMALGAM was identified as the best performing evolutionary algorithm (although the differences were small) embedded in GOMORS.

2.3.4 Expensive Evaluation Point Selection - Step 2.4

Step 2.4 of the GOMORS algorithm determines which of the candidate points are evaluated by the expensive objective functions, $F(x)$, within an algorithm iteration. The candidate points are obtained from Steps 2.2 and 2.3, and are denoted as $\widehat{P}_m^A$ and $\widehat{P}_m^B$ respectively. As mentioned earlier, selection of points for expensive evaluation is a critical step in the algorithm because this is usually by

far the most computationally expensive step.

A balance between *exploration* [21], *exploitation* and *diversification* [7] is crucial for selecting points for expensive evaluations from candidate points, $\widehat{P}_m^A$ and $\widehat{P}_m^B$. *Exploration* of the decision space aims at selecting points in unexplored regions of the decision space. *Exploitation* aims at exploiting the inexpensive response surface approximations of Step 2.1 to assist in choosing appropriate points for expensive evaluations. *Diversification* strives to ensure that the non-dominated evaluated points are spread out in the objective space.

GOMORS employs a multi-rule based strategy for selection of a small subset of candidate points ($\widehat{P}_m^A$ and $\widehat{P}_m^B$) for actual functions evaluations (computing F via simulation). The various "rules" employed in the strategy target a balance between *exploration*, *exploitation* and *diversification*. A detailed framework of Step 2.4 of the algorithm is given below which gives an overview of the selection rules (The i^{th} rule is referred as Rule i) (Refer Table 4.1 for definitions of sets and symbols):

Details of Step 2.4 (in Section 2.3.1:)

Inputs from previous steps: (m, S_m, P_m, $\widehat{P}_m^A$, $\widehat{P}_m^B$)

Evaluation Points selection: Select $t = \sum_{i=0}^{4} t_i$ points for expensive evaluations via rules 0-4. Let S_{curr} be the selected points. So there are t_i points generated using Rule i, where the Rules are listed below.

Apply Rule 0 - Random Sampling: Pick t_0 points for expensive function evaluation via random sampling from a uniform distribution. Ensures exploration of decision space.

Apply Rule 1- Hypervolume Improvement: Use hypervolume improvement to choose t_1 points from the candidate points $\widehat{P}_m^A$. Let $H_V(P_m)$ denote the hypervolume [2] of the objective space dominated by the vectors in the set P_m (see Figure 2.1(a)). A new point for expensive function evaluation is selected as follows (see Table 4.1 for definition of

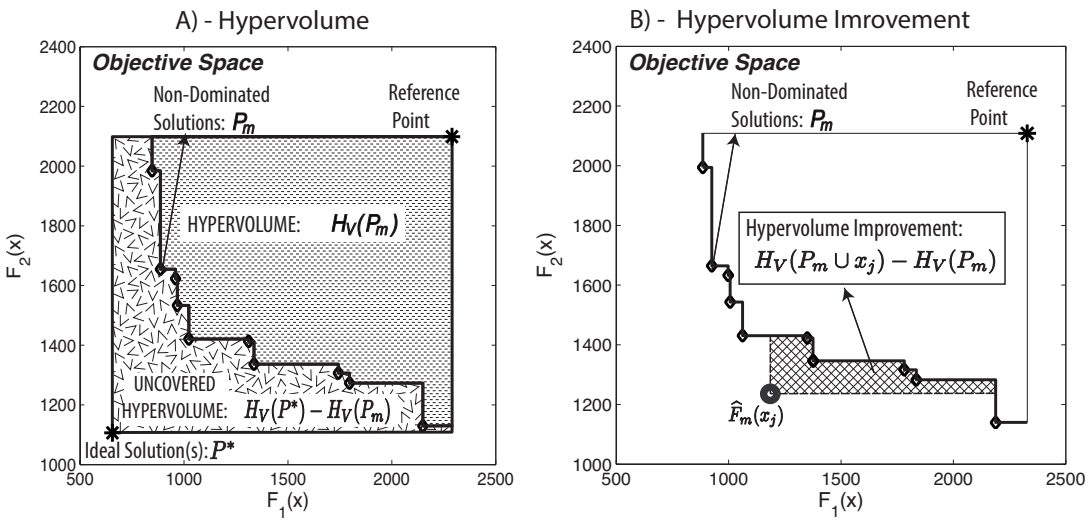

Figure 2.1: Visualization of a) Hypervolume and Uncovered Hypervolume and b) Hypervolume Improvement employed in Rule 1 and Rule 4 of Step 2.4 (discussed in Section 2.3.4). Maximization of Hypervolume Improvement implies selection of a point for function evaluation that predicts maximum improvement in hypervolume coverage of a non-dominated set as per RBF approximation.

sets and variables):

$$x^* = \arg\max_{x_j \in \widehat{P}_m^A} \left[H_V(P_m \cup x_j) - H_V(P_m) \right] \tag{2.4}$$

Equation 2.4 exploits the RBF approximation and chooses a point for function evaluation which maximizes the improvement in hypervolume in the objective space as per the RBF approximations of the candidate points, as illustrated in Figure 2.1(b).

Apply Rule 2- Maximize Minimum Domain Euclidean Distance: Let $x_{i,A} \in P_m^A$ and $x_j \in S_m$. Choose t_2 points for expensive evaluation such that the maximum of minimum distances of each point from already evaluated points is maximized, i.e.:

$$x^* = \arg\max_{x_{i,A} \in \widehat{P}_m^A} \left[\min_{x_j \in S_m} \|x_{i,A} - x_j\| \right] \tag{2.5}$$

Apply Rule 3- Maximize Minimum Objective Space Euclidean Distance: Choose t_3 points such that the maximum of minimum distances of each point from already eval-

uated points, as per the objective function space, is maximized, i.e.:

$$x^* = \arg\max_{x_{i,A} \in \widehat{P}_m^A} \left[\min_{x_j \in S_m} \|\widehat{F}_m(x_{i,A}) - F(x_j)\| \right] \tag{2.6}$$

Rule 2 and Rule 3 are a hybrid of exploration and exploitation.

Apply Rule 4 - Hypervolume Improvement in "Gap Optimization" candidates: Use "Rule 1" to select t_4 points from the candidate points obtained via "*Gap Optimization*", i.e, $\widehat{P}_m^B$. Select points for expensive function evaluations as follows (See Table 4.1 for definition of sets and variables):

$$x^* = \arg\max_{x_j \in \widehat{P}_m^B} \left[H_V(P_m \cup x_j) - H_V(P_m) \right] \tag{2.7}$$

The difference between (2.4) and (2.7) is that $\widehat{P}_m^A$ in (2.4) comes from Step 2.2 and $\widehat{P}_m^B$ in (2.7) comes from Step 2.3.

The number of points to be selected for expensive evaluation via each rule i.e, t_i may vary. However, we performed all our computer experiments with $t_i = 1$, for Rules 1-4. A point is selected via Rule 0 in each algorithm iteration with a probability of 0.1. Hence, either four or five points are selected for expensive function evaluations in each iteration of the algorithm. The expensive evaluations of points are performed in parallel to further speed up the algorithm.

In order to assess the individual effectiveness of the rules, we performed computer experiments on eleven test problems with the exclusive use of all rules (i.e, one point is selected from one rule, in all algorithm iterations). Furthermore, we also performed experiments with the idea of cycling between all rules in subsequent iterations of GOMORS framework. These different selection strategies were compared against the multiple rule selection strategy for simultaneous

selection of evaluation points (described above). Results of the computer experiments indicated that if the value of parallelization is considered, the multiple rule selection strategy outperforms the other strategies employed on our analysis (See Section B.4 of Appendix B for details.). However, if GOMORS is used in a serial setting, i,e, one point is evaluated in each algorithm iteration, cycling between rules, and use of Rule 3 are most beneficial (see Section B.4 of Appendix B).

2.4 Test Problems

In order to test the performance of GOMORS, computational experiments were performed on various test functions and a groundwater remediation problem. Certain characteristics of various problems can lead to inadequate convergence or poor distribution of points on the Pareto front. These characteristics include high dimensionality, non-convexity and non-uniformity of the Pareto front [7], the existence of multiple locally optimal Pareto fronts [5], low density of solutions close to the Pareto front, and existence of complicated Pareto sets (this implies that the Pareto solutions are defined by a complicated curve in the decision space) [24]. We have employed eleven test problems in our experimental analysis which incorporate the optimization challenges mentioned above. Five test problems are part of the ZDT test suite [5], while six were derived from the work done by Li and Zhang [24]. Mathematical formulations and optimization challenges of the test problems are discussed in detail in Section B.2.1 of Appendix B. These problems are collectively referred as synthetic test problems in subsequent discussions.

Table 2.2: Two MO variants, GWDA and GWDM, of the groundwater optimization problem (Section 2.4.1). M is total number of finite element grid nodes, N is the total number of wells, T is the total number of management periods, X_t is the pumping decision in management period t, $s_{m,t}$ is the contaminant concentration at grid m and at the end of the last time period T, and $C_t(X_t)$ is the cost of cleanup for the pumping decision X_t:

GWDA	**GWDM**
minimize: $f_1(X) = \sum_{t=1}^{T} C_t(X_t)$	minimize: $f_1(X) = \sum_{t=1}^{T} C_t(X_t)$
minimize: $f_2(X) = \frac{\sum_{m=1}^{M} s_{m,T}(X)}{M}$	minimize: $f_2(X) = \max_{m=1}^{M} s_{m,T}(X)$
subject to: $X \in [0,1]^{N \times T}$	subject to: $X \in [0,1]^{N \times T}$

2.4.1 Groundwater Remediation Design Problem

Optimization problems pertaining to groundwater remediation models usually require solving complex and computationally expensive PDE systems to find objective function values for a particular input [29]. The groundwater remediation problem used in our analysis is based on a PDE system which describes the movement and purification of contaminated groundwater given a set of bioremediation and pumping design decisions [30]. Detailed description of the problem is provided in Section B.2.2 of Appendix B. The decision variables of the problem are the injection rates of remediation agent at 3 different well locations during each of m management periods. The input dimension size ranges between 6 and 36 variables, depending upon the number of management periods incorporated in the numerical computation model.

The remediation optimization problem can be formulated as two separate multi-objective optimization problems, called GWDA and GWDM (Table 2.2).

In the first formulation, GWDM, the aim is to minimize the maximum contaminant concentration at the end of the pumping time, and to minimize the cost of remediation. The second formulation, GWDA, aims to minimize the average contaminant concentration, along with cost. The mathematical formulation of GWDA and GWDM are depicted in Table 2.2.

2.5 Results and Analysis

2.5.1 Experimental Setup

We tested our algorithm on the test functions and groundwater problems discussed in the previous section. The performance of GOMORS was compared against the Non-Dominated Sorting Algorithm-II (NSGA-II) proposed by Kalyanmoy Deb (2001) [6] (discussed in Section 2.3.3) and the kriging-based ParEGO algorithm proposed by Knowles [23]. All three algorithms are quite different. ParEGO is a multi-objective version of EGO [21] where the multi-objective problem is converted into many single objective optimization problems through Tchebycheff weighting [47]. One single objective problem is chosen at random from the predefined set of decomposed problems and EGO is applied to it to for selection of one point for evaluation per algorithm iteration. ParEGO is not designed for high dimensional problems (more than 15 variables). GOMORS on the other hand embeds RBFs within an evolutionary framework, and selects multiple points (from various rules defined in Section 2.3.4) for simultaneous (parallel) evaluations in each iteration and is designed for low and higher (15-25 decision variables) dimensional problems.

Since the objective of GOMORS is to find good solutions to MO problems within a limited function evaluation budget, our experiments were restricted to

400 function evaluations. Since all algorithms compared are stochastic, ten optimization experiments were performed for each algorithm, on each test problem, and results were compared via visual analysis of fronts and a performance metric based analysis. A detailed description of the experimental setup, including parameter settings for all algorithms and source code references for ParEGO and NSGA-II, is provided in Section B.5 of Appendix B.

The **uncovered hypervolume** metric [2] was used to compare the performance of various algorithms. Uncovered hypervolume is the difference between the total feasible objective space (defined by the reference and ideal points in Figure 2.1(a)) and the objective space dominated by estimate of the Pareto front obtained by an algorithm. A lower value of the metric indicates a better solution and the ideal value is zero.

Results from the synthetic test problems were analyzed in combination through the metric. Experiments were performed for each test problem with 8 decision variables, 16 decision variables and 24 decision variables to highlight performance differences of GOMORS, ParEGO and NSGA-II with varying problem dimensions. Results of all synthetic test problems were compiled for analysis by summing the metric values of each of the ten optimization experiments performed on each test problem. Since the uncovered hypervolume metric values obtained for each individual test problem and algorithm combination could be considered as independent random variables, a sum of them across a single algorithm is another random variable which is a convolution of the independent random variables. This convolution based metric summarizes overall performance of an algorithm on all synthetic test problems and is used as basis of our analysis methodology.

Results from the two variants of the groundwater remediation problem

(GWDA and GWDM) were also analyzed through the uncovered hypervolume metric in a similar manner. Experiments were performed with 6 decision variables, 12 decision variables and 24 decision variables for each groundwater problem. Results from individual subsequent experiments performed on each problem were summed to obtain a convoluted metric analysis of the groundwater problems. Performance of GOMORS and ParEGO on the GWDM test problem was further assessed through visual analysis of non-dominated solutions obtained from each algorithm (median solution). The numerical comparisons are based on the number of objective function evaluations and hence do not incorporate parallel speedup. ParEGO does not have a parallel implementation so its wall-clock time is much longer than GOMORS. However, the comparisons here focus on the number of function evaluations and evaluate an estimate of total CPU time. These comparisons do not consider the additional advantage of GOMORS (parallel) in wall-clock time.

2.5.2 Results: Synthetic Test Problems

The metric analysis performed on the synthetic test problems is depicted via box plots in Figure 2.2. The vertical axis in Figure 2.2 is the convoluted uncovered hypervolume metric described in Section 2.5.1. The lower the uncovered hypervolume, the better is the quality of the Non-dominated Front that the algorithm has found.

Figure 2.2 aims to 1) visualize speed of convergence of all algorithms (by comparing results of 200 and 400 objective function evaluations) and 2) understand the effect of increasing decision space dimensions on performance of algorithms (by comparing problems with 8, 16 and 24 decision variables).The box plot visualization of each algorithm within each sub-plot of Figure 2.2 corre-

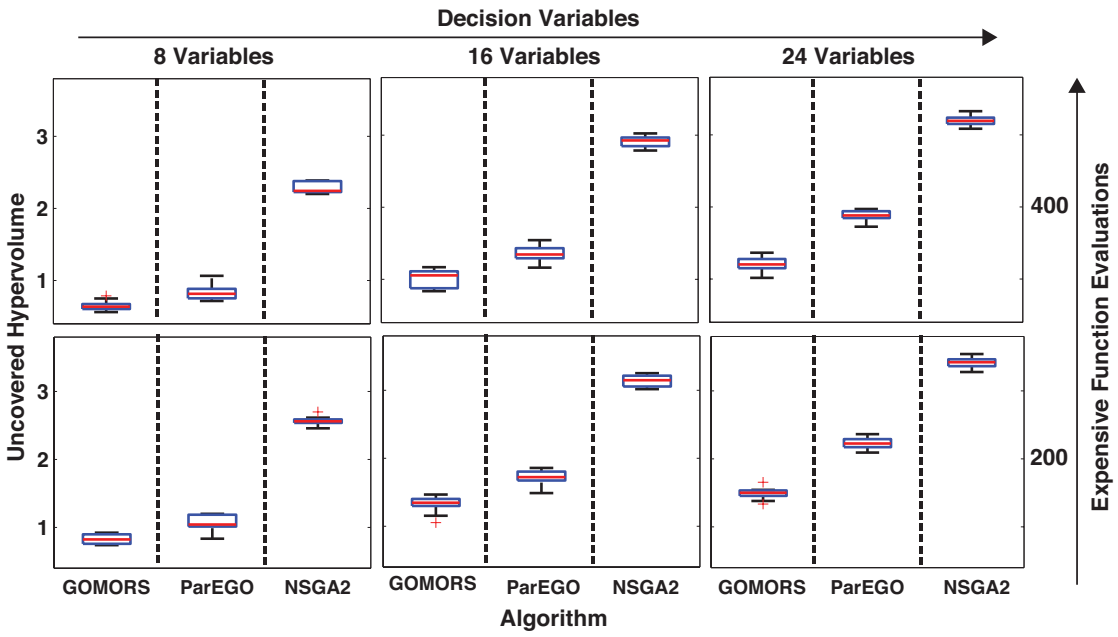

Figure 2.2: Synthetic Test Problems: Box plots of uncovered hypervolume metric values summed over the eleven test problems. Axis scales are identical across all sub-plots, which describe problems ranging from 8 to 24 decisions and from 200 to 400 function evaluations. Upper row is for 400 evaluations and lower row is for 200 evaluations. In all cases lower values of uncovered hypervolume are best.

sponds to the uncovered hypervolume metric values summed over the eleven test problems (see Section 2.5.1). Lower values of the metric signify superiority of performance and a lower spread within a box plot depicts robustness of performance. Each sub-plot within Figure 2.2 compares performance of all three algorithms for a specified number of decision variables (number of decision variables vary between 8 and 24) and a fixed number of function evaluations (either 200 or 400). Traversing from bottom to top, one can visualize the change in performance of algorithms as function evaluations increase (from 200 to 400), and a left to right traversal can help in visualizing performance differences with an increase in decision space dimensions (8, 16 and 24 decision variables).

Figure 2.2 clearly illustrates that GOMORS outperforms ParEGO for the 24 variable versions of the synthetic problems, both in terms of speed of convergence (at 200 evaluations) and at algorithm termination after 400 function evaluations (see top right sub-plot of Figure 2.2). GOMORS' convergence to a good solution is faster than ParEGO for the 8 and 16 variable versions of the synthetic test problems, but the difference is not as distinguishable (from Figure 2.2) as in the case of the 24 variable versions. The Wilcoxon rank-sum test [16] (at 5 percent significance) confirms that performance of GOMORS is better than ParEGO for the 8, 16 and 24 variable versions of the test problems after 100, 200 and 400 function evaluations. Figure 2.2 also indicates that both GOMORS and ParEGO significantly outperform NSGA-II with reference to the synthetic problems when evaluations are limited to 400.

2.5.3 Results: Groundwater Remediation Design Problem

The box-plot analysis methodology for the groundwater remediation problem is similar to the one employed for the synthetic test problems and the analysis is summarized in Figure 2.3. Results summarized in Figure 2.3 are consistent with the findings observed with the synthetic test functions in Figure 2.2. GOMORS and ParEGO both outperform NSGA-II for the two MO groundwater problems defined in Table 2.2, GWDA and GWDM, when function evaluations are limited to 400. Performance of GOMORS (as per Figure 2.3) is superior to ParEGO with application to the 12 and 24 variable versions of GWDA and GWDM, both in terms of speed of convergence (at 200 evaluations) and upon algorithm termination after 400 function evaluations. The difference between GOMORS and ParEGO is not visually discernible for the 6 variable versions of the problems, but the rank-sum test (at 5 percent significance) confirms that performance of

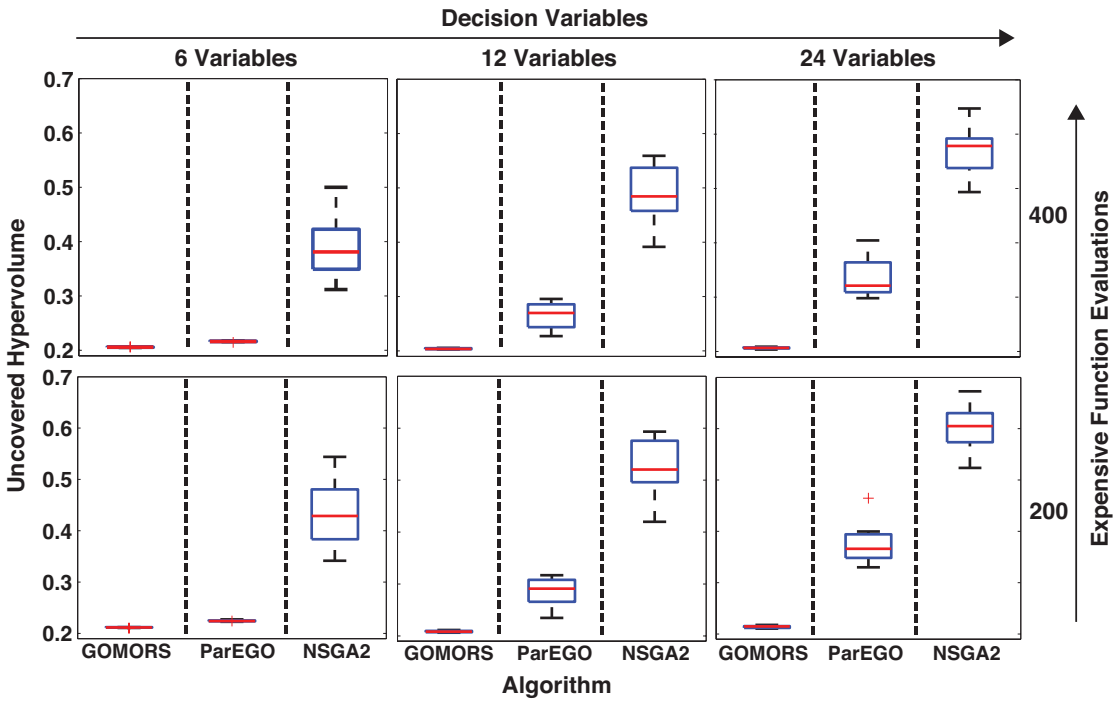

Figure 2.3: Groundwater application problems: Box plots of uncovered hypervolume metric values summed over the two groundwater remediation design optimization problems, i.e GWDA and GWDM. See Figure 2.2 caption for figure explanation.

GOMORS is better, which is supported by Figure 2.4.

Figure 2.4 provides a visual comparison of non-dominated trade-offs obtained from GOMORS and ParEGO with application to the GWDM groundwater problem. The red line within each sub-plot is an estimate of the Pareto front of GWDM. Since knowledge of the true front is not known, we obtained our estimate of true Pareto front through a single trial of NSGA-II with 50000 function evaluations. The green dots within each sub-plot correspond to the non-dominated solutions obtained via application of the algorithm referenced in the sub-plot. There are two sub-figures within Figure 2.4 and four sub-plots within each sub-figure. Sub-figure 2.4a corresponds to the median (the fronts ranked fifth in ten experiments of each algorithm) non-dominated fronts ob-

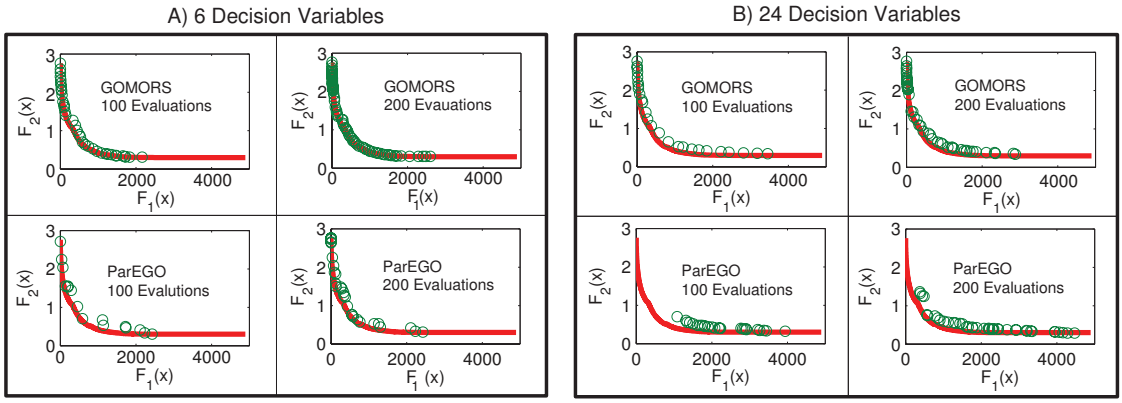

Figure 2.4: Estimated Non-dominated Front for Groundwater Problem GWDM: Visual comparison of median non-dominated fronts (as per uncovered hypervolume) obtained from GOMORS and ParEGO for GWDM with 6 and 24 decisions, and after 100 and 200 expensive function evaluations. Red line is result of 50,000 evaluations with NSGA-II. Green circles are non-dominated solutions from GOMORS or ParEGO.

tained from each algorithm for GWDM with 6 decision variables and sub-figure 2.4b corresponds to the median fronts obtained from each algorithm for GWDM with 24 decision variables. A clock-wise traversal of sub-plots within each sub-figure depict results in the following order: 1) GOMORS after 100 evaluations, 2) GOMORS after 200 evaluations, 3) ParEGO after 200 evaluations, and 4) ParEGO after 100 evaluations.

Figure 2.4 indicates that both algorithms seem to converge to the estimated Pareto front and also manage to find diverse solutions on the front for the 6 variable version of GWDM. More diverse solutions are obtained from GOMORS however, and with better speed of convergence than with ParEGO (depicted by the 6 variable version results after 100 and 200 evaluations). Performance of GOMORS is significantly better than ParEGO for the 24 variable version of the problem both in terms of speed of convergence (depicted by 24 variable GWDM version results after 100 and 200 evaluations) and diversity of solutions.

In case of computationally expensive real-world problems it may not be possible to run multiple MO experiments. Hence, we also visually analyzed trade-offs of worst case solutions (as per uncovered hypervolume) obtained from GOMORS and ParEGO for both GWDM and GWDA. The visualizations are provided in Section B.6 of Appendix B. After performing a metric based analysis and a visual analysis, it can be concluded that performance of GOMORS is superior to ParEGO on the test problems examined here, especially on the relatively higher dimensional problems.

2.6 Conclusion

GOMORS is a new parallel algorithm for multi-objective optimization of black box functions that improves efficiency for computationally expensive objective functions by obtaining good solutions with a relatively small number of evaluations (less than 500). GOMORS accomplishes this with the construction in each iteration of a surrogate response surface based on all the values of the objective functions computed in the current and previous iterations. Then evolution and non-dominated sorting are applied to the inexpensive function describing this surrogate surface. It is then possible to evaluate a large number of points on the surrogate inexpensively so multiple Rules can be used to select a diversity of points for expensive evaluation on parallel processors. The use of these multiple Rules is the innovative aspect of GOMORS and contributes to its effectiveness and robustness with limited numbers of evaluations (less than 500). GOMORS is very different from ParEGO, which solves multiple single objective problems.

Our numerical results indicate that GOMORS was significantly more effective than ParEGO when the number of evaluations was quite limited relative

to the difficulty of the problem (Figures 2.2-2.4) for both the test functions and the groundwater partial differential equation models. Computational demands for nonlinear optimization will grow rapidly as the dimension increases, so the effectiveness of GOMORS becomes more obvious on higher dimensional problems, as is evident in Figures 2.2-2.4.

There are many real application models with computational times that are so large that the evaluations will be greatly limited, especially for multi-objective problems. For example, analysis of a carbon sequestration monitoring problem required global optimization with seven decisions of a nonlinear, multiphase flow transient PDF model that took over 2 hours per simulation [13]. So even 100 evaluations for a problem like this is probably more evaluations than are feasible. Hence the GOMORS approach to multi-objective optimization is a contribution in the area of surrogate-assisted multi-objective optimization of objective functions based on computationally expensive, multimodal, black box models.

2.7 References

[1] Taimoor Akhtar and Christine A. Shoemaker. Multi objective optimization of computationally expensive multi-modal functions with rbf surrogates and multi-rule selection. *Journal of Global Optimization*, pages 1–16, 2015.

[2] Johannes M. Bader. *Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods*. CreateSpace, Paramount, CA, 2010.

[3] N Beume, B Naujoks, and M Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, September 2007.

[4] Martin D Buhmann. *Radial Basis Functions*. Cambridge University Press, New York, NY, USA, 2003.

[5] Kalyanmoy Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evol. Comput.*, 7(3):205–230, September 1999.

[6] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, PPSN VI, pages 849–858, London, UK, UK, 2000. Springer-Verlag.

[7] Kalyanmoy Deb and Deb Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 1 edition, June 2001.

[8] Kalyanmoy Deb and Pawan Nain. An Evolutionary Multi-objective Adaptive Meta-modeling Procedure Using Artificial Neural Networks. In

Shengxiang Yang, Yew-Soon Ong, and Yaochu Jin, editors, *Evolutionary Computation in Dynamic and Uncertain Environments*, volume 51, chapter 13, pages 297–322. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[9] Francesco di Pierro, Soon-Thiam Khu, Dragan Savić, and Luigi Berardi. Efficient multi-objective optimal design of water distribution networks on a budget of simulations using hybrid algorithms. *Environ. Model. Softw.*, 24(2):202–213, February 2009.

[10] Alan Diaz-Manriquez, Gregorio Toscano-Pulido, and Wilfrido Gomez-Flores. On the selection of surrogate models in evolutionary optimization algorithms. *IEEE Congress of Evolutionary Computation (CEC)*, pages 2155–2162, June 2011.

[11] M. T M Emmerich, K.C. Giannakoglou, and B. Naujoks. Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *Evolutionary Computation, IEEE Transactions on*, 10(4):421–439, Aug 2006.

[12] Michael Emmerich, Alexios Giotis, Mutlu zdemir, Thomas Bck, and Kyriakos Giannakoglou. Metamodel-assisted evolution strategies. In *In Parallel Problem Solving from Nature VII*, pages 361–370. Springer, 2002.

[13] A Espinet, Christine A Shoemaker, and C Doughty. Estimation of Plume distribution for Carbon Sequestration using parameter estimation with limited monitoring data. *Water Resources Research*, 49(7):4442–4464, 2013.

[14] H. Fang, M. Rais-Rohani, Z. Liu, and M. F. Horstemeyer. A comparative study of metamodeling methods for multiobjective crashworthiness optimization. *Comput. Struct.*, 83(25-26):2121–2136, September 2005.

[15] Chariklia A. Georgopoulou and Kyriakos C. Giannakoglou. A multi-objective metamodel-assisted memetic algorithm with strength-based local refinement. *Engineering Optimization*, 41(10):909–923, 2009.

[16] Jean D Gibbons and S Chakraborti. *Nonparametric Statistical Inference*. Chapman and Hall, Boca Raton, FL, 2010.

[17] H.-M. Gutmann. A radial basis function method for global optimization. *J. of Global Optimization*, 19:201–227, March 2001.

[18] D. Huang, T. T. Allen, W. I. Notz, and N. Zeng. Global optimization of stochastic black-box systems via sequential kriging meta-models. *J. of Global Optimization*, 34(3):441–466, March 2006.

[19] R. Jin, W. Chen, and T. W. Simpson. Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and Multidisciplinary Optimization*, 23(1):1–13, December 2001.

[20] Yaochu Jin, M. Olhofer, and B. Sendhoff. A framework for evolutionary optimization with approximate fitness functions. *Evolutionary Computation, IEEE Transactions on*, 6(5):481–494, Oct 2002.

[21] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *J. of Global Optimization*, 13(4):455–492, December 1998.

[22] Marios K Karakasis and Kyriakos C Giannakoglou. On the use of metamodel-assisted , multi-objective. *Engineering Optimization*, 38(8):941–957, 2006.

[23] J. Knowles. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 8(5):1341–66, February 2006.

[24] Hui Li and Qingfu Zhang. Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, April 2009.

[25] Dudy Lim, Yaochu Jin, Yew-Soon Ong, and Bernhard Sendhoff. Generalizing surrogate-assisted evolutionary computation. *IEEE Trans. Evol. Comp*, 14(3):329–355, June 2010.

[26] S. Zapotecas Martinez and Carlos Artemio Coello Coello. Combining surrogate models and local search for dealing with expensive multi-objective optimization problems. In *IEEE Congress on Evolutionary Computation*, pages 2572–2579. IEEE, 2013.

[27] Saúl Zapotecas Martínez and Carlos A. Coello Coello. A memetic algorithm with non gradient-based local search assisted by a meta-model. In *Proceedings of the 11th international conference on Parallel problem solving from nature: Part I*, PPSN'10, pages 576–585, Berlin, Heidelberg, 2010. Springer-Verlag.

[28] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):pp. 239–245, 1979.

[29] Barbara S. Minsker and Christine A. Shoemaker. Differentiating a finite element biodegradation simulation model for optimal control. *Water Resources Research*, 32(1):187–192, 1996.

[30] Barbara S Minsker and Christine A Shoemaker. Dynamic Optimal Control of In-Situ Bioremediation of Ground Water. *Water Resources Planning and Management*, 124(3):149–161, 1998.

[31] Gerardo Montemayor-garcía and Gregorio Toscano-pulido. A Study of Surrogate Models for their use in Multiobjective Evolutionary Algorithms. In *8th International Conference on Electrical Engineering Computing Science and Automatic Control (CCE)*, 2011.

[32] Juliane Mueller and Christine A Shoemaker. SO-I: A Surrogate Model Algorithm for Expensive Nonlinear Integer Programming Problems Including Global Optimization Applications. *Journal of Global Optimization*, (In press and online), 2013.

[33] Juliane Mueller, Christine A Shoemaker, and R Piche. SO-MI: A Surrogate Model Algorithm for Computationally Expensive Nonlinear Mixed-Integer, Black-Box Global Optimization Problems. *Computers and Operations Research*, 40(5):1383–1400, 2009.

[34] Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, and Markus Vincze. Multiobjective optimization on a limited budget of evaluations using model-assisted s-metric selection. In *Proceedings of the 10th international conference on Parallel Problem Solving from Nature: PPSN X*, pages 784–794, Berlin, Heidelberg, 2008. Springer-Verlag.

[35] M. J. D. Powell. *The Theory of Radial Basis Function Approximation in 1990*, pages 105–210. Oxford University Press, USA, May 1992.

[36] Kevin L. Priddy and Paul E. Keller. *Artificial Neural Networks: An Introduction*. SPIE Press, 2005.

[37] R Regis, C Shoemaker, and A. stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J. of Computing*, 19:497–509, 2007.

[38] R.G. Regis and C.a. Shoemaker. Local Function Approximation in Evolutionary Algorithms for the Optimization of Costly Functions. *IEEE Transactions on Evolutionary Computation*, 8(5):490–505, October 2004.

[39] Rommel G Regis and Christine A Shoemaker. Constrained Global Optimization of Expensive Black Box Functions Using Radial Basis Functions. *Journal of Global Optimization*, pages 153–171, 2005.

[40] Rommel G Regis and Christine A Shoemaker. Improved strategies for radial basis function methods for global optimization. *Journal of Global Optimization*, 37(1):113–135, 2007.

[41] Rommel G Regis and Christine A Shoemaker. Parallel Stochastic Global Optimization Using Radial Basis Functions. *INFORMS J. of Computing*, 21(3):411–426, 2009.

[42] Rommel G Regis and Christine A Shoemaker. A Quasi-Multistart Framework for Global Optimization of Expensive Functions using Response Surface Models. *Journal of Global Optimization*, (In press), 2013.

[43] Rommel G Regis and Christine A Shoemaker. Combining Radial Basis Function Surrogates Dynamic Coordinate Search in High Dimensional Expensive Black-box Optimization. *Engineering Optimization*, 45(5):529–555, 2013.

[44] Alejandro Rosales-p, Carlos A Coello Coello, Jesus A Gonzalez, Carlos A Reyes-garcia, and Hugo Jair Escalante. A Hybrid Surrogate-Based Ap-

proach for Evolutionary Multi-Objective Optimization. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 2548–2555, 2013.

[45] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and Analysis of Computer Experiments. *Statistical Science*, 4(4):409–423, November 1989.

[46] L.V. Santana-Quintero, V.A. Serrano-Hernandez, Carlos A. Coello Coello, A.G. Hernandez-Diaz, and J. Molina. Use of radial basis functions and rough sets for evolutionary multi-objective optimization. In *Computational Intelligence in Multicriteria Decision Making, IEEE Symposium on*, pages 107–114, April 2007.

[47] Ralph Steuer and Eng-Ung Choo. An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26:326–344, 1983.

[48] Vladimir N. Vapnik. *Statistical learning theory*. Wiley, 1 edition, September 1998.

[49] Jasper a Vrugt and Bruce a Robinson. Improved evolutionary optimization from genetically adaptive multimethod search. *Proceedings of the National Academy of Sciences of the United States of America*, 104(3):708–11, January 2007.

[50] Stefan M Wild, Rommel G Regis, and Christine A Shoemaker. ORBIT: optimization by Radial Basis Function Interpolation in Trust-Regions. *SIAM Journal of Scientific Computing*, 30(6):3197–3219, 2007.

[51] Stefan M Wild and Christine A Shoemaker. Global Convergence of Radial

Basis Function Trust-Region Algorithms for Derivative-Free Optimization. *SIGEST Article, SIAM Review*, 55(2):349–371, 2013.

[52] Saúl Zapotecas Martínez and Carlos A. Coello Coello. Moea/d assisted by rbf networks for expensive multi-objective optimization problems. In *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference*, GECCO '13, pages 1405–1412, New York, NY, USA, 2013. ACM.

[53] Qingfu Zhang and Hui Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.

[54] Qingfu Zhang, Wudong Liu, Edward Tsang, and Botond Virginas. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Trans. Evol. Comp*, 14(3):456–474, June 2010.

CHAPTER 3

MEANINGFUL MULTI OBJECTIVE PARAMETER ESTIMATION OF COMPUTATIONALLY EXPENSIVE WATERSHED MODELS WITH SURROGATE ALGORITHMS AND THE DISTRIBUTED CARDINALITY INDEX

3.1 Introduction

Watershed simulation models typically involve, (a) physical parameters that are difficult to measure directly, and/or (b) conceptual parameters that are impossible to measure. Conceptual parameters stem from considerable simplifications employed in modeling of the natural processes within the watershed. Model calibration is a process that can be employed to adjust values of such parameters, where the aim is to mimic reality, via comparison of model response to historically observed measurements. Traditional calibration methodologies typically involve manual trial-and-error, with expert opinion of a hydrologist within an interactive calibration framework. The value of expert opinion cannot be disregarded; however, manual methodologies can be extremely time consuming and complicated.

A bulk of prior and contemporary research has focused on automatic calibration schemes involving single objective optimization, e.g, [8, 26, 34, 52]. The simulation-optimization problem for a complex watershed model formulated within the automatic calibration framework is typically non-linear and probably has multiple local minima (i.e. is "multi-modal") [53], and various studies have employed heuristic search techniques in a search for the globally optimal solution.

However, watershed calibration based on a single aggregated calibration

performance metric could lead to a significant loss of information within the calibration process. The calibration process could be highly sensitive to various factors, especially the proposed objective function used to assess the adequacy of a set of parameter values.

The advent of multi-objective optimization has added a new dimension to the hydrological model calibration process. Gupta et al. [21] mention that the multi criteria approach is a potential way forward towards automatic calibration where the process of automatic calibration can emulate the manual calibration process by simultaneously incorporating numerous measures for assessing model performance. Many recent studies have focused on employing multi-objective optimization for watershed model calibration [16], highlighting the effectiveness of multi-objective analysis in deducing various "Pareto" optimal calibrations [33, 64], providing added insight into model ambiguity resulting from model imperfections and parameter uncertainty [59] and understanding modeling limitations [22]. Efstratiadis and Koutsoyiannis [17] discuss the similarities between the "Pareto dominance" concept inherent in multi-objective calibration, and the "equifinality" concept inherent in the GLUE framework [6, 39], highlighting the ability of multi-objective calibration to help understand model uncertainties.

Numerous research contributions have been made in proposing algorithms for multi-objective optimization, within the simulation-optimization framework [10, 13]. Various algorithmic contributions, within the water resources community have also been made, focusing primarily on evolutionary strategies [43, 46, 53, 54]. Evolutionary strategies are frequently referred as multi-objective evolutionary algorithms (MOEA) in contemporary literature. Tang et al. [53] provide a comparative analysis of various evolutionary algorithms, in order to

deduce their effectiveness in hydrological model calibration.

While prior research and current industry calibration techniques indicate the inherent multi-criteria nature of the hydrological model calibration problem [8, 22, 32], the computational complexity of distributed hydrological models pose a huge challenge to the use of multi-criteria optimization algorithms in the calibration process.

It should be noted that the calibration optimization problem can be a computationally expensive simulation optimization problem, since the objective function(s) are evaluated via simulation and a simulation evaluation for watershed models can take a very long time. Hence, algorithms that require fewer model evaluations to produce good trade-off solutions, are desirable.

The use of surrogate models within an optimization algorithm can be highly effective in reducing time for computing objectives for multi-objective calibration of complex watershed problems. (The terms surrogate, response surface and meta model are all used to describe the use of existing information to build an approximation of the objective function, which then guides the optimization search.) Surrogate based single optimization algorithms have been widely used in water resources calibration applications. Most optimization methods used in these applications include Radial Basis Function (RBF) based methods [23, 47, 48] , Kriging based methods [25] and Artificial Neural Network (ANN) [67, 69] based methods. Some of these methods have been extended to incorporate parallelism [49]; local optimization [50, 61, 62]; integer problem handling [35, 36] etc. The prevalent calibration application areas in prior research incorporating the use of surrogate optimization methods predominantly include watershed model calibration [7, 27, 45, 52], groundwater model calibration [37, 38] and carbon sequestration model calibration [18, 19]. Razavi et al. [44] provide a

comprehensive review of literature on the use of surrogates in water resources.

Surrogates have also been used in application of multi objective optimization to complex water resources problems. [31] combine an ANN with an evolutionary algorithm for multi objective calibration of a rainfall-runoff model. Bau and Mayer [3] employ kriging based surrogates within a multi objective framework for optimal design of groundwater (pump-and-treat) remediation systems. Behzadian et al. [4] combine a multi objective evolutionary algorithm (MOEA) with an ANN to efficiently deduce optimal sampling locations of pressure loggers for a water distribution system. di Pierro et al. [15] explore the use of surrogate based MO algorithms with application to water distribution network design. Castelletti et al. [9] incorporate the use of numerous surrogate methods for efficient multi objective optimization with application to water quality planning in reservoirs and lakes.

This study aims at understanding the value of surrogates in efficient multi objective calibration of computationally expensive watershed models via a comprehensive comparison of numerous surrogate and non-surrogate based algorithms. The algorithms compared in this study include the widely used non-dominated sorting genetic algorithm (NSGA-II) [14], the multi-objective evolutionary algorithm with decomposition (MOEA/D) [65] and AMALGAM [60], which are popular evolutionary algorithms that have been applied to water resources problems.

In order to assess the ability of surrogate methods to improve the computational efficiency of multi-objective, watershed calibration, we compare these non surrogate algorithms against two response surface-assisted optimization methods, namely GOMORS [1] and ParEGO [28]. Both GOMORS and ParEGO build new approximations after each iteration in which new expensive evalu-

ations of the objective functions are completed. ParEGO converts the multi-objective problem into a series of single objective functions, whereas GOMORS solves the multiple objective problem using non-dominated sorting and a set of rules for selecting the next expensive function evaluation. Our focus is on multi-objective optimization of relatively computationally expensive objectives, so we have a limited evaluation budget (less than 1000 evaluations). We also introduce a new metric, referred as the Distributed Cardinality index which is designed for evaluating performance of algorithms in terms of their effectiveness in watershed model calibration.

Our results indicate that GOMORS outperforms all other algorithms within a limited model evaluation budget (less than 1000), when used to calibrate flow parameters of two distributed SWAT watershed model case studies, namely the Townbrook SWAT Model, and the Cannonsville Swat Model, developed by Tolson and Shoemaker [56]. Hence this analysis is focused on different kinds of problems than the comprehensive analysis in Tang et al. [53] that compares performance of various evolutionary algorithms MOEAs, for a large simulation evaluation budget (up to 100,000 evaluations), which is not a feasible number for a large watershed like the Cannonsville.

3.2 Multi-Objective Calibration Decision Analysis

The multi-objective calibration framework discussed in this study, can be formulated as a box-constrained optimization problem:

$$\min_{\theta} F(\theta) = [f_1(\theta), \ldots, f_k(\theta)]^T \tag{3.1}$$
$$\text{subject to } \theta_i^{min} \leq \theta_i \leq \theta_i^{max} \text{ , } i = 1 \ldots n$$

Where $\theta = [\theta_1 \dots \theta_n]$ is the vector of decisions, or more specifically, the vector of model parameters to be calibrated, bounded by the vectors θ_{min} and θ_{max}. $F(\theta)$ is the vector of calibration objectives, which can be subjectively defined by a calibration expert. In order to evaluate the objective $F(\theta)$ for a candidate decision vector θ, a computationally expensive run of the watershed simulation model is to be performed. Hence, a desirable property of an optimization methodology is the ability to produce good solutions within a limited budget of simulation evaluations. The budget on simulation evaluations depends on the computation time of a model and the total time available for the calibration process.

3.3 Optimization Algorithms

Within the water resources literature, several recent studies have shown that in general, watershed model calibration problems can be highly multi-modal, and existence of false non-dominated Pareto fronts (locally optimal fronts) can pose a huge threat to the addedaccuracy of the optimization process [30, 46]. Hadka and Reed [24] state that multi-modality is a severe challenge for most multi-objective evolutionary algorithms (MOEAs). In our analysis, we aim to compare optimization algorithms with varying search capabilities in order to understand their capabilities in tackling the numerous optimization challenges of watershed calibration within a very limited simulation evaluation budget.

3.4 Multi-Objective Evolutionary Algorithms (MOEAs)

The optimization literature contains various multi-objective optimization algorithms, specifically for tackling simulation optimization problems. These search based meta-heuristics are dominated by multi-objective Evolutionary Al-

gorithms (MOEAs). Deb [13] and Coello et al. [10] suggest that use of MOEAs can be highly beneficial, since the population based structure of an evolutionary algorithm can be exploited to simultaneously converge towards the Pareto front, and maintain a diverse set of trade-offs.

Various search methodologies within evolutionary optimization can be employed to tackle the two-fold aim of multi-objective optimization. Local search, decomposition of the objective vector into multiple single objective optimization problems, and employing multi-method search are some of them. Three multi-objective evolutionary algorithms (MOEAs) are used for comparison in this study. They are discussed in Sections 3.4.1 - 3.4.3.

3.4.1 Non-Dominated Sorting Genetic Algorithm -II

NSGA-II, proposed by Deb et al. [14] is a revolutionary and extremely popular multi-objective (MOEA) . NSGA-II has been applied to various applied engineering problems across numerous disciplines [10] including water resources [5]. NSGA-II handles the evolutionary search optimization process by ranking and archiving parent and child populations according to a non-domination sorting (a measure of convergence), and crowding distance, which is a measure of diversity of a solution, on a particular front. The NSGA-II is a *posteriori* algorithm [10], aimed at using Pareto-dominance to move the the non-dominated front towards convergence, during the search process.

3.4.2 Multi Objective Evolutionary Algorithm -Decomposition (MOEA/D)

Aggregate functions can also be used in the multi-objective optimization process in order to convert a multi-objective problem into many single-objective problems. The aggregate methodology falls under a separate class of MOEAs within the classification of Coello et al. [10], where the aim is to solve many single objective optimization problems, with different aggregation priorities, to converge to the Pareto front, and maintain divergence. The MOEA/D [65] uses aggregate functions, and simultaneously solves many single-objective Tchebycheff decompositions of multi-objective problems in a single run. Tchebycheff decomposition is a preference based aggregation method for converting a vector of objectives into a single objective. The MOEA/D is an established benchmark algorithm and has won the 2009 IEEE Congress on Evolutionary Computation (CEC 2009) competition, which allowed a large number of function evaluations.

3.4.3 AMALGAM - A Multi-algorithm Genetically Adaptive Multi-Objective Optimization Method

AMALGAM [60] is a multi-method evolutionary algorithm, which incorporates search mechanics of various algorithms. The key feature of AMALGAM is simultaneous multi-method search and the search mechanism selection is self-adaptive. AMALGAM incorporates four candidate MOEAs, NSGA-II, Particle Swarm Optimization (PSO), Differential Evolution (DE), and Adaptive Metropolis Search (AMS). Recent watershed literature has seen various applications of AMALGAM, including the work of Zhang et al. [68] which includes application of AMALGAM for multi-site calibration and comparison against

SPEA2 and NSGA-II.

3.5 Surrogate Assisted MO Optimization

Since we are interested in efficient multi-objective calibration of expensive watershed models, incorporation of surrogate assisted optimization methodologies is important in this analysis. Surrogate assisted search methods typically employ computationally inexpensive response surface models (or "surrogate" models), within the iterative search process, in order to guide search towards optimal solutions. While surrogate based single objective optimization has been successfully used within the water resources research community [37, 52], use of surrogate assistance in multi-objective applications is scarce. Within the context of model calibration the only application of surrogate assisted MO optimization we found in prior literature is the work by Liong et al. [31] who incorporate an ANN with an MOEA for calibration of a rainfall-runoff model. In our comparative analysis, we compare two surrogate based algorithms (ParEGO and GOMORS), along with the evolutionary algorithms mentioned above for watershed model calibration. This is the first published application of either ParEGO or GOMORS on a watershed model calibration problem.

3.5.1 ParEGO

ParEGO [28] uses a Kriging-based [51] surrogate surface for multi-objective optimization, and it is specifically designed for applications involving a very limited evaluation budget. ParEGO is an iterative search method, which uses aggregate function to divide a multi-objective optimization problem into several single objective problems. During each iteration of the algorithm, a subproblem

is chosen randomly and solved via the Efficient Global Optimization (EGO) algorithm proposed by Jones et al. [25]. Applications of ParEGO to test problems and comparison against NSGA-II show very promising results within a limited evaluation budget.

3.5.2 GOMORS - Gap Optimized Multi-Objective Optimization with Response Surfaces

GOMORS [1] is another iterative scheme which employs Radial Basis Functions (RBF) as a surrogate model to guide multi-objective search towards the optimal set of solutions. GOMORS embeds surrogate approximation via RBFs, within an MOEA, in order to improve algorithm efficiency.

During each iteration of GOMORS, an RBF based surrogate model (fitted from already evaluated points) is used to approximate the expensive objective functions. An evolutionary algorithm is subsequently applied to the surrogate model for identification of new potential points for expensive evaluation. However, not all solutions of the surrogate evolutionary optimization are actually evaluated, and only some solutions are selected for expensive function evaluations in each algorithm iteration. A novel methodology (referred as multi-rule selection) is introduced that simultaneously selects multiple evaluation points through different rules including Approximate Hypervolume Improvement, Maximizing minimum domain distance, Maximizing minimum objective space distance, and surrogate-assisted local search.

GOMORS differs from ParEGO in that GOMORS solves the multi-objective problem directly while ParEGO converts the MO problem into multiple single objective problems.

Application of GOMORS to a hypothetical groundwater remediation design problem depict promising results, where GOMORS outperforms ParEGO and NSGA-II with 100, 200 and 400 evaluations for a 6 dimensional problem, a 12 dimensional problem and a 24 dimensional problem [1].

3.6 Calibration Formulations

The intensified interest in multi-objective calibration within the hydrological model calibration research community results from the realization that multiple criteria should be considered within the calibration process (for instance, volume error, low flow calibration, peak flow calibration, seasonal calibration etc are all reasonable criteria). Recent research efforts [5, 22, 53, 64] have indicated that significant conflicts and trade-offs might exist between calibration criteria and that visualization of trade-offs can assist decision makers in understanding model limits and choosing appropriate calibrations. Numerous performance metrics can be employed to quantify the potentially conflicting calibration criteria. The Nash-Sutcliffe Efficiency (NSE) [40], bias and mean absolute error [22] are some of the metrics that can be used as objective functions in multi-objective optimization formulations. Two multi-objective flow calibration formulations are used in this study, with the formulation focus on visualizing trade-offs between conflicting objectives.

3.6.1 Formulation 1 - Threshold Based Flow Separation (2-objective)

This formulation aims at highlighting the trade-off between high flow calibration and moderate/low flow calibration. The relative importance of different

hydrological processes varies between high flow and low flow situations so a given set of model parameters might do relatively well at matching data under high flow conditions and relatively poorly under low flow conditions (or vice versa). Hence it is reasonable to consider the goals of getting adequate fits to the data under low and high flow conditions as two different objectives for which there will be a tradeoff.

The measured data is divided into two categories, "high flow observations" and "non-high flow observations". The data is segregated via identification of a flow threshold, beyond which all observations are tagged as "high flow calibrations". The flow threshold is typically identified by a decision maker. By dividing the data into various categories, we can devise various objectives for a multi-objective optimization formulation:

$$\begin{aligned} f_1(\theta) &= \sum_{i \in N_1} [y_i - \widehat{y_i}(\theta)]^2 \text{ , where } N_1 = \{i \mid y_i \leq Y, 1 \leq i \leq n\} \\ f_2(\theta) &= \sum_{i \in N_2} [y_i - \widehat{y_i}(\theta)]^2 \text{ , where } N_2 = \{i \mid Y \leq y_i, 1 \leq i \leq n\} \end{aligned} \tag{3.2}$$

Equation 3.2 describes the objectives used in our first formulation, "Formulation 1", where y_i and $\widehat{y_i}$ are the measured and simulated flows on day i, and Y is the flow threshold (identified by decision maker). The two objectives in this formulation are "1) High Flow Sum of Squared Errors ($f_2(\theta)$)" and "2) Moderate-to-Low Flow Sum of Squared Errors ($f_1(\theta)$)". Previous research efforts have shown that a significant trade-off exists between varying flow regimes [30]. The flow threshold values, Y, employed in all our case studies are the 95-th percentile values of the calibration data sets.

3.6.2 Formulation 2 - Decomposition of NSE of Flow (3-objective)

Gupta et al. [20] highlight that the Nash-Sutcliffe Efficiency (NSE) criterion can be decomposed into three components, 1) linear correlation, 2) relative bias, and 3) relative variability, where each component focuses on calibrating a different and potentially conflicting aspect of flow. The relative bias component of the criterion tends to minimize volume balance errors, the relative variability tends to mimic the flashiness of the hydrograph, inherently focusing on capturing extreme flows, while the correlation criterion, in combination with relative variability tends to capture the shape of the hydrograph. The potential for conflict between the decomposed NSE metrics is acknowledged by Gupta et al. [20]. The multi-objective calibration formulation of the three objective NSE decomposition [20] is employed as "Formulation 2" in our algorithm comparison effort, and is as follows:

$$\begin{aligned} f_1(\theta) &= [r-1]^2 \text{ , where } r = \frac{\sum_{i=1}^{n} y_{obs,i} y_{sim,i} - n\bar{\mu}_{obs}\bar{\mu}_{sim}}{(n-1)\bar{\sigma}_{obs}\bar{\sigma}_{sim}} \\ f_2(\theta) &= [\alpha-1]^2 \text{ , where } \alpha = \bar{\sigma}_{sim}/\bar{\sigma}_{obs} \\ f_3(\theta) &= [\beta-1]^2 \text{ , where } \beta = \bar{\mu}_{sim}/\bar{\mu}_{obs} \end{aligned} \tag{3.3}$$

where $y_{obs,i}$ and $\widehat{y}_{sim,i}$ are the measured and simulated flows on day i, $\bar{\mu}_{obs}$ and $\bar{\mu}_{sim}$ are the estimated mean values of measured and simulated flows, and $\bar{\sigma}_{obs}$ and $\bar{\sigma}_{sim}$ are the estimated values of standard deviation of measured and simulated flows. Both multi-objective formulations employed in this study are designed for calibration of the model to measured data obtained from a single location/site. However, approach can be extended to multi-site calibration formulations.

3.7 Case Studies

The two Watershed Model case studies used in our analysis are derived from the Cannonsville Watershed modeling case study, by Tolson and Shoemaker [55]. The Soil and Water Assessment Tool (SWAT) [41, 42] is used for model development. SWAT is a physically based, deterministic and semi-distributed watershed modeling tool.

3.7.1 Case Study I: Cannonsville Watershed

Tolson and Shoemaker [56] introduce two scaled variations of the Cannonsville SWAT model, as flow calibration case studies. Case Study I incorporates a computationally expensive calibration model (a single simulation run takes around 2-3 minutes for a 10 year simulation period), which constitutes 43 subbasins and predicts flow within the Cannonsville Watershed. The model calibration exercise is carried out at the Walton flow monitoring location, which drains upto 860 km^2 of the watershed. A preliminary sensitivity analysis of the model flow parameters is discussed by Tolson and Shoemaker [55]. While some of the parameters are spatially variable, scaling factors are employed to correlate such parameters, and subsequently reduce the number of parameters to be calibrated [56]. Tolson and Shoemaker [56] identify 15 parameters, that are to be calibrated, for flow prediction in Case Study I. Case Study I employs a 9 year time period, for daily flow calibration at the Walton flow monitoring station.

3.7.2 Case Study II: Townbrook

Townbrook is a subwatershed within the Cannonsville Watershed , covering an area of around 37 km^2. Case Study II is derived from the single subbasin Town-

Table 3.1: The flow calibration test case suite employed in comparative algorithm analysis with formulations defined in Section 3.6.1 and 3.6.2

Problem Name	**Formulation**	**Case Study**	**Decisions Variables**	**Objectives**	**Simulation Time/Eval(s)**
SW-2	1. Threshold	II	15	2	10
SW-3	2. NSE	II	15	3	10
FW-2	1. Threshold	I	15	2	150
FW-3	2. NSE	I	15	3	150

brook SWAT model developed by Tolson and Shoemaker [56]. The Townbrook SWAT model is a relatively inexpensive model (one simulation run takes up to 10 seconds for a 10 year modeling period), and can be used extensively for algorithm comparison. The model predicts flow within the Townbrook watershed, and is employed as a flow calibration case study. The Townbrook subwatershed is monitored for flow by USGS, and Case Study II employs a 10 year time period, for daily flow calibration of 15 parameters of the Townbrook SWAT model.

Given two calibration formulations, and two case studies, we developed 4 watershed calibration test case studies, as our test suite, for comparative algorithm analysis. The nomenclature used for the test case studies, along with a brief overview of the problems, is provided for reference in table 3.1. SW is an abbreviations for "Sub-Watershed", which is a reference to the Townbrook sub-watershed case study (Case Study II). FW is an abbreviation for "Full-Watershed", which is a reference to the Cannonsville cast study (Case Study I).

3.8 Algorithm Comparison Methodology

The effectiveness of a multi-objective optimization algorithm can be assessed via analysis of algorithm ”efficiency”. Efficiency corresponds to the effectiveness of an algorithm in identifying a set of good ”quality” trade-off solutions, quickly and within a limited simulation evaluation budget. Due to the stochastic nature of all algorithms compared in this analysis, it is also important to understand and compare algorithm ”reliability”, i.e the ability of an algorithm to produce good ”quality” solutions, consistently over multiple trial runs.

Deb [13] defines ”quality” of a multi-objective solution (also called approximate front and non-dominated front) to be a combination of two (potentially conflicting) properties: 1) ”convergence” and 2) ”diversity”. Convergence can be described as the proximity of an algorithm's solutions, to the Pareto front, and ”diversity” is the extent of the true trade-off represented by an algorithm's solutions (see Figure 3.1 for illustration).

A visual comparison of trade-off solutions (commonly referred as the Pareto front) obtained via different algorithms is the most common comparison methodology employed in prior literature [13, 53, 66]. Efforts have also been employed to quantify ”quality” of trade-off solutions, within a single performance measure [2, 58]. In our analysis, we employ a combination of visual trade-off analysis, and performance metric based analysis, in order to understand and compare performance of algorithms on the test suite.

We also compare the ability of algorithms in terms of identification of ”meaningful” trade-offs / calibrations from multi-objective optimization, within a limited evaluation budget. The definition of ”meaningful” trade-offs is discussed in Section 3.8.3.

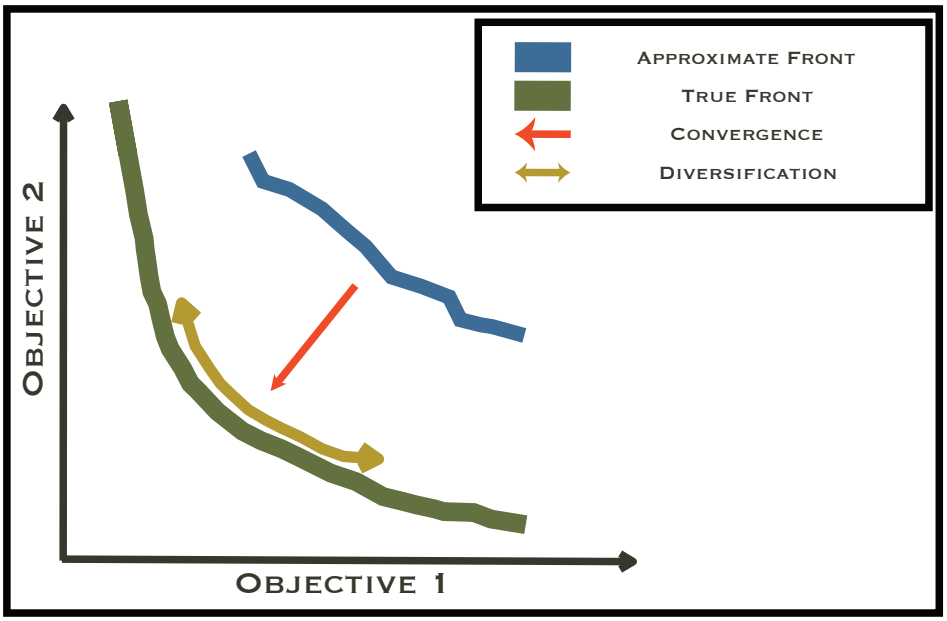

Figure 3.1: Illustration of the meanings of convergence and diversity. The red arrow is a depiction of convergence, i.e., the progress of an approximate front towards the Pareto front. The yellow line depicts diversity, i.e., extent of trade-off captured by an approximate front.

3.8.1 Experimental Setup

We initiated the algorithm comparison methodology by deducing suitable values for parameters of all algorithms under discussion. A small trial-and-error exercise was performed to tune population sizes for all multi-objective evolutionary algorithms (MOEAs). Since performance of MOEAs is highly dependent on population sizes, we ran multiple trials of NSGA-II, MOEA/D and AMALGAM on the Townbrook (sub-watershed) test case studies, *SW-2* and *SW-3*, with population sizes of 20, 50, 100 and 200, and an evaluation limit of 1000. The initial trial-and-error analysis showed that within the limited evaluation budget of 1000, a population size of 20 was desirable for all MOEAs, in the case of the bi-objective Townbrook case study, *SW-2*, and a population size of 50 was desirable for all MOEAs, in the case of the 3-objective Townbrook

case study, *SW-3*. The same population sizes were used for the respective Cannonsville case studies, *FW-2* and *FW-3*. ParEGO's configuration recommended by Knowles [28] was employed, and the GOMORS configuration recommended byAkhtar and Shoemaker [1] was used.

Due to the stochastic nature of all algorithms, multiple trial runs were performed for all watershed case studies. We performed 10 trials for each algorithm with 1000 function evaluations for all case studies. Our analysis was focused at comparing algorithms' performances, in terms of efficiency and effectiveness, within a simulation evaluation budget of 1000.

Efficiency of all algorithms was compared via plotting uncovered hypervolume (see section 3.8.2) performance metric values (averaged over multiple trial runs) against number of function evaluations. These plots are called *progress graphs* in our discussion. A visual illustration of the uncovered hyervolume metrics is provided in Figure 3.2. Section 3.9 provides a discussion on comparative efficiency of all algorithms (as per the uncovered hypervolume metric), upon application to the watershed calibration case studies discussed in this study.

For the two three-objective test problems (*SW-3* and *FW-3*), effectiveness of algorithms was compared within the context of watershed model calibration, via the meaningful trade-off analysis (see Section 3.8.3). Section 3.10 discusses the performance of GOMORS, ParEGO and AMALGAM in terms of their ability in identifying meaningful and diverse calibrations, within a limited evaluation budget. The Distributed Cardinality performance index is introduced (see Section 3.8.3) for analyzing algorithm performance in this context.

For the two bi-objective test problems (*SW-2* and *FW-2*), effectiveness of algorithms was compared via a visual comparison of best and worst trade-offs obtained through each algorithm after a fixed number of function evaluations.

The best and worst trade-offs obtained by an algorithm refer to the best and worst approximations of the Pareto front obtained by an algorithm in multiple trials, according to the uncovered hypervolume metric value. The best and worst trade-off approximations obtained from each algorithm were also plotted against the approximated true Pareto front, in order to visually assess the level of convergence and diversity attained by each algorithm within a limited evaluation budget. This comparative analysis was performed for GOMORS, ParEGO and AMALGAM. Section 3.11 provides details of the findings of the visual trade-off analysis.

3.8.2 The Uncovered Hypervolume Metric

The uncovered hypervolume [2] metric has been used to compare algorithm performance in Section 3.9. Figure 3.2 provides a visual illustration of the meaning of the uncovered hypervolume metric. Let P (illustrated in Figure 3.2) be the set of non dominated solutions obtained as an approximation to the Pareto front from an algorithm. Let P^* be the ideal solution(s) of the multi-objective optimization problem being solved. The ”ideal solution” of a multi-objective optimization problem is the vector depicting minimum attainable values of all objectives. In our case studies, all objectives are measures of error. Hence, the ”ideal point” for all the watershed problems is the zero vector. The ”reference point” depicted in Figure 3.2 is the worst attainable solution of the optimization problem. Since, the worst attainable values are not known in the problems discussed in this study, the ”reference point” vector is estimated by the worst values of all objectives obtained in our computer experiments. Hence, the total feasible objective space is bounded by the reference and ideal points.

The Uncovered hypervolume [2] is the difference between the total feasible

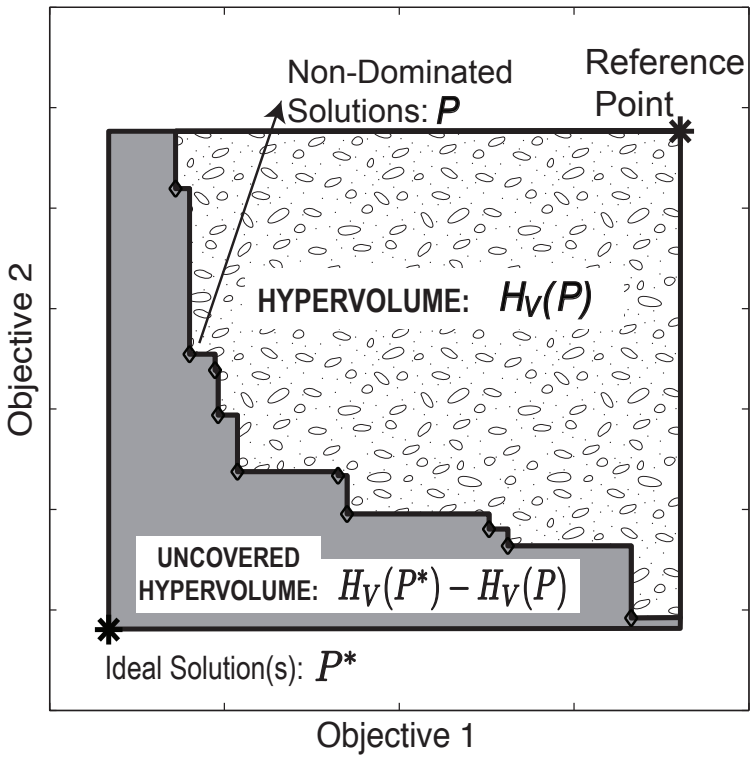

Figure 3.2: Illustration of the Uncovered Hypervolume metric which provides a comprehensive quantification of algorithms' convergence and diversification abilities.

objective space (defined by the reference and ideal points in Figure 3.2(a)) and the objective space dominated by estimate of the Pareto front (depicted as P in the figure) obtained by an algorithm. A lower value of the uncovered hypervolume index indicates a better solution and the ideal value is zero. The uncovered hypervolume metric incorporates both convergence to the ideal front as well as diversity of solutions on the front.

3.8.3 Meaningful Trade-offs and the Distributed Cardinality Index

The term ”meaningful trade-off” has been defined and used by various authors in the water resources literature [6, 17, 30]. A ”meaningful trade-off” essentially refers to a trade-off (non-dominated) solution obtained by multi-objective optimization, which is acceptable as a calibration. Hence, the definition of a meaningful trade-off is subjective. Prior literature includes various techniques for identification of a meaningful trade-off [6, 17, 46].

The meaningful trade-off analysis is performed for the two three objective calibration problems, i.e, *SW-3* and *FW-3*. Bounds (or value ranges) are defined for each objective function to identify meaningful solutions from trade-offs obtained from different algorithms. Solutions on the Pareto Front which lie within the defined bounds are collectively referred as the bounded Pareto Front, or alternatively as the set of meaningful trade-offs. The heavy line in Figure 3.3 illustrates the bounded Pareto Front (for a hypothetical problem).

The three objectives for test problems *SW-3* and *FW-3* are based on the decomposition of NSE (see Section 3.6.2 and Equation A.8) with correlation (r), relative variability (α), and relative bias (β) as key components of each objective. We define three levels of meaningfulness in our analysis by delineating three sets of bounds on the three objectives. All trade-off solutions with correlation greater than 0.7, relative variability less 15 percent, and relative bias less than 10 percent, are defined as *Level-I* meaningful trade-offs. The corresponding objective function ranges for *Level-I* meaningful trade-offs are [0-0.09], [0-0.0225] and [0-0.01] respectively. Trade-off solutions with correlation greater than 0.75, relative variability less than 15 percent, and relative bias less than 8 percent, are defined as *Level-II* meaningful. Consequently, the corresponding objective

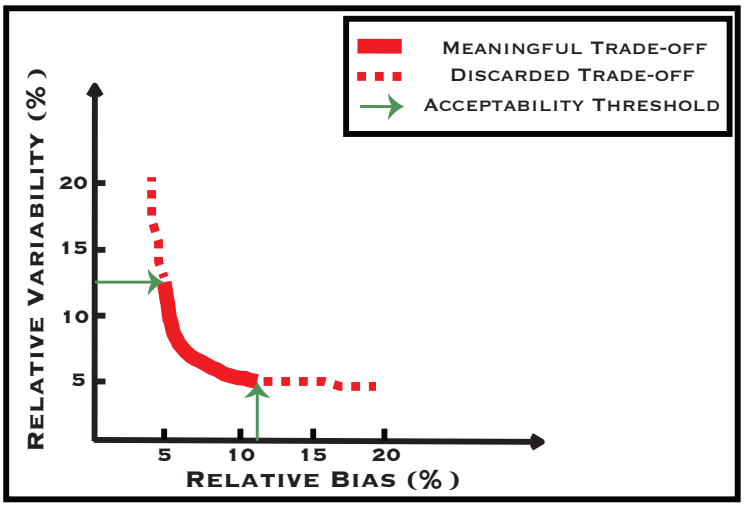

Figure 3.3: Illustration of a meaningful trade-off between two objectives, relative bias and relative variability. Bounds on each objective functions (identified by the green arrows) are defined according to user preference and solutions in the trade-off set which are within the defined bounds, are called meaningful.

function ranges / bounds for *Level-II* meaningful solutions are [0-0.0625], [0-0.0225] and [0-0.0064] respectively. Trade-off solutions with correlation greater than 0.8, relative variability less than 15 percent and relative bias less than 8 percent are defined as *Level-III* meaningful with [0-0.04], [0-0.0225] and [0-0.0064] being the respective objective function ranges / bounds for *Level-III* meaningful solutions. (See Table 3.2 for definitions of meaningful levels). It should be noted that a higher meaningful level inherently describes a higher aspiration level in terms of the quality of calibrations desired.

Various authors [30, 57] show that while many trade-off solutions might exist between competing objectives for hydrological model calibration, the number of meaningful trade-off solutions might be very limited. This is illustrated in the trade-offs obtained in our analysis as well. Figure 3.4, for instance, plots simulated hydrographs of trade-off calibration solutions obtained via a single

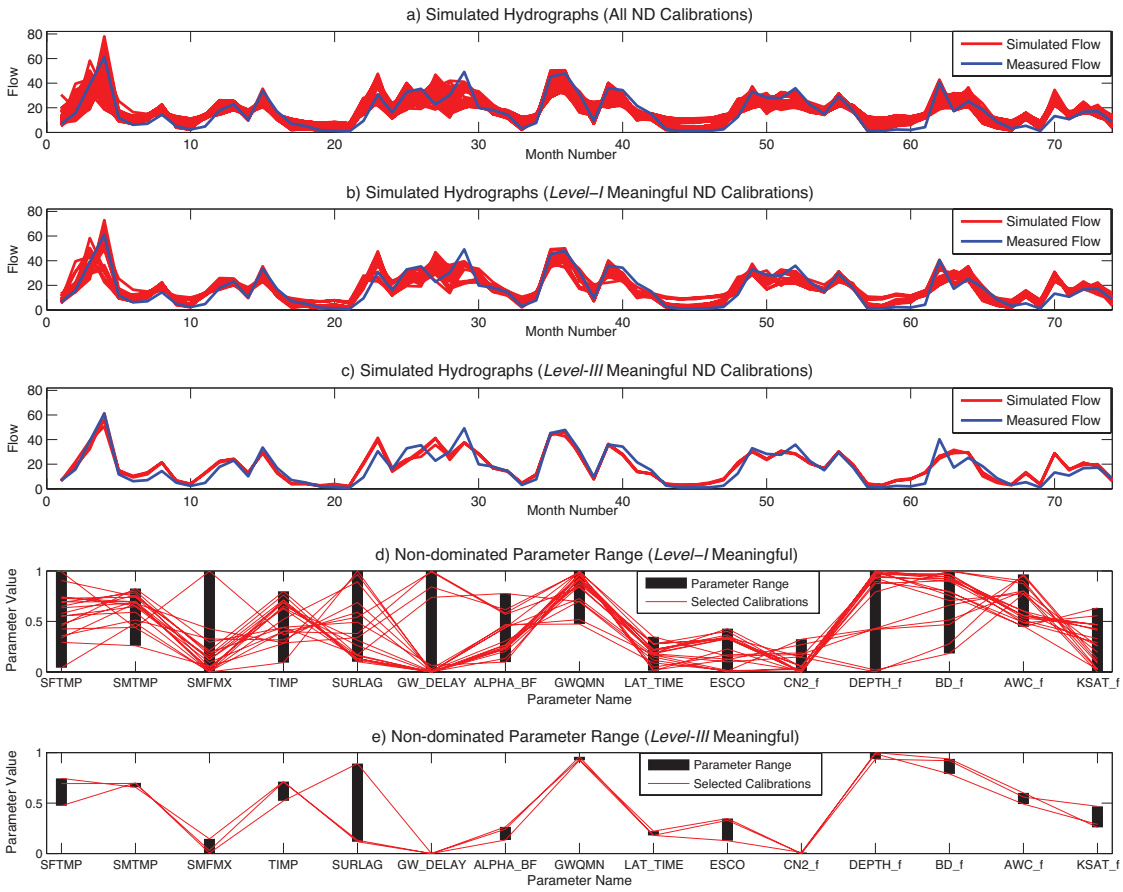

Figure 3.4: Simulated hydrographs and corresponding calibration alternatives obtained from multi-objective optimization (Problem *FW-3*) via GOMORS after 500 function evaluations (single trial): a) Hydrographs visualized for all non-dominated calibrations (89 ND calibrations found), b) Hydrographs visualized for *Level-I* meaningful ϵ-*box* non-dominated calibrations (19 ND calibrations found), c) Hydrographs visualized for *Level-III* meaningful ϵ-*box* non-dominated calibrations (3 ND calibrations found), d) Meaningful calibrations associated with the hydrographs of (b), e) Meaningful calibrations associated with the hydrographs of (c).

run of GOMORS (after 500 function evaluations), for problem *FW-3*. As is illustrated, Figure 3.4(a) visualizes hydrographs of all non-dominated calibrations, 3.4(b) visualizes hydrographs of *level-I* meaningful calibrations and 3.4(c) plots hydrographs of *level-III* meaningful calibrations. While 89 non-dominated cali-

brations are identified via GOMORS, only 3 of these calibrations achieve *level-III* meaningfulness. Please note that the solutions shown in Figure 3.4(b-e) correspond to "ϵ-*box*" non dominated calibrations (definition of ϵ-*box* non domination is provided later in this section).

Figure 3.4 illustrates that within the context of watershed model calibration, a good measure of effectiveness of multi-objective optimization algorithms is their ability in identifying meaningful non-dominated calibrations, within a limited evaluation budget. A quantifiable measure in this regard is the number of meaningful non-dominated calibrations obtained by an MO algorithm within a limited evaluation budget. It is also important that the meaningful non-dominated calibrations obtained by an algorithm are significantly different from each other (or diverse).

We introduce a new metric, Distributed Cardinality, denoted as $|ND(L, \epsilon)|$, to measure the success of the MO search given a meaningfulness specified by level L. The set $ND(L, \epsilon)$ is the collection of all ϵ-*box non dominated* solutions obtained from an MO algorithm that are meaningful at a Level L. $|ND(L, \epsilon)|$ is then the cardinality of the set $ND(L, \epsilon)$.

Meaningful levels are essentially user-defined bounds on the Pareto set, and a calibration expert may subjectively define meaningful levels. (See Table 3.2 for the meaningful levels defined for our case studies.)

The ϵ in the set $ND(L, \epsilon)$ is based on the concept of ϵ-*box dominance archiving* used by Deb et al. [12] and Kollat and Reed [29]. The ϵ-*box dominance archiving* concept divides the m-dimensional objective space into hypercubes / boxes of size ϵ (which is a user-defined vector, i.e, $\epsilon = [\epsilon_1, \ldots, \epsilon_m]$). If there are multiple non-dominated solutions inside a box, then only one of the solutions is retained in the ϵ-*box dominance archive*.The lowest objective function value of a

box (lowest left corner) represents its ϵ-*non domination* value. Hence, if a box is dominated by another box (as per the ϵ-*non domination* value), non-dominated solutions residing in the dominated box are removed from the ϵ-*box dominance archive*. The purpose of this sorting procedure is to identify diverse solutions from the set of meaningful non-dominated solutions. More details are given in Deb et al. [12] and Kollat and Reed [29].

The distributed cardinality index is a performance metric that we define to be the number of ϵ-*box dominance archiving* points that are also meaningful solutions. Hence a higher value of the distributed cardinality index for Algorithm A than for Algorithm B indicates that Algorithm A has been successful at finding more solutions that are both meaningful and diverse. As explained earlier, "meaningful" is user defined, so in this analysis we consider three definitions of "meaningful" called *Level-K, K=1,2,3*. The size of the ϵ vector is also user-defined and in this analysis $\epsilon = [0.002, 0.001, 0.0001]$. Discussion of results pertaining to meaningful calibrations is in Section 3.10.

Table 3.2: Definitions of the three levels of meaningful solutions employed in the meaningful trade-off analysis

Meaningful Level	**Correlation (r)**	**Relative Variability (α)**	**Relative Bias(β)**
Level-I	≥ 0.70	$\leq 15\%$	$\leq 10\%$
Level-II	≥ 0.75	$\leq 15\%$	$\leq 8\%$
Level-III	≥ 0.80	$\leq 15\%$	$\leq 8\%$

3.9 Results and Analysis - Hypervolume Metric

3.9.1 Progress Graphs

The relative efficiency of the algorithms discussed in this analysis can be summarized by *progress graphs*. *Progress graphs* plot values of the uncovered hypervolume metric against number of function evaluations, to visualize algorithm progress with time. Before moving towards a detailed discussion of comparative performance of the algorithms, we summarize their relative performances on all watershed test problems in Figure 3.5 through *progress graphs*.

Each sub-figure corresponds to one watershed test problem and provides visualizations of algorithm progress with number of function evaluations according to the uncovered hypervolume metric. As was mentioned earlier, the uncovered hypervolume metric tends to highlight both convergence and diversification and lower values of the metric are desirable.

Two deductions are evident from the analysis in Figure 3.5: 1) Overall average performance of GOMORS is better than all other algorithms for all watershed test problems within a limited watershed simulation evaluation budget of 1000 and 2) NSGA-II and MOEA/D are the least efficient algorithms for all watershed test problems within a limited evaluation budget of 1000.

The *progress graphs* clearly indicate that average performance of GOMORS is better than that of ParEGO and AMALGAM. The *progress graphs* also indicate relative superiority of GOMORS, ParEGO and AMALGAM over NSGA-II and MOEA/D for a limited evaluation budget of 1000.

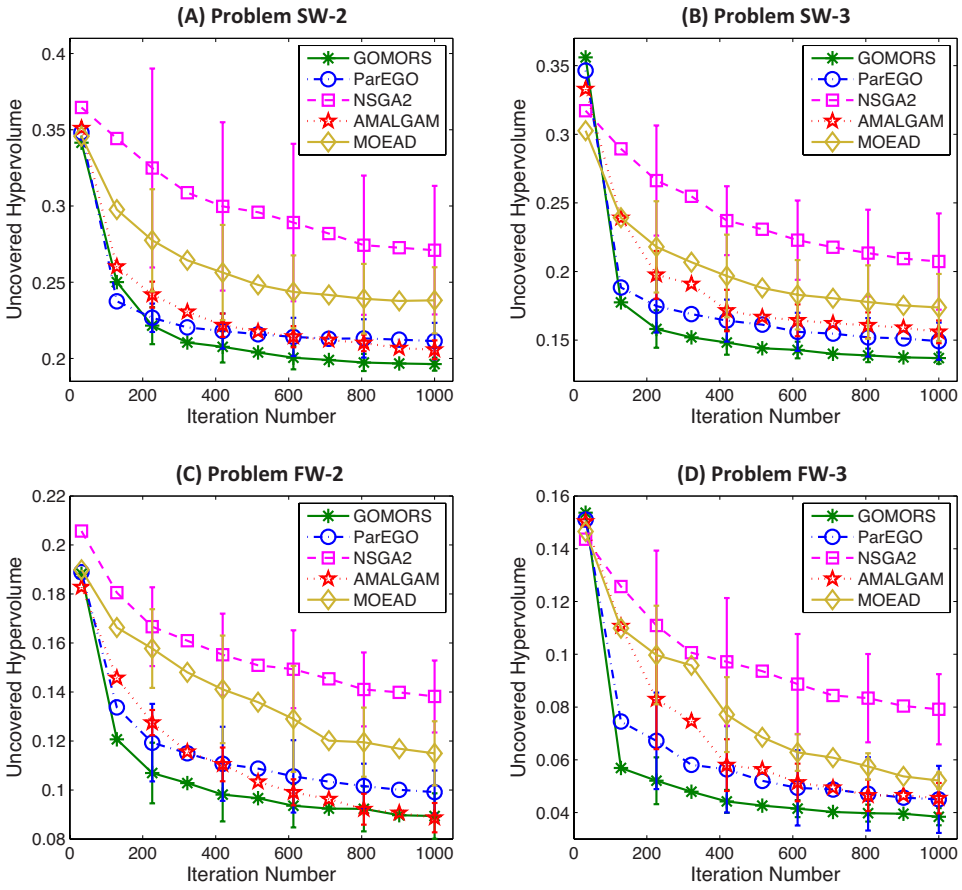

Figure 3.5: ***Progress Graphs for all case studies*** (Table 4.1): Progress graphs with plots of average uncovered hypervolume values against number of function evaluations, averaged over 10 trials, with application to: A) Problem SW-2, B) Problem SW-3, C) Problem FW-2 and D) Problem FW-3. The error bars show standard deviations for each algorithm. Lower curves are best.

3.9.2 Statistical Analysis

In order to analyze the difference in performance between GOMORS, ParEGO and AMALGAM in further detail, the Mann-Whitney Rank Sum test [11] was performed over the uncovered hypervolume metric values obtained for each algorithm in multiple trials. The Rank Sum test is a non-parametric statistical hypothesis test for deducing whether results obtained from one algorithm in multiple trial runs are significantly different from results obtained from another

algorithm in multiple trials. The algorithms are compared in pairs, the uncovered hypervolume metric value is used as the performance quality measure and the Rank Sum Test is performed for all watershed problems after 200, 500 and 1000 evaluations of each algorithm are complete. Hence, there are 24 Rank Sum tests for each algorithm, 6 for each test problem.

A summary of the Mann-Whitney Rank Sum Test is provided in figure 3.6. The columns headings of all tables in the figure depict the algorithms that were compared against each other and the corresponding entries of the columns show the results of the Rank Sum Test for 200, 500 and 1000 evaluations. The indicated results in the table cells are the p-values of the Rank Sum Test.

The p-value is positive if the first algorithm listed is better than the second for some significance level. The p given indicates that the alternative hypothesis (that the first algorithm has a better mean than the second algorithm) is accepted at the significance level indicated by the p-value. The statistical evidence that the first algorithm is better than the second algorithm, is strongest if the p value is small.

We see in Figure 3.6, that for the tests with GOMORS listed first the p-values are less than 0.1 (indicating GOMORS is better than the alternative algorithm at a 10 percent significance level) in 21 our of 24 cases. In the remaining 3 cases the p is positive so none of alternatives to GOMORS is better than GOMORS, but the advantage of GOMORS is not statistically significant.

Figure 3.6 has positive p-values for all the tests where GOMORS is listed first, indicating that none of the algorithms is shown to be statistically better than GOMOR for any of the 12 combinations of problem (SW-2, SW-3, FW-2, FW-3) and numbers of evaluations (200, 500, or 1000). For 10 out of the 12 cases of GOMORS versus AMALGAM comparison, the p value is very low (less than

(A) - Problem *SW-2*

Algorithms / Evaluations	*GOMORS - ParEGO*	*GOMORS - AMALGAM*	*ParEGO - AMALGAM*
200	0.473	0.006	0.0004
500	0.026	0.001	0.734
1000	0.007	0.002	0.385

(B) - Problem *SW-3*

Algorithms / Evaluations	*GOMORS - ParEGO*	*GOMORS - AMALGAM*	*ParEGO - AMALGAM*
200	0.021	0.0003	0.009
500	0.007	0.0006	0.521
1000	0.011	0.0002	0.186

(C) - Problem *FW-2*

Algorithms / Evaluations	*GOMORS - ParEGO*	*GOMORS - AMALGAM*	*ParEGO - AMALGAM*
200	0.064	0.001	0.026
500	0.031	0.054	0.385
1000	0.038	0.345	- 0.014

(D) - Problem *FW-3*

Algorithms / Evaluations	*GOMORS - ParEGO*	*GOMORS - AMALGAM*	*ParEGO - AMALGAM*
200	0.076	0.0003	0.162
500	0.104	0.001	0.186
1000	0.344	0.009	0.241

Figure 3.6: Summary of statistical comparison via Mann-Whitney Rank Sum Test applied to GOMORS, ParEGO and AMALGAM, according to uncovered hypervolume metric. Columns in all tables correspond to the pair of algorithms which are compared, and rows correspond to the number of function evaluations after which algorithms are compared. Table cells correspond to the p-values obtained from the Rank Sum Test. Small and positive p-values strongly support the hypothesis that the first algorithm (listed in column head) is better than the second algorithm.

0.009), indicating the hypothesis that GOMORS is better than AMALGAM at a 98 percent significance level, which is a very strong result. The overall performance of GOMORS is better than ParEGO since all the p values in the GOMORS

versus ParEGO tests are positive and 7 out of 12 are below 0.05 (95 percent significance level).

While ParEGO's performance may not be as good as GOMORS but it is distinctly better than AMALGAM as indicated by the positive p values in 11 out of the 12 cases (in the ParEGO-AMALGAM columns in Figure 3.6). GOMORS, ParEGO and AMALGAM all decisively outperform NSGA-II and MOEA/D (see Fig. 3.5). Hence, NSGA-II and MOEA/D were not included in the rank sum test analysis. Our analysis highlights that GOMORS frequently outperforms AMALGAM and ParEGO. However, none of the other algorithms outperforms GOMORS on any case study.

3.10 Meaningful Trade-off Analysis

3.10.1 Algorithm Comparison

The analysis from section 3.9 indicates that the surrogate based search algorithm GOMORS performs well across all case studies in terms of robustness and efficiency, and its overall performance is better than all other algorithms discussed in this study. The analysis also highlights that surrogate based algorithms (GOMORS and PArEGO) and multi method evolutionary algorithms (AMALGAM) can be more efficient than traditional MOEAs, for hydrological model calibration within a limited simulation evaluation budget. In order to further assess the comparative effectiveness of GOMORS, ParEGO and AMALGAM, within the context of watershed model calibration, we employ the meaningful trade-off analysis discussed in Section 3.8.3.

We investigate the ability of GOMORS, AMALGAM and ParEGO, to produce acceptable or meaningful solutions, by analyzing the bounded Pareto

Fronts obtained from each algorithm. This excludes solutions the user defines to not be desirable. The meaningful trade-off analysis is performed for the *SW-3* and *FW-3* problems (NSE decomposition formulations).

As discussed in Section 3.8.3, the first step in this process is to define bounds for 1) correlation between observed data and simulated model output, 2) relative bias of simulated data and observed data, and 3) relative variability of simulated data and observed data. Three sets of bounds are defined in our analysis, and are referred as *level-I*, *level-II* and *level-III* meaningful bounds respectively (see Table 3.2 for definitions of meaningful levels).

Table 3.3: Distributed cardinality index values for comparing performance of all algorithms in terms of identifying *level-I*, *level-II* and *level-III* meaningful and diverse trade-offs (highest median values of $|ND(L, \epsilon)|$ are best) within 500 simulation evaluations - ***Problem SW-3***

Meaningful Level	**Algorithm**	$\|ND(L, \epsilon)\|$ **Index**		
		Min	**Median**	**Max**
Level-I	GOMORS	3	**5**	10
	ParEGO	1	5	10
	AMALGAM	1	3	9
Level-II	GOMORS	1	**3**	5
	ParEGO	0	0	1
	AMALGAM	0	0	0
Level-III	GOMORS	0	0	0
	ParEGO	0	0	0
	AMALGAM	0	0	0

Table 3.4: Distributed cardinality index values for comparing performance of all algorithms in terms of identifying *level-I*, *level-II* and *level-III* meaningful and diverse trade-offs (highest median values of $|ND(L,\epsilon)|$ are best) within 500 simulation evaluations - ***Problem FW-3***

Meaningful Level	**Algorithm**	$\|ND(L,\epsilon)\|$ **Index**		
		Min	**Median**	**Max**
Level-I	GOMORS	6	**13.5**	19
	ParEGO	2	7	12
	AMALGAM	3	6.5	11
Level-II	GOMORS	5	**8**	14
	ParEGO	2	6	11
	AMALGAM	2	3.5	7
Level-III	GOMORS	0	**4**	9
	ParEGO	0	2.5	11
	AMALGAM	0	1	7

As is discussed in Section 3.8.3, the distributed cardinality index is introduced for comparison of GOMORS, ParEGO and AMALGAM in terms of producing meaningful trade-offs. The distributed cardinality index (also referred as $|ND(L,\epsilon)|$) is an indicator of the number of meaningful and diverse (diversity is addressed through the concept of ϵ-*box* non domination) non-dominated calibrations identified by an algorithm. Hence, higher values of the metric are desirable. Table 3.3 provides a performance metric summary (in terms of producing meaningful and diverse trade-off solutions) of algorithms on a limited evaluation budget of 500 evaluations, with application to the *SW-3* test problem.

The metric analysis shows that GOMORS is significantly better than ParEGO for identification of *level-II* meaningful calibrations. While ParEGO fails in identifying even a single *level-II* meaningful calibration in 9 out of 10 algorithm runs, GOMORS successfully identifies at least one *level-II* meaningful calibration in 10 algorithm runs. GOMORS and ParEGO perform reasonably well, though, in terms of identifying *level-I* meaningful calibrations.

Performance of GOMORS is better than AMALGAM in terms of producing *level-I* and *level-II* meaningful calibrations. None of the algorithms is able to produce even a single *level-III* meaningful calibration in multiple trials. This may be due to the structural deficiencies inherent in the structure of the simulation model (Section 3.10.2 discusses the insights that may be derived from the distributed cardinality index at different meaningful levels).

Table 3.4 provides a similar metric analysis summary for the *FW-3* case study. Table 3.4 also shows that GOMORS is more effective than both ParEGO and AMALGAM, within a limited evaluation budget, in the ability to produce meaningful and diverse trade-offs / calibrations. It can be noted that the number of meaningful and diverse calibrations (deduced through the distributed cardinality index) obtained from each algorithm decrease with an increase in the meaningful level. This is logical since a higher meaningful level signifies a higher level of aspiration towards obtaining good quality solutions in terms of all objectives (via administration of stricter bounds on the non-dominated front).

While the distributed cardinality index is a good measure for comparing effectiveness of different algorithms within the context of watershed model calibration, it can also be used to derive numerous insights during the multi objective calibration process. The next section provides a short illustration of how the

distributed cardinality index ($|ND(L, \epsilon)|$) may be effectively used by a calibration expert in deriving insights from the multi-objective calibration process.

3.10.2 Deriving Insights from the Distributed Cardinality index

Numerous studies have highlighted the value of multi criteria analysis in watershed model calibration in terms of identification of structural inadequacies, model limitations, parameter sensitivity, parameter uncertainty etc [8, 16, 17, 22, 59]. This section analyzes multi objective calibration results of all non dominated calibrations and meaningful calibrations (deduced via the distributed cardinality index) obtained via GOMORS to derive insights on some of the aspects mentioned above. For this purpose, we use the median solutions (as per uncovered hydervolume) obtained via GOMORS (after 500 evaluations), for both three objective case studies, i.e, *SW-3* and *FW-3*.

Figure 3.7 provides an insightful illustration of the value of information provided by multi-objective calibration analysis and the distributed cardinality index for the *SW-3* case study. Figure 3.7(a) provides a visualization of the simulated hydrographs of all non dominated calibrations obtained via GOMORS (after 500 evaluations), plotted against the observed hydrograph. The uncertainty associated with the multi objective trade-off is evident here. An insight that can be derived from Figure 3.7(a) is that the model does not adequately capture the flashiness and shape of the observed hydrograph. This could be a result of model limitation / structural inadequacy.

Figure 3.7 also gives an illustration of how a small set of equally good calibration alternatives may be identified by the the distributed cardinality index.

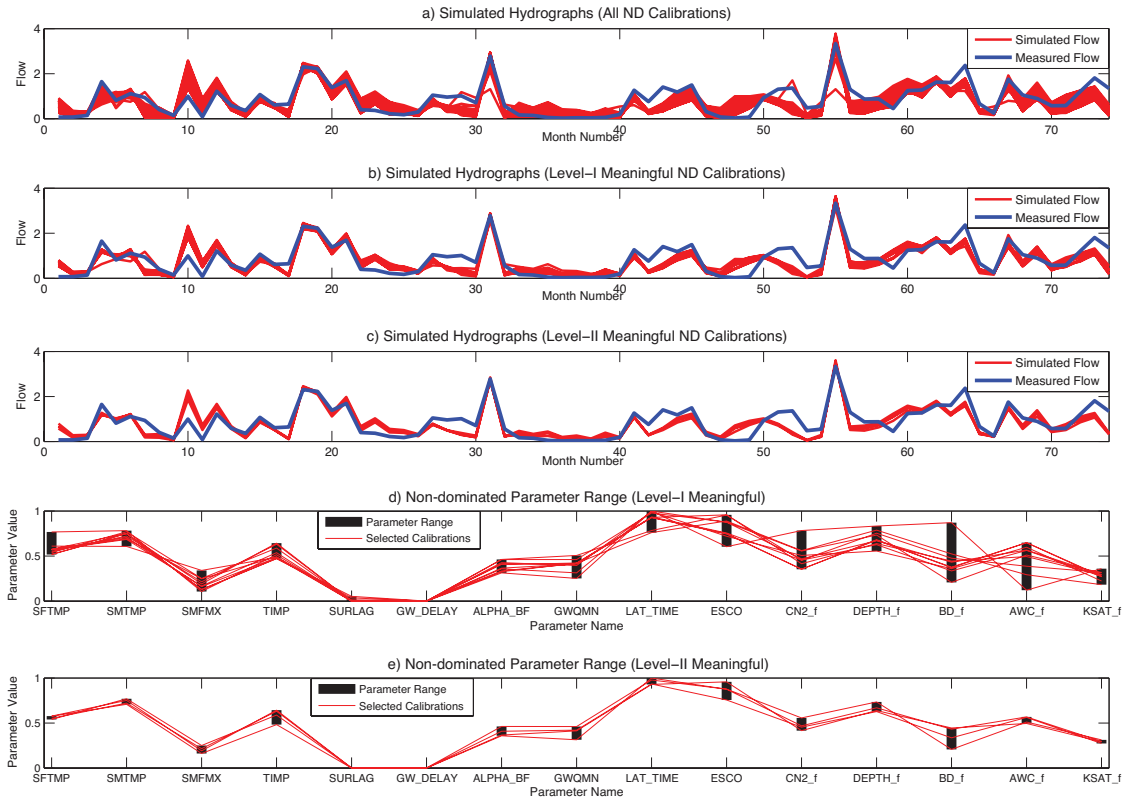

Figure 3.7: Simulated hydrographs and corresponding calibration alternatives obtained from multi-objective optimization (Problem *SW-3*) via GOMORS after 500 function evaluations (median trial): a) Hydrographs visualized for all non-dominated calibrations (54 ND calibrations found), b) Hydrographs visualized for *Level-I* meaningful ϵ-*box* non-dominated calibrations (10 ND calibrations found), c) Hydrographs visualized for *Level-II* meaningful ϵ-*box* non-dominated calibrations (4 ND calibrations found), d) Meaningful calibrations associated with the hydrographs of (b), e) Meaningful calibrations associated with the hydrographs of (c).

For instance, visualization of hydrographs of the *level-II* meaningful calibrations (see Figure 3.7(c)) depicts a narrow band of uncertainty. (Four *level-II* meaningful calibrations are identified by the distributed cardinality index). Furthermore, sensitivity of the model parameters can also be assessed by visualizing the equally good parameter sets (see Figure 3.7(e)). For instance, Figure 3.7(e)

illustrates that SURLAG is a sensitive model parameter and BD_F is not a very sensitive model parameter for the *SW-3* case study.

Figure 3.4 provides an illustrative view of the value of information that can be extracted from multi objective calibration and the distributed cardinality index for the *FW-3* case study. Visualization of simulated hydrographs of all non dominated calibrations (from the median run of GOMORS after 500 evaluations) plotted against the observed hydrograph (see Figure 3.4(a)) depicts the uncertainty associated with the multi objective trade-off. Here we observe that the uncertainty band of the simulated hydrographs adequately encapsulates the shape of the observed hydrograph (apart from the rising limb on the hydrograph in some instances). Hence, an insight that may be derived from Figure 3.4(a) is that the model structure adequately simulates flow for the *FW-3* case study.

The distributed cardinality index is effective in identifying a small set of meaningful calibration alternatives for the *FW-3* case study. Figures 3.4(b) and 3.4(c) illustrate how the uncertainty in simulations is reduced by identification of meaningful calibrations via the distributed cardinality index. At *level-III* meaningfulness, the distributed cardinality index identified three calibration alternatives for the *FW-3* case study. Figure 3.4(e) plots these calibration alternatives together to illustrate the inherent uncertainty and sensitivity of the model parameters. For instance, Figure 3.4(e) illustrates that CN2_F is a sensitive model parameter and SURLAG is not a sensitive model parameter.

The illustrative analysis of Figures 3.4 and 3.7 highlights the potential advantages of efficient multi-objective calibration of computationally expensive models and the value of information that can be derived from multi objective calibration and the distributed cardinality index. GOMORS is an efficient multi-

objective optimization algorithm which can be effectively used in combination with the distributed cardinality index by modeling experts in the process of model calibration.

3.11 Visual Trade-off Analysis

Section 3.10 provides a comprehensive assessment of effectiveness of algorithms in terms of their ability to identify meaningful calibrations, with reference to the two three-objective test problems. For comparing algorithms' effectiveness with reference to the two bi-objective test problems, we use the visual trade-off analysis (since it is easy to visualize and analyze bi-objective trade-offs).

Figure 3.8 provides a comparison of GOMORS, ParEGO and AMALGAM with 500 or 1000 evaluations, against the estimated Pareto front (estimated via a single run of AMALGAM with 10,000 evaluations) for the *SW-2* problem. Parts A-C of Figure 3.8 visualize the best and worst trade-offs obtained by each algorithm after 500 function evaluations, and D-F depict best and worst trade-offs obtained after 1000 evaluations. The best and worst trade-off solutions correspond to the best and worst runs of each algorithm in multiple trials and are determined by the uncovered hypervolume metric value. The red line in each subfigure depicts the trade-off obtained via AMALGAM after 10,000 function evaluations, which is assumed to be the estimated Pareto optimal trade-off.

The figure indicates that while all algorithms tend to converge quickly, the convergence efficiency obtained via GOMORS is the best since it does best for 500 evaluations. The closeness of the best and worst trade-off curves obtained via GOMORS, also supports reliability of the algorithm. Reliability here means that there is less variability in algorithm results on one problem over multiple trials. Figure 3.8 shows that the best solutions obtained via all algorithms

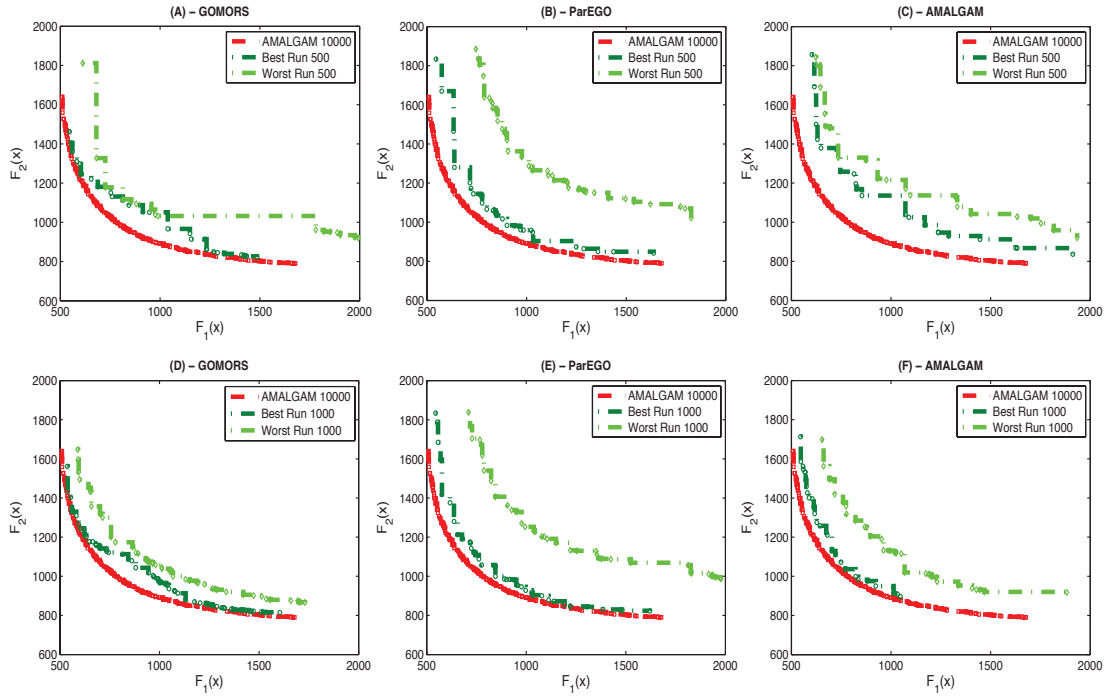

Figure 3.8: ***Problem SW-2***: Comparison of best and worst trade-offs obtained via GOMORS, ParEGO and AMALGAM in 10 trials, plotted against the "estimated true" Pareto front, approximate via an AMALGAM run with 10,000 evaluations: A) GOMORS trade-offs after 500 evaluations, D) GOMORS trade-offs after 1000 evaluations, B) ParEGO trade-offs after 500 evaluations, E) ParEGO trade-offs after 1000 evaluations, C) AMALGAM trade-offs after 500 evaluations, and F) AMALGAM trade-offs after 1000 evaluations.

after 1000 evaluations compare well against the estimated Pareto front, with ParEGO's performance being the best. However, ParEGO's worst trade-off solution depicts relatively poor performance, indicating that the algorithm is not as reliable as GOMORS, in terms of application to the *SW-2* case study.

Figure 3.9 provides a similar comparison for the *FW-2* case study. It is evident from Figure 3.9 as well, that GOMORS converges more quickly than ParEGO and AMALGAM at both 500 and 1000 evaluations, and produces more diverse solutions within a limited evaluation budget. Furthermore GOMORS is

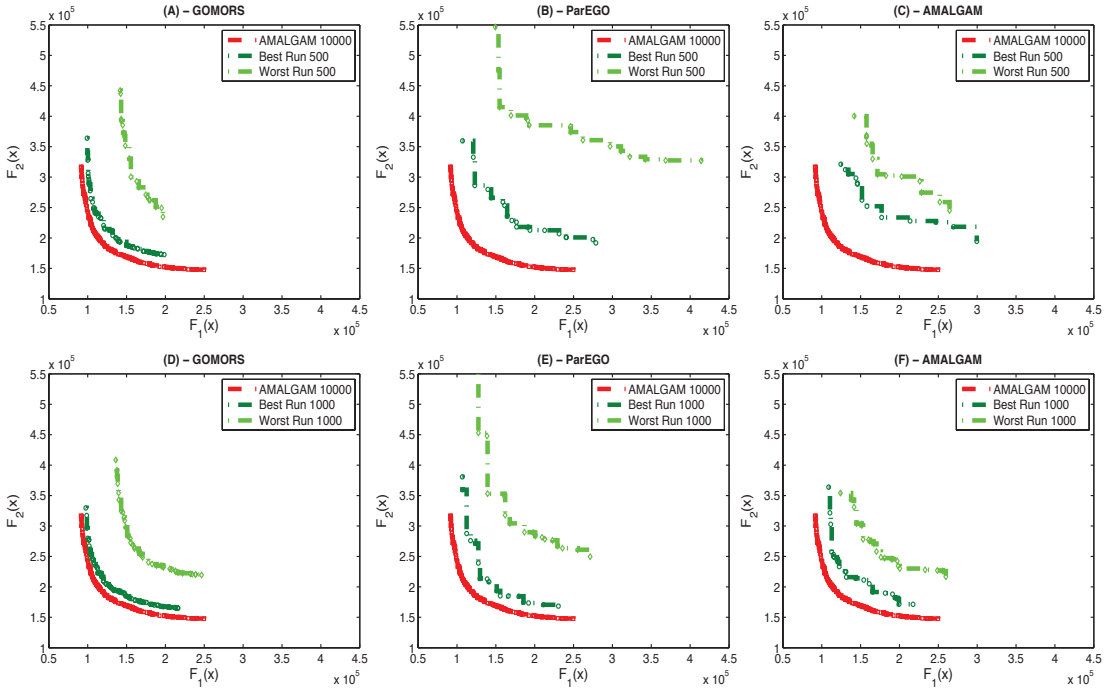

Figure 3.9: ***Problem FW-2***: Comparison of best and worst trade-offs obtained via GOMORS, ParEGO and AMALGAM in 10 trials, plotted against the "estimated true" Pareto front, approximate via an AMALGAM run with 10,000 evaluations: A) GOMORS trade-offs after 500 evaluations, D) GOMORS trade-offs after 1000 evaluations, B) ParEGO trade-offs after 500 evaluations, E) ParEGO trade-offs after 1000 evaluations, C) AMALGAM trade-offs after 500 evaluations, and F) AMALGAM trade-offs after 1000 evaluations.

relatively more reliable than ParEGO, and has the ability to produce comparable solutions to AMALGAM (after 10,000 function evaluations), with 90 percent fewer function evaluations. AMALGAM was selected to estimate the Pareto optimal front based on a 10,000 evaluation test run, because both GOMORS and ParEGO are specifically designed for a limited evaluation budget only.

3.12 Conclusion

This study provides a brief introduction to various popular and state-of-the art algorithms, including MOEAs and surrogate assisted search methods, and understands their relative effectiveness in hydrological model calibration, within a limited budget of simulation evaluations. We introduce a new multi-objective algorithm, GOMORS, and compared it two older multi-objective algorithms (AMALGAM and ParEGO) that have been shown in previous papers to be more efficient than the widely used NSGA-II. GOMORS and ParEGO make use of a surrogate surface to enhance computational efficiency. AMALGAM is an evolutionary algorithm with no surrogate surface.

The algorithms were applied to the multi-objective calibration of a large watershed SWAT model using many years of flow data. AMALGAM and ParEGO have been used in water resources applications, but GOMORS has not been applied previously in water resources [15, 63]. This study has focused on multi-objective optimization of costly watershed models for which it is not feasible to do thousands of simulations. Hence, the performance of the algorithms were compared on budgets of 500 and 1000 evaluations. Furthermore, the effectiveness of all algorithms were evaluated in terms of their ability to reliably produce trade-off solutions with good convergence and diversification capabilities, within a limited evaluation budget.

A statistical testing analysis shows that GOMORS is not outperformed by any algorithm, within an evaluation budget of 1000, after application to 4 watershed calibration case studies. Further analysis also shows that GOMORS, ParEGO and AMALGAM are significantly more efficient that NSGA-II and MOEA/D.

A further analysis, with identification of meaningful trade-offs only, also

shows that the two surrogate methods GOMORS and ParEGO are successful in producing various meaningful calibration alternatives within a limited evaluation budget. The performance of GOMORS is better than ParEGO in this regard. A new performance, referred as the $|ND(L, \epsilon)|$ index is used for this analysis. In this analysis we have also illustrated how the $|ND(L, \epsilon)|$ index can be used for deriving numerous insights from efficient multi objective calibration.

We also visually compare the non dominated fronts (for bi-objective case studies) obtained by GOMORS, ParEGO and AMALGAM after 1000 evaluations with an approximation of the true Pareto Front (approximated via AMALGAM solutions obtained after 10,000 evaluations). This analysis shows that solutions obtained by GOMORS converge more quickly to the approximated Pareto front than ParEGO and AMALGAM. Hence the comparison depicts that, even within a limited evaluation budget, GOMORS can converge quickly, and provide an acceptable set of calibration alternatives.

Given the computational burden associated with multi-objective calibration of large semi-distributed watershed models, computationally efficient algorithms can be employed to produce good and meaningful calibration alternatives within a very limited simulation evaluation budget. The surrogate methods GOMORS and ParEGO are very promising algorithms, which can be used more frequently to efficiently calibrate watershed models. While both algorithms are very promising, GOMORS can be easily modified to incorporate modest parallelization, and hence can be more efficient in producing good calibration alternatives. Due to the increasing complexity of watershed models, GOMORS can be efficiently employed within the calibration process to produce good and meaningful calibration alternatives.

3.13 References

[1] Taimoor Akhtar and Christine A. Shoemaker. Multi objective optimization of computationally expensive multi-modal functions with rbf surrogates and multi-rule selection. *Journal of Global Optimization*, pages 1–16, 2015.

[2] Johannes M. Bader. *Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods*. CreateSpace, Paramount, CA, 2010.

[3] Domenico A. Bau and Alex S. Mayer. Stochastic management of pump-and-treat strategies using surrogate functions. *Advances in Water Resources*, 29(12):1901 – 1917, 2006.

[4] Kourosh Behzadian, Zoran Kapelan, Dragan Savic, and Abdollah Ardeshir. Stochastic sampling design using a multi-objective genetic algorithm and adaptive neural networks. *Environmental Modelling and Software*, 24(4):530 – 541, 2009.

[5] E Bekele and J Nicklow. Multi-objective automatic calibration of SWAT using NSGA-II. *Journal of Hydrology*, 341(3-4):165–176, August 2007.

[6] Keith Beven and Andrew Binley. The future of distributed models: Model calibration and uncertainty prediction, July 1992.

[7] Nikolay Bliznyuk, David Ruppert, Christine Shoemaker, Rommel Regis, Stefan Wild, and Pradeep Mugunthan. Bayesian calibration and uncertainty analysis for computationally expensive models using optimization and radial basis function approximation. *Journal of Computational and Graphical Statistics*, 17(2):270–294, 2008.

[8] Douglas P. Boyle, Hoshin V. Gupta, and Soroosh Sorooshian. Toward im-

proved calibration of hydrologic models: Combining the strengths of manual and automatic methods. *Water Resources Research*, 36(12):null+, 2000.

[9] A. Castelletti, F. Pianosi, R. Soncini-Sessa, and J. P. Antenucci. A multiobjective response surface approach for improved water quality planning in lakes and reservoirs. *Water Resources Research*, 46(6):n/a–n/a, 2010. W06502.

[10] C.A. Coello Coello, G.L. Lamont, and D.A. van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation. Springer, Berlin, Heidelberg, 2nd edition, 2007.

[11] W. J. Conover. *Practical Nonparametric Statistics*. John Wiley & Sons, December 1998.

[12] K Deb, M Mohan, and S Mishra. A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions. Technical Report 2003002, KanGAL, February 2003.

[13] Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 1 edition, June 2001.

[14] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, PPSN VI, pages 849–858, London, UK, UK, 2000. Springer-Verlag.

[15] Francesco di Pierro, Soon-Thiam Khu, Dragan Savić, and Luigi Berardi. Efficient multi-objective optimal design of water distribution networks on

a budget of simulations using hybrid algorithms. *Environ. Model. Softw.*, 24 (2):202–213, February 2009.

[16] Qingyun Duan, Hoshin V. Gupta, Soroosh Sorooshian, Alain N. Rousseau, and Richard Turcotte. *Calibration of Watershed Models*. American Geophysical Union, 2013.

[17] Andreas Efstratiadis and Demetris Koutsoyiannis. One decade of multi-objective calibration approaches in hydrological modelling: a review. *Hydrological Sciences Journal*, 55(1):58–78, February 2010.

[18] A Espinet, Christine A Shoemaker, and C Doughty. Estimation of Plume distribution for Carbon Sequestration using parameter estimation with limited monitoring data. *Water Resources Research*, 49(7):4442–4464, 2013.

[19] Antoine J. Espinet and Christine A. Shoemaker. Comparison of optimization algorithms for parameter estimation of multi-phase flow models with application to geological carbon sequestration. *Advances in Water Resources*, 54(0):133 – 148, 2013.

[20] Hoshin V Gupta, Harald Kling, Koray K Yilmaz, and Guillermo F Martinez. Decomposition of the mean squared error and NSE performance criteria : Implications for improving hydrological modelling. *Journal of Hydrology*, 377(1-2):80–91, 2009.

[21] Hoshin V. Gupta, Soroosh Sorooshian, Terri S. Hogue, and Douglas P. Boyle. *Advances in Automatic Calibration of Watershed Models*, pages 9–28. American Geophysical Union, 2013.

[22] Hoshin Vijai Gupta, Soroosh Sorooshian, and Patrice Ogou Yapo. Toward

improved calibration of hydrologic models: Multiple and noncommensurable measures of information. *Water Resources Research*, 34(4):751, 1998.

[23] H.-M. Gutmann. A radial basis function method for global optimization. *J. of Global Optimization*, 19:201–227, March 2001.

[24] David Hadka and Patrick Reed. Borg: An auto-adaptive many-objective evolutionary computing framework. *Evol. Comput.*, 21(2):231–259, May 2013.

[25] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *J. of Global Optimization*, 13 (4):455–492, December 1998.

[26] N Kajami, H Gupta, T Wagener, and S Sorooshian. Calibration of a semi-distributed hydrologic model for streamflow estimation along a river system. *Journal of Hydrology*, 298(1-4):112–135, October 2004.

[27] Soon-Tham Khu, Dragan Savic, Yang Liu, and Henrik Madsen. A fast evolutionary-based meta-modelling approach for the calibration of a rainfall-runoff model. In *In Pahl-Wostl, C., Schmidt, S. and Jakeman, T. (eds) iEMSs 2004 International Congress: Complexity and Integrated Resources Management*. International Environmental Modelling and Software Society, 2004.

[28] J. Knowles. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 8(5):1341–66, February 2006.

[29] J B Kollat and P M Reed. A computational scaling analysis of multiobjec-

tive evolutionary algorithms in long-term groundwater monitoring applications. *Adv. Water Resour.*, 30(3):335–353, 2007.

[30] J. B. Kollat, P. M. Reed, and T. Wagener. When are multiobjective calibration trade-offs in hydrologic models meaningful? *Water Resources Research*, 48 (3):1–19, March 2012.

[31] Shie-Yui Liong, Soon-Tham Khu, and Weng-Tat Chan. Derivation of pareto front with genetic algorithm and neural network. *Journal of Hydrologic Engineering*, 6(1):52–61, 2001.

[32] H Madsen. Automatic calibration of a conceptual rainfallrunoff model using multiple objectives. *Journal of Hydrology*, 235(3-4):276–288, August 2000.

[33] H Madsen. Parameter estimation in distributed hydrological catchment modelling using automatic calibration with multiple objectives. *Advances in Water Resources*, 26(2):205–216, February 2003.

[34] Henrik Madsen, Geoffrey Wilson, and Hans Christian Ammentorp. Comparison of different automated strategies for calibration of rainfall-runoff models. *Journal of Hydrology*, 261:48–59, 2002.

[35] Juliane Mueller and Christine A Shoemaker. SO-I: A Surrogate Model Algorithm for Expensive Nonlinear Integer Programming Problems Including Global Optimization Applications. *Journal of Global Optimization*, 59: 865–889, 2013.

[36] Juliane Mueller, Christine A Shoemaker, and R Piche. SO-MI: A Surrogate Model Algorithm for Computationally Expensive Nonlinear Mixed-Integer, Black-Box Global Optimization Problems. *Computers and Operations Research*, 40(5):1383–1400, 2013.

[37] Pradeep Mugunthan and Christine A Shoemaker. Assessing the impacts of parameter uncertainty for computationally expensive groundwater models. *Water Resources*, 42:1–15, 2006.

[38] Pradeep Mugunthan, Christine A. Shoemaker, and Rommel G. Regis. Comparison of function approximation, heuristic, and derivative-based methods for automatic calibration of computationally expensive groundwater bioremediation models. *Water Resources Research*, 41(11):n/a–n/a, 2005. W11427.

[39] M Muleta and J Nicklow. Sensitivity and uncertainty analysis coupled with automatic calibration for a distributed watershed model. *Journal of Hydrology*, 306(1-4):127–145, May 2005.

[40] J.E. Nash and J.V. Sutcliffe. River flow forecasting through conceptual models part i a discussion of principles. *Journal of Hydrology*, 10(3):282 – 290, 1970.

[41] S.L. Neitsch, J.G. Arnold, J.R. Kiniry, and J.R. Williams. *Soil and Water Assessment Tool Input/Output File version 2005*. US Department of Agriculture Agricultural Research Service, September 2004.

[42] S.L. Neitsch, J.G. Arnold, J.R. Kiniry, and J.R. Williams. *Soil and Water Assessment Tool theoretical documentation version 2005*. US Department of Agriculture Agricultural Research Service, January 2005.

[43] John Nicklow, Patrick Reed, Dragan Savic, Tibebe Dessalegne, Laura Harrell, Amy C. Hilton, Mohammed Karamouz, Barbara Minsker, Avi Ostfeld, Abhishek Singh, and Emily Zechman. State of the Art for Genetic Algo-

rithms and beyond in Water Resources Planning and Management. *Journal of Water Resources Planning and Management*, 1(1):27, 2009.

[44] Saman Razavi, Bryan a. Tolson, and Donald H. Burn. Review of surrogate modeling in water resources. *Water Resources Research*, 48(7):n/a–n/a, July 2012.

[45] Saman Razavi, Bryan A. Tolson, and Donald H. Burn. Numerical assessment of metamodelling strategies in computationally intensive optimization. *Environmental Modelling and Software*, 34(0):67 – 86, 2012. Emulation techniques for the reduction and sensitivity analysis of complex environmental models.

[46] P. M. Reed, D. Hadka, J. D. Herman, J. R. Kasprzyk, and J. B. Kollat. Evolutionary multiobjective optimization in water resources: The past, present, and future. *Advances in Water Resources*, 51:438–456, January 2013.

[47] R Regis, C Shoemaker, and A. stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J. of Computing*, 19:497–509, 2007.

[48] R.G. Regis and C.a. Shoemaker. Local Function Approximation in Evolutionary Algorithms for the Optimization of Costly Functions. *IEEE Transactions on Evolutionary Computation*, 8(5):490–505, October 2004.

[49] Rommel G Regis and Christine A Shoemaker. Parallel Stochastic Global Optimization Using Radial Basis Functions. *INFORMS J. of Computing*, 21 (3):411–426, 2009.

[50] Rommel G Regis and Christine A Shoemaker. A Quasi-Multistart Frame-

work for Global Optimization of Expensive Functions using Response Surface Models. *Journal of Global Optimization*, (In press), 2013.

[51] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and Analysis of Computer Experiments. *Statistical Science*, 4(4):409–423, November 1989.

[52] Christine A Shoemaker, Rommel G Regis, and Ryan C Fleming. Watershed calibration using multistart local optimization and evolutionary optimization with radial basis function approximation. *New York*, 52(December): 450–465, 2007.

[53] Y. Tang, P. Reed, and T. Wagener. How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration? *Hydrology and Earth System Sciences*, 10(2):289–307, May 2006.

[54] Y. Tang, P.M. Reed, and J.B. Kollat. Parallelization strategies for rapid and robust evolutionary multiobjective optimization in water resources applications. *Advances in Water Resources*, 30(3):335–353, March 2007.

[55] B Tolson and C Shoemaker. Cannonsville reservoir watershed swat2000 model development, calibration and validation, J. *of Hydrology*, 337:68–86, 2007.

[56] Bryan A Tolson and Christine A Shoemaker. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. *Water Resources*, 43:1–16, 2007.

[57] Kathryn van Werkhoven, Thorsten Wagener, Patrick Reed, and Yong Tang. Characterization of watershed model behavior across a hydroclimatic gradient. *Water Resources Research*, 44(1):1–16, January 2008.

[58] David A. Van Veldhuizen and David A. Van Veldhuizen. Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations. Technical report, Evolutionary Computation, 1999.

[59] Jasper a. Vrugt. Improved treatment of uncertainty in hydrologic modeling: Combining the strengths of global optimization and data assimilation. *Water Resources Research*, 41(1):1–17, 2005.

[60] Jasper a Vrugt and Bruce a Robinson. Improved evolutionary optimization from genetically adaptive multimethod search. *Proceedings of the National Academy of Sciences of the United States of America*, 104(3):708–11, January 2007.

[61] Stefan M Wild and Christine A Shoemaker. Global Convergence of Radial Basis Function Trust-Region Algorithms for Derivative-Free Optimization. *SIGEST Article, SIAM Review*, 55(2):349–371, 2013.

[62] Stefan M Wild, Rommel G Regis, and Christine A Shoemaker. ORBIT: optimization by Radial Basis Function Interpolation in Trust-Regions. *SIAM Journal of Scientific Computing*, 30(6):3197–3219, 2007.

[63] Raghavan Srinivasan Xuesong Zhang and Michael Van Liew. On the use of multi-algorithm, genetically adaptive multi-objective method for multi-site calibration of the swat model. *Hydrological Processes*, 24:955–969, 2010.

[64] P Yapo, H Gupta, and S Sorooshian. Multi-objective global optimization for hydrologic models. *Journal of Hydrology*, 204(1-4):83–97, January 1998.

[65] Qingfu Zhang and Hui Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.

[66] Qingfu Zhang, Aimin Zhou, Shizheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Wudong Liu. Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition. *Mechanical Engineering*, pages 1–30, 2009.

[67] Xuesong Zhang, Raghavan Srinivasan, and Michael Van Liew. Approximating swat model using artificial neural network and support vector machine1. *JAWRA Journal of the American Water Resources Association*, 45(2): 460–474, 2009.

[68] Xuesong Zhang, Raghavan Srinivasan, and Michael Van Liew. On the use of multi-algorithm, genetically adaptive multi-objective method for multi-site calibration of the SWAT model. *Hydrological Processes*, 24(8):955–969, April 2010.

[69] Rui Zou, Wu-Seng Lung, and Jing Wu. An adaptive neural network embedded genetic algorithm approach for inverse water quality modeling. *Water Resources Research*, 43(8):n/a–n/a, 2007. W08427.

CHAPTER 4

EFFICIENT MULTI-OBJECTIVE OPTIMIZATION THROUGH PARALLEL SURROGATE ASSISTED LOCAL SEARCH

4.1 Introduction

It is the purpose of this paper to describe a new algorithm for multi objective optimization that is more efficient because it uses surrogate response surfaces as well as a parallel surrogate-assisted local search mechanism. Use of non-surrogate hybrid algorithms for multi-objective optimization, and the recent surge in development of multi-objective optimization algorithms (for simulation-optimization) has been dominated by multi-objective Evolutionary Algorithms (MOEA) [5]. The population based structure of evolutionary algorithms assists in maintaining convergence and diversity for multi-objective optimization [9]. Additionally, evolutionary algorithms can be easily parallelized using a simple synchronous master-slave framework [13] [22] (described in Section 4.3.2), and can significantly increase efficiency in solving computationally expensive simulation-optimization problems.

Local search methodologies have also been used frequently within hybrid evolutionary strategies and can be especially beneficial for specific problems [5]. Coello et al. (2007) [5] discuss the inherent advantages of incorporating local search schemes (for instance, hill climbing, simulated annealing, Tabu search, etc.) in hybrid algorithms and their success in terms of improving convergence, diversity, robustness and efficiency in multi-objective optimization.

While focus on local search for multi-objective optimization has significantly increased, there has been little progress in exploiting surrogate assistance in local search. Regis and Shoemaker (2007) [27] introduced the concept

of surrogate-assised candidate search for computationally expensive single objective optimization. Application of their work to various test problems and watershed calibration problems showed very promising results for a limited simulation evaluation budget [27]. Wild and Shoemaker [38] employed surrogate assistance within a trust region framework to assist in efficient optimization of computationally expensive problems.

We propose a "Multi-Objective Parallel Local Surrogate-Assisted" search algorithm (MOPLS) for efficient multi-objective optimization of computationally expensive simulation optimization problems. The primary aim of this study is to exploit the powers of a synchronous master-slave parallel framework, and surrogate-assisted local search for optimization within a limited budget of evaluations. An additional motivation of the study is to highlight the power of surrogate-assisted local search in terms of its potential advantages of hybridization with multi-method evolutionary algorithms (for example AMALGAM [36], Borg [16]). Vrugt et al. (2007) [36] propose the multi-method evolutionary strategy AMALGAM, which has been successfully applied in watershed model calibration studies [39] [22]. Hadka et al. (2007) [16] apply Borg, an auto-adaptive hybrid evolutionary algorithm for multi-objective optimization, to rainfall-runoff model calibration and groundwater monitoring problems with promising results in terms of scalability and robustness [26]. Five test problems and the watershed calibration test suite from Chapter 3 were used to compare performance of MOPLS with ParEGO [19], AMALGAM [36] and GOMORS on a limited budget of evaluations (less than 1000).

4.2 Problem Description

Let $\mathcal{D}$ (decision space) be a unit hypercube and a subset of $\mathbb{R}^d$ and x be a variable in the decision space, i.e., $x \in [0,1]^d$ (Any hyperrectangular domain can be mapped into the unit hypercube). Let the number of objectives equal k and let $F(x) = \{f_1(x), \ldots, f_k(x)\}$ be the vector of objectives such that $F(x) \in \mathcal{F}$ (objective space). So f_i denotes the ith objective, where f_i is a function of x and $f_i : \mathcal{D} \mapsto \mathbb{R}$ for $1 \leq i \leq k$. The framework of the multi-objective optimization problem we aim to solve is as follows:

$$\begin{aligned} &\text{minimize } F(x) = [f_1(x), \ldots, f_k(x)]^T \\ &\text{subject to } x \in [0,1]^d \end{aligned} \tag{4.1}$$

Note that any box-constraint domain can be normalized to a unit hypercube. The above problem can be solved by aggregating the objective vector by assigning weights to each objective and thereby converting the problem into a single objective optimization problem. However, this approach only yields one solution (determined by the weights chosen) and does not provide trade-off information of Pareto optimal solutions. Our focus is on employing a *posteriori* [6] preference optimization, where the purpose of the multi-objective optimization problem is to find a set of Pareto-optimal (or trade-off) solutions, $P^* = \{x^* \mid x^* \in \mathcal{D}\}$.

Definition 1 . *A solution $x_1 \in \mathcal{D}$ **dominates** another solution $x_2 \in \mathcal{D}$ ($x_1 \preceq x_2$) if and only if $f_i(x_1) \leq f_i(x_2)$ for all $1 \leq i \leq k$, and $f_i(x_1) < f_i(x_2)$ for some $1 \leq i \leq k$.*

Definition 2 *Given a set of solutions $S = \{x \mid x \in \mathcal{D}\}$, a subset of solutions $S^* = \{x^* \mid x^* \in S\}$ is **non dominated** in S if there does not exist a solution $x \in S$ which dominates $x^* \in S^*$, i.e, $S^* = \{x^* \in S \mid \nexists x \in S, x \preceq x^*\}$.*

Definition 3 *A set of solutions $P^* = \{x^* \in \mathcal{D}\}$ which is non-dominated in the decision space, i.e, $\mathcal{D}$ is called a* ***Pareto optimal set****. Hence, $P^* = \{x^* \in \mathcal{D} \mid \nexists x \in \mathcal{D}, x \preceq x^*\}$.*

As per the above definitions, the aim of a multi-objective optimization algorithm is to find the set of Pareto optimal solutions P^*. The set of objective vectors corresponding to P^* is called the the *Pareto front* and is denoted as P^*_{front}. Since computation of the objective function vector is expensive, the aim of our analysis is to deduce a set P^*_{approx} which closely approximates P^*, within a limited number of function evaluations.

4.3 Surrogate-assisted - Multi-objective Parallel Local Search (MOPLS)

The key elements of a search algorithm for multi-objective optimization are convergence to the true front and diversity with respect to the objective space. Various algorithms address the issue of establishing convergence and maintaining diversity. Elitism, diversity maintenance through crowding distance, hypervolume metric, IGD metric, and indicator-based evolutionary methods are some mechanisms employed in past literature, for assistance in the search for a good approximation of P^*. (For a detailed discussion on different search mechanisms, see [5] and [16].)

We are, however, interested in multi-objective optimization of computationally expensive functions, where the number of simulation evaluations available is limited, and hence, MOPLS employs surrogate- assisted search within a synchronous parallel framework in order to achieve good convergence and diversity within a limited number of simulations. When employing response surfaces (surrogate models) within the search mechanics, it is important that the search

maintains a balance between *exploration* and *exploitation* [17] within the decision space. *Exploration* aims to ensure that the search remains global by moving in unexplored areas of the decision space. *Exploitation* tends to exploit the nature of the (inexpensive) surrogate model to guide the search towards potentially optimal solutions.

The synchronous Mater-slave parallel framework of MOPLS (described in Section 4.3.2) does not require derivative information and is suitable for computationally expensive functions, where computation times of the expensive functions do not vary across the decision space. We assume that there are N_c processors available and simulation times for functions evaluations are roughly the same. The search mechanics for *exploration*, *convergence* and *diversity* are highly sensitive to N_c, and the balance of *exploration* and *exploitation* is obtained via maintenance of a tabu archive [12]. The general framework of the algorithm, and details of the search mechanism and tabu archive maintenance are provided in Sections 4.3.1 - 4.3.4.

4.3.1 General Algorithm Framework

The expensive optimization problem is solved via a simultaneous surrogate-assisted local *candidate search* (discussed further in Section 4.3.4) around various evaluated points, called *center points*. The parallel surrogate-assisted search methodology is iterative and the algorithm terminates after a fixed number of function evaluations. The master process controls other processes called slaves. The algorithm is divided into three major sections, namely 1) Initialization, 2) Iterative improvement and 3) Termination. The general framework of the algorithm and each of its major segments is as follows (see Table 4.1 for definitions of sets and variables):

Step 0 - Define Algorithm Inputs:

E_T - Maximum number of expensive function evaluations

E_I - Number of initial expensive evaluations

r_{init} - The initial local search radius (discussed in Step 2.2 of the algorithm)

N_c - Number of *center points* for simultaneous surrogated-assisted local search

Step 1 - Initial Evaluation Points Selection: The master task is initialized, and selects, using an experimental design, an initial set (ordered) of points $\{x_1, \ldots, x_m\}$, where $x_i \in \mathcal{D}$, for $1 \leq i \leq m$, and $m = E_I$. The master initiates N_c slave tasks and evenly distributes function evaluation tasks to the slaves. The slaves evaluate the objectives $F = [f_1, \ldots, f_k]^T$ at the selected E_I points, via expensive simulations, and return results to the master.

Let $\{y_1, \ldots, y_m\}$ be the objective evaluations corresponding to $\{x_1, \ldots, x_m\}$, i.e, $y_i = F(x_i)$ for $1 \leq i \leq m$. The master initiates the memory structures (as lists) $\{r_1, \ldots, r_m\}$, $\{c_1, \ldots, c_m\}$ and $\{c_1^{'}, \ldots, c_m^{'}\}$, where $r_i = r_{int}$, $c_i = 0$ and $c_i^{'} = 0$ for $1 \leq i \leq m$. Let $S_m = \{z_1, \ldots, z_m\}$ be defined as a multi-attribute ordered set, where $z_i = (x_i, y_i, r_i, c_i, c_i^{'})$, and y_i, r_i, c_i, and $c_i^{'}$ are the objective functions values (y_i) and memory attributes (r_i, c_i and $c_i^{'}$) corresponding to x_i for $1 \leq i \leq m$. The master also initializes $S_{taboo} = \{\}$, where S_{taboo} is the set of evaluated points not considered for local search within the iterative framework of the algorithm (Step 2 below). Let $P_m = \{z_i \in S_m \mid x_i \text{ is non-dominated in } \{x_1, \ldots, x_m\}\}$ be the set of non-dominated points from S_m.

Step 2 - Iterative Improvement: Run algorithm iteratively until termination condition is satisfied:

While $m \leq E_T$

> **Step 2.1 -Select Center Points:** The master task selects N_c center points from S_m based on non-dominated sorting and hypervolume contributions (discussed in Section 4.3.3 and Algorithm Step 2.1). Let $I_{cen} = \{ind_1 \ldots ind_{N_c}\}$ be the indices of the selected centers corresponding to the ordered set S_m.

Step 2.2 - Parallel Surrogate-Assisted Local Candidate Search: Let S_{cen} denote the set of evaluated points chosen as center points in Step 2.1, i.e, $S_{cen} = \{z_i \in S_m \mid j \in I_{cen}, i = j, 1 \leq j \leq N_c\}$. The master initiates N_c slave tasks and sends a center point, $z_i \in S_{cen}$ to each slave task, i, where $1 \leq i \leq N_c$:

For $i = 1, \ldots, N_c$

The slave task, i, performs a Surrogate-Assisted Local Candidate Search (discussed in detail in Section 4.3.4 and Algorithm Step 2.2) around the selected center point $z_i \in S_{cen}$ and selects (and evaluates) a point , x_i^*, for expensive function evaluation. The slave, i, returns z_i^* to the master task. Note that $z_i^* = (x_i^*, y_i^*, r_i^*, c_i^*, c_i^{*'})$ is a multi attribute element, where x_i^* is the point selected for expensive evaluation, $y_i^* = F(x_i^*)$, and r_i^*, c_i^*, and $c_i^{*'}$ are memory attributes corresponding to x_i^*.

End Loop

Let the ordered set $S_{curr} = \{z_1^* \ldots z_{N_c}^*\}$ correspond to all the evaluated points obtained from the parallel surrogate-assisted local search.

Step 2.3 - Update Tabu List and Memory archive: The tabu list, i.e, S_{taboo} is updated. For $i = 1, \ldots, m$, the memory attributes i.e, r_i, c_i and c_i', of $z_i = (x_i, y_i, r_i, c_i, c_i')$ are updated, where $z_i \in S_m$. Details on the procedure for updating the tabu list and the memory attributes, are discussed in Section 4.3.5 and Algorithm Step 2.3.

Step 2.4 - Update the evaluated set and the non-dominated set: The set of evaluated points is updated , i.e, $S_m = \{S_m\} \cup \{S_{curr}\}$ and $m = m + N_c$. Update the non-dominated set P_m, i.e, compute $P_m = \{z_i \in S_m \mid x_i$ is non-dominated in $\{x_1, \ldots, x_m\}\}$.

End Loop

Step 3 - Return Best Approximated Front: Return $P_{approx} = \{x_i \mid x_i$ is non-dominated in $\{x_1, \ldots, x_m\}\}$ as an approximation to the globally optimal solution set, i.e, $P^* = \{x_i^* \mid x_i^*$ is non-dominated in $\mathcal{D}\}$.

Table 4.1: Definitions of Sets and Variables

Item	Description
$F(x) = [f_1, \ldots, f_k]$	Expensive objectives for the multiple-objective optimization problem
$z_i = (x_i, y_i, r_i, c_i, c_i^{'})$	A multi-attribute element where x_i is the decision variable, y_i is the objective function vector corresponding to x_i, r_i is the local search radius of x_i, and c_i and $c_i^{'}$ are the failure count and tabu count, respectively, of x_i.
$S_m = \{z_1, \ldots, z_m\}$	Multi-attribute ordered set corresponding to all points, $\{x_1, \ldots, x_m\}$, which have been evaluated via costly simulation until iteration m of the algorithm.
$S_{cen} = \{z_1, \ldots, z_{N_c}\}$	The subset of evaluated points ($S_{cen} \subset S_m$) chosen as center points in Step 2.1.
$z_i^* = (x_i^*, y_i^*, r_i^*, c_i^*, c_i^{*'})$	A multi-attribute element, where x_i^* is the point selected for expensive evaluation after surrogate-assisted local search around the center point, x_i, in Step 2.2 (x_i is an attribute of $z_i \in S_{cen}$), $y_i^* = F(x_i^*)$, $r_i^* = r_{init}$, $c_i^* = 0$, and $c_i^{*'} = 0$.
$S_{curr} = \{z_1^* \ldots z_{N_c}^*\}$	Multi-attribute ordered of all evaluated points obtained from the parallel surrogate-assisted local search in Step 2.2.
S_{taboo}	The subset of evaluated points ($S_{taboo} \subset S_m$) which are in the taboo archive. The taboo archive is updated in Step 2.3.
$P_m = \{z \in S_m\}$	The *non-dominated* solutions in S_m based on expensive evaluations.

The search mechanics of MOPLS is dependent upon numerous parameters which are listed and defined in Table 4.2.

The algorithm initialization phase (Step 1) starts off with sampling of E_I points $x_1 \ldots x_{E_I}$ from decision space, $\mathcal{D}$, via an experimental design method. We use Latin hypercube sampling to generate the initial sample points [21]. Expensive simulation is employed to evaluate the values of all objectives for the initially sample points. Step 2, i.e, the iterative loop is the core component of the algorithm and is discussed in detail in Section 4.3.2 below.

4.3.2 Iterative Improvement

According to the search classifications discussed by Coello et al. [5], MOPLS falls under the category of Pareto sampling techniques, where the primary objective of the search is to move the set of non-dominated points towards the

Table 4.2: Definitions of Algorithm Parameters

Item	Description
E_I	The number of initial expensive evaluations.
E_T	The number of total expensive evaluations.
N_c	The number of *center points* (assumed to be equal to the number of available processors) to be employed for simultaneous local surrogate-assisted candidate search in each algorithm generation / iteration.
r_{init}	Search parameter which controls the size of the local search neighborhood, and the dynamics of the tabu list, and will be discussed further in section 4.3.2.

Pareto front and maintain diversity amongst the non-dominated set. This is done by performing simultaneous local searches around a subset of evaluated points (also called *center points*) in each algorithm iteration.

Step 2 of the algorithm constitutes the iterative framework of MOPLS. The iterative loop begins with selection of *center points* (Step 2.1), which is based on ranking of evaluated points in terms of their convergence and diversity potential (discussed in detail in Section 4.3.3). The process of selection of *center points* within the MOPLS iterative framework is synonymous with the process of selection of the parent population in an evolutionary algorithm. Selection of center points (or parent population) is followed by simultaneous surrogate-assisted local searches (Step 2.2) around each point within the population (discussed in detail in Section 4.3.4). A memory archive and tabu list are also maintained, and updated (Step 2.3) in each algorithm iteration / generation. The memory archive adaptively changes the local search neighborhood of the evaluated points. The tabu list is a secondary memory archive, which prohibits the inclusion of certain evaluated points, in the population of center points. The procedure of maintaining and updating the memory archive and tabu list is discussed in detail in Section 4.3.5.

The iterative improvement loop within MOPLS, hence, includes three major components, i.e, 1) selection of population of center points for local search, 2) simultaneous surrogate-assisted local searches around population of center points to choose and evaluate new points, and 3) updating of a memory archive and a *tabu list* of *center points*. The master process within the synchronous parallel framework selects the population of center points and maintains the *tabu list*, while the slave processes perform surrogate-assisted local search, evaluate new points, and send the information back to the master process. The archive of evaluated points, S_m, is subsequently updated by the master process along with the set of non-dominated points, P_m, and a new iteration of the improvement loop is initiated.

4.3.3 Selection of Center Points

The iterative framework (Step 2) of MOPLS incorporates simultaneous surrogate-assisted local searches around a subset of already evaluated points, S_m. This subset of points, denoted as S_{cen} (where $S_{cen} \in S_m$), is referred as the set of center points or the parent population in subsequent discussions. N_c is an algorithm parameter, which denotes the number of center points (or population size) chosen for parallel local search in each algorithm iteration. Algorithm Step 2.1 provides a detailed account of the procedure for selection of S_{cen}.

As mentioned in the previous sections, center points for local search are ranked, and subsequently selected, based on their convergence and diversity potential. Goldberg [13] suggested the use of *non-dominated ranking*. Deb et al. [8] built upon Goldberg's idea by introducing *non-dominated sorting* [9]. The *non-dominated sorting* algorithm computes the non-dominated solutions in the evaluation set (S_m), assigns them the highest rank, removes them from contention,

and assigns the next rank to the non-dominated solutions amongst the remaining evaluated solutions. This process is repeated to rank all solutions (See Figure 4.2(a) for visual illustration of non-dominated sorting).

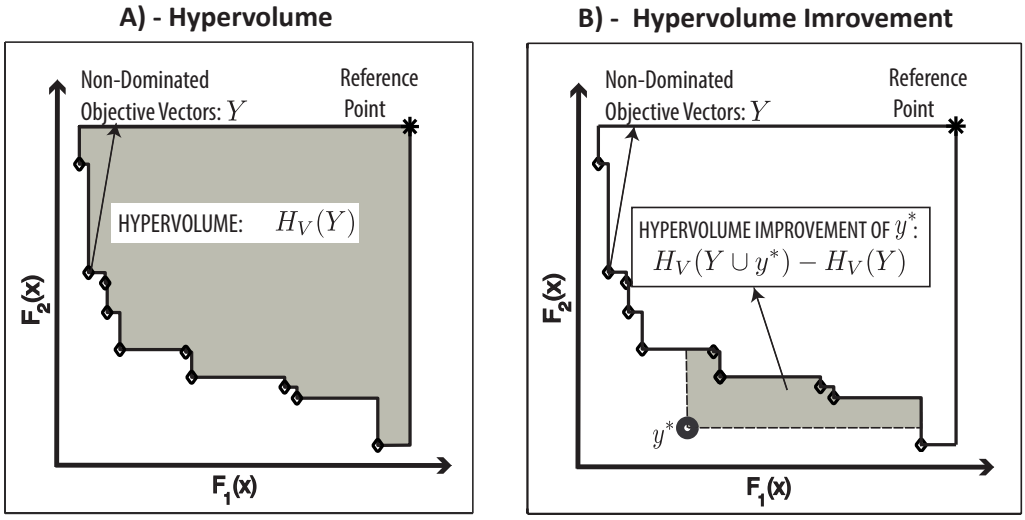

Figure 4.1: Visualization of a) Hypervolume and b) Hypervolume Improvement employed in Step 2.2 for selection of new evaluation points (discussed in Section 4.3.4).

While *non-dominated sorting* helps in ranking evaluated solutions according to their contribution towards convergence, it provides no information regarding a solution's contribution towards diversity. Furthermore, *non-dominated sorting* does not differentiate between solutions which have the same non-dominated ranking. Various metrics have been discussed in past literature that provide a measure of diversity contribution, and, in conjunction with non-dominated sorting, these metrics can provide a unique ranking of evaluated solutions according to convergence and diversity. Some of the metrics include crowding distance [8], niching [9], indicator based contribution, [5] etc. Hence, typically, search-based multi-objective optimization algorithms employ a two-layered strategy to rank evaluated solutions and subsequently guide the search process.

MOPLS also employs a two-layered strategy to rank evaluated solutions and subsequently choose *center points* (controlled by the master process) for simultaneous surrogate-assisted local search (performed by slave processes). While the first layer of ranking is based on *non-dominated sorting*, the second layer involves ranking all solutions within a front according to their hypervolume (also called S-Metric) contributions [23] [42].

Definition 4 *Let $X = \{x_1, \ldots, x_n\}$ be a set of ordered solutions and let $Y = \{F(x_1), \ldots, F(x_n)\}$ be the corresponding set of objective vectors. The hypervolume of the set X is the volume of region of the objective space dominated by Y. Hypervolume is denoted as $H_V(Y)$. The objective space is bounded by a reference vector, b (see Figure 4.1(a) for illustration).*

Definition 5 *Let $X = \{x_1, \ldots, x_n\}$ be a set of non-dominated solutions and let $Y = \{F(x_1), \ldots, F(x_n)\}$ be the corresponding set of objective vectors. The hypervolume contribution of a point $x_i \in X$ (denoted as HC_i) is the difference in hypervolume of Y and hypervolume of Y excluding the objective vector $F(x_i)$, i.e, $HC_i = H_V(Y) - H_V(Y \backslash F(x_i))$ (see Figure 4.2(b) for illustration).*

The detailed algorithmic framework of the center selection step , i.e, Step 2.1, (including inputs and outputs) of MOPLS is provided in Algorithm Step 2.1 above. As depicted in Algorithm Step 2.1, the center selection strategy employs two layers in order to rank quality of already evaluated solutions (S_m), 1) non-dominated sorting (line 6) and 2) hypervolume contributions-based ranking (lines 7 and 8) (illustrated in Figure 4.2).

One major advantage of ranking solutions (within a non-dominated front) via the hypervolume contribution metric lies in the ability of the metric to quantify both convergence and diversity of a particular solution. However, compu-

Algorithm Step 2.1: Center Selection

Input : 1) N_c - Number of centers per generation.
2) S_m - Set of evaluated points.
3) S_{taboo} - Set of *taboo* evaluated points. $S_{taboo} \subset S_m$.
4) d_{thresh} - Scalar value defined in Step 2.1 on page 131.

Output: 1) I_{cen} - Indices of the selected centers that correspond to the ordered archive of evaluated points, S_m.

```
1  begin
2      S = S_m;
3      S_cen = {};
4      while |S_cen| ≤ N_c do
5          X = {x_i | (x_i, y_i, r_i, c_i, c_i')∈S, 1≤i≤|S|} ;
6          S* = {(x_i, y_i, r_i, c_i, c_i')∈S | x_i is non-dominated in X};
7          HC = HypervolumeContributions(S*);
8          S* = SortFront(S*, HC);
9          for i = 1 to |S*| and (x_i, y_i, r_i, c_i, c_i') ∈ S* do
10             Let z_i = (x_i, y_i, r_i, c_i, c_i') ;
11             if z_i ∉ S_taboo then
12                 if |S_cen| == 0 then
13                     S_cen = {S_cen} ∪ {z_i};
14                 else if |S_cen| ≤ N_c then
15                     for j = 1 to |S_cen| and (x̂_j, ŷ_j, r̂_j, ĉ_j, ĉ_j') ∈ S_cen do
16                         if ||x̂_j − x_i|| ≤ r̂_j * d_thresh then
17                             go to line 9;
18                     S_cen = {S_cen} ∪ {z_i};
19         S = {S}\{S*};
20     I_cen = {i | a_i∈S_m, b_j∈S_cen, a_i = b_j, 1≤j≤N_c};

1  Procedure HypervolumeContributions(S*)
2      Y* = {y_i | (x_i, y_i, r_i, c_i, c_i')∈S*, 1≤i≤|S*|} ;
3      for i = 1 to |Y*| and y_i ∈ Y* do
4          HC_i = [H_V(Y*) − H_V({Y*}\{y_i})];
5      return HC;

1  Procedure SortFront(S*, HC)
2      Sort (descending order) the elements of S* according to the hypervolume
       contribution values recorded in HC;
3      return S*;
```

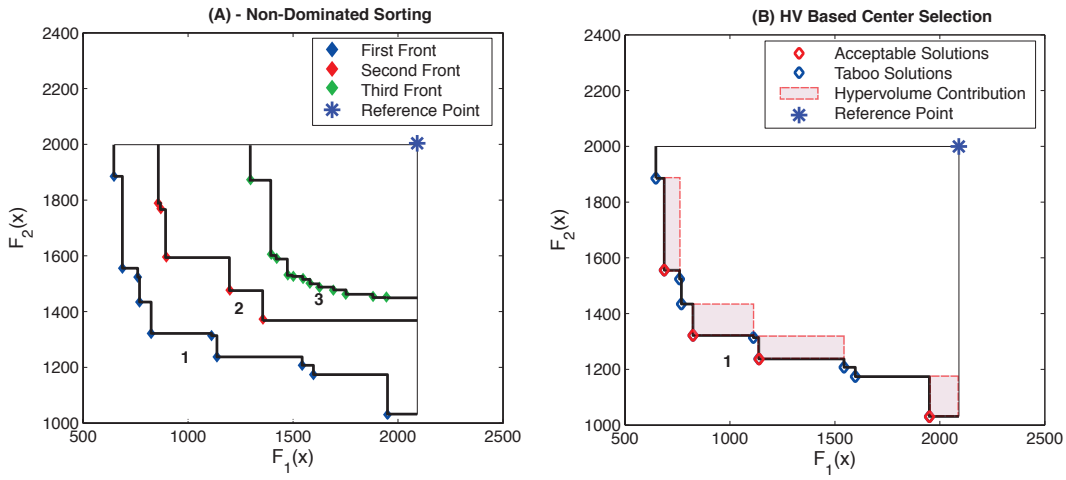

Figure 4.2: Illustration of how centers are chosen in Step 2.2 of MOPLS: a) The selection routine starts with Non-Dominated sorting of evaluated solutions, b) Centers from within a front are selected from the Non-Tabu set on the front, based on their hypervolume contributions to the front.

tation of hypervolume contribution is an NP-Hard problem [3], and the computational intensity increases exponentially with an increase in the number of objectives. Figure 4.2 provides a visual depiction of the two-layered center selection strategy. As can be seen, the hypervolume contribution metric is essentially an integral. To reduce computational effort, we employ a Monte Carlo experiment to efficiently calculate hypervolume contributions. (For further discussion on hypervolume computations, see [2]).

Selection of centers based purely on their ranking is a greedy approach. Such an approach could bias the search towards exploration and exploitation around good quality solutions only, and hence could lead to convergence to what is only a locally optimal front.

In order to avoid such a scenario, MOPLS maintains a list of tabu solutions (S_{taboo}), which cannot be selected as centers. The procedure for updating the

tabu list is discussed in Section 4.3.5 and Algorithm Step 2.3. Figure 4.2, depicts the tabu solutions in blue, and the center selection algorithmic framework incorporates the tabu list within the selection mechanism (see line 11 of Algorithm Step 2.1). In addition to maintenance of a tabu list, the center selection strategy also ensures that a point that is close to another selected center (in the current iteration) is not considered for selection (see lines 15-18 of Algorithm Step 2.1), since a local search will be performed around the other center anyway. The closeness is deduced by a threshold, d_{thresh} (where $d_{thresh} = 1 - \frac{m-E_I}{E_T-E_I}$), and the local search radius r_i. At the beginning of the algorithm the value of d_{thresh} is equal to 1. Hence, points which lie within the search neighborhoods of already selected center points, are not considered for selection as centers. However, this value is reduced to 0, as the algorithm progresses, in order to allow points which are close to each other in the decision space to be selected as center points simultaneously. This enables concentrated local search for improved diversification.

4.3.4 Parallel Surrogate-Assisted Local Candidate Search

Surrogate-assisted optimization has been extensively used in single objective simulation optimization problems. Jones et al. [17] proposed Efficient Global Optimization (EGO), employing kriging [32] as a surrogate model in order to guide the search process. Radial Basis Functions (RBF) [4] [15] [14] [29] [27] have also been widely used for efficient simulation optimization, within a limited evaluation budget.

Surrogate-assisted search methods have also been incorporated into multi-objective optimization (MO) [19] [23] [40] [18]. Some authors have also employed local search within surrogate-assisted MO algorithms. For in-

stance, Georgopoulou and Giannakoglou [11] use gradient based refinements of promising solutions highlighted by RBF approximation during each algorithm generation. Santana et. al. [33] divide the heuristic search into two phases where the first phase employs a global surrogate-assited evolutionary search, and the second phase employs rough set theory for local refinements. The local search mechanism employed in MOPLS is considerably different from the local search methodologies employed in previous surrogate-assisted algorithms. Surrogate-assisted local search is employed exclusively in MOPLS, within a synchronous parallel framework, to select multiple new points for expensive evaluation in each algorithm generation.

The surrogate-assisted local search mechanism employed in Step 2.2 of MOPLS is inspired from the works of Regis and Shoemaker [27] [30]. These works [27] [30] demonstrate the effectiveness of surrogate-assisted local search in efficient single objective optimization by employing a Radial Basis Function (RBF) based surrogate model within a multi start local search mechanism. Additionally, they also demonstrated that the methodology could be parallelized to further improve efficiency.

Step 2.2 of MOPLS also employs a parallel surrogate-assisted local search to choose new points for expensive evaluations. As is depicted in the general algorithm framework, the master process sends one point each from the set (or parent population) of center points to each slave process in Step 2.2. Each slave process selects a new point for expensive evaluation after performing a surrogate-assisted search in the local neighborhood of the center point (sent by the master process). The slave process subsequently evaluates the selected point via expensive evaluation and sends the evaluated point back to the master process. The procedure adopted by the slave process in selection and evaluation of

the new point is depicted in Algorithm Step 2.2 and discussed below.

Response Surface Model

The first major task of the slave process is approximation of the costly functions, F, via the inexpensive surrogates, $\widehat{F}(x) = [\widehat{f_1}, \ldots, \widehat{f_k}]$ (line 4 of Algorithm Step 2.2). Various approximation methods, including artificial neural networks [25], Support Vector Machines (SVM) [34], kriging [32], and radial basis functions (RBFs) [4][24] could be employed to fit the response surface model.

Various authors [31] [10] demonstrate the relative effectiveness of RBF approximation in tackling high dimensional problems (approximately defined as problems with more than 15 decision variables). The slave process of MOPLS hence employs RBFs as the surrogate modeling methodology for approximating the costly functions.

The training set (denoted as S) used for fitting the RBF model is a subset of expensively evaluated points (denoted as S_m). The training set, S, includes up to 500 points which are closest (as per euclidean distance) to the center point (denoted as x_{cen}) in the decision space. The size of the training set is kept limited to avoid excessive computational burden of fitting the RBF model. Hence, the surrogate model is referred as local surrogate model in subsequent discussions.

Generation of Random Candidate Points

Fitting of the local surrogate model is followed by random generation of numerous points around the *center point* (see line 5 of Algorithm Step 2.2). These randomly generated points are referred as *Candidate points* in subsequent discussions. Regis and Shoemaker [27] introduced the idea of generating candidate points via a Gaussian perturbation of the center point, i.e., $x_{cand,i} \sim \mathcal{N}(x_{cen}, \sigma^2 I_d)$,

Algorithm Step 2.2: Surrogate-Assisted Candidate Search

Input : 1) S_m - Set of evaluated points.
2) $z_{cen} \in S_m$ - Evaluated point selected for local search in Step 2.1.
Please note that $z_{cen} = (x_{cen}, y_{cen}, r_{cen}, c_{cen}, c'_{cen})$
3) P_m - Set of *non-dominated* evaluated points. $P_m \subset S_m$.

Output: $(x^*, y^*, r^*, c^*, c^{*'})$ - Point selected for evaluation and evaluated via expensive simulation.

begin
  p_1 = generate a uniformly distributed real number in [0, 1];
  if $p_1 \leq prob_{cand}$ **then** perform surrogate-assisted search
    $\widehat{F}$ = `FitRbfModel(`x_{cen}, S_m`)`;
    X_{cand} = `GenerateCandidates(`x_{cen}, r_{cen}`)`;
    $X^*_{cand} = \{x \in X_{cand} \mid x$ is non-dominated in X_{cand} as per the approximate function $\widehat{F}\}$;
    p_2 = generate a uniformly distributed real number in [0, 1];
    if $p_2 \leq prob_{hv}$ **then**
      x^* = `BestHypervolume(`X^*_{cand}, P_m`)`;
    else
      x^* = `MaxMinMethod(`X^*_{cand}, S_m`)`;
  else mutate center
    x^* = `mutate(`x_{cen}`)` ;
  $y^* = F(x^*)$, $r^* = r_{init}$, $c^* = 0$, $c^{*'} = 0$;

Procedure `FitRbfModel(`x_{cen}, S_m`)`
  Choose a subset, S, from already evaluated solutions, S_m, which are closest to x_{cen}, as per euclidean distance ;
  Fit response surface models (RSM) for each objective based on a set S. Let $\widehat{F}(x) = [\widehat{f_1}, \ldots, \widehat{f_k}]$ denote the inexpensive RSMs ;
  return $\widehat{F}$;

Procedure `GenerateCandidates(`x_{cen}, r_{cen}`)`
  $X_{cand} = \{\}$, p = generate a uniformly distributed real number in [0, 1];
  if $p \leq \frac{1}{2}$ **then** */* perform hyper-spherical local search */*
    $\sigma = r_{cen}$;
  else /* perform hyper-eliptical local search */
    $\sigma = |a|$, where $a \sim \mathcal{N}(\sigma, \sigma^2/4)$ and $\mathcal{N}$ denotes the normal distribution ;
  $N_{cand} = 500 * d$, where d is the number of decision variables;
  for $i = 1$ *to* N_{cand} **do** generate candidate points
    $x_{cand,i} = x_{cen} + z$ where z is a random vector s.t $z \sim \mathcal{N}(0, \sigma^2 I_d)$;
    $X_{cand} = \{X_{cand}\} \cup \{x_{cand,i}\}$;
  return X_{cand} ;

Procedure `BestHypervolume(`X_{cand}, P_m`)`
  $Y_m = \{y_i \mid (x_i, y_i, r_i, c_i, c'_i) \in P_m, 1 \leq i \leq |P_m|\}$;
  $x^* = \arg\max_{x_j \in X^*_{cand}} \left[H_V(Y_m \cup \widehat{F}(x_j)) - H_V(Y_m) \right]$;
  return x^*;

Procedure `MaxMinMethod(`X^*_{cand}, S_m`)`
  $X_m = \{x_i \mid (x_i, y_i, r_i, c_i, c'_i) \in S_m, 1 \leq i \leq |S_m|\}$;
  $x^* = \arg\max_{x_i \in X^*_{cand}} \left[\min_{x_j \in X_m} \|x_i - x_j\| \right]$;
  return x^*;

Procedure `mutate(`x_{cen}`)`
  Generate a point $x^* \in \mathcal{D}$ via either *gaussian mutation* or *uniform mutation* (with equal probability) of x_{cen};
  return x^*;

where $x_{cand,i}$ is a candidate point. If a randomly generated candidate point is out of the unit hypercube bounds, it is forced to a corner point. Hence, candidate points are concentrated in a hypersphere around the center point (as depicted in Figure 4.3(A)).

MOPLS employs a similar candidate generation methodology. Please note that the standard deviation of the gaussian perturbation, σ, is equal to the memory variable r_{cen}. The memory variable, r_{cen}, hence corresponds to the local search neighborhood of the center point, x_{cen}. Every evaluated point x_i in the set S_m includes an associated memory variable r_i, which denotes the standard deviation of the gaussian perturbation for generating candidate points in the neighborhood of x_i. r_i is a dynamic variable which is initialized to be equal to r_{init}, and is updated in Step 2.3 of MOPLS as per the procedure described in Algorithm Step 2.3 and Section 4.3.5.

We also propose a variation of the original candidate generation methodology in MOPLS, i.e, $x_{cand,i} \sim \mathcal{N}(x_{cen}, |\mathcal{N}(\sigma, \sigma/2)|)$. With this variation, candidates are concentrated in a hyperellipse and search concentration varies across the decisions. Such a search methodology might be particularly useful in the presence of varying sensitivities of decision variables. The difference between the two search strategies is depicted in Figure 4.3. In each iteration of MOPLS, each slave chooses one of the two candidate generation methodologies with equal probability.

The number of candidate points generated by the slave process is equal to $500 * d$, where d denotes the number of decision variables. After generation of the candidate points the local surrogate model is used to approximate their objective functions. The *candidate points* approximated by the surrogate model in the vicinity of the *center point* which are non-dominated (as per the surrogate

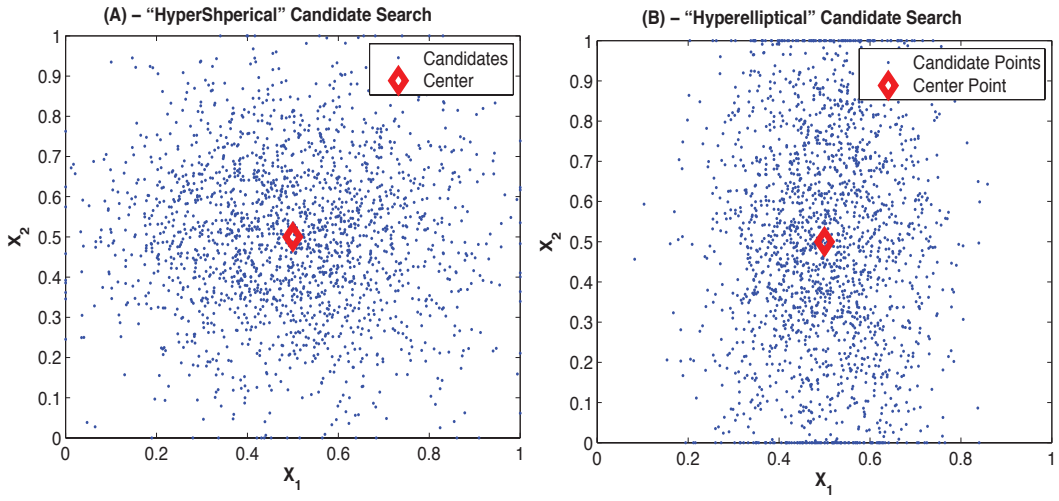

Figure 4.3: An example illustrating the local candidate points generation methodology proposed by [27]: a) The "Hyperspherical" search mechanism originally proposed by [27], b) The "Hyperelliptical" search mechanism proposed in this study. The algorithm generates candidates from either mechanism with equal probability

approximation), are identified (see line 6 of Algorithm Step 2.2), and a new point is subsequently chosen for evaluation.

Selection of Expensive Evaluation Point

Let X^*_{cand} denote the set of non-dominated candidate points (as per the surrogate approximation). A new point for expensive evaluation is selected from the set X^*_{cand}. Selection of a new point for actual evaluation is based on one of two strategies, namely, hypervolume improvement and Max-Min method (see Lines 7-11 of pseudo-code of Step 2.2).

The hypervolume improvement selection method computes the approximate hypervolume improvement (we use the word approximate since objective function values are approximated by the surrogate model) of every RBF

approximated non-dominated candidate solution, and chooses the one with the best hypervolume improvement. Pseudo code for this method is depicted in the procedure "BestHypervolume" of Algorithm Step 2.2 and is illustrated in Figure 4.1(b). Hypervolume improvement is calculated via a Monte Carlo experiment, in order to reduce computational burden.The approximate hypervolume improvement can quantify the potential value added by a candidate in terms of both convergence and diversity, and hence, selection based on approximate hypervolume is purely based on exploitation of the surrogate model within the local search. Hypervolume improvement based selection is used in various other surrogate and non-surrogate optimization algorithms [2].

The Max-Min selection methodology is depicted in the procedure "MaxMinMethod" of Algorithm Step 2.2. As per the max-min method, an RBF-approximated non-dominated candidate point (with an RBF-approximated objective function) which is furthest from already evaluated points is selected for function evaluation. While the candidate is chosen from amongst the approximated non-dominated set, i.e, X^*_{cand} (hinting at exploitation), it is relatively furthest from the already evaluated points (hinting at exploration). Hence, the max-min selection method incorporates both exploration and exploitation. The probability of hypervolume improvement based candidate selection (denoted as $prob_{hv}$ in Algorithm Step 2.2) is set to 0.65, for all experiments.

Mutation

The methodology for surrogate-assisted local search is depicted in lines 3-11 of Algorithm Step 2.2. As is evident from the pseudo code depicted in Algorithm Step 2.2, selection of a new point for expensive evaluation is not necessarily based on surrogate-assisted local search. The slave process employs surrogate-

assisted local search with a probability, $prob_{cand}$,where the value of $prob_{cand}$ is set as 0.9 for all computational experiments. The slave process employs mutation as the alternative to surrogate-assisted local search, for selecting a new point for expensive evaluation.

The use of mutation is a critical feature in evolutionary algorithms, ensuring that the search remains global and has the potential to move to any point in the decision space. Mutation is employed in MOPLS (see line 13 of Algorithm Step 2.2) to ensure that the search mechanics remains global. We employ a Gaussian, and a uniform mutation within the selection methodology.

After the slave process selects a new point for expensive evaluation (via either surrogate-assisted local search or mutation), the slave expensively evaluates the selected point (see line 14 of Algorithm Step 2.2). Furthermore, the memory attributes of the newly evaluated point are also initiated (see line 14 of Algorithm Step 2.2) and the newly evaluated point is sent to master process.

4.3.5 Tabu List and Memory Archive

In Section 4.3.1, we defined the ordered set of evaluated points as $S_m = \{z_i \mid z_i = (x_i, y_i, r_i, c_i, c_i'), 1 \leq i \leq m\}$, where z_i is a multi-attribute element of S_m, x_i is the ith decision variable, y_i is the objective vector corresponding to x_i, and r_i, c_i and c_i' are memory attributes corresponding to x_i. r_i, c_i and c_i' (for $1 \leq i \leq m$) are collectively referred as the memory archive of the evaluated set, S_m. The memory attribute r_i is generically referred as the "local search radius" and corresponds to the standard deviation of gaussian perturbations that generate the candidate points of Step 2.2 (discussed previously in Section 4.3.4). The memory attribute c_i is generically referred as the "failure count", and is discussed further later in this section. The memory attribute c_i' is generically referred as the "tabu count",

and is also discussed further later in this section. The Step 2.3 of the general algorithm framework of MOPLS updates the memory archive, after new points have been selected and evaluated via the parallel local search of Step 2.2. The procedure for updating the memory archive is depicted in Algorithm Step 2.3.

Algorithm Step 2.3: Update Memory Archive

Input : 1) S_m - Ordered set of points evaluated before current algorithm generation.
2) I_{cen} - Indices of the *centers* chosen for parallel local search that correspond to the ordered archive of evaluated points, S_m.
3) S_{curr} - New points selected (via parallel local search around *centers*) and evaluated in Algorithm Step 2.2.
4) S_{taboo} - Set of *taboo* evaluated points. $S_{taboo} \subset S_m$.
5) P_m - Subset of *non-dominated* in S_m.

Output: 1) S_m - Ordered set of points evaluated before current algorithm generation, with updated memory attributes.
2) S_{taboo} - Updated list / set of *taboo* evaluated points. $S_{taboo} \subset S_m$.

1 **begin**
2 **for** $k = 1$ *to* $|S_{curr}|$ **and** $(x_k^*, y_k^*, r_k^*, c_k^*, c_k'^*) \in S_{curr}$ **and** $j_k \in I_{cen}$ **do**
3 $Y_m = \{y_i \mid (x_i, y_i, r_i, c_i, c_i^{'}) \in P_m, 1 \leq i \leq |P_m|\}$;
4 $HI_k = \left[H_V(Y_m \cup y_k^*) - H_V(Y_m) \right]$;
5 **if** $HI_k == 0$ **then**
6 $i = j_k$;
7 $z_i = (x_i, y_i, r_i, c_i, c_i^{'}) \in S_m$, i.e, x_i is the center point (step 2.1) around which local search (Step 2.2) was performed to obtain the new evaluated point x_k^* ;
8 $r_i = \frac{r_i}{2}$;
9 $c_i = c_i + 1$;
10 **for** $i = 1$ *to* $|S_m|$ **and** $(x_i, y_i, r_i, c_i, c_i^{'}) \in S_m$ **do**
11 Let $z_i = (x_i, y_i, r_i, c_i, c_i^{'})$;
12 **if** $c_i^{'} > 0$ **then**
13 $c_i^{'} = c_i^{'} - 1$;
14 **if** $c_i^{'} == 0$ **then**
15 $S_{taboo} = \{S_{taboo}\} \backslash \{z_i\}$;
16 **else if** $c_i > c_{thresh}$ **then**
17 $c_i^{'} = c_{tenure}$;
18 $r_i = r_{init}$;
19 $c_i = 0$;
20 $S_{taboo} = \{S_{taboo}\} \cup \{z_i\}$;

The process of updating the memory archive is divided into two phases. In the first phase (depicted by lines 2-9 of Algorithm Step 2.3) we traverse through the list of center points that were chosen for local search in Step 2.1 of current MOPLS iteration, and assess the performance of local search around each center

point. Performance of a *center point* is assessed by computing the hypervolume improvement registered by the new point generated from local search around the center point (see line 4 of Algorithm Step 2.3). If a hypervolume improvement is not registered, the "local search radius" (r_i) is reduced by half, and the "failure count" (c_i) is incremented by one. Hence, it can be intuitively said that the "failure count", c_i, archives the number of times a local search around x_i did not contribute to an improvement in the non-dominated solution set. When the "failure count" of a point exceeds the number, c_{thresh}, the center is added to the Tabu list, S_{taboo} (discussed further in the following paragraph).

As mentioned earlier, the memory attribute c_i' is referred as the "tabu count" and corresponds to the number of algorithm iterations remaining for which a point x_i will remain in the tabu list, S_{taboo}. A non-zero value of "tabu count", thus, implicitly means that the point is in the taboo list. In the second phase of the memory archive update procedure (depicted in lines 10-20 of Algorithm Step 2.3) we traverse through all evaluated points (not including the points evaluated in the current algorithm generation) to 1) update the tabu list, and 2) update the tabu count. For all points which are in the tabu list the "tabu count" is reduced by one. If the "tabu count" for a point in the tabu list is reduced to zero, the point is removed from the tabu list. For points which are not in the tabu list, we check if the "failure count" exceeds the number, c_{thresh}. If this is the case, the point is added to the tabu list (S_{taboo}), the "tabu count" is set equal to the number, c_{tenure}, the "failure count" is reset to zero, and the "local search radius" is set equal to the initial search radius, r_{init} (Please note that r_{init} is an input parameter for MOPLS). It can be deduced from the above discussion that c_{tenure} is the tabu tenure for a point that is added to the tabu list.

The self-adaptive failure count method (inspired by Regis and Shoemaker

[27]) ensures that the algorithm does not get stuck in searching around locally optimal solutions, and tends to promote *exploration* within the search mechanics. Since our parallel search methodology revolves around effective selection of *centers*, exploration within center selection could be vital to maintain a global search. However, we would not like to discard a center forever, hence. after a fixed number of algorithm iterations, c_{tenure}, a tabooed center is removed from the *tabu* list. In all our experiments, c_{thresh} and c_{tenure} are set to 3 and 5, respectively.

Multi-objective Parallel Local Surrogate-Assisted Search (MOPLS) is a stochastic metaheurisitc algorithm aimed at efficient optimization of computationally expensive multi-objective optimization problems. The algorithm aims to maintain an efficient and dynamic search mechanism by 1) exploiting local surrogate models for efficient convergence, 2) maintaining exploration through mutation and 3) maintaining a balance between exploration, exploitation, convergence and diversity through Tabu assisted selection of simultaneous local search *centers*.

4.4 Computational Experiments

Assessing the performance of a multi-objective optimization algorithm is considerably more challenging than assessing the performance of a single-objective optimization algorithm both because there are more factors to evaluate for a MO problem and because each trial is more computationally expensive. Coello et al. [5] recommend that the quality of a multi-objective optimization algorithm be assessed via two major factors: efficiency and effectiveness. Efficiency signifies the speed with which the algorithm obtains good quality solutions, and effectiveness measures the quality of solutions, addressed in terms of both

convergence and diversity (See Figure 3.1 for an illustration of convergence and diversity). Since MOPLS is a stochastic algorithm, it is important that the reliability of the algorithm is also assessed, i.e., the algorithm's ability to consistently produce good solutions.

Comparison against some benchmark algorithms and application to various test problems can help in assessing the strengths and weaknesses of an algorithm. In the following section, we summarize the experimental setup employed to assess the performance of MOPLS, which includes a brief overview of benchmark algorithms, test problems, and the methodology employed to assess effectiveness and efficiency of MOPLS when applied to the various test problems and compared against benchmark algorithms.

4.4.1 Alternate Algorithms

The performance of MOPLS is compared to GOMORS, ParEGO [19] and AMALGAM [37]. GOMORS is our own algorithm, introduced in [1] and Chapter 2 of this thesis. ParEGO [19] is a kriging based efficient multi-objective optimization algorithm, which is discussed in Chapters 2 and 3. AMALGAM is a multi-method evolutionary algorithm, which employs various search mechanics for efficient evolutionary optimization of multi-objective problems. Our analysis in Chapter 3 clearly indicated that the performance of GOMORS, ParEGO and AMALGAM is superior to other benchmark MOEAs, namely NSGA-II and MOEA/D, when evaluation budget is limited to less than 1000. Since in this analysis we are interested in comparing algorithm performance within a limited evaluation budget (1000), we consider comparison against GOMORS, ParEGO and AMALGAM only.

Three different versions of MOPLS are employed in the analysis; MOPLS-

4, MOPLS-8 and MOPLS-16, where the N in MOPLS-N refers to the number of simultaneous centers. Hence, MOPLS-16 employs 16 simultaneous centers (i.e, 16 synchronous parallel processes within each algorithm iteration), and local surrogate-assisted search is performed around each center. Please note that theoretically, a synchronous master-slave parallel framework could allow N simultaneous local searches to be executed (if N cores are available).

4.4.2 Test Problems

We have chosen five unconstrained (or box constrained) test problems for algorithm performance analysis. All of the test problems chosen pose different challenges to the search optimization process. Hence, if performance of an algorithm is consistently good across all problems, one can be more confident of the algorithm's capabilities.

The first two test problems, ZDT3 and ZDT6 are part of the Deb et. al's ZDT test suite [7]. ZDT3 has a disconnected Pareto front, and ZDT6 has a low density of solutions, around the Pareto front. Additionally, the distribution of solutions on the ZDT6 Pareto front is non-uniform, and the Pareto front is non-convex.

The next three test problems used in our study (UF1, UF2 and UF3) were proposed by Li et al. [20], and highlighted by [41], as potentially hard multi-objective optimization problems, where the Pareto-optimal sets are complicated and difficult to find. Additionally, it should be noted that UF3 has many locally optimal fronts.

All test problems have two objectives and the number of decision variables can vary between 2 and 30. Mathematical formulations and optimization challenges of the test problems are discussed in detail in Section B.2.1 of Appendix B. We test and compare performance of MOPLS on the test problems with 8 and

16 decision variables. This is done to assess the differences in performance of MOPLS with other algorithms on low (less than 10 decision variables) and high (greater than 10 decision variables) dimensional problems. In order to differentiate between the same test problem's 8 and 16 variable versions, we use the convention 'NAME-D' to refer to the test problem, where 'D' corresponds to the number of decision variables of the test problem. Consequently, the two variants of, for instance, ZDT3 are referred as ZDT3-8 and ZDT3-16 in subsequent discussions.

4.4.3 Watershed Model Calibration Test Suite

In Chapter 3, we discussed the development of a Watershed Model Calibration test suite, comprising of two bi-objective optimization problems, and two 3-objective optimization calibration problems. These test problems were part of two watershed model case studies, Town Brook, and Cannonsville. In an effort to analyze the performance of MOPLS on real world problems, we employ the watershed model calibration test suite for comparative analysis against benchmark algorithms. (For further discussion on the test suite, refer to Chapter 3 of the thesis.)

4.4.4 Comparison Methodology

We have defined a set of benchmark algorithms for comparison against MOPLS, and a set of test problems on which the algorithm performance will be assessed. As part of this comparison methodology, it is important to incorporate comprehensive testing of algorithm effectiveness and efficiency.

Table 4.3: The flow calibration test case suite employed in comparative algorithm analysis

Problem Name	Formulation	Case Study	Decisions Variables	Objectives	Simulation Time/Eval(s)
TBROOK1	Threshold	II	15	2	10
TBROOK2	NSE	II	15	3	10
CVILLE1	Threshold	I	15	2	150
CVILLE2	NSE	I	15	3	150

Experimental Setup

Since all algorithms are stochastic, we have performed multiple trial runs of each algorithm on all test problems and the watershed calibration suite. 10 trials each were performed for all algorithms on the test problems and the watershed test suite. Algorithm performance comparison is limited to a maximum of 400 function evaluations per algorithm for the test problems and 1000 function evaluations per algorithm for the watershed calibration problems. For the test problems, performance of MOPLS is compared against GOMORS and ParEGO only, whereas, for the watershed calibration problems, performance of MOPLS is compared against GOMORS, ParEGO and AMALGAM. Since evaluation budget is limited to 400 function evaluations for the test problems and AMALGAM is a non-surrogate based algorithm which is not designed for a small evaluation budget, we do not include AMALGAM in the comparison of test problems.

Table 4.4: Parameter Settings for MOPLS

Name	Description	Setting
E_I	The number of initial expensive evaluations	$2d + 2$
E_T	The number of total expensive evaluations	400 or 1000
N_c	The number of *center points*	4, 8 or 16
r_{init}	Candidate / local search parameter /radius	0.2

Algorithm Parameters

Our algorithm comparison methodology incorporates the use of suitable values for parameters of all algorithms under discussion with the aim of conducting a fair comparison. ParEGO's parameter configuration recommended in Chapter 2 and Section B.5.2 is employed, and the GOMORS configuration recommended in Chapter 2 and Section B.5.3 is used. A small trial-and-error exercise was performed to tune population sizes for AMALGAM. Since performance of MOEAs is highly dependent on population sizes, we ran multiple trials of AMALGAM on the watershed calibration test suite, with population sizes of 20, 50, 100 and 200, and an evaluation limit of 1000. The trial-and-error analysis showed that within the limited evaluation budget of 1000, a population size of 20 is desirable in the case of the bi-objective watershed problems (TBROOK1 and CVILLE1), and a population size of 50 is desirable in the case of the 3-objective watershed problems (TBROOK2 and CVILLE2). The remaining parameters of AMALGAM were set as per the recommendations provided in [36].

As mentioned in Table 4.2, there are four parameters in the MOPLS algorithm. MOPLS' parameter configuration is provided in Table 4.4. The number of initial function evaluations is fixed at $(2d + 2)$ as per the recommendations provided by Regis and Shoemaker [28], where d is the number of decision vari-

ables. This is equal to the number of initial function evaluations of GOMORS and ParEGO. Since MOPLS, GOMORS and ParEGO use an experimental design for sampling of initial points, the fairest way to compare them is to have the same number of initial sampling points for all of them (for a particular problem). Hence, the same number of samples are used for MOPLS, GOMORS and ParEGO on each problem tested.

In case of the five test problems, function evaluations for MOPLS (and all other algorithms) are limited to 400, while for the watershed calibration test suite, the function evaluations are limited to 1000. As mentioned earlier, three different versions of MOPLS are analyzed, i.e, with 4, 8 and 16 center. Value of the candidate search radius / parameter, r_{init}, for MOPLS is set to 0.2 for all test instances. This value of the search radius is recommended by Regis and Shoemaker [28].

Visual Comparison of Trade-offs

The comparison methodology includes a visual comparison of non-dominated solutions obtained via each algorithm. This is possible since our analysis is limited to 2-objective and 3-objective test instances. For the 3-objective test instances non-dominated solutions are compared in two dimensions and results are displayed in three sub-figures, i.e, trade-offs between two objectives are compared in one sub-figure. The visual comparisons are performed after a fixed number of function evaluations and are supported by box plots (for watershed problems only) for performance metrics (metrics used in our analysis are discussed below). This comparison helps in visualizing the effectiveness of all algorithms in producing good solutions within a prescribed evaluation budget.

Performance Metrics

The performance metrics employed in the analysis include 1) the uncovered hypervolume metric and 2) the Inverses Generational Distance metric. Figure 4.4 gives a visual illustration of the meaning of the performance metrics. Let P (in Figure 4.4) be the set of non dominated solutions obtained as an approximation to the Pareto front from an algorithm and let P^* be the set of ideal solution(s) of the multi-objective optimization problem being solved. The "ideal solution(s)" of a multi-objective optimization problem is the Pareto front, if known (the Pareto front is known for the test problems), or the vector depicting minimum attainable values of all objectives. In our watershed calibration problems, all objectives are measures of error. Hence, the "ideal solution(s)" for all the watershed calibration problems is the zero vector. The "reference point" depicted in Figure 4.4 is the worst attainable solution of the optimization problem. Since, the worst attainable values are not known in the problems discussed in this study, the "reference point" vector is estimated as the worst values of all objectives obtained in our computer experiments. Hence, the total feasible objective space is the volume bounded by the reference point and ideal solution(s).

The Uncovered hypervolume [2] is the difference between the total feasible objective space (defined by the reference and ideal solution(s) in Figure 4.4(a)) and the objective space dominated by estimate of the Pareto front (depicted as P in Figure 4.4) obtained by an algorithm. A lower value of the index indicates a better solution and the ideal value is zero. The uncovered hypervolume metric incorporates both convergence to the ideal solution(s) as well as diversity of solutions within a single metric.

The The Inverse Generational Distance (IGD) [35] metric (defined by the equation depicted in Figure 4.4(b)) is the minimum distance of estimate of the

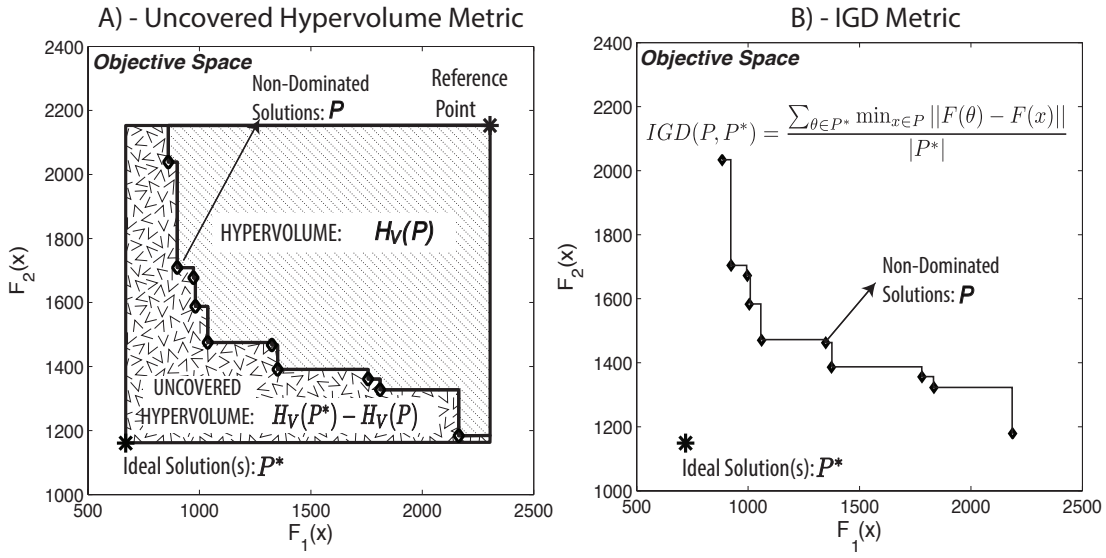

Figure 4.4: Illustration of performance metrics used in analysis: a) The Hypervolume provides a comprehensive quantification of algorithms' convergence and diversification abilities, b)The Inverse Generational Distance (IGD) (from ideal point) is a good measure of convergence. c) The Generational Distance (GD) also predominantly captures convergence

Pareto front obtained by an algorithm, P, from the "Ideal Solution(s)", P^*. The IGD metric is a good measure of convergence of an algorithm to the "Ideal Solution(s)".

Progress Plots

The performance metrics incorporated in our analysis are also used to analyze the relative efficiency of MOPLS. This is done through *progress plots* of the mean metric values plotted against the number of function evaluations. Since the focus of our analysis is on optimization of problems with computationally expensive objectives, the evaluation time is dominated by the number of function evaluations. Hence, progress plots analyze the relative efficiency of the algorithms by plotting the average metric-based performance of each algorithm

against the number of function evaluations. With progress plots one can easily see the impact total evaluations has on the quality of the convergence and how well for a fixed limit on number of evaluations, each algorithm performs.

Wall Clock Time

The comparison methodology depicted above focuses on comparing algorithm performance in function evaluations. However, an advantage inherent in MOPLS is the added efficiency that might be obtained via parallelization. Hence, we also use wall clock time to effectively compare performance of algorithms in a parallel setting. Our definition of wall clock time is defined in the following paragraph. Please note that this definition is valid with the assumption that function evaluations are computationally expensive.

Lets assume that that computation time for each function evaluation is considerably high such that other computational overheads arising, for example, from updating the response surface, are negligible. Furthermore, we also assume that N, i.e, the number of expensive function evaluations being evaluated in one algorithm iteration can be evaluated simultaneously. This means that if N_{core} is the number of cores available for parallel computing, $N_{core} \geq N$. Given these assumptions, one unit of wall clock time is equivalent to one algorithm iteration.

The algorithm comparison in wall clock time is done for MOPLS, GOMORS and ParEGO and for all problems tested. Since the initial number of function evaluations for these algorithms is equal, the time required for initial sampling and evaluation is not included in the wall clock time budget. We set the maximum wall-clock time to be 60 units. The total number of function evaluations will consequently be equal to $(2d + 2) + 60 * N$, where d is number of decision

variables of the problem, $(2d + 2)$ is the number of initial function evaluations and N is the number of expensive function evaluations being evaluated in one algorithm iteration.

ParEGO is a serial algorithm. Hence, $N = 1$ for ParEGO. GOMORS is a parallel algorithm where four expensive evaluations are performed in each algorithm iteration. Hence, $N = 4$ for GOMORS. In case of MOPLS-4, MOPLS-8 and MOPLS-16, $N = 4$, 8 and 16, respectively. Our wall clock time analysis for all problems includes visual comparison of non-dominated solutions obtained via each algorithm after 60 wall clock time units, and *progress plots* of the mean uncovered hypervolume metric values plotted against the number of wall clock time units.

4.5 Results and Discussion

4.5.1 Test Problems

Analysis in Function Evaluations

The algorithms were first tested on the five test problems. Each test problem has two objectives and poses a different optimization challenge (discussed in Section 4.4.2). The number of decision variables for each test problem is set to 8 or 16. Hence, comparison was performed on two instances of each test problem. In order to account for the stochastic search nature of all algorithms, 10 trials were performed for each algorithm (on each problem instance), with a budget of 400 objective function evaluations per trial.

Figures 4.5 - 4.9 provide summaries of performance of all algorithms, for 10 trial runs of 400 evaluations each. Each figure depicts results obtained for

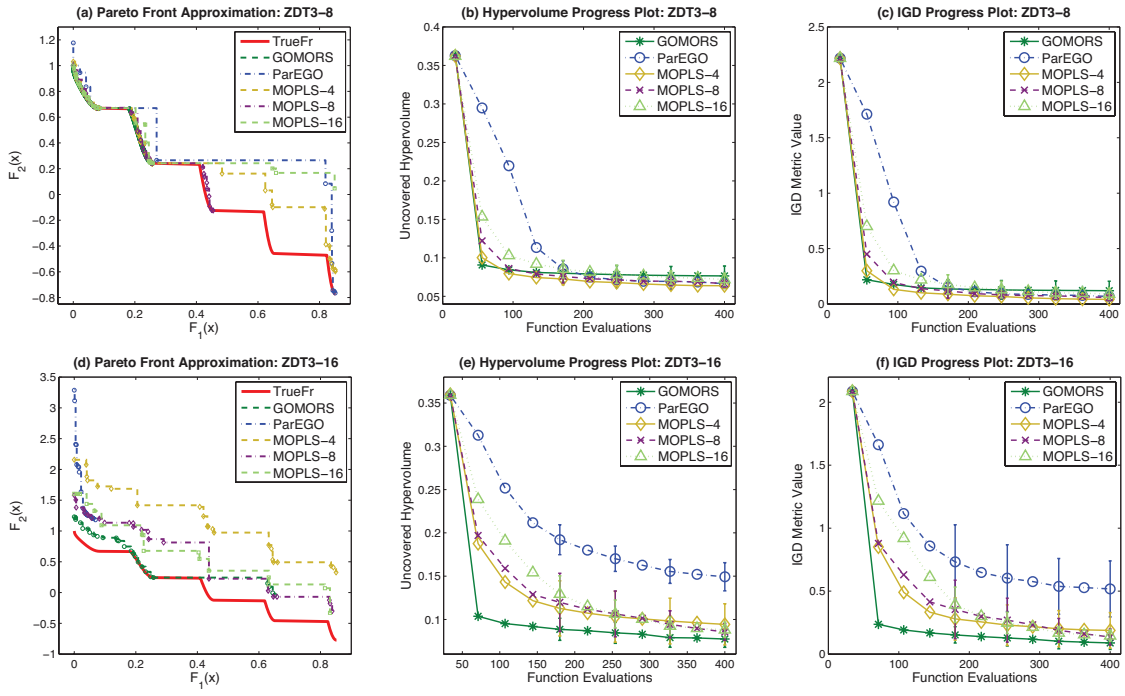

Figure 4.5: ***Test Problems ZDT3-8 and ZDT3-16***: Comparison of algorithm results after 400 function Evaluations: (a and d) Visualization of non-dominated solutions with worst hypervolume metric value in 10 trials and after 400 function evaluations, (b and e) Hypervolume progress plots, (c and f) Progress plots with mean IGD values. Vertical error bars indicate standard deviation.

the two variants of each test problem (i.e, with 8 and 16 decision variables). The summarized analysis of Figures 4.5 - 4.9 includes 1) visualization of worst trade-offs (according to hypervolume metric) obtained in 10 trials by each algorithm after 400 evaluations (see sub-figures (a) and (d)), 2) progress plot of mean hypervolume metric value for 10 trials (see sub-figures (b) and (e)), and 3) progress plot of mean IGD metric value for 10 trials (see sub-figures (c) and (f)). The trade-off visualization gives a visual depiction of convergence and diversity of all algorithms, and the progress plots depict relative average efficiency of all algorithms. The solid red lines in all Pareto front approximation plots depict

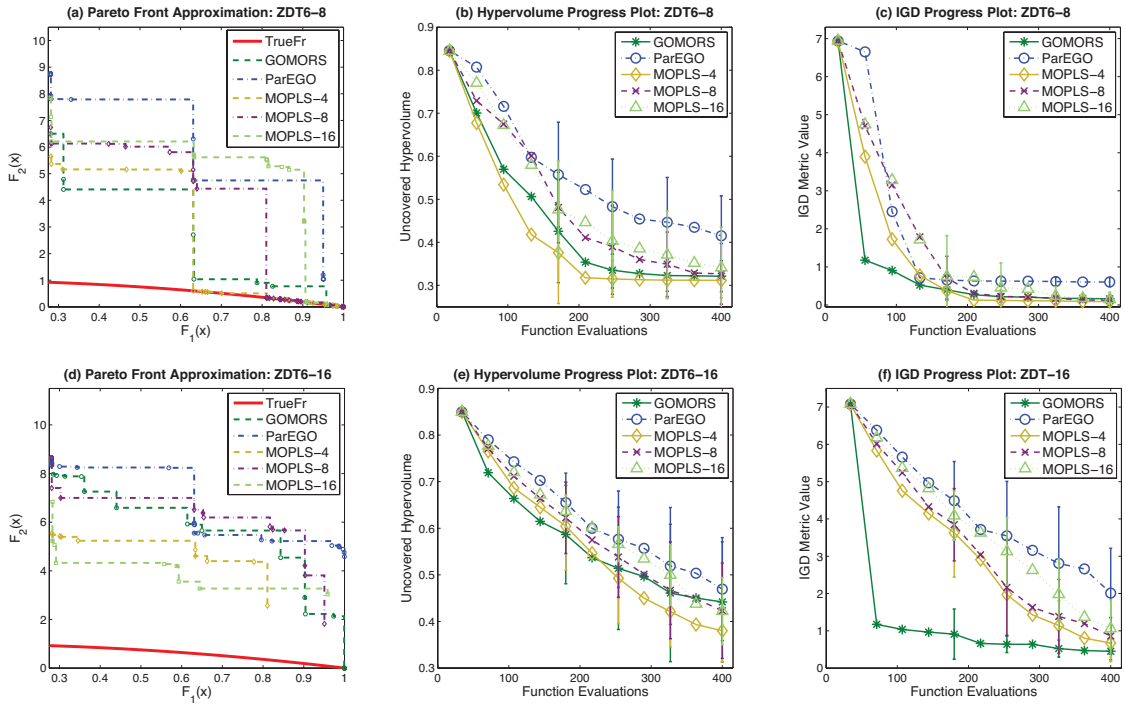

Figure 4.6: ***Test Problems ZDT6-8 and ZDT6-16***: Comparison of algorithm results after 400 function Evaluations: (a and d) Visualization of non-dominated solutions with worst hypervolume metric value in 10 trials and after 400 function evaluations, (b and e) Hypervolume progress plots, (c and f) Progress plots with mean IGD values. Vertical error bars indicate standard deviation.

the known Pareto front.

Figure 4.5 provides a summary of the performance of all algorithms after 400 function evaluations on the ZDT3-8 and ZDT3-16 test problems (8 and 16 signify number of decision variables). The optimization challenge of ZDT3 is that the known Pareto front (depicted in red) is disconnected. In case of the ZDT3-8 test problem, visualizations of the worst trade-offs obtained by each algorithm after 400 evaluations, and the metric progress plots indicate that all algorithms converge to the Pareto front. However, GOMORS and MOPLS-4 seem to perform better in terms of speed of convergence as per the progress plots (Figures

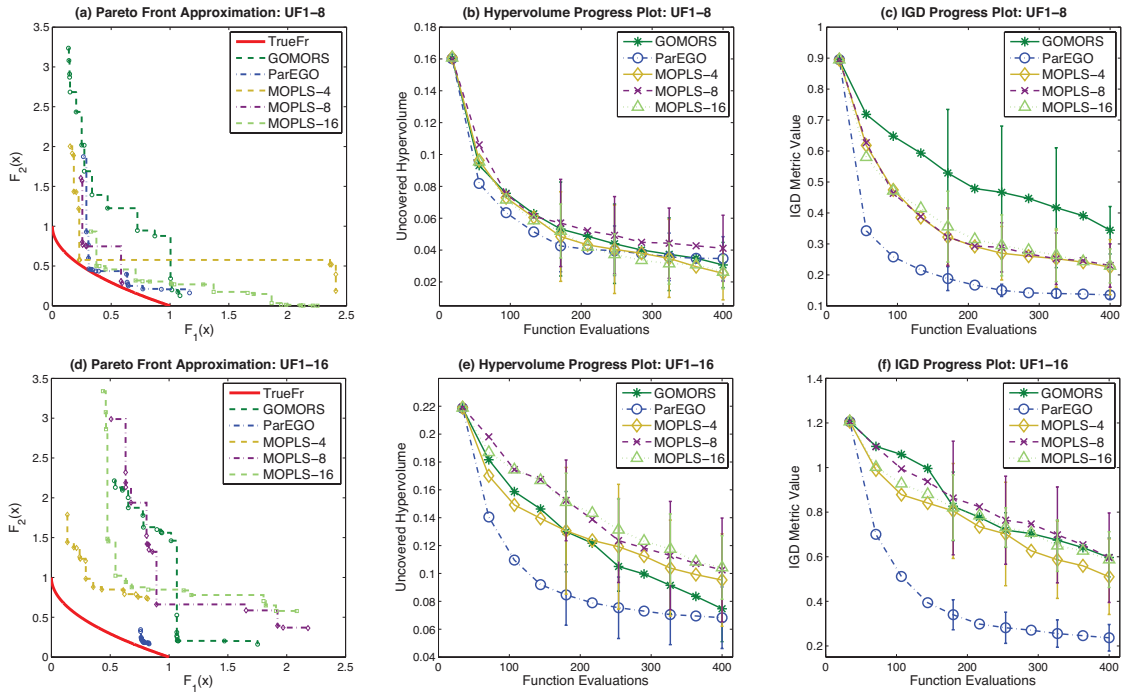

Figure 4.7: ***Test Problems UF1-8 and UF1-16***: Comparison of algorithm results after 400 function Evaluations: (a and d) Visualization of non-dominated solutions with worst hypervolume metric value in 10 trials and after 400 function evaluations, (b and e) Hypervolume progress plots, (c and f) Progress plots with mean IGD values. Vertical error bars indicate standard deviation.

4.5(b) and 4.5(c)). Furthermore, analysis of the worst Pareto Approximation plot, i.e, Figure 4.5(a) depicts that MOPLS-4 is better than GOMORS in terms of producing a diverse trade-off set.

For the higher dimensional variant of ZDT3, i.e, ZDT3-16, performance of GOMORS is clearly better than that of all other algorithms in terms of both speed of convergence as depicted in Figures 4.5(e) and 4.5(f), and in terms of obtaining a trade-off which is close to the Pareto Front as depicted in Figure 4.5(d). Hence, overall serial performance of GOMORS is better than other algorithms for the low and high dimensional variants of the ZDT3 test problem. The

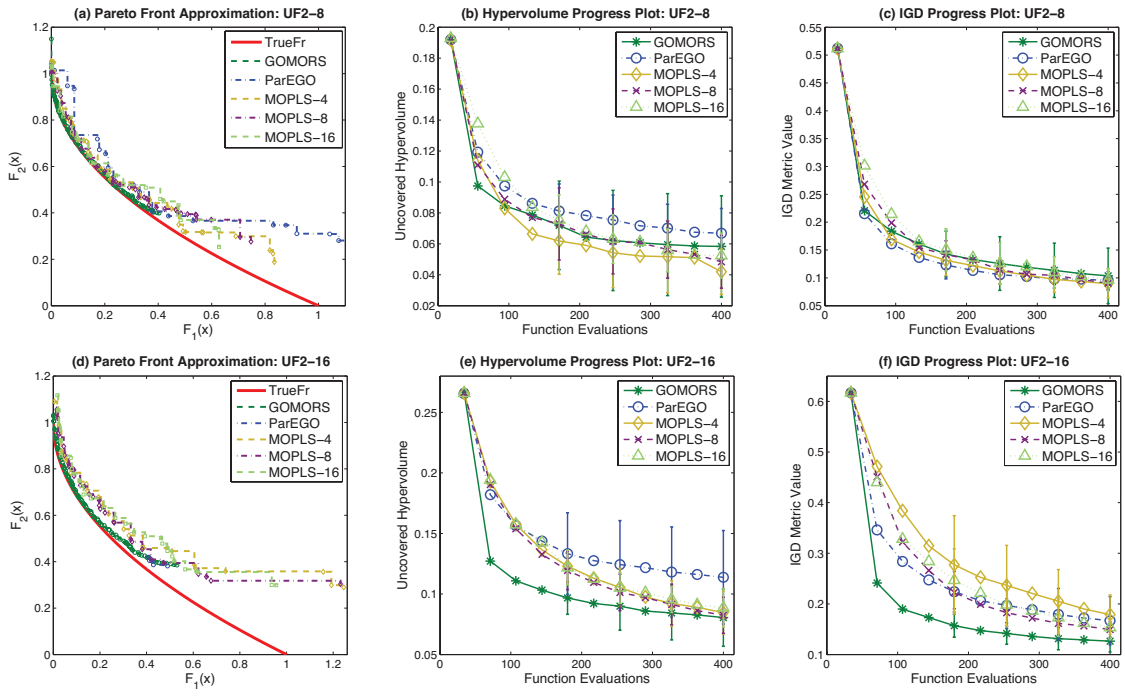

Figure 4.8: ***Test Problems UF2-8 and UF2-16***: Comparison of algorithm results after 400 function Evaluations: (a and d) Visualization of non-dominated solutions with worst hypervolume metric value in 10 trials and after 400 function evaluations, (b and e) Hypervolume progress plots, (c and f) Progress plots with mean IGD values. Vertical error bars indicate standard deviation.

overall serial performance of ParEGO is the worst.

Figure 4.6 provides a summary of performance of all algorithms after 400 function evaluations for ZDT6-8 and ZDT6-16. Results are inconclusive in terms of identifying the best algorithm for the ZDT6 test problems. As per the IGD metric (see Figure 4.6(c) and 4.6(f)) GOMORS is clearly more efficient in terms of speed of convergence to the Pareto front. However, as per the uncovered hypervolume metric (see Figure 4.6(b) and 4.6(e)) MOPLS-4 seems slightly better than GOMORS in terms of both speed of convergence and quality of Pareto front approximation after 400 function evaluations. Furthermore, the Pareto ap-

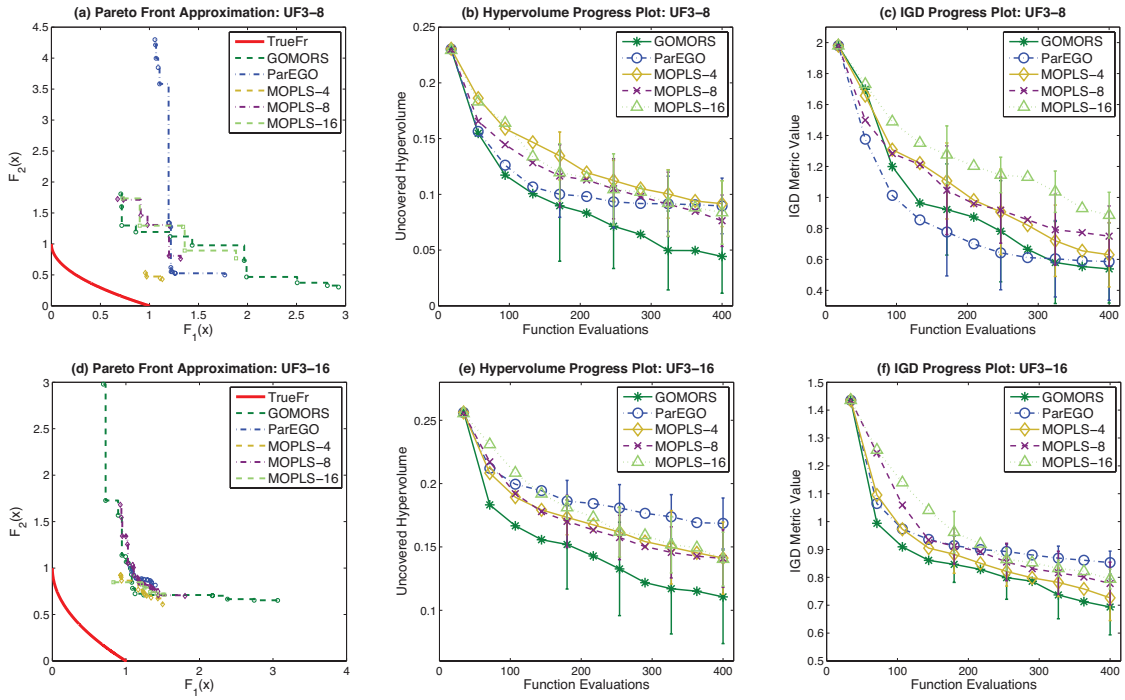

Figure 4.9: ***Test Problems UF3-8 and UF3-16***: Comparison of algorithm results after 400 function Evaluations: (a and d) Visualization of non-dominated solutions with worst hypervolume metric value in 10 trials and after 400 function evaluations, (b and e) Hypervolume progress plots, (c and f) Progress plots with mean IGD values. Vertical error bars indicate standard deviation.

proximation plots of ZDT6-8 and ZDT6-16 (see Figure 4.6(a) and 4.6(d)) depict that MOPLS-4 is most effective in producing a diverse set of trade-off solutions. Performance of ParEGO for the ZDT6 test problems is worse than the other algorithms.

Figure 4.7 shows the results of serial comparison (i.e, in function evaluations) of all algorithms for the UF1 test problems. The results here clearly indicate the relative superiority of ParEGO, especially with application to UF1-16. UF1-8 and UF1-16 are the only test problems where performance of ParEGO (in function evaluations) is better than all other algorithms.

Figure 4.8 shows the results of serial comparison (in function evaluations) of all algorithms for the UF2 test problems. The comparative results for the UF2 test problems are very similar to the results observed for the ZDT3 test problems. In case of UF2-8 (see Figures 4.8(a), 4.8(b) and 4.8(c)) MOPLS-4 has the best performance, however, marginally. In case of UF2-16 (see Figures 4.8(d), 4.8(e) and 4.8(f)) GOMORS clearly has the best performance.

For the two variants of the UF3 test problem (see Figure 4.9), performance of GOMORS is better in terms of speed of convergence as is depicted in sub-figures 4.9(b) and 4.9(d). However the difference is not discernible as per the Pareto approximation depicted in sub-figures 4.9(a) and 4.9(c).

Our analysis of the results obtained from all test problems (in function evaluations) shows that overall performance of GOMORS is best amongst all algorithms, especially for the higher dimensional test problems (16 variables). In case of the lower dimensional test problems, both GOMORS and MOPLS-4 perform reasonably well. ParEGO's performance is best only for the UF1 test problems only.

Analysis in Wall Clock Time

The analysis depicted in Figures 4.5 - 4.9 focuses on comparing algorithm performance in function evaluations. However, since a potential advantange of MOPLS and GOMORS is the added efficiency that might be obtained via parallelization, it is important to analyze performance of algorithms in wall clock time (see Section 4.4.4 for our definition of wall clock time). The following analysis focuses on analyzing comparative performance of algorithms in wall clock time for the five test problems, ZDT3, ZDT6, UF1, UF2 and UF3.

Figures 4.10 - 4.14 provide a comparative analysis of all algorithms in wall

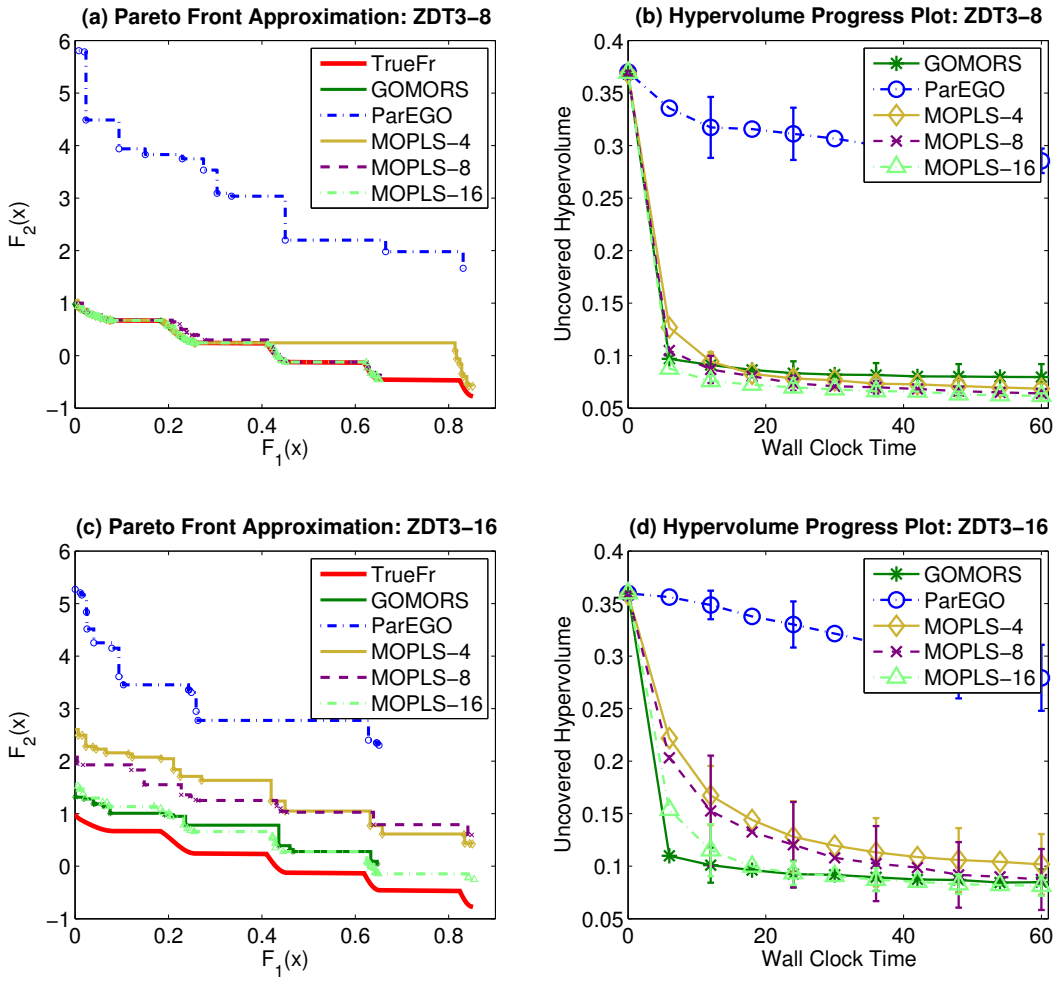

Figure 4.10: ***Test Problems ZDT3-8 and ZDT3-16***: Comparison of algorithm results in wall clock time: (a and c) Visualization of non-dominated solutions with worst hypervolume metric value in 10 trials and 60 wall clock time units, (b and d) Hypervolume progress plots for up to 60 wall clock time units. Vertical error bars indicate standard deviation.

clock time, for 10 trial runs of 60 units of wall clock time each. Each figure depicts results obtained for the two variants of each test problem, i.e, with 8 and 16 decision variables. The summarized analysis of Figures 4.10 - 4.14 includes 1) visualizations of worst trade-offs (according to hypervolume metric) obtained in 10 trials by each algorithm after 60 units of wall clock time (see sub-figures

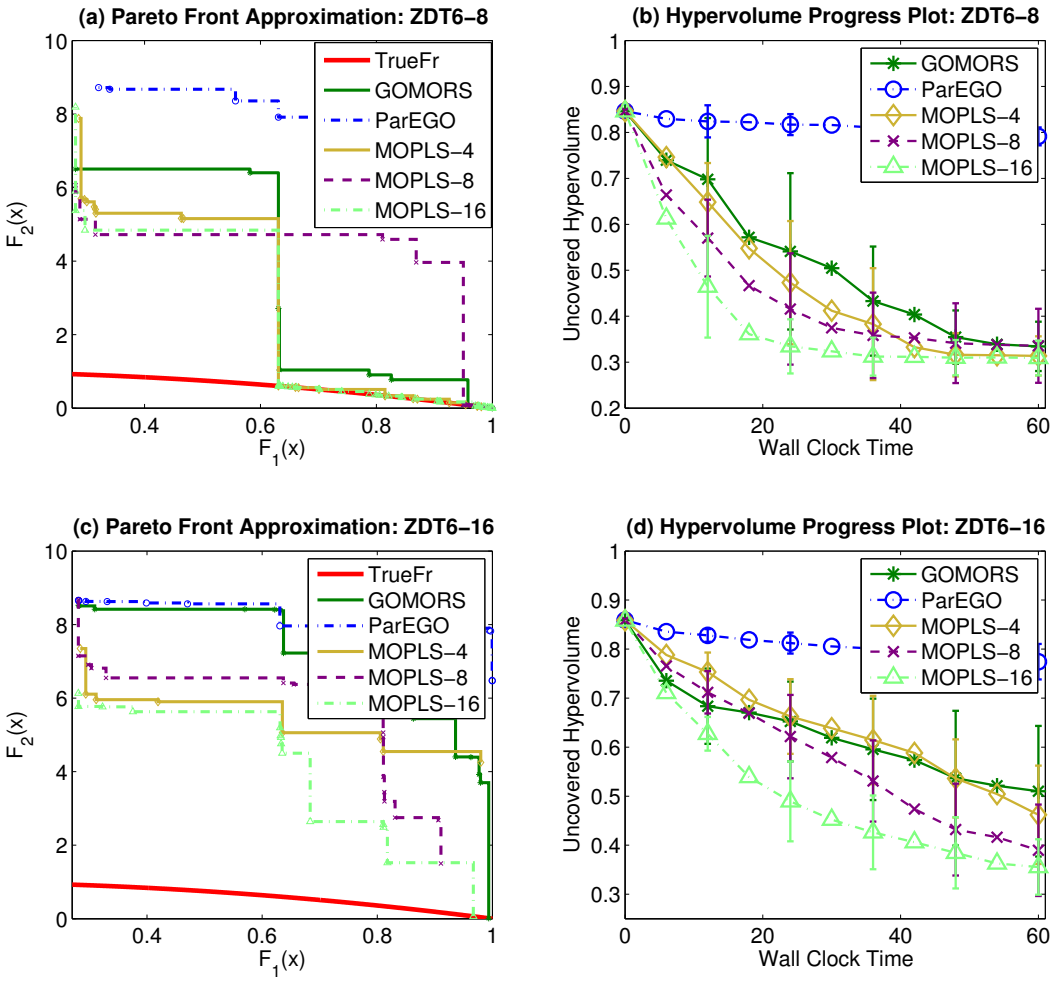

Figure 4.11: ***Test Problems ZDT6-8 and ZDT6-16***: Comparison of algorithm results in wall clock time: (a and c) Visualization of non-dominated solutions with worst hypervolume metric value in 10 trials and 60 wall clock time units, (b and d) Hypervolume progress plots for up to 60 wall clock time units. Vertical error bars indicate standard deviation.

(a) and (c)), and 2) progress plots of mean hypervolume metric value for 10 trials (see sub-figures (b) and (d)). The trade-off visualization gives a visual depiction of convergence and diversity of all algorithms in wall clock time, and the progress plots depict relative average efficiency of all algorithms in wall clock time. The solid red lines in all Pareto front approximation plots depict the

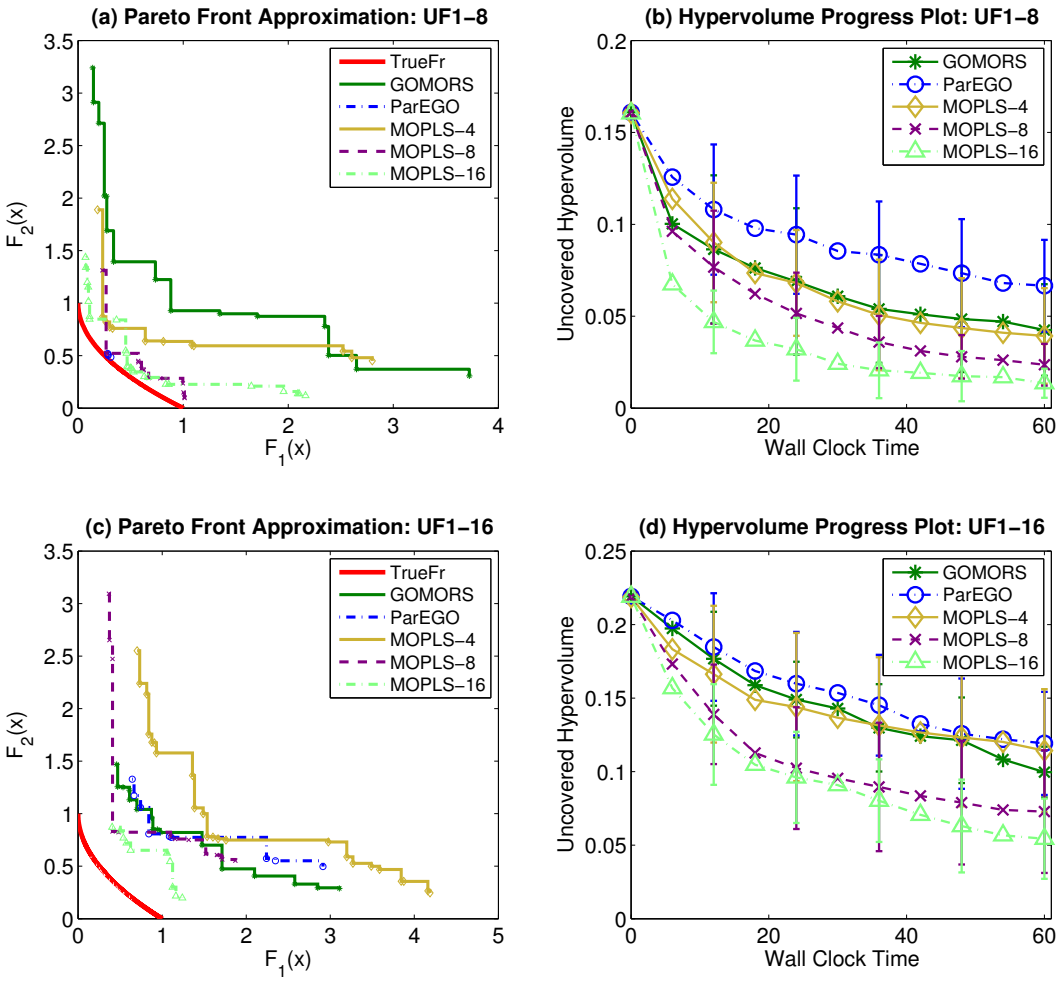

Figure 4.12: ***Test Problems UF1-8 and UF1-16***: Comparison of algorithm results in wall clock time: (a and c) Visualization of non-dominated solutions with worst hypervolume metric value in 10 trials and 60 wall clock time units, (b and d) Hypervolume progress plots for up to 60 wall clock time units. Vertical error bars indicate standard deviation.

known Pareto front. Please note that for sub-figures (a) and (c) the best solution is the line closest to the red line and for sub-figures (b) and (d) the best line is the lowest line.

Figures 4.10 - 4.14 clearly indicate the relative superiority of MOPLS-16 over all other algorithms for all test problems in wall clock time and in terms of both

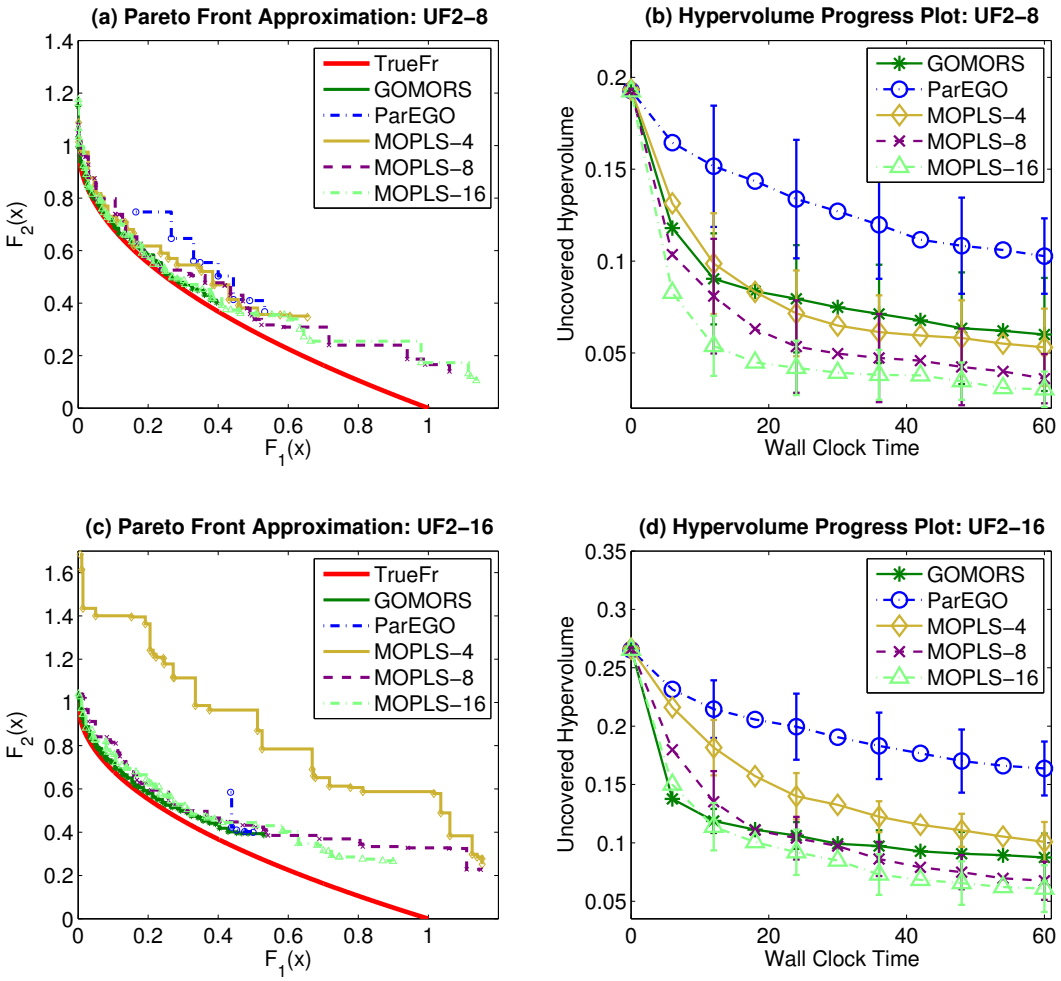

Figure 4.13: ***Test Problems UF2-8 and UF2-16***: Comparison of algorithm results in wall clock time: (a and c) Visualization of non-dominated solutions with worst hypervolume metric value in 10 trials and 60 wall clock time units, (b and d) Hypervolume progress plots for up to 60 wall clock time units. Vertical error bars indicate standard deviation.

speed of convergence (as depicted by the progress plots) and the Pareto approximation obtained after 60 wall clock units. While our comparison in function evaluations depicted that overall performance of GOMORS is best and performance of ParEGO is best for the UF1 test problem variants, Figures 4.10 - 4.14 show that MOPLS-16 clearly has the best overall performance when comparison

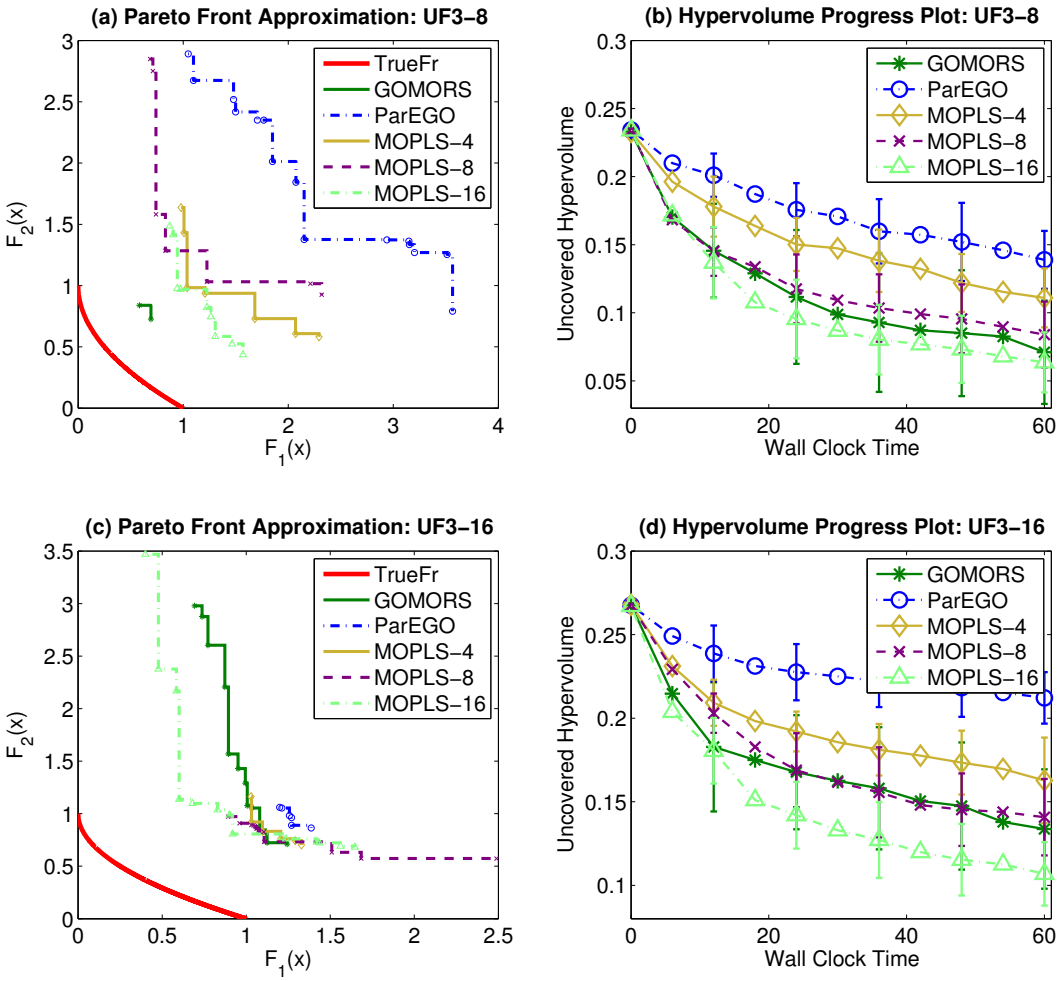

Figure 4.14: ***Test Problems UF3-8 and UF3-16***: Comparison of algorithm results in wall clock time: (a and c) Visualization of non-dominated solutions with worst hypervolume metric value in 10 trials and 60 wall clock time units, (b and d) Hypervolume progress plots for up to 60 wall clock time units. Vertical error bars indicate standard deviation.

is performed in wall clock time.

In case of both the 8 and 16 variable versions of ZDT6, UF1, UF2 and UF3 (see Figures 4.11 - 4.14) performance of MOPLS-16 is the best amongst all algorithms compared. For the ZDT3 problems, performances of MOPLS-16 and GOMORS are comparable (see Figure 4.10). The wall clock time analysis also shows that

performance of MOPLS-8 is better than other algorithms (apart from MOPLS-16) for four out of five test problems (ZDT6, UF1, UF2 and UF3). Hence, the wall clock analysis indicates that increasing the number of centers results in improved performance of MOPLS in wall clock time and performance of MOPLS is best amongst all algorithms with application to the test problems when comparison is made in wall clock time.

4.5.2 Watershed Calibration Test Suite

This section provides an analysis of the effectiveness and efficiency of MOPLS in producing good trade-off solutions within a limited evaluation budget when applied to the Watershed Calibration Test Suite (described in detail in Chapter 3). Our analysis with test problems (Section 4.5.1) indicates that increasing the number of simultaneous local search centers in MOPLS results in improved algorithm performance when results are analyzed in wall clock time. An increase in the number of centers also results in an increase in the number simultaneous local searches in different regions of interest, and subsequently increases the potential global nature of the search. Given the potential increase in the global nature of the search and benefits of parallelization, we focus on analyzing performance of MOPLS on the watershed problems with 4, 8 and 16 simultaneous centers. As mentioned earlier, these versions of the algorithm are called MOPLS-4, MOPLS-8 and MOPLS-16 respectively.

Analysis in Function Evaluations

The evaluation budget for the analysis is restricted to 1000 (The number of function evaluations means the number of objective vector evaluations), and results are compared against GOMORS, ParEGO and AMALGAM, for comparative

analysis. Due to the stochastic nature of all algorithms, 10 trial runs were performed for each algorithm, on all watershed test problems (See section 3.7 for a detailed discussion on development of the Watershed Test Suite).

Performance of all versions of MOPLS was assessed via 1) a visual depiction of trade-offs obtained by the worst trials of all algorithms (according to uncovered hypervolume metric value), 2) box plot comparison of the quality of trade-off solutions obtained by all algorithms, and 3) progress plots of the hypervolume quality metric and the IGD metric, depicting the mean hypervolume and IGD values plotted against number of function evaluations. The visual comparison of trade-offs provides a depiction of the effectiveness of all algorithms in producing good solutions within a limited simulation evaluation budget. The box plot analysis helps in analyzing reliability of algorithms in consistently producing good solutions within a limited evaluation budget. The progress plot analysis compares the efficiency of all algorithms in their ability to converge as the algorithm search progresses.

Figures 4.15 - 4.18 provide a summary of the visual trade-off comparison and box plot analysis with application to the watershed calibration test suite. For the two bi-objective watershed problems, TBROOK1 and CVILLE1 (see Figures 4.15 and 4.17) ,the worst trade-offs obtained from all algorithms (identified by the algorithm trial with worst hypervolume metric value) after 200, 500 and 1000 simulations, respectively, are shown in subfigures (A-C). The corresponding box plots in subfigures (D-F) show the variations in hypervolume metric across multiple trial runs. It should be noted that the hypervolume metric corresponds to the uncovered hypervolume illustrated in Figure 4.4, and a lower value of the hypervolume metric is desirable.

For the two 3-objective watershed problems, TBROOK2 and CVILLE2 (see

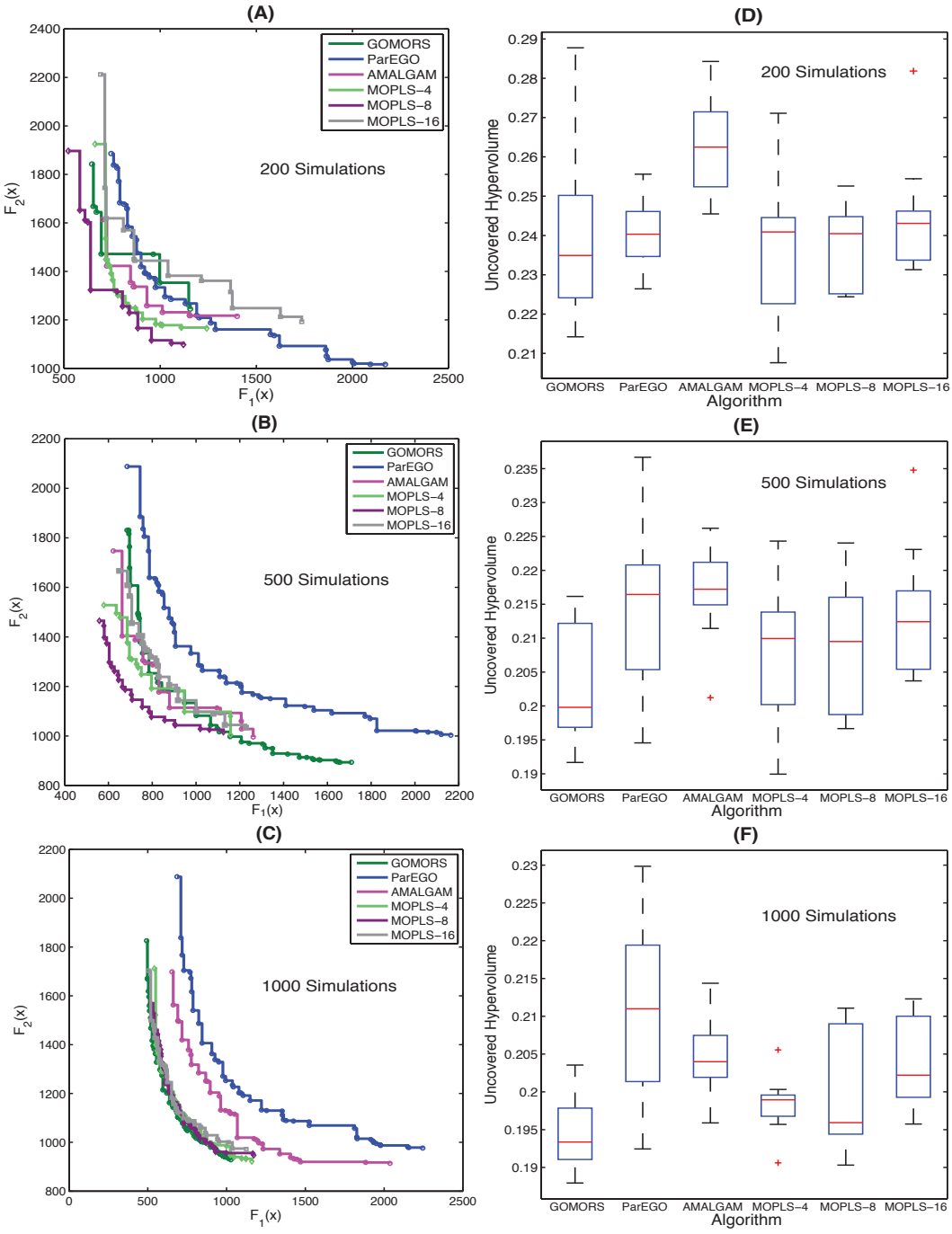

Figure 4.15: TBROOK1: Comparison of algorithm results after 200, 500, and 1000 function Evaluations (10 trials): (a-c) Visualization of non-dominated solutions obtained from worst trials of each algorithm, (d-f) Box Plot algorithm comparison of the uncovered hypervolume metric.

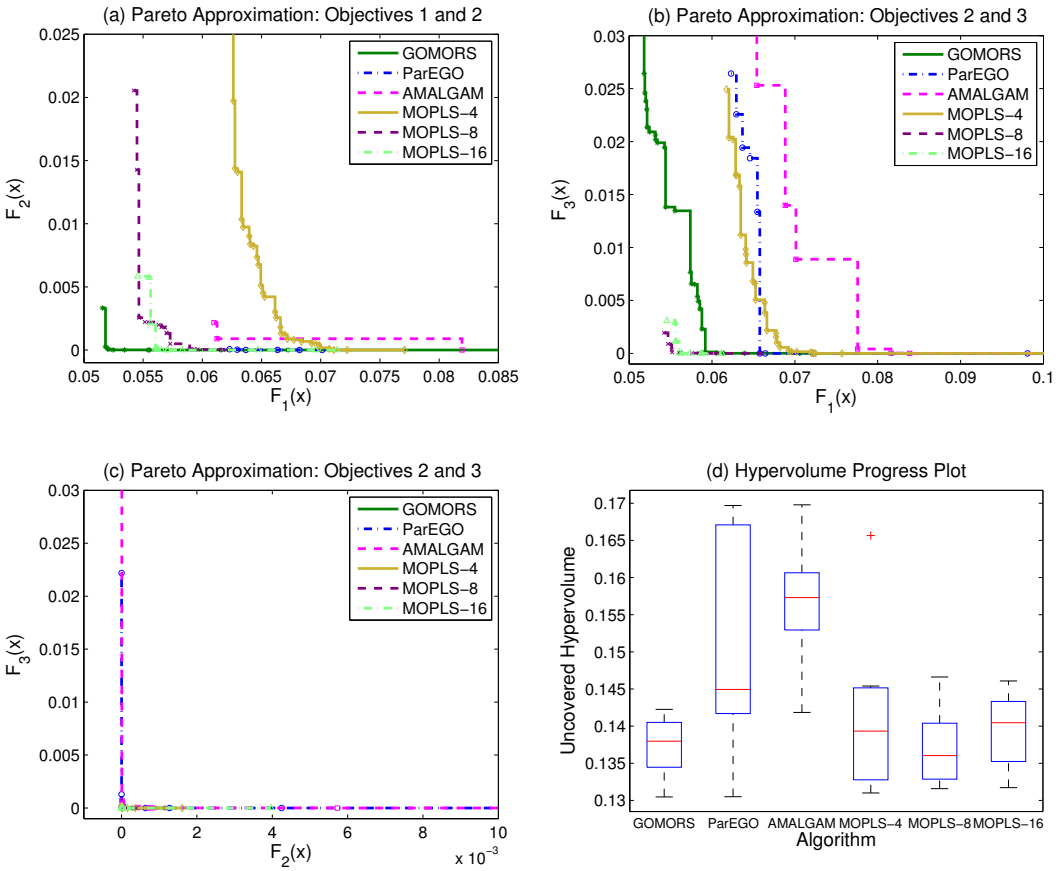

Figure 4.16: TBROOK2: Comparison of algorithm results after 1000 function Evaluations (10 trials): (a-c) Visualization of non-dominated solutions obtained from worst trials of each algorithm (two Objectives at a time), (d) Box Plot algorithm comparison of the uncovered hypervolume metric.

Figures 4.16 and 4.18) ,the worst trade-off obtained from all algorithms (identified by the algorithm trial with worst hypervolume metric value) after 1000 simulations, respectively, is visualized in subfigures (A-C). Trade-off between two objectives is visualized in each subfigure. The corresponding box plot in subfigure (D) shows the variations in hypervolume metric across multiple trial runs after 1000 function evaluations.

Figures 4.15 - 4.18 consistently indicate that the performance of GOMORS,

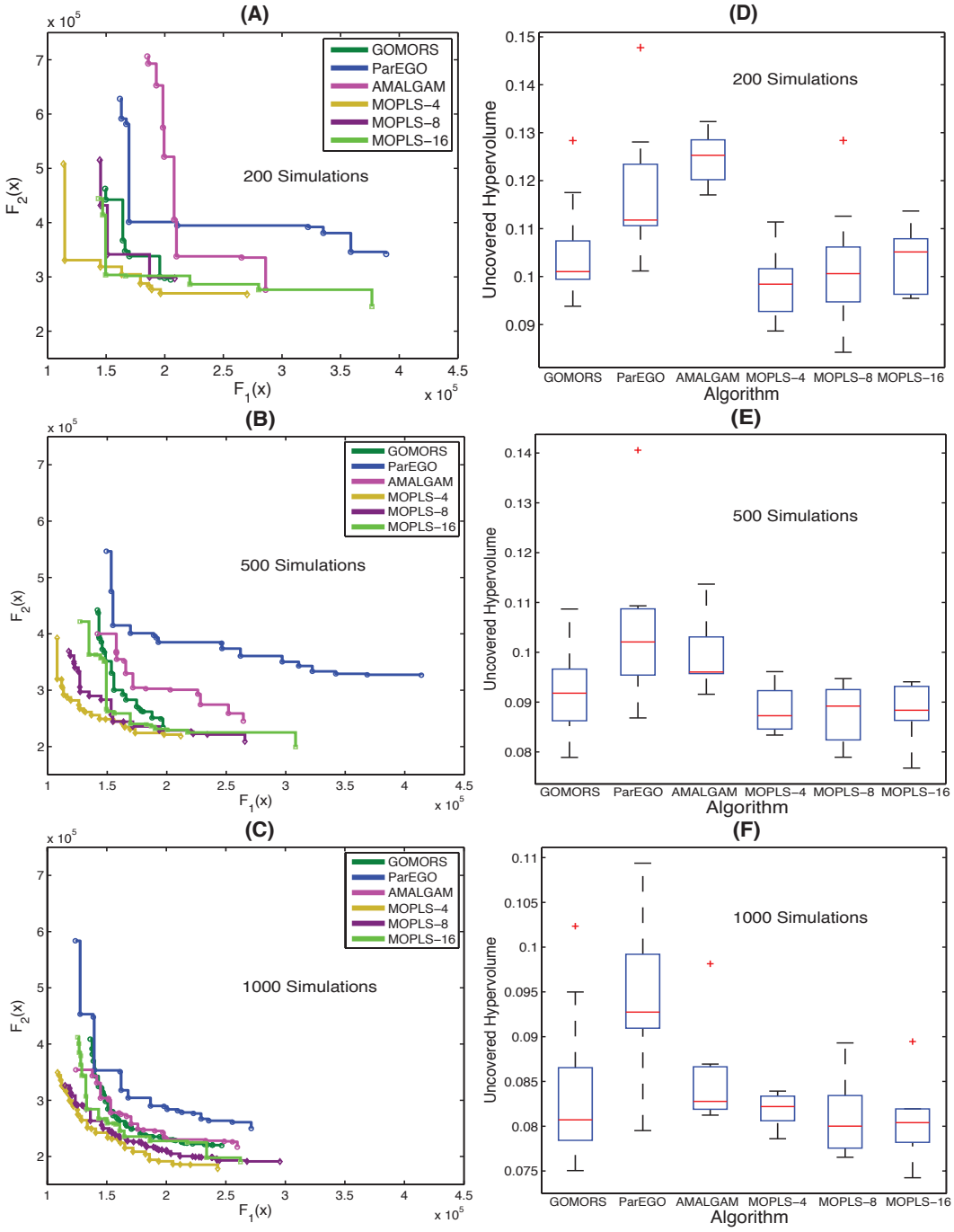

Figure 4.17: CVILLE1: Comparison of algorithm results after 200, 500, and 1000 function Evaluations (10 trials): (a-c) Visualization of non-dominated solutions obtained from worst trials of each algorithm, (d-f) Box Plot algorithm comparison of the uncovered hypervolume metric.

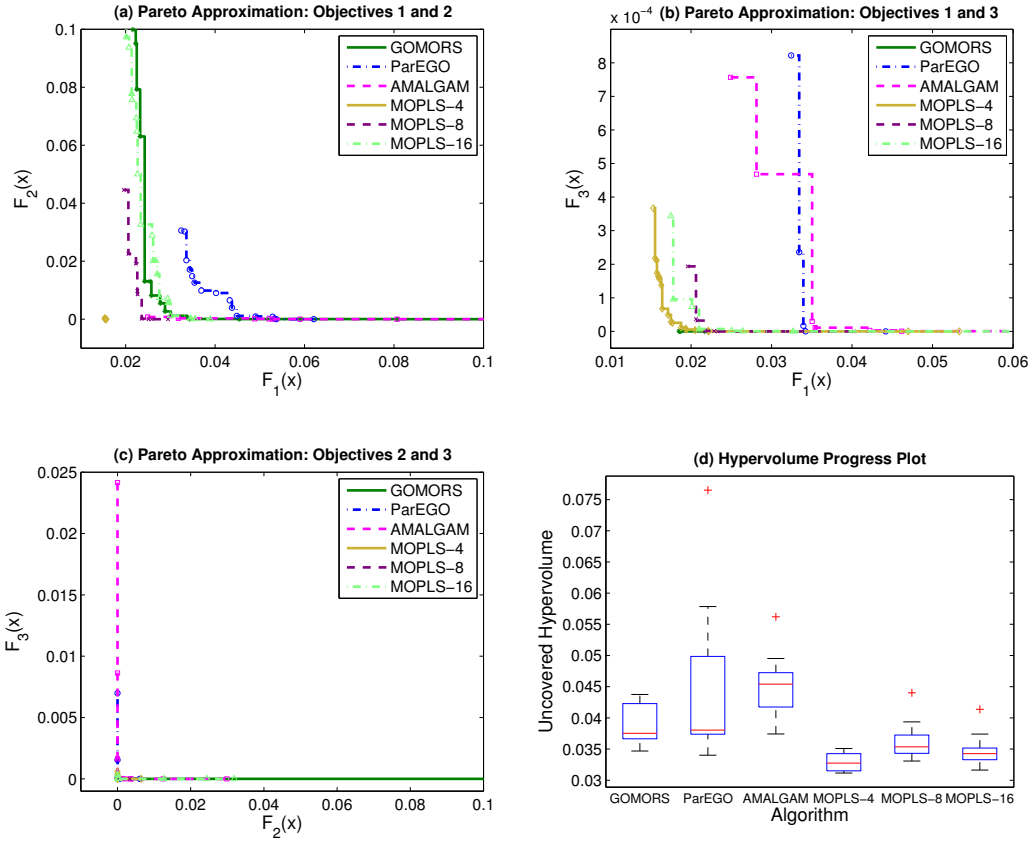

Figure 4.18: CVILLE2: Comparison of algorithm results after 1000 function Evaluations (10 trials): (a-c) Visualization of non-dominated solutions obtained from worst trials of each algorithm (two Objectives at a time), (d) Box Plot algorithm comparison of the uncovered hypervolume metric.

MOPLS-4, MOPLS-8 and MOPLS-16 is good in terms of efficiency and reliability, for all watershed test problems. The comprehensive analysis of Chapter 3, which included a statistical comparison of performance of GOMORS, ParEGO and AMALGAM (Figure 3.6), indicated that GOMORS either outperforms ParEGO and AMALGAM with statistical significance, or, it is not outperformed by any other algorithm with respect to application to the watershed calibration test suite. This notion is echoed in Figures 4.15 - 4.18. Additionally, this analysis also indicates that the performance of MOPLS-4, MOPLS-8,

and MOPLS-16 is comparable to that of GOMORS for all watershed calibration problems. The box-plot analysis of CVILLE1 and CVILLE2 further indicates that MOPLS-4, MOPLS-8 and MOPLS-16 are more reliable than GOMORS and consistently produce relatively good trade-off solutions.

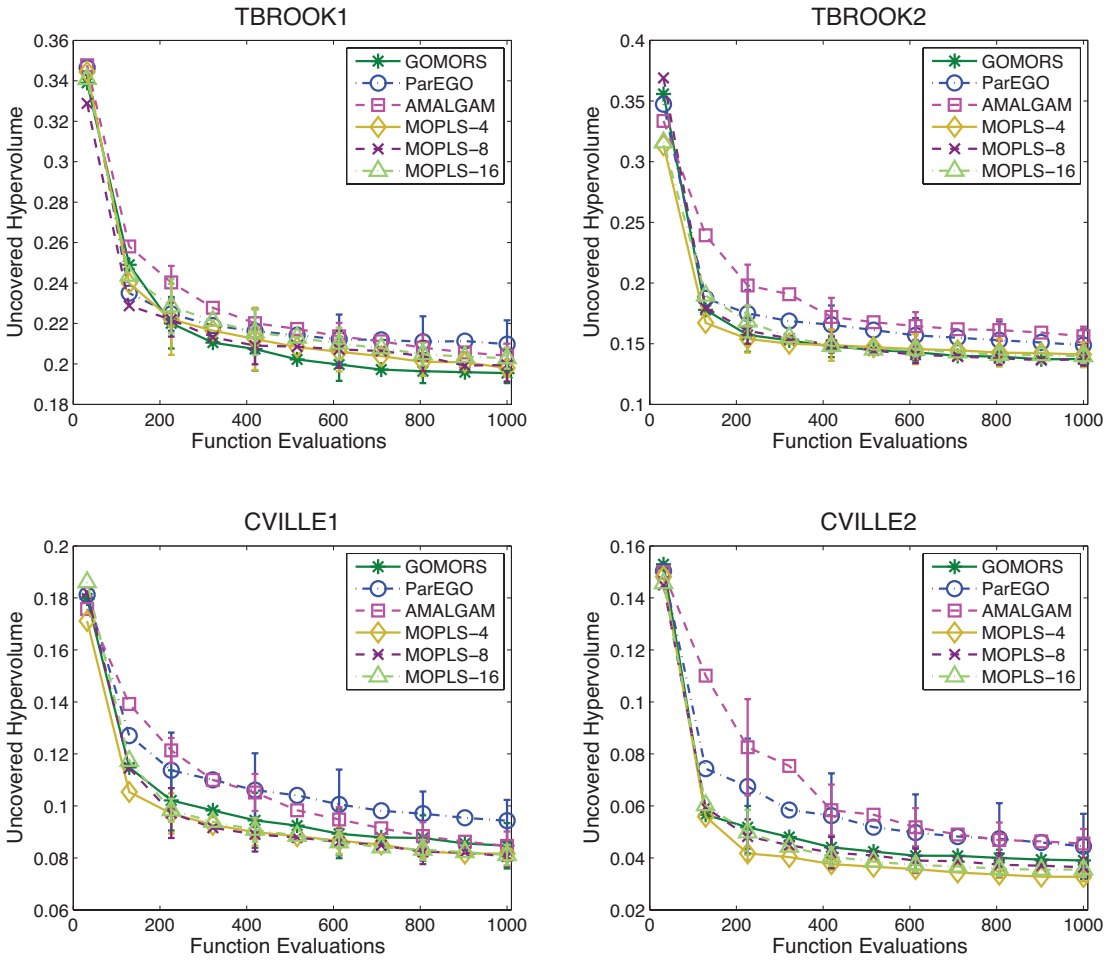

Figure 4.19: Comparison of efficiency of algorithms for the Watershed calibration test suite: Hypervolume metric averaged over multiple trials, as a function of number of function evaluations (iterations). Vertical error bars indicate standard deviation.

The efficiency of algorithms is also analyzed for the watershed calibration test suite. Figures 4.19 and 4.20 depict the efficiency of all algorithms via progress plots of mean metric values plotted against the number of function

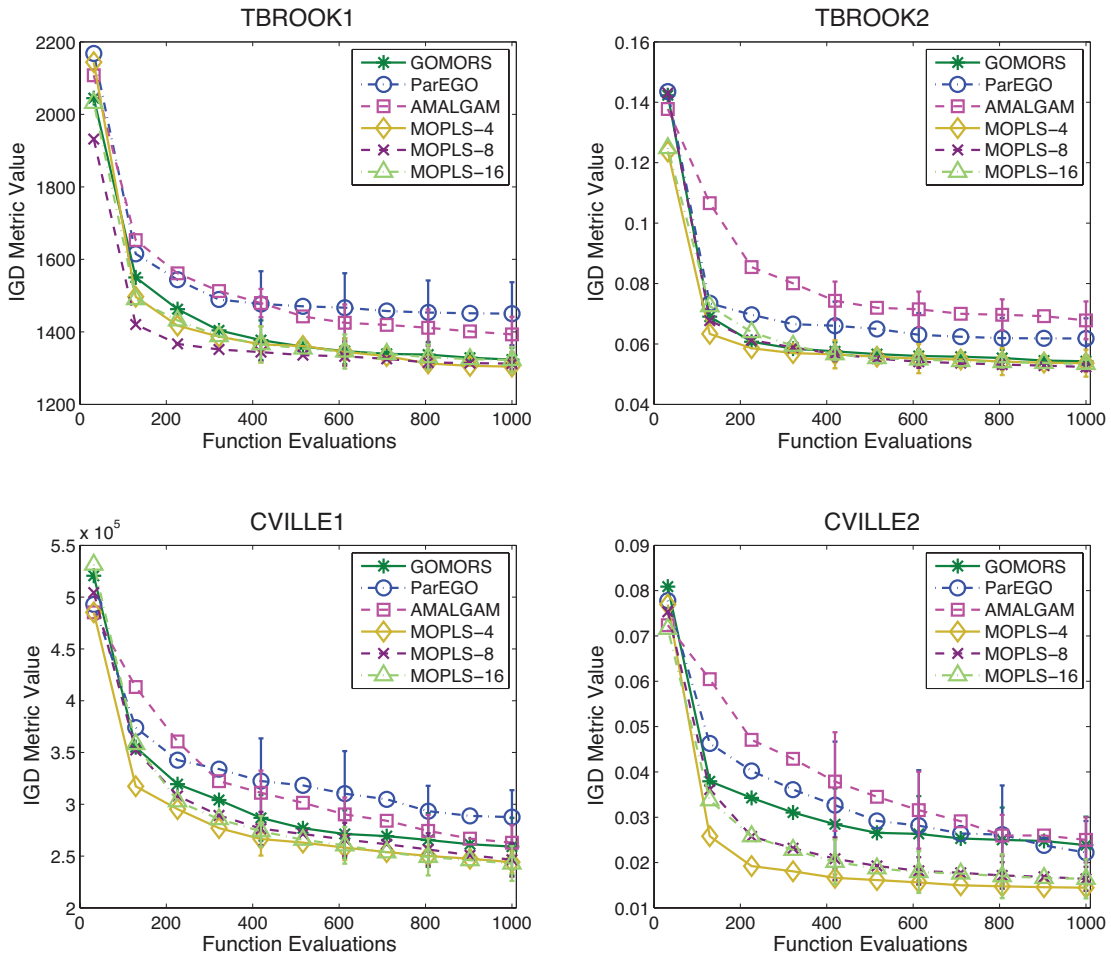

Figure 4.20: Comparison of efficiency of algorithms for the Watershed calibration test suite: IGD metric averaged over multiple trials, as a function of number of evaluations (iterations). Vertical error bars indicate standard deviation.

evaluations. A clear indication from both progress plot figures is that GOMORS, MOPLS-4, MOPLS-8 and MOPLS-16 outperform ParEGO and AMALGAM in terms of their efficiency of convergence. Additionally, the visual trade-off comparison of Figures 4.15 - 4.18 showed that GOMORS, MOPLS-4, MOPLS-8 and MOPLS-16 are effective in producing diverse trade-offs.

The comprehensive analysis of Figures 4.15 - 4.20 indicates that GOMORS, MOPLS-4, MOPLS-8 and MOPLS-16 are efficient, effective and reliable, in

comparison to ParEGO and AMALGAM, with application to all problems in the watershed calibration test suite. Furthermore, MOPLS could outperform GOMORS in wall clock time if the speed-ups obtained via parallelization are considered. A performance analysis in wall clock time is also performed and discussed later in our analysis. Another critical observation of this analysis is that the performance of MOPLS remains consistent with an increase in the number of centers. This indicates that the algorithm could have immense potential to be highly parallelized. This observation does make sense, because increasing the number of centers, increases the number of simultaneous local searches, resulting in a more global search within each algorithm iteration. However, the value of adding centers could be highly sensitive to the dynamics of the problem being solved.

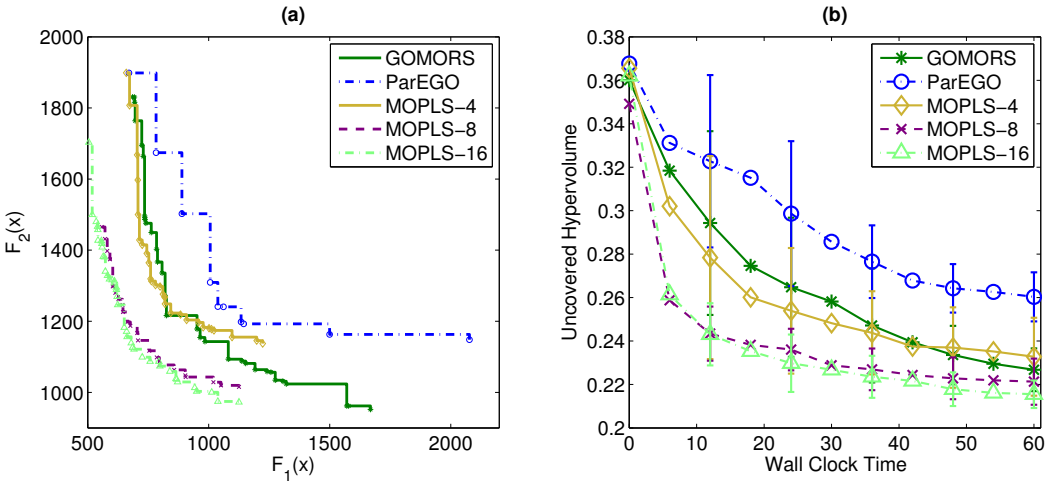

Figure 4.21: ***Problem TBROOK1***: Comparison of algorithm results in wall clock time: (a) Visualization of non-dominated solutions with worst hypervolume metric value in 10 trials and after 60 wall clock time units, (b) Hypervolume progress plots for 60 wall clock time units. Vertical error bars indicate standard deviation.

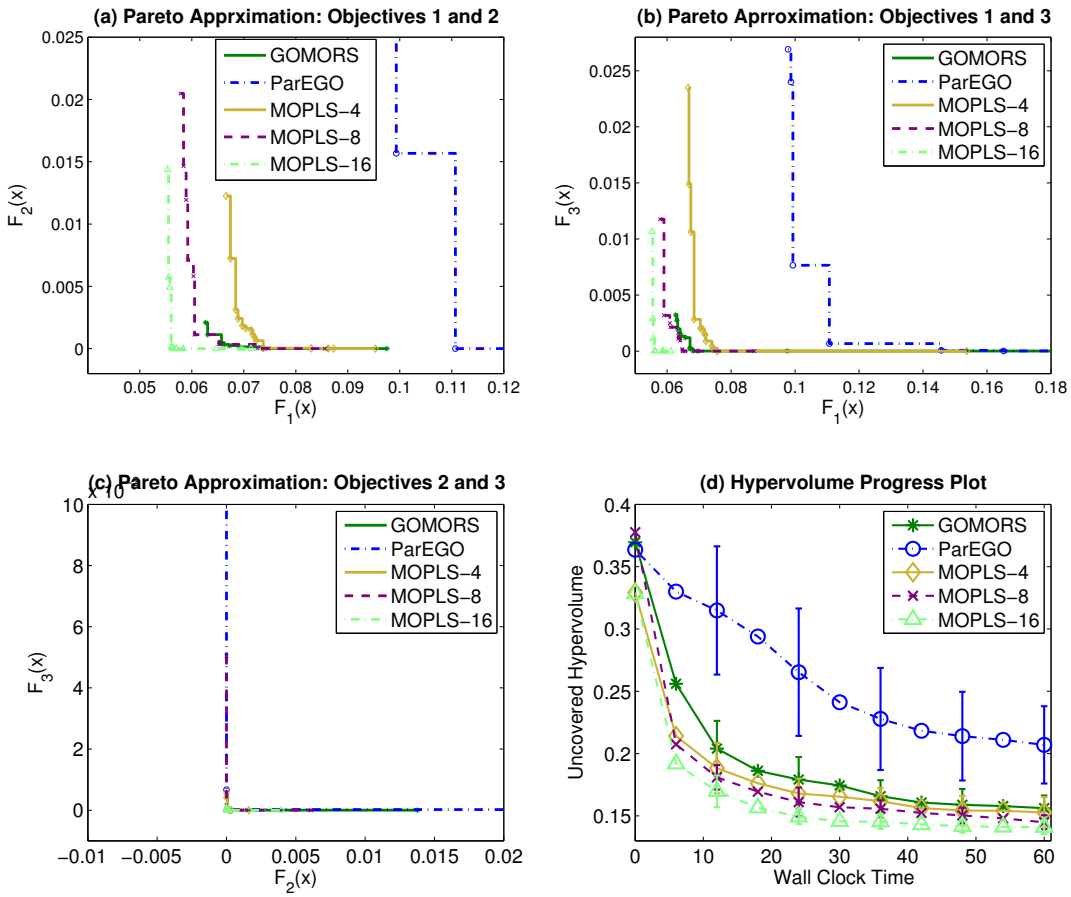

Figure 4.22: ***Problem TBROOK2***: Comparison of algorithm results in wall clock time: (a-c) Visualization of non-dominated solutions with worst hypervolume metric value in 10 trials and after 60 wall clock time units (2 Objectives at a time), (d) Hypervolume progress plots for 60 wall clock time units. Vertical error bars indicate standard deviation.

Analysis in Wall Clock Time

The following analysis focuses on analyzing comparative performance of algorithms in wall clock time for the watershed calibration test suite. This comparative analysis is performed for MOPLS-4, MOPLS-8, MOPLS-16, GOMORS and PArEGO. Since MOPLS (all three versions, i.e, MOPLS-4, MOPLS-8 and MOPLS-16) clearly outperforms AMALGAM in terms of function evaluations,

within a budget of 1000 evaluations, AMALGAM is not included in the wall clock time analysis.

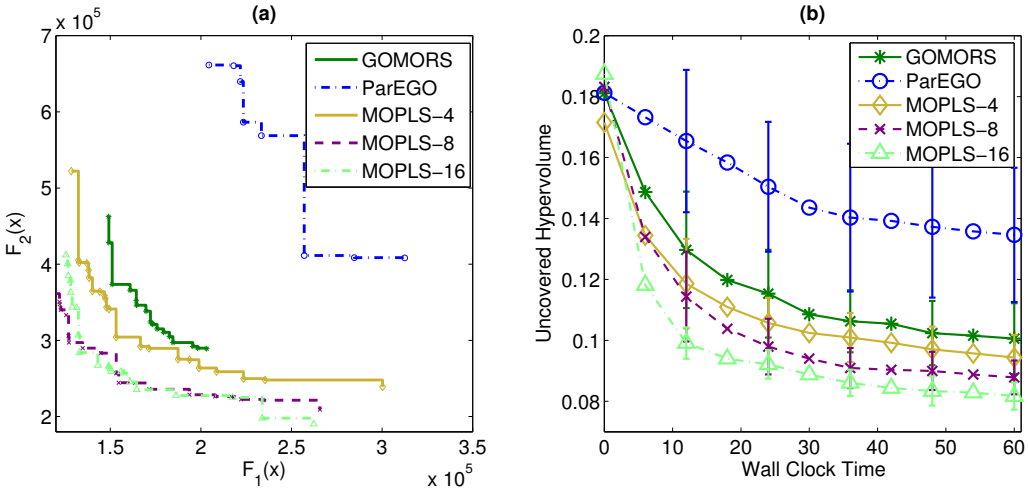

Figure 4.23: ***Problem CVILLE1***: Comparison of algorithm results in wall clock time: (a) Visualization of non-dominated solutions with worst hypervolume metric value in 10 trials and after 60 wall clock time units, (b) Hypervolume progress plots for 60 wall clock time units. Vertical error bars indicate standard deviation.

Figures 4.21 - 4.24 provide a comparative analysis of all algorithms in wall clock time, for 10 trial runs of 60 units of wall clock time each. Each figure depicts results obtained for a watershed calibration test problem. The summarized analyses of Figures 4.21 - 4.24 include 1) visualizations of worst trade-offs (according to hypervolume metric) obtained in 10 trials by each algorithm after 60 units of wall clock time, and 2) progress plots of mean hypervolume metric value for 10 trials. Please note that for figures depicting visualizations of trade-offs, the best trade-off line is the one that is closest to the zero vector (i.e, lower left corner of figure) and is diverse, and for figures showing uncovered hypervolume progress plots, the best line is the lowest line.

Figures 4.21 and 4.23 depict results for the two bi-objective watershed prob-

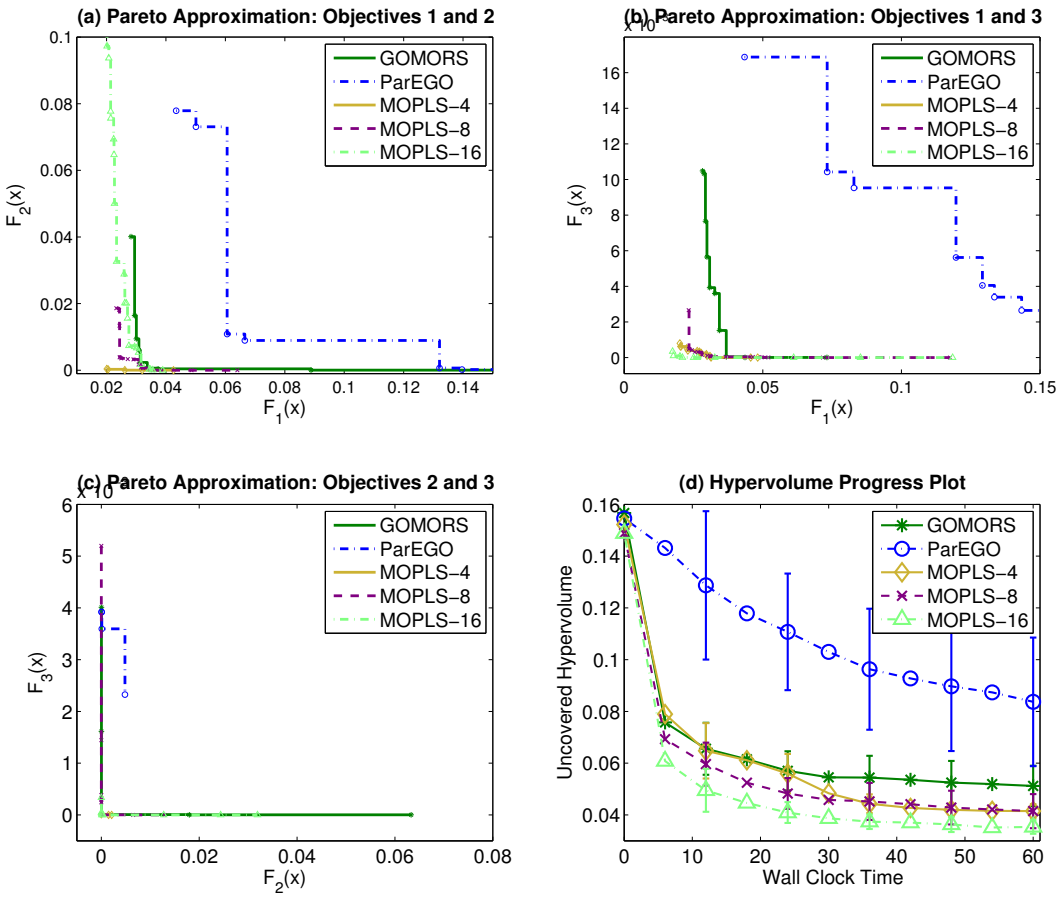

Figure 4.24: ***Problem CVILLE2***: Comparison of algorithm results in wall clock time: (a-c) Visualization of non-dominated solutions with worst hypervolume metric value in 10 trials and after 60 wall clock time units (2 Objectives at a time), (d) Hypervolume progress plots for 60 wall clock time units. Vertical error bars indicate standard deviation.

lems, i.e, TBROOK1 and CVILLE1. Each figure has two sub-figures. Sub-figure(a) depicts a visual comparison of the worst non-dominated solutions (in terms of uncovered hypervolume) obtained from each algorithm after 60 wall clock time units. Sub-figure(b) depicts a comparison of algorithms via progress plot of uncovered hypervolume plotted against wall clock time. The comparative analysis of both TBROOK1 and CVILLE1 (as depicted in Figures 4.21 and 4.23) in wall clock time clearly shows that MOPLS-16 outperforms all other algo-

rithms in terms of both speed of convergence and quality of trade-offs obtained after 60 wall clock time units.

Figures 4.22 and 4.24 show results for the two 3-objective watershed problems, i.e, TBROOK2 and CVILLE2. Each figure has four sub-figures. Sub-figures(a-c) provide a visual comparison of the worst non-dominated solutions (in terms of uncovered hypervolume) obtained from each algorithm after 60 wall clock time units. Sub-figure(d) shows a comparison of algorithms via progress plot of uncovered hypervolume plotted against wall clock time. The comparative analysis of TBROOK2 and CVILLE2 (as depicted in Figures 4.22 and 4.24) also shows that performance of MOPLS-16 is better than all other algorithms. Hence, it can be concluded that in terms of application to test problems and the watershed calibration test suite, MOPLS is an efficient, effective and reliable algorithm, especially when its potential for parallelization is taken into consideration.

4.6 Conclusion

Expensive multi-objective optimization problems, involving many decisions (more than 10) can benefit from surrogate-assisted local search. In Chapter 2, we incorporated surrogate-assisted local search with global surrogate search to guide the heuristic search process. Our results indicated that local surrogate-assisted search (also called Gap Optimization in Chapter 2) could assist in efficient multi-objective optimization of computationally expensive problems.

Regis and Shoemaker [27] introduced the idea of local candidate search, by incorporating radial basis functions as surrogates to assist in efficient single objective optimization. Their results were promising for optimization within a very limited evaluation budget.

In this chapter, we expanded on the local candidate search idea proposed by Regis and Shoemaker [27] to develop a Multi-Objective Parallel Local Surrogate Search (MOPLS) algorithm. The development and subsequent performance analysis of MOPLS shows that the algorithm produces promising results within a limited evaluation budget. The algorithm analysis on a limited budget of 400 evaluations with application to various test problems, shows that the algorithm compares well against other surrogate-assisted search algorithms, namely GOMORS and ParEGO. With incorporation of the speed-ups obtained via parallelization of MOPLS (analyzed in terms of wall clock time), we show that the algorithm's performance is either comparable or superior to other surrogate based search algorithms.

Application of MOPLS to a watershed flow calibration test suite shows even better results. The algorithm compares well against GOMORS, ParEGO and AMALGAM within a limited budget of 1000 evaluations. Furthermore, MOPLS is superior to GOMORS and ParEGO when algorithm comparison is performed in wall clock time (see Section 4.4.4 for our definition of wall clock time).

Since there is an increased interest in multi-method search algorithms for efficient multi-objective optimization of computationally expensive problems, a local surrogate-assisted search can be incorporated into multi method algorithms to further improve optimization efficiency. The synchronous parallel framework of MOPLS makes it an attractive search methodology for implementation on cloud clusters. While this analysis gives an indication of the immense promise of MOPLS and parallel local surrogate-assisted search, especially with application to computationally expensive watershed calibration problems, further innovations can be introduced to make the algorithm efficient and effective for many-objective (more than three) optimization of computationally expen-

sive problems.

4.7 References

[1] Taimoor Akhtar and Christine A. Shoemaker. Multi objective optimization of computationally expensive multi-modal functions with rbf surrogates and multi-rule selection. *Journal of Global Optimization*, pages 1–16, 2015.

[2] Johannes M. Bader. *Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods*. CreateSpace, Paramount, CA, 2010.

[3] Karl Bringmann and Tobias Friedrich. Approximating the least hypervolume contributor: Np-hard in general, but fast in practice. In Matthias Ehrgott, CarlosM. Fonseca, Xavier Gandibleux, Jin-Kao Hao, and Marc Sevaux, editors, *Evolutionary Multi-Criterion Optimization*, volume 5467 of *Lecture Notes in Computer Science*, pages 6–20. Springer Berlin Heidelberg, 2009.

[4] Martin D Buhmann. *Radial Basis Functions*. Cambridge University Press, New York, NY, USA, 2003.

[5] C.A. Coello Coello, G.L. Lamont, and D.A. van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation. Springer, Berlin, Heidelberg, 2nd edition, 2007.

[6] Jared L. Cohon and David H. Marks. A review and evaluation of multiobjective programing techniques. *Water Resources Research*, 11(2):208–220, 1975.

[7] Kalyanmoy Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evol. Comput.*, 7(3):205–230, September 1999.

[8] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, PPSN VI, pages 849–858, London, UK, UK, 2000. Springer-Verlag.

[9] Kalyanmoy Deb and Deb Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 1 edition, June 2001.

[10] Alan Diaz-Manriquez, Gregorio Toscano-Pulido, and Wilfrido Gomez-Flores. On the selection of surrogate models in evolutionary optimization algorithms. *IEEE Congress of Evolutionary Computation (CEC)*, pages 2155–2162, June 2011.

[11] Chariklia A. Georgopoulou and Kyriakos C. Giannakoglou. A multi-objective metamodel-assisted memetic algorithm with strength-based local refinement. *Engineering Optimization*, 41(10):909–923, 2009.

[12] Fred Glover, Manuel Laguna, and Rafael Mart. Tabu search, 1997.

[13] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

[14] H Gutmann. On the Semi-norm of Radial Basis Function Interpolants. *Journal of Approximation Theory*, 111(2):315–328, August 2001.

[15] H.-M. Gutmann. A radial basis function method for global optimization. *J. of Global Optimization*, 19:201–227, March 2001.

[16] David Hadka and Patrick Reed. Borg: An auto-adaptive many-objective

evolutionary computing framework. *Evol. Comput.*, 21(2):231–259, May 2013.

[17] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *J. of Global Optimization*, 13(4):455–492, December 1998.

[18] Marios K Karakasis and Kyriakos C Giannakoglou. On the use of metamodel-assisted , multi-objective. *Engineering Optimization*, 38(8):941–957, 2006.

[19] J. Knowles. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 8(5):1341–66, February 2006.

[20] Hui Li and Qingfu Zhang. Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *Trans. Evol. Comp*, 13(2):284–302, April 2009.

[21] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):pp. 239–245, 1979.

[22] John Nicklow, Patrick Reed, Dragan Savic, Tibebe Dessalegne, Laura Harrell, Amy C. Hilton, Mohammed Karamouz, Barbara Minsker, Avi Ostfeld, Abhishek Singh, and Emily Zechman. State of the Art for Genetic Algorithms and beyond in Water Resources Planning and Management. *Journal of Water Resources Planning and Management*, 1(1):27, 2009.

[23] Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, and Markus Vincze. Multiobjective optimization on a limited budget of evaluations using

model-assisted s-metric selection. In *Proceedings of the 10th international conference on Parallel Problem Solving from Nature: PPSN X*, pages 784–794, Berlin, Heidelberg, 2008. Springer-Verlag.

[24] M. J. D. Powell. *The Theory of Radial Basis Function Approximation in 1990*, pages 105–210. Oxford University Press, USA, May 1992.

[25] Kevin L. Priddy and Paul E. Keller. *Artificial Neural Networks: An Introduction*. SPIE Press, 2005.

[26] P. M. Reed, D. Hadka, J. D. Herman, J. R. Kasprzyk, and J. B. Kollat. Evolutionary multiobjective optimization in water resources: The past, present, and future. *Advances in Water Resources*, 51:438–456, January 2013.

[27] R Regis, C Shoemaker, and A. stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J. of Computing*, 19:497–509, 2007.

[28] R Regis, C Shoemaker, and A. stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J. of Computing*, 19:497–509, 2007.

[29] R.G. Regis and C.a. Shoemaker. Local Function Approximation in Evolutionary Algorithms for the Optimization of Costly Functions. *IEEE Transactions on Evolutionary Computation*, 8(5):490–505, October 2004.

[30] Rommel G Regis and Christine A Shoemaker. Parallel Stochastic Global Optimization Using Radial Basis Functions. *INFORMS J. of Computing*, 21(3):411–426, 2009.

[31] Rommel G Regis and Christine A Shoemaker. Combining Radial Basis Function Surrogates Dynamic Coordinate Search in High Dimensional Ex-

pensive Black-box Optimization. *Engineering Optimization*, 45(5):529–555, 2013.

[32] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and Analysis of Computer Experiments. *Statistical Science*, 4(4):409–423, November 1989.

[33] L.V. Santana-Quintero, V.A. Serrano-Hernandez, Carlos A. Coello Coello, A.G. Hernandez-Diaz, and J. Molina. Use of radial basis functions and rough sets for evolutionary multi-objective optimization. In *Computational Intelligence in Multicriteria Decision Making, IEEE Symposium on*, pages 107–114, April 2007.

[34] Vladimir N. Vapnik. *Statistical learning theory*. Wiley, 1 edition, September 1998.

[35] David A. Van Veldhuizen and David A. Van Veldhuizen. Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations. Technical report, Evolutionary Computation, 1999.

[36] Jasper a Vrugt and Bruce A Robinson. Improved evolutionary optimization from genetically adaptive multimethod search. *Proceedings of the National Academy of Sciences of the United States of America*, 104(3):708–11, January 2007.

[37] Jasper a Vrugt and Bruce a Robinson. Improved evolutionary optimization from genetically adaptive multimethod search. *Proceedings of the National Academy of Sciences of the United States of America*, 104(3):708–11, January 2007.

[38] Stefan M Wild, Rommel G Regis, and Christine A Shoemaker. ORBIT: optimization by Radial Basis Function Interpolation in Trust-Regions. *SIAM Journal of Scientific Computing*, 30(6):3197–3219, 2007.

[39] Raghavan Srinivasan Xuesong Zhang and Michael Van Liew. On the use of multi-algorithm, genetically adaptive multi-objective method for multi-site calibration of the swat model. *Hydrological Processes*, 24:955–969, 2010.

[40] Qingfu Zhang, Wudong Liu, Edward Tsang, and Botond Virginas. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Trans. Evol. Comp*, 14(3):456–474, June 2010.

[41] Qingfu Zhang, Aimin Zhou, Shizheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Wudong Liu. Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition. *Mechanical Engineering*, pages 1–30, 2009.

[42] Eckart Zitzler, Dimo Brockhoff, and Lothar Thiele. The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization*, volume 4403 of *Lecture Notes in Computer Science*, chapter 64, pages 862–876. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

CHAPTER 5

CONCLUSION

This research was aimed at introducing methods for tackling calibration problems, within a Multi-Objective Optimization framework. Multi-objective methods have recently attracted a high level of interest across various applied disciplines[2], and have been in focus in this thesis. The multi-objective problem proposes a set of alternative or trade-off solutions to the decision maker, after which he/she can make an informed decision.

Within the hydrological model calibration literature, various authors have identified the need to employ multi-objective methods [4] [8] [12] [3] to identify alternate calibrations. Given the computational burden associated with evaluation of objectives through complex models, our focus has been on developing algorithms for efficient calibration of computationally expensive hydrological models. This requires that good solutions be obtained within a limited number of evaluations. Chapters 2 and 4 of the thesis discuss details of the developed algorithms, and document their effectiveness in producing reasonable calibration alternatives within a limited evaluation budget. The work depicted in the thesis also highlights the value that can be added, within meta-heuristics search methodologies, by incorporating surrogate assistance, especially, when evaluation of objectives is very time consuming. Furthermore, Radial Basis Functions can be used effectively as surrogates for problems with more than 10 decision variables. The analysis in Chapter 4 also highlights the advantage of incorporating surrogates within a local search based synchronous parallel optimization framework to efficiently solve expensive multi-objective optimization problems.

The aim of an effective multi-objective calibration analysis is not the identification of many trade-off solutions but the identification of meaningful trade-off solutions [7] (also referred as behavioral solutions in prior literature [1]). Identification of meaningful calibration solutions can assist decision makers in understanding and quantifying model uncertainties (The GLUE framework is an example [1] [3]). Furthermore, the existence of numerous meaningful solutions could also highlight structural imperfections of a model, and help decision makers in understanding model limitations.

Since, the focus of our research has been on efficient multi-objective optimization of watershed calibration problems, a critical question within the analysis is "Can we obtain meaningful trade-offs within a limited evaluation budget?". Chapter 3 of the thesis introduces a new metric for this purpose, Distributed Cardinality, that defines and identifies distributed meaningful trade-offs and also quantifies an algorithm's effectiveness in producing meaningful trade-offs. Furthermore, the analysis in Section 3.10 (based on the Distributed Cardinality index) shows that GOMORS is more effective than ParEGO [6] and AMALGAM [11] in identifying meaningful solutions for multi objective watershed model calibrations problems.

This thesis depicts a significant progress towards efficient multi-objective optimization of watershed calibration problems. We have successfully highlighted the value of surrogate assisted search in multi-objective optimization, and its potential application to expensive multi-objective simulation optimization problems. We showed that meaningful calibrations could be obtained within a limited evaluation budget through efficient optimization, for models based on the Soil and Water Assessment Tool (SWAT) [10] and [9]. Overall, this thesis contributes to multi-objective calibration of watershed models, by

proposing algorithms for efficient calibration, and by proposing a new metric for comparing performance of algorithms in terms of producing meaningful calibrations.

In addition to the aim of providing insights through current work, this thesis also opens up avenues of further research in various directions. While surrogate-assisted search has been discussed extensively in prior literature, the potential for its hybridization with other meta-heuristics has not been explored extensively. Recent heuristic algorithm development research efforts have shown interest in hybridized algorithms [2]. AMALGAM [11] and Borg[5] are amongst some recent algorithms that incorporate multi method search within the optimization process. While these algorithms are efficient, and effective, their effectiveness could further improve by incorporation of surrogate assistance in the search process. MOPLS can be employed within a Multi-method search for efficient optimization. Given the recent advances in cloud computing, MOPLS could be effectively used in multi-method search optimization of computationally expensive Multi-criteria problems.

The optimization formulations incorporated in this research typically employ two or three objectives. However, our algorithms can be modified to handle many objectives. This might be critical within the calibration framework. Our focus has been on calibration of a single constituent i.e flow. However, multiple constituents could be incorporated into the calibration analysis, resulting in an increase in the number of calibration objectives. Hence, another avenue of future research could include incorporation of many objectives in the calibration analysis, and assessing if any modifications in the proposed algorithms described in this thesis are necessary for improving computational efficiency in order to analyze many objectives.

5.1 References

[1] Keith Beven and Andrew Binley. The future of distributed models: Model calibration and uncertainty prediction, July 1992.

[2] C.A. Coello Coello, G.L. Lamont, and D.A. van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation. Springer, Berlin, Heidelberg, 2nd edition, 2007.

[3] Andreas Efstratiadis and Demetris Koutsoyiannis. One decade of multi-objective calibration approaches in hydrological modelling: a review. *Hydrological Sciences Journal*, 55(1):58–78, February 2010.

[4] Hoshin Vijai Gupta, Soroosh Sorooshian, and Patrice Ogou Yapo. Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information. *Water Resources Research*, 34(4):751, 1998.

[5] David Hadka and Patrick Reed. Borg: An auto-adaptive many-objective evolutionary computing framework. *Evol. Comput.*, 21(2):231–259, May 2013.

[6] J. Knowles. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 8(5):1341–66, February 2006.

[7] J. B. Kollat, P. M. Reed, and T. Wagener. When are multiobjective calibration trade-offs in hydrologic models meaningful? *Water Resources Research*, 48(3):1–19, March 2012.

[8] H Madsen. Parameter estimation in distributed hydrological catchment modelling using automatic calibration with multiple objectives. *Advances in Water Resources*, 26(2):205–216, February 2003.

[9] S.L. Neitsch, J.G. Arnold, J.R. Kiniry, and J.R. Williams. *Soil and Water Assessment Tool Input/Output File version 2005*. US Department of Agriculture Agricultural Research Service, September 2004.

[10] S.L. Neitsch, J.G. Arnold, J.R. Kiniry, and J.R. Williams. *Soil and Water Assessment Tool theoretical documentation version 2005*. US Department of Agriculture Agricultural Research Service, January 2005.

[11] Jasper a Vrugt and Bruce a Robinson. Improved evolutionary optimization from genetically adaptive multimethod search. *Proceedings of the National Academy of Sciences of the United States of America*, 104(3):708–11, January 2007.

[12] P Yapo, H Gupta, and S Sorooshian. Multi-objective global optimization for hydrologic models. *Journal of Hydrology*, 204(1-4):83–97, January 1998.

APPENDIX A

WATERSHED MODEL CALIBRATION THROUGH MULTI-OBJECTIVE OPTIMIZATION

A.1 Introduction

Watershed model calibration is a process that has evolved into a critical exercise in the development of effective deterministic models for prediction and forecasting of watershed behaviors. These models are employed to assess watershed (quality and quantity) response to prevalent and varying management practices and climate patterns [1]. Numerous watershed modeling methodologies are prevalent in the research world and industry today, which include distributed and lumped models, physically based models, stochastic models, deterministic models, event-based models, etc. [25]

Recent advances in computing power have led to development of complex, physically based models to encapsulate finer details of flow and transport behavior at very large spatial scales and smaller time steps. Even with the computing capabilities available to us today, such models can be computationally expensive [39] [29], and incorporate a significant number of parameters [46] [12] which are either impossible to measure, or hard to understand, or both. The process of identification of suitable values for these parameters has been a critical topic of research studies and comes under the realm of model calibration. The calibration process essentially incorporates comparison of model response with historical measured data. In situations where historical data is not available, calibrated parameter values of watersheds with similar physical attributes to the one under consideration, are used as a benchmark for calibration. This study focuses on model calibration with available historical watershed response

data.

"Manual methodologies versus automatic strategies" for watershed model calibration is a topic of immense debate amongst researchers and practitioners. Manual methodologies typically employ expert opinion of hydrologists coupled with a hit and trial scheme [15][4] to ensure that model predictions replicate measured reality as closely as possible. Boyle et al. (2000) [4] highlight the importance and complexity of the process of replicating reality as closely as possible. Manual methodologies employ various techniques simultaneously in the process of replication, for instance goodness of fit measures [17] [31], visual hydrograph analysis, understanding of hydrology, etc. [37]

Automatic calibration strategies employ use of optimization techniques to propose an automatic alternative to the process of replicating measured reality. Goodness of fit measures are typically used as calibration optimization objectives and are often the sole criterion for assessing calibration quality [14]. Since the watershed model calibration problem can essentially be viewed as a simulation-optimization problem, search based algorithms have been dominantly used in development of automatic calibration strategies. The key challenge in the optimization process is tackling computational complexity of simulation models by devising efficient algorithms which produce good solutions within a limited number of simulation evaluations. The Dynamically Dimensioned Search Algorithm proposed by Tolson and Shoemaker (2007) is a computationally efficient search algorithm for automatic watershed model calibration [41].

Historically, single objective optimization techniques have been prevalent in the automatic calibration process. However, the calibration problem itself is inherently a multi-objective optimization problem, where the objectives are

to minimize the absolute difference between each observation and corresponding simulated output, assuming that there is no measurement error. Yapo et al. (1998) [47] and Gupta et al. (1998) [17] highlight the inherent multi-objective nature of a watershed model calibration problem, indicating that a single numerical performance criterion seldom ensures overall closeness of simulated data with measured data. While various performance criteria have been identified as suitable for automatic calibration of watershed models, no single criterion ensures overall optimality in the true sense. Hence, hydrologists prefer a manual approach towards calibration, where the inherent multi-objective nature of the calibration process can be highlighted through an interactive, subjective and complex decision making process. The inherent multi-objective nature of a calibration problem adds new challenges namely 1) numerical formulation of objectives and 2) computational complexity of multi-objective optimization.

While a manual methodology has the advantage of use of visual and hydrological analysis, flexibility, and availability of expert opinion in the process of calibration, it can be cumbersome, time-consuming and requires expert human interaction. On the other hand, automatic techniques are computationally efficient and may not require human interaction in the calibration process. Quality of single objective-based automatic strategies, however, relies heavily on the calibration optimization objective which might not be desirable. Multi-objective strategies can bridge the gap between automatic and manual calibration techniques [27] [4], but may be limited by computational complexity of simulation models [39], and the added need of a strategy for choosing a parameter combination from a set of equally optimal parameter combinations.

The aim of this study is to analyze and understand some of the major, aforementioned challenges of calibrating computationally expensive hydrological

models involving numerous unmeasurable parameters, and subsequently, to propose a generic calibration strategy to counter the identified major challenges. We proposed a "Hybrid Automatic Manual Strategy" (HAMS) for watershed model calibration. The proposed strategy focuses on calibration via a multi-objective optimization framework. A novel efficient surrogate model based optimization algorithm is employed for multi-objective optimization, and a novel technique for selection of a parameter combination from various equally optimal combinations, is proposed.

A.2 Calibration Problem Description

Calibration of any models is an effort in tweaking unmeasurable and /or unknown parameters of a model, such that model predictions are as close to reality as is possible. A popular way employed in the process of calibrating a model is to involve historical measurements in order to guide the process of calibration. While recent advances in research have led to development of highly sophisticated watershed models, completely accurate replication of reality has not been achieved. Errors within models originate from assumptions made during model development and through errors in model input. If historical measurements are employed to guide the model calibration process, one should also be aware of the existence of measurement errors.

In this analysis, our focus is on calibration of deterministic watershed flow models, meaning the model output is fixed for a given input. Additionally, we do not consider measurement error in the calibration process, and hence assume that historical measurements depict reality.

Let $\mathcal{D}$ (domain space) be a hypercube and a subset of $\mathbb{R}^d$ and $\theta \in \mathcal{D}$, where $\theta = \{\theta_1, \ldots, \theta_d\}$ represents the d model parameters that require calibration. Let

$Y = \{y_1, \ldots, y_n\}$ be the vector that denotes observed watershed responses measured at one location and n different times. Let $\widehat{Y}(\theta) = \{\widehat{y_1}(\theta), \ldots, \widehat{y_n}(\theta)\}$ be the vector that denotes the corresponding response of the simulation model. Since measurement error is not considered in the parameter calibration process, the calibration problem can be depicted as follows:

$$\{\theta^* \in \mathcal{D} : \widehat{Y}(\theta^*) \approx Y\} \tag{A.1}$$

"$\approx$" is a measure of closeness of simulated output to measured output, and consequently indicates *model performance* against historical measured information. Hence, based on the discussed assumptions, the focus of our calibration study is to deduce a parameter combination or a set of parameter combinations such that Equation A.1 is satisfied.

A.2.1 The Manual Calibration Process

Boyle et. al. (2000) [4] define manual calibration as a 3-stage process. Stage-0 incorporates identification of prior uncertainty bounds on the model parameters to be calibrated. Stage-I involves refinement of model parameter ranges via assessing effects of individual parameters, on *model performance*. This assessment typically entails a trial and error scheme, where parameters are individually tweaked to re-assess *model performance* and its sensitivity to individual parameters. Stage-II employs parameter interactions within a more sophisticated and complex decision framework to further refine parameter ranges and further improve *model performance*.

The manual calibration process essentially aims at solving the problem depicted in equation A.1. However, manual calibration is subjective and varies according to the preferences of the calibration expert. Calibration experts make

use of information regarding the proposed use of the model, a detailed understanding of the model, and various techniques for evaluating *model performance* (for instance visual hydrograph analysis, goodness of fit measures, etc.) to interactively adjust parameter values.

While the manual calibration process is thorough and interactive, it can be highly inefficient and requires high level human expert opinion which might not be readily available. The time constraint in the calibration process has emerged as a critical issue in recent history due to the advent of highly sophisticated distributed and semi-distributed models. Depending on the scale and complexity of the underlying watershed, the computational cost of a simulation, using the new and improved models, can be extremely high (order of minutes and hours). Hence, it is desirable that the calibration process produce good results within a limited number of simulation evaluations.

A.2.2 Automatic Calibration

Automatic calibration techniques assume logical equivalence of the "$\approx$" measure (equation A.1) to a goodness of fit measure, in order to convert the closeness of simulated output to measured data, into a single quantifiable performance measure. Subsequently it is assumed that:

$$\widehat{Y}(\theta) \approx Y := \max_{\theta} f(\theta) \tag{A.2}$$

where $f(\theta)$ is a goodness of fit measure prescribed for the automatic calibration process. Automatic optimization techniques have emerged as an alternate to the manual calibration process, where sophisticated search methods can be employed to calibrate a model to a predefined objective, i.e., $f(\theta)$. Various single objective optimization algorithms have been used in prior research as well as

in industry either to assist in, or serve as an alternative to, manual calibration [18]. Evolutionary search algorithms, simulated annealing and tabu search [9] are some of the many algorithms that have been used for automatic calibration of watershed flow models through single objective optimization. Considerably efficient algorithms have been proposed recently in order to tackle the computational intensity of contemporary simulation models. [42] proposed the Dynamically Dimensioned Search algorithm for efficient calibration of parameters of large watershed models. Surrogate model based optimization algorithms have also been proposed to further speed up the automatic calibration process [19] [36].

Since single objective techniques are heavily dependent on the objection function chosen, practitioners generally prefer manual techniques. However, some prior research indicates better performance via automatic single objective techniques than manual calibration strategies [24]. While single objective automatic calibration has been very popular in past literature, recent studies have also highlighted the advantages of using them iteratively to replicate the manual calibration process [20].

A.2.3 Multi-Objective Calibration Decision Analysis

Despite all the advantages of automatic techniques, e.g., their ease of use, computational efficiency, and ability to produce results without high level knowledge from hydrologists, manual calibration is the preferred methodology in industry today. Since hydrologists consider various aspects / criteria during the calibration process, automatic strategies especially single-objective automatic strategies are not preferred by many. While automatic optimization strategies have the advantage of finding near-optimal solutions for a particular objective

within a limited budget of simulation evaluations (a methodology which might be considerably efficient compared to hit and trial techniques employed in manual calibration), they only consider a single objective in deducing a parametric combination during the calibration process.

Various authors ([13], [47] and [17]) highlight the inherent multi-objective nature of a calibration problem, emphasizing that one particular parameter combination might not be the optimal calibration for a particular model. Gupta et al. (1998) [17] aptly identify that a model calibration problem as prescribed by equation A.1 is essentially a multi-objective optimization, where, ideally, the calibration process aim is to subsequently minimize the difference between each measured observation and corresponding simulation response. Equation A.3 provides a mathematical representation of the "ideal" equivalence (signified by "≡") of the calibration problem to multi-objective optimization.

$$\{\theta^* \in \mathcal{D} : \widehat{Y}(\theta^*) \equiv Y\} \;:=\; \min_{\theta} \{\, |y_1 - \widehat{y_1}(\theta)| \,, \ldots, \, |y_n - \widehat{y_n}(\theta)| \,\} \tag{A.3}$$

Assuming that a simulation model can replicate reality accurately and accurate historical data is available (assuming there is no measurement error), solving the "ideal" multi-objective problem depicted in equation A.3 could produce the optimal calibration. However, watershed simulation model development is based on various assumptions which tend to simplify natural phenomena, and it is highly unrealistic to assume existence of a single optimum solution of the "ideal" multi-objective problem. Multi-objective optimization problems tend to produce a set of equally optimal solutions upon solving. This solution set is also referred as a *pareto optimal* set or a set of *trade-off* solutions. Let P^* be the *pareto optimal* set for the "ideal" multi-objective calibration problem:

$$P^* = \{\theta \in \mathcal{D} : \min_{\theta}\{\, |y_1 - \widehat{y_1}(\theta)| \,, \ldots, \, |y_n - \widehat{y_n}(\theta)| \,\}\} \tag{A.4}$$

It is safe to assume that P^* realistically contains more than one *pareto optimal* solution. "Ideal" equivalence of the calibration problem to multi-objective optimization, as depicted in equations A.3 and A.4, signifies the immense advantage of incorporating multi-objective optimization in the watershed calibration process. However, the "ideal" multi-objective optimization problem typically incorporates a large set of objectives, and hence cannot practically be solved, even via use of efficient optimization techniques (discussed further in section A.5).

The "ideal" multi-objective calibration can be subjectively re-defined to considerably reduce the number of competing calibration objectives. If the number of objectives is reduced to one, the methodology essentially becomes an automatic scheme, as depicted in equation A.2. Even with a subjective redefinition of the "ideal" calibration formulation, multi-objective optimization can be critical in visualization of trade-offs between competing calibration objectives, and assist practitioners in choosing a calibration combination accordingly. Bekele et al. (2007) [2] indicate that a particular calibration combination might be optimal or near optimal according to a particular objective (e.g., peak events), but might be less than satisfactory for other objectives (e.g., drought events), signifying the underlying trade-off between two objectives.

A well established calibration consideration amongst hydrologists is the proposed application of a model (for instance prediction of total pollutant loading at a point). Madsen et al. (2003) [27] emphasize the advantage of using multi-objective analysis for calibration according to the proposed application of the model. This consideration is the primary focus within the prevalent manual calibration methodology. The importance of using multi-objective optimization in bridging the gap between automatic and manual techniques for calibration

has been further highlighted by [4]. Khu et al. (2008) [23] propose strategies for incorporating multi-objective optimization into calibration of watershed models involving measurement information at multiple sites.

The existence of multiple *trade-off* solutions for the "ideal" multi-objective calibration problem reveals the inherent structural ambiguity of the simulation model. Gupta et al. [15] highlight the use of multi-objective optimization in quantifying structural ambiguity of model predictions, which can be critical to the eventual selection of a calibration combination. Tolson and Shoemaker [43] highlight the concept of "equifinality" in the calibration process, where it is noted that various calibration combinations could be equally good. The "equifinality" concept is further used to quantify uncertainty in model response. Vrugt et al. [44] emphasize the importance of measurement error in uncertainty quantification, highlighting the importance of parametric uncertainty in the process. Muleta et al. employ the GLUE methodology to quantify model uncertainty [30].

While multi-objective optimization can provide added trade-off information (in comparison with automatic single objective techniques) to practitioners during the process of calibration, effectiveness of the multi-objective approach is highly dependent on various factors. Some of these include: 1) choice of objectives in the multi-objective problem formulation, 2) choice and efficiency of the optimization algorithm, and 3) a methodology for choosing an appropriate calibration combination from alternatives obtained via multi-objective analysis. Prior research methodologies focus on some of these aspects separately. For instance Fenicia et al. (2007) [11] emphasize use of a step-wise, single-objective optimization approach and compare with a multi-objective optimization approach, emphasizing that the multi-objective approach can help in highlighting

structural uncertainty of the underlying model. We propose a Hybrid Automatic Manual Strategy (HAMS), which employs multi-objective optimization within a hybrid calibration strategy.

A.2.4 A Hybrid Automatic Manual Strategy - HAMS

Posteriori [7] Multi-objective strategies consider multiple objectives in an automatic optimization process and can provide a set of solutions which are equally optimal as per the objectives specified (quantification of objectives is the critical process). However, there is an added challenge of deciding what calibration combination amongst the pareto optimal solutions is best suited for a particular model and its proposed use. In this paper we propose a hybrid multi-objective strategy that can be employed to 1)Deduce a set of parameter combinations suitable for a particular model, and 2) Decide upon a single parameter combination as the best possible calibration for a particular model, a decision contingent upon the proposed use of the model and the results obtained through the multi-objective optimization process. This strategy aims at bridging the gap between automatic and manual methods for calibration and is called HAMS. Figure 1 provides a flow description of the proposed strategy.

As depicted by Figure A.1, HAMS is a sequential methodology which is initiated by 1) formulating a calibration problem in a multi-objective framework, 2) efficiently solving the multi-objective calibration problem via surrogate based optimization and 3) using an interactive analysis as well as an expert analysis (automatically mimicking manual calibration) for choosing a calibration combination from various alternatives identified by multi-objective optimization. Sections A.5 - A.8 discuss HAMS in detail and describe how the strengths of automatic and manual calibration are incorporated into the Hybrid Automatic

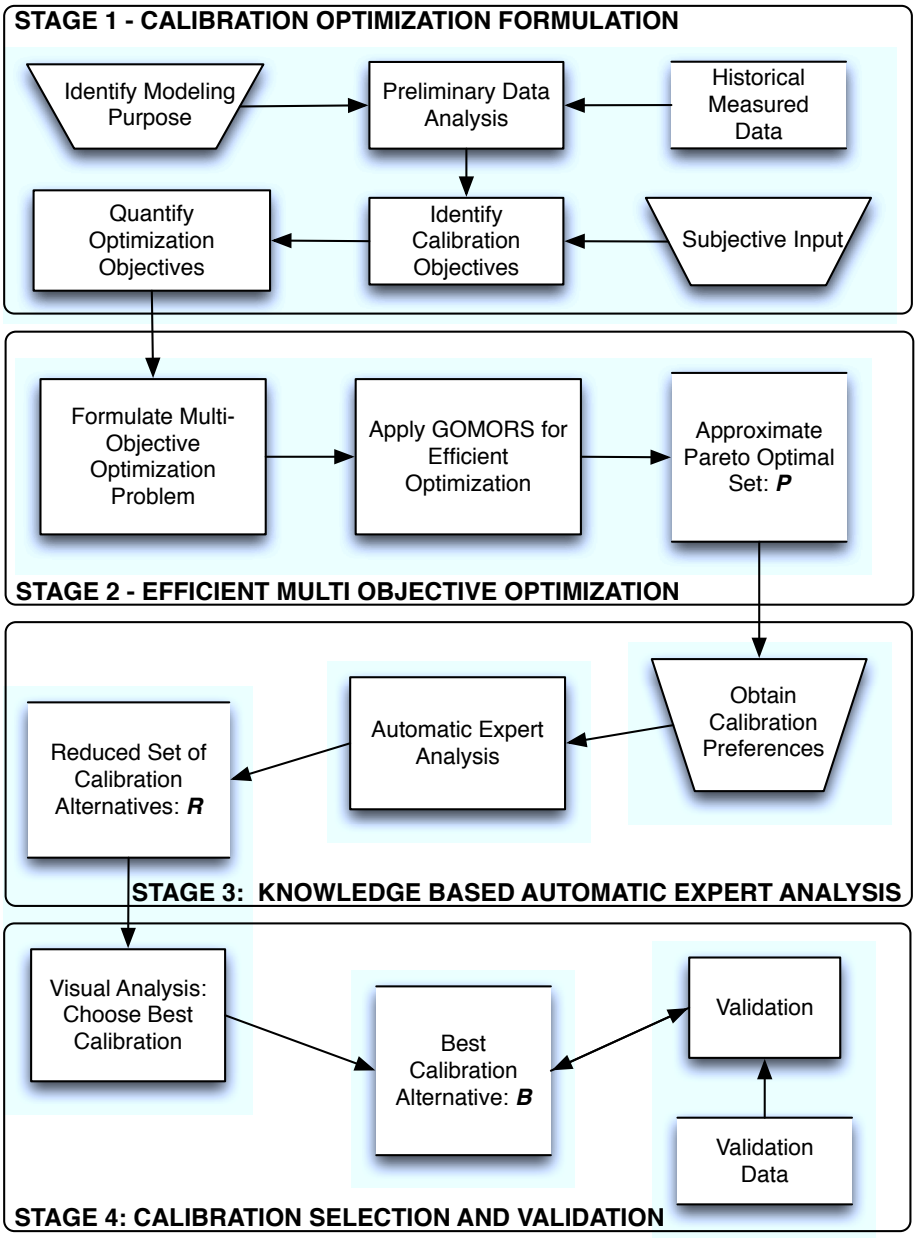

Figure A.1: Flow chart of the Hybrid Automatic Manual Scheme for Watershed model calibration

Manual Strategy for watershed flow calibration of the Cannonsville Reservoir SWAT model, developed to simulate flow, sediment and phosphorus transport, to the Cannonsville Reservoir. Results are discussed during discussion of the hybrid strategy. Sections A.3 and A.4 briefly discuss the case studies used in

analyzing the proposed calibration strategy.

A.3 Case Study I: Cannonsville Watershed

[25] provides a brief introduction to the different types of watershed modeling methodologies prevalent in the research world and industry today, which include distributed and lumped models, physically based models, stochastic models, deterministic models, event-based models, etc.

Various authors [35] [22] discuss the varying nature of the calibration process for lumped, distributed and semi-distributed models. Distributed models tend to involve a large set of parameters that might need calibration. Considerable work has been done to identify sensitive parameters in distributed models, for an effort to reduce parametric dimensionality. Werkhoven et al. (2009) [46] discuss a methodology for reducing parametric dimensionality via sensitivity analysis.

The Cannonsville Watershed SWAT Model is the primary case study used in analyzing the HAMS calibration strategy. Tolson and Shoemaker (2007) [40] discuss the development of the Cannonsville Watershed Model through the Soil and Water Assessment Tool (SWAT).

The Soil and Water Assessment Tool is a physically based, deterministic, semi-distributed watershed model [33] and [32]. The aim of modeling the Cannonsville Watershed through SWAT is to predict phosphorus loading into the Cannonsville Reservoir, which is a direct water supply source for New York City. The modeling prediction exercise through SWAT is carried out to forecast phosphorus loading into the watershed sink and introduce best management practices if the forecast indicates significant deterioration in watershed quality. In this study our focus is only on the calibration of flow parameters of the model.

A preliminary sensitivity analysis of the model flow parameters is discussed in [40] and the final number of parameters to be calibrated is 15.

A.4 Case Study II: Townbrook

Townbrook is one of 43 sub-basins of the Cannonsville Watershed in upstate New York, also modeled via SWAT. The primary aim of proposing new best management practices for the Townbrook and Cannonsville Watersheds is reductions of pollutant loading at the least cost. Hence, the primary aim of a SWAT model that can assist in analyzing alternate BMPs is good prediction of pollutant loading. Our aim in this study is to calibrate the two models for flow, since flow predictions dictate pollutant loading predictions. The advantages and importance of introducing new BMPs are highlighted by [13]. Since the Townbrook model is relatively inexpensive, in terms of computational cost (computational time of a single run is in the order of seconds), we use the Townbrook model to compare the efficiency of our optimization strategy in Stage-II of HAMS (see Figure A.1), to various other algorithms.

A.5 Stage I: Calibration Optimization Formulation

The aim of Stage 1 (see Figure A.1) of the Hybrid Automatic Manual Strategy for watershed flow calibration is to formulate the calibration problem as a multi-objective optimization problem. As mentioned previously, the "ideal" multi-objective optimization problem depicted in equation A.3 typically involves a large number of objectives (equal to the number of observations, i.e., n) which could potentially be highly correlated. Due to the dimensional complexity of calibration problems and computational complexity of simulation models, the

"ideal" multi-objective problem is virtually unsolvable. Additionally, multi-collinearity between competing objectives make the "ideal" problem undesirable [17]. Since the "ideal" problem is practically unsolvable, we can subjectively choose potentially unrelated calibration criteria to formulate a reduced multi-objective calibration optimization problem. Based on a subjectively defined multi-objective formulation, we can assume logical equivalence to the "$\approx$" measure in equation A.1:

$$\{\theta^* \in \mathcal{D} : \widehat{Y}(\theta^*) \approx Y\} \;:=\; \min_{\theta} \{f_1(\theta), \ldots, f_k(\theta)\} \tag{A.5}$$

Once the purpose or planned use of a model is known, hydrologists focus on various aspects / criteria in the model calibration process, which typically include: 1) Volume Flow Calibration, 2) Base Flow Calibration, 3) Peak Flow Calibration, 4) Hydrograph Shape Calibration. There are many subjective and objective methods that have been used in prior research and in practice to analyze the quality of calibration, and calibrate according to the above-mentioned objectives. Objective metrics for quality analysis include statistical criteria, for instance R squared value, the Nash-Sutcliffe Efficiency [31], Bias, mean absolute error, etc. [17], and can all be used as objective functions in single or multi-objective optimization algorithms. Subjective criteria are employed, especially, to calibrate models for shape and to capture extreme events more accurately.

A.5.1 Threshold Based Formulation

In this analysis we investigate three multi-objective calibration formulations, the first one being "threshold based formulation". In this formulation, the measured data is analyzed in order to segregate the observations based on thresholds.

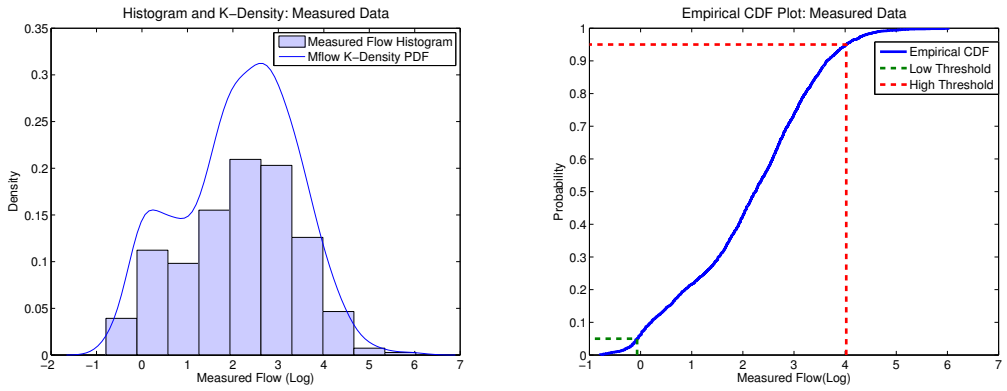

Figure A.2: Cannonsville Measured Data Distribution and ananlysis

Figure A.2 depicts the distribution and empirical CDF of the measured data available for calibration of the Cannonsville Watershed Flow model. The measured data can be divided into various categories (for instance peak flow observations, low flow observations, etc.), upon subjective identification of thresholds, as depicted in the right side of figure A.2. By dividing the data into various categories, we can devise various objectives, for our multi-objective optimization formulation.

$$
\begin{aligned}
f_1(\theta) &= \sum_{i \in N_1} [y_i - \widehat{y_i}(\theta)]^2 \text{ , where } N_1 = \{i \mid y_i \leq y_{max}, 1 \leq i \leq n\} \\
f_2(\theta) &= \sum_{i \in N_2} [y_i - \widehat{y_i}(\theta)]^2 \text{ , where } N_2 = \{i \mid y_{max} \leq y_i, 1 \leq i \leq n\}
\end{aligned}
\tag{A.6}
$$

Equation A.6 describes the objectives used in the first formulation (Formulation 1) proposed for HAMS. The two objectives in this formulation are ”1) Peak Flow Sum of Squared Errors” and ”2) Other Flow Sum of Squared Errors”. This formulation is not used in the HAMS analysis, but has only been used to briefly demonstrate the efficiency of the novel multi-objective algorithm used in stage-II of the proposed calibration strategy.

For the second formulation, the measured data is divided into three cate-

gories. The first category is low flows, second is medium flows and third is high flows. Since the high flows significantly impact overall error (if a single objective sum of squared formulation is employed), and low flow calibration is critical to overall volume errors, separating these flows through thresholds allows the performance of the model within each flow range to be considered as a separate objective in a multi-objective optimization problem. Figure A.2 depicts the subjectively identified thresholds obtained in this analysis, from the empirical distribution of the measured data for the Cannonsville Model. Consequently, a three objective threshold based formulation is devised and is depicted in equation A.7.

$$
\begin{aligned}
f_1(\theta) &= \sum_{i \in N_1} [y_i - \widehat{y_i}(\theta)]^2 \text{ , where } N_1 = \{i \mid y_i \leq y_{min}, 1 \leq i \leq n\} \\
f_2(\theta) &= \sum_{i \in N_2} [y_i - \widehat{y_i}(\theta)]^2 \text{ , where } N_2 = \{i \mid y_{min} \leq y_i \leq y_{max}, 1 \leq i \leq n\} \\
f_3(\theta) &= \sum_{i \in N_3} [y_i - \widehat{y_i}(\theta)]^2 \text{ , where } N_3 = \{i \mid y_{max} \leq y_i, 1 \leq i \leq n\}
\end{aligned}
\tag{A.7}
$$

A.5.2 Decomposition of NSE

The third calibration formulation (Formulation 3) discussed in our analysis is inspired by the work done by Gupta et al. (2009) [16] in identifying various components of the Nash Sutcliffe Efficiency (NSE) [31] criterion. As per the analysis of Gupta et al. [16], the NSE criterion can be divided into three components, 1) relative bias, 2) relative variability, and 3) linear correlation. The relative bias component of the criterion tends to minimize volume balance errors, the relative variability tends to mimic the flashiness of the hydrograph, inherently focusing on capturing extreme flows, while the correlation criterion, in combination with relative variability, tends to capture the shape of the hydrograph.

As is mentioned by Gupta et al. [16], there is an inherent disadvantage of

using all three criteria as a combination in the NSE component. Since the NSE component describes relative bias as a fraction of measured data variability, using the NSE as a single objective gives less weight to the objective of minimizing model bias, potentially resulting in significant volume errors. Gupta et al. [16] also show that within the NSE metric relative variability is always bounded by the correlation, which is always less than unity, and hence, NSE tends to underestimate the variability of flows. While the KGE metric proposed by Gupta et al. [16] can be used as an alternative, to automatically calibrate the model via single objective optimization, it devoids the calibration process of analysis of the interplay between these objectives, which can potentially be conflicting. The advantage of using this methodology is that one does not need to define thresholds as in Formulations 1 and 2. The mathematical description of the three objectives for Formulation 3 are as follows:

$$\begin{aligned} f_1(\theta) &= [r-1]^2 \text{ , where } r = \frac{\sum_{i=1}^{N}(y_{obs,i}-\bar{\mu}_{obs}) * (y_{sim,i}-\bar{\mu}_{sim})}{\sqrt{\sum_{i=1}^{N}(y_{obs,i}-\bar{\mu}_{obs})^2 * \sum_{i=1}^{N}(y_{sim,i}-\bar{\mu}_{sim})^2}} \\ f_2(\theta) &= [\alpha-1]^2 \text{ , where } \alpha = \bar{\sigma}_{sim}/\bar{\sigma}_{obs} \\ f_3(\theta) &= [\beta-1]^2 \text{ , where } \beta = \bar{\mu}_{sim}/\bar{\mu}_{obs} \end{aligned} \tag{A.8}$$

Table A.1 briefly describes the multi-objective optimization calibration problems used in our analysis. Note that TBROOK1 is only used to briefly depict the relative efficiency of the multi-objective optimization algorithm used in stage 2 of HAMS.

A.6 Stage II: Efficient Multi-Objective Optimization

Prior research has primarily focused on using evolutionary algorithms for multi-objective calibration of watershed models [38] [34]. Tang et al. (2006) [39] assess the effectiveness of evolutionary algorithms towards multi-objective

Table A.1: List of multi-objective problems used in this paper for analysis of HAMS

Problem Name	Description
TBROOK1	Formulation 1 applied to the town brook case study
TBROOK2	Formulation 2 applied to the town brook case study
CVILLE1	Formulation 1 applied to the Cannonsville case study
CVILLE2	Formulation 2 applied to the Cannonsville case study
CVILLE3	Formulation 3 applied to the Cannonsville case study

calibration of distributed watershed models with large parameter sets. While the use of evolutionary algorithms for multi-objective calibration of distributed watershed models has been extensively explored, surrogate model-based algorithms have not been tested extensively in this regard [21]. It is highlighted in various texts [39], that use of evolutionary strategies could be computationally very expensive and is largely dependent on the calibration problem at hand. Surrogate model-based algorithms could significantly reduce the computational complexity of the multi-objective optimization process. HAMS introduces the use of GOMORS, which is a surrogate model-based multi-objective optimization algorithm aimed at finding good approximations to the Pareto front within a limited budget of simulations of a complex, distributed watershed model. We further discuss the algorithm in this section and also discuss results of GOMORS in comparison with AMALGAM [45] for the TBROOK1 problem formulation described in Table A.1.

The proposed strategy, GOMORS is iterative, and the response surface model, which is used to approximate the costly objective functions, is updated after each iteration. Radial Basis Function Approximation [5] [6] is employed

for development of the surrogate model / response surface model. One point is selected for function evaluation in each iteration and subsequently used to update the response surface model. Selection of the evaluation point is based on a trade-off between three criteria, namely 1) selected point should not be very close to already evaluated points, 2) selected point should be close to or amongst the points that form a near-optimal front on the response surface model, and 3) selected point should be picked such that unexplored areas on the response surface model close to the near-optimal front are explored.

GOMORS consists of 4 major segments: 1) Picking initial sample points, 2) Response Surface Model, 3) Evolutionary algorithm for solving multi-objective problem and 4) Picking additional points from estimated pareto front established on approximated functions . The algorithm output is a non-dominated set of solutions P^{best} which is an approximation to the true pareto solution of the problem i.e P^{*}. The algorithm framework allows for picking multiple points at one instance for simultaneous simulation evaluation. Hence, a modestly parallel version (involving up to 4 cores) of the algorithm was developed particularly for this study to further improve algorithm efficiency.

A.6.1 Algorithm Comparison

Table A.1 provides a list of case studies / test problems used in the analysis of HAMS. Stage-II of the strategy employs GOMORS, an efficient strategy for multi-objective optimization of the calibration formulation developed in Stage-I. In this section we analyze the algorithmic efficiency of GOMORS by assessing its application to TBROOK1 (see Table A.1). Furthermore, performance of GOMORS is compared against AMALGAM and NSGA-II. AMALGAM is a hybrid algorithm proposed by Vrugt et al. [45] as a relatively efficient multi-

method evolutionary strategy. NSGA-II [8] is a well known and widely used evolutionary algorithm for multi-objective optimization.

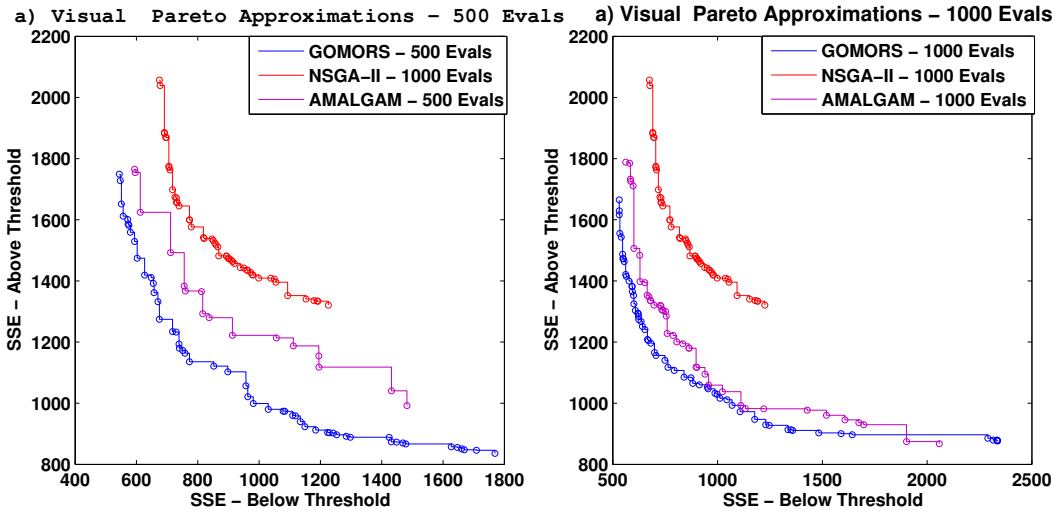

Figure A.3: Visual Pareto Approximation Comparison - GOMORS, AMALGAM and NSGA-II: a) Comparison for TBROOK1 problem after 500 simulation evaluations, b) Comparison for TBROOK1 problem after 1000 simulation evaluations

We performed 10 test runs of each of the algorithms for TBROOK1. Each test run was terminated after 1000 simulation evaluations. Figure A.3 provides a visual comparison of the best fronts attained for each algorithm after 10 test runs (front quality was gauged via Inverse Generational Distance (IGD) metric analysis). Algorithm performance is also visually compared after completing 1000 simulation evaluations. Figure A.3 clearly indicates that GOMORS converges to a better set of non-dominated solutions than both AMALGAM and NSGA-II after a limited number of evaluations. AMALGAM tends to perform better than NSGA-II; howeve,r it is relatively inefficient in comparison to GOMORS for the TBROOK1 test problem.

The relative inefficiency of AMALGAM in comparison to GOMORS can be

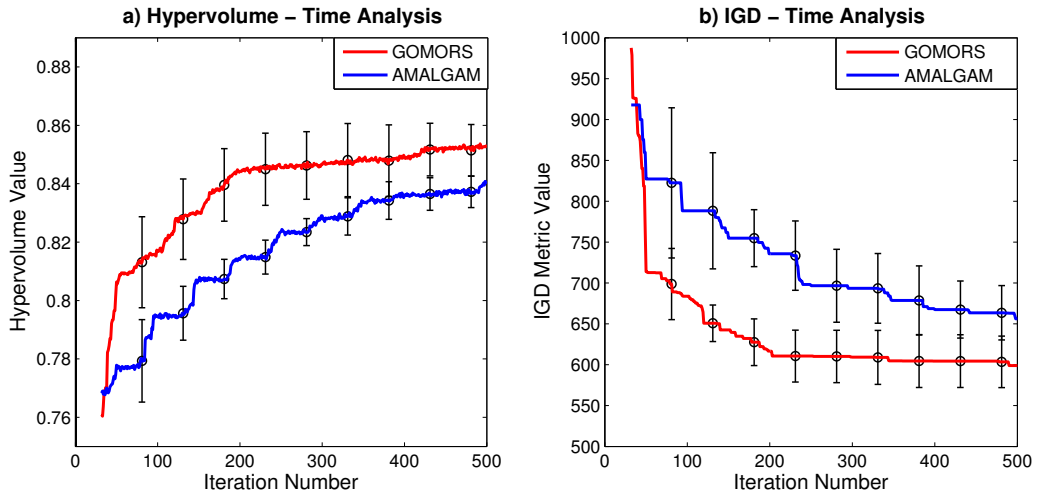

Figure A.4: Time Analysis via pareto set performance metrics - GOMORS and AMALGAM a) Hypervolume Metric Comparison for TBROOK1 problem after 500 simulation evaluations, b) IGD metric Comparison for TBROOK1 problem after 500 simulation evaluations

further illustrated through a progress plot, where the performance metric value obtained for an algorithm is plotted against the number of function evaluations. In this analysis we employ the IGD metric and the hypervolume index for a time comparison of the quality of different algorithms. Note that a lower value of the IGD metric is desired and a high value of the hypervolume index is desired. Figure A.4 provides a visual summary of the progress plot analysis of GOMORS and AMALGAM. The red and blue lines in each subplot indicate metric values averaged over 10 test runs for each algorithm. The error bars correspond to the variation in each quality metric estimate. Figure A.4 clearly indicates that GOMORS converges quickly to a better solution, within a limited evaluation budget of 500 simulations, as per the performance metrics used in this analysis.

A.6.2 Parametric Sensitivity Analysis

In the previous section, we established the relative efficiency of GOMORS, by testing the algorithm on TBROOK1 and CVILLE1 (see table A.1). These test problems were primarily established to compare algorithm performance. In this section we focus on analyzing the results of Stage-II of the HAMS strategy, i.e., the multi-objective optimization. Our analysis focus is on Case Study - I: flow parameter calibration of the Cannonsville Reservoir SWAT model. Furthermore, we analyze multi-objective optimization results for two formulations discussed previously, CVILLE2 and CVILLE3. GOMORS was applied to the two different formulations and algorithm was terminated after 500 simulation evaluations.

The idea behind stage-II of HAMS is to produce a set, P, of alternative calibration combinations which are equally optimal (or non-dominated), via multi-objective optimization. The set, P, is generated through application of GOMORS. A visual analysis of parameter interactions and trade-offs between competing objectives for the CVILLE2 problem is depicted in Figure A.5.

The figure depicts very interesting trade-offs between competing objectives. Note that the multi-objective formulation for CVILLE2 is the three objective threshold based formulation depicted in equation A.7, which lumps the observations into categories of low flow, average flow and peak flow observations. Figure A.5 indicates that while there is a significant trade-off between the sum of squared errors (SSE) of low flow and average flow observations, and peak flow and average flow observations, a significant trade-off between peak flow SSE and low flow SSE does not exist. Since we use historical rainfall information (indicating that rainfall errors might not be significant) as simulation input during calibration, a better low flow calibration could essentially lead to better prediction of peak flows as well. Hydrologists tend to fit base flow or low flow

observations to ensure better encapsulation of variation in flow hydrology.

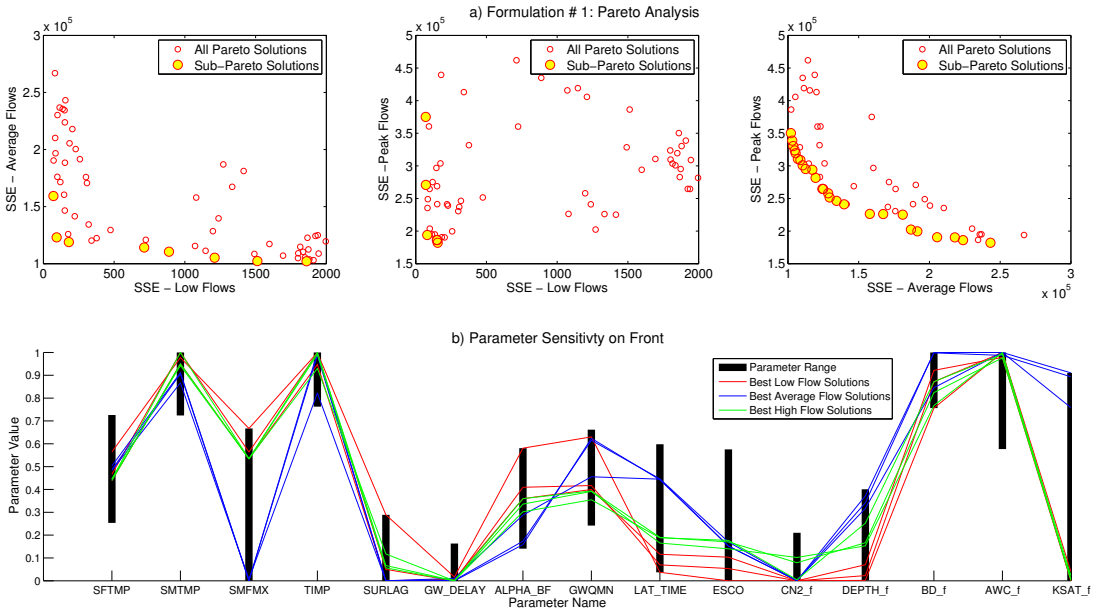

Figure A.5: Visual Pareto Analysis and Parametric Sensitivity Analysis: Formulation CVILLE2 - Cannonsville Reservoir SWAT Model

Figure A.5(b) depicts the range of parameters on the optimal front. Additionally, the figure depicts parameter calibration combinations for extreme solutions on the optimal front. For instance the green lines in Figure A.5(b) depicts parameter combination values for best peak flow solutions obtained via GOMORS. Figure A.5(b) highlights a significant interplay between between the parameters LATTIME and SMFMX when different parts of the hydrograph are being fitted. LATTIME affects recharge from groundwater, and hence can be critical in fitting low flow events. SMFMX on the other hand, affects snowmelt and can be critical in fitting other flow regimes. Note that while parameter values of SMFMX and LATTIME for best low flow and peak flow calibration are similar, corresponding values for best average flow calibrations are significantly different. This interplay could be further explored by hydrologists to assist in

the calibration process.

The CVILLE3 problem, which incorporates the decomposed objectives of NSE proposed by Gupta et al. [16] was also analyzed via HAMS. Figure A.6 provides a visual depiction of the set, P, of alternative calibrations obtained via HAMS stage-II application of multi-objective optimization to CVILLE3. Figure A.6(a) is a visualization of trade-offs between competing objectives. A significant trade-off exists between correlation error and bias, and correlation error and relative variability, indicating that a correlation-based calibration of the model could result in relatively high volume balance and variability errors. The variability errors could indicate poor peak flow and low flow calibration. Hence, Stage-II of HAMS provides important information to hydrologists in choosing a parameter set for their desired model application.

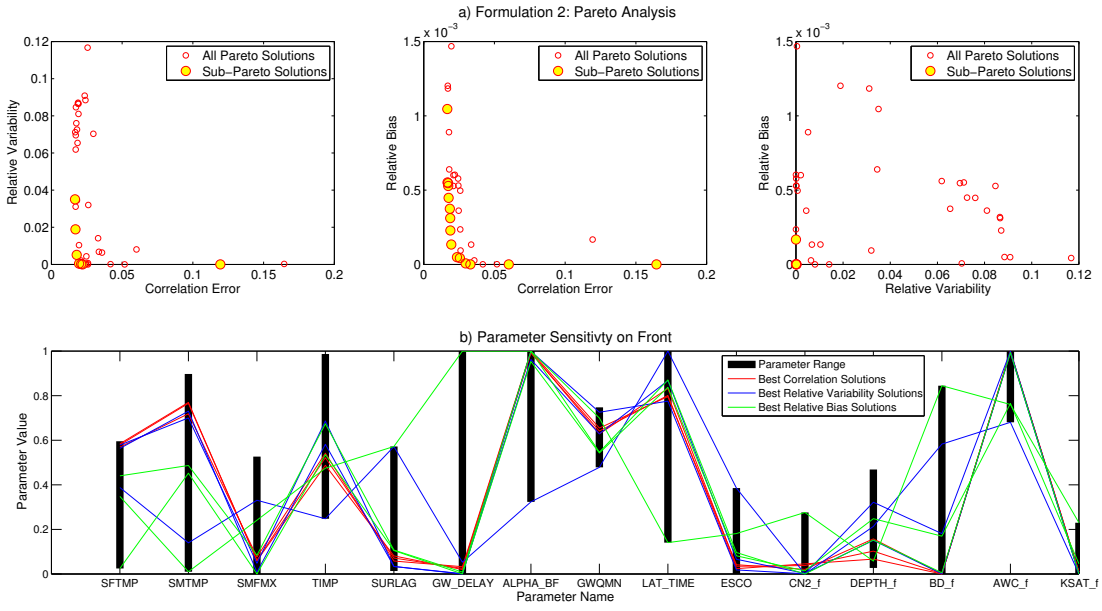

Figure A.6: Visual Pareto Analysis and Parametric Sensitivity Analysis: CVILLE3 - Cannonsville Reservoir SWAT Model

Figure A.6(b) depicts the range of parameter values in the approximately op-

timal set of solutions generated by Stage-II multi-objective optimization applied to the CVILLE3 problem formulation (the NSE decomposition). Furthermore, the figure provides a view of parameter combinations which are best for each individual objective of the CVILLE3 formulation (3 best combinations for each objective are visualized). The figure indicates that different parameter combinations seem to produce good volume balance or bias (illustrated by the green lines), and consequently shows that different parameter combinations can provide reasonable goodness of fit values for a particular objective.

Stage-II of HAMS provides a set of parameter combinations, P, which are equally good (non-dominated) as per a prescribed multi-objective calibration formulation developed in Stage-I. The application of GOMORS to CVILLE2 produces 60 alternative parameter calibrations, while application to CVILLE3, produces 39 alternative parameter calibrations. In an effort to tackle the calibration problem, as depicted by equation A.1, Stage-II of HAMS produces a set of reasonable alternatives, P, which could satisfy the "$\approx$" measure of closeness (referred in Equation A.1) in a subjective sense. However, in a more practical setting, a single parameter combination might be desired. Stages III and IV of HAMS, discuss a methodology for selecting one calibration combination from P.

A.7 Stage III: Reduction of Alternates

The selection of a single parameter combination from a set of alternatives obtained via multi-objective optimization is a critical topic which has been explored in previous research studies. Dumedah et al. (2010) [10] make use of a clustering methodology to automatically choose a solution from a set of optimal alternatives. Shrestha et al. (2008) [34] make use of fuzzy preference selection,

in conjunction with multi-objective optimization, to completely automate the selection process of a calibration parameter from a set of non-dominated solutions. Some studies prefer a manual approach to the selection of an alternative, where trade-offs between competing objectives are visualized and an alternative is chosen via subjective preference. Madsen et al. (2000) [26] suggest a manual selection procedure, based on knowledge of purpose or proposed use of model. This knowledge is critical in the selection process. For instance, if the model purpose is to eventually predict pollutant loading at specific locations in the watershed, the primary aim of flow calibration would be to minimize volume balance errors.

Given the importance of model purpose, Stage-III of HAMS employs two different strategies for reducing the number of reasonable alternatives. Reducing the number of alternatives (rather than selecting one) provides a calibration expert with a small set of alternatives, after which a visual calibration analysis can be performed on a manageable set of alternatives to select a suitable parameter combination.

A.7.1 Interactive Decision Support System

Coon et al. (2008) [37] provide a practitioners' methodology for manual calibration of watershed flow parameters. After an understanding of model purpose is established, a practitioners' calibration strategy typically aims at focusing on the following criteria in the model calibration process: 1) Volume Calibration, 2) Base Flow Calibration, 3) Peak Flow Calibration and 4)Hydrograph Shape Calibration. As highlighted by Boyle et al. (2000) [4], an intelligent and interactive hit and trial methodology can be employed in the manual calibration framework to satisfy various calibration criteria simultaneously. We propose the use

of an Interactive Graphical User interface to mimic the interactive process followed by practitioners in Stage-III of HAMS. This version of HAMS is called HAMS-IS.

HAMS-IS aims at reducing the set of alternative, P, to a smaller (sub)set, R, through an interactive elimination process. As defined earlier, manual flow calibration focuses on 1) volume balance, 2) base flow calibration, and 3) peak flow calibration. We use the same strategy within an interactive framework to reduce the set, P, of solutions obtained in Stage-II of HAMS, to a smaller set of alternatives, R.

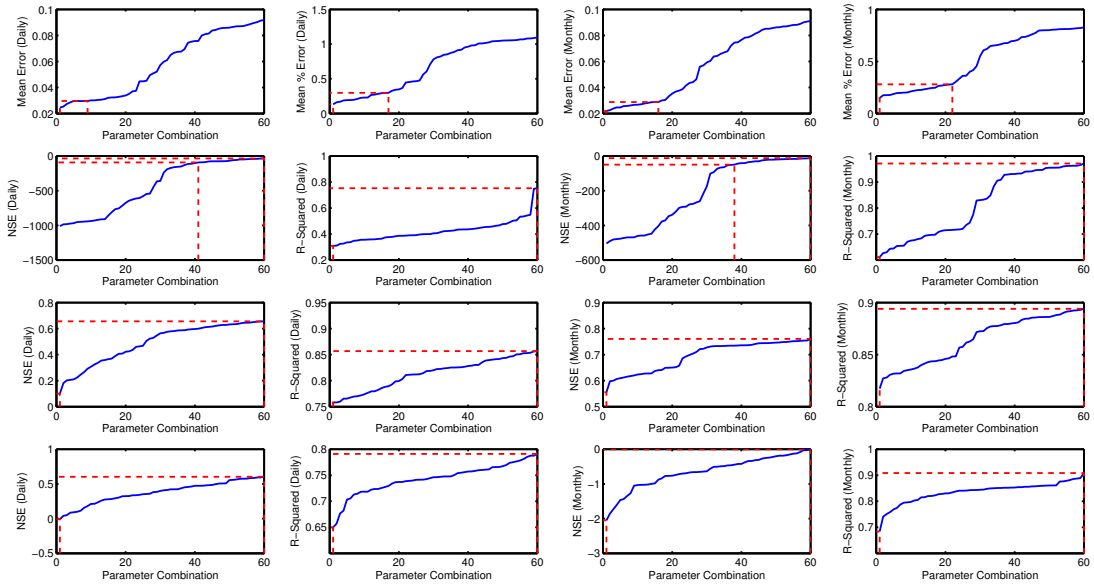

Figure A.7: Range of statistical values for different statistics: Formulation CVILLE2: Cannonsville Reservoir Model

Goodness of fit measures are employed in the interactive analysis, to ensure good 1) volume balance, 2) base flow calibration, 3) peak flow calibration and 4) average flow calibration. Relative bias and mean percent error are statistics used to ensure good volume balance, while the Nash Sutcliffe Efficiency (NSE)

and Correlation Coefficient, R, are used to assess the calibration quality of the different flow regimes. The interactive scheme starts with visualization of an initial range (i.e, over all calibration alternatives in P) of values for each goodness of fit measure and then allows the user to define a more narrow range for each measure. The user can specify a desired range for each measure (based on knowledge of model purpose) interactively, and is subsequently notified of the number of alternatives in the reduced set, R, for the specified ranges defined for each measure. Within this interactive interface, subjective analysis can be employed by a user to reduce the initial set of alternatives to a reduced set, after which a visual analysis can be performed to select one suitable alternative.

When applied to CVILLE2, Stage-II of HAMS-IS produced 60 alternative parameter combinations. Figure A.7 provides a visual snippet of the final range of acceptable values, after application of the interactive system, via Stage-III of HAMS-IS. A total of sixteen goodness of fit measures are employed in Stage-III of HAMS to assess performance of calibration alternatives in terms of volume balance (four measures are used and the ranges are visualized in the four sub-figures of the first row from the top in Figure A.7), base flow calibration (four measures are used and the ranges are visualized in the four sub-figures of the second row from the top in Figure A.7), peak flow calibration (four measures are used and the ranges are visualized in the four sub-figures of the third row from the top in Figure A.7) and average flow calibration (four measures are used and the ranges are visualized in the four sub-figures of the fourth row from the top in Figure A.7). The red lines depict the range of goodness of fit measures for each of the 16 measures. The blue line depicts a trend of the goodness of fit values in the set P. Since the purpose of the Cannonsville Model is the prediction of phosphorus loading in the Cannonsville reservoir, higher preference is given

to volume error minimization during the interactive alternatives reduction process. The number of solutions in the reduced set, R, is seven. We can also reduce the number of solutions in the set, R, to one and choose a single solution from amongst the various alternatives through the HAMS-IS interactive analysis.

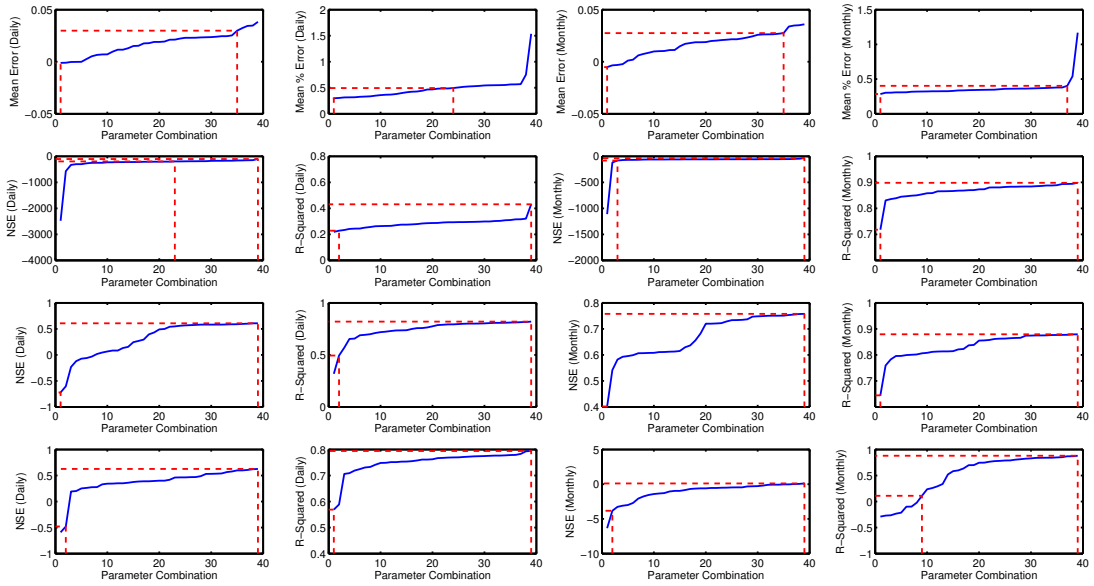

Figure A.8: Range of statistical values for different statistics: CVILLE3: Cannonsville Reservoir Model

Stage-II of HAMS-IS when applied to CVILLE3, produced 39 alternative parameter combinations. Figure A.8 provides a visual snippet of the final range of acceptable values, after application of the interactive system in Stage-III of HAMS-IS (arrangement of sub-figures in Figure A.8 is identical to the arrangement of sub-figures in Figure A.7). The number of solutions in the reduced set, R, is ten. A visual analysis is performed in stage-IV of HAMS-IS to choose one solution as the best calibration combination.

A.7.2 Knowledge Based Strategies

As discussed earlier, while knowledge based strategies, multi-objective optimization, automatic single objective optimization and manual calibration have separately been employed in deducing values of uncertain parameters in distributed watershed models, a hybrid comprehensive strategy which uses the strengths of each can further improve the calibration process. Madsen et al. (2002) [28] compare knowledge based methods with single and multi-objective calibration strategies, indicating that no method is superior in terms of all performance metrics. While such a strategy can be desirable when expert opinion is not available, an alternative strategy could be employed to reduce the set of alternatives to a small number through automatic expert analysis or fuzzy preference. We suggest an alternative to the HAMS-IS, which is the HAMS-ES. Stage-III of HAMS-ES differs from HAMS-IS, where instead of using manual interaction, we use an expert system-based methodology to mimic the manual alternatives reduction process and subsequently reduce alternatives to a smaller number for further manual analysis or choose the best according to the expert analysis ranking systems. This process is not very different from the fuzzy preference methodology proposed by Shrestha et al. (2008) [34]; however, we use a conditional elimination system prevalent in expert analysis.

Figure A.9 describes the automatic alternative reduction scheme, prescribed by HAMS-ES. The results obtained via application of HAMS-ES to CVILLE2 and CVILLE3, were very similar to the results obtained via HAMS-IS, indicating that an automatic expert analysis strategy can be an efficient alternative to the HAMS-IS methodology.

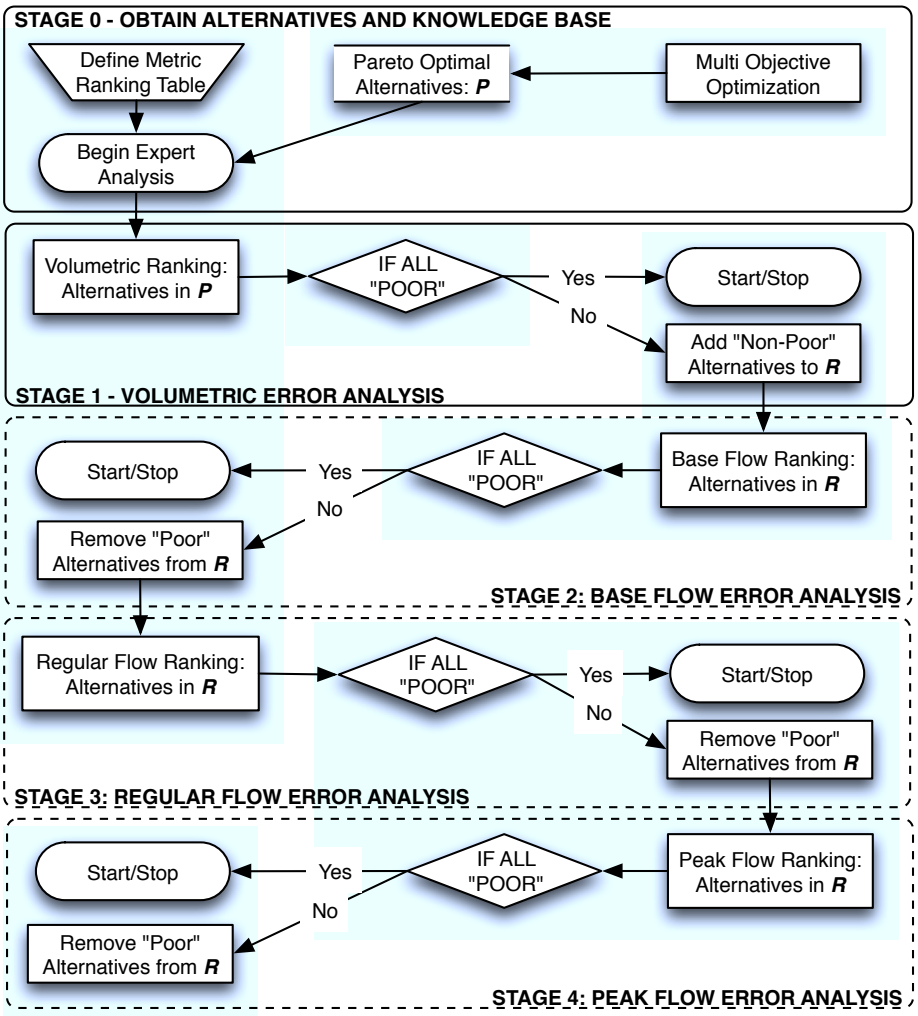

Figure A.9: Flow description of the Knowledge Based Pareto Reduction Scheme in HAMS-ES

A.8 Stage IV: Manual Decision Analysis

The final stage of the HAMS strategy is choosing one calibration alternative from amongst a very small set obtained through the first three stages of the hybrid strategy. Various multi-objective analysis research efforts highlight that manually choosing a solution from various alternatives is considerably simple, if the number of alternatives is less than 10.

Application of HAMS to CVILLE2 results in 7 alternatives, after Stage-III of

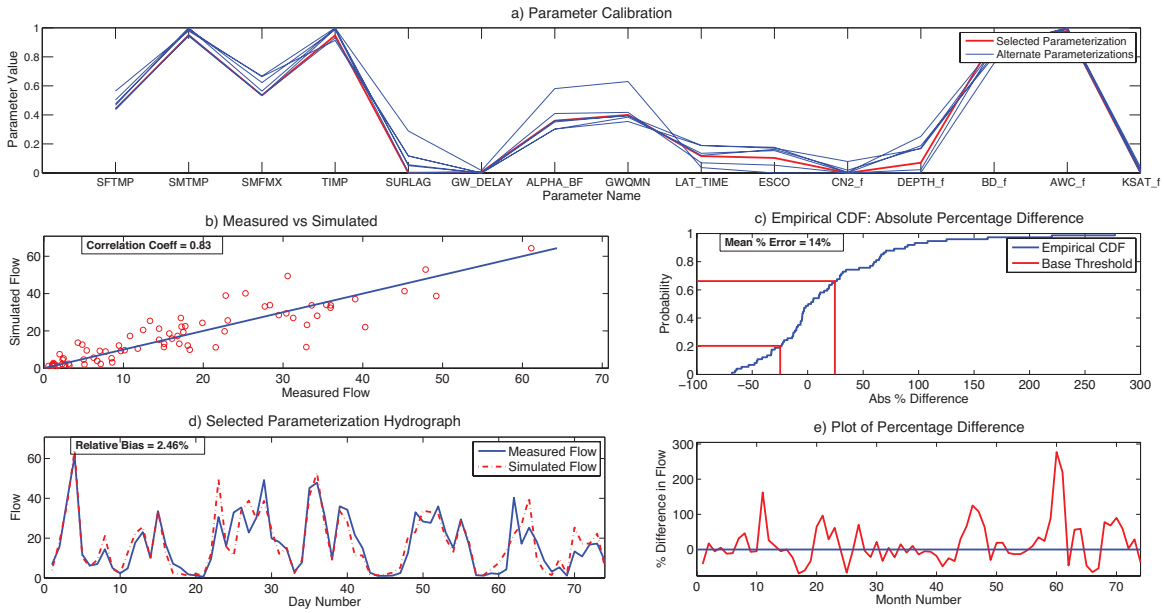

Figure A.10: Best Solution Selected after manual evaluation: CVILLE2 - Cannonsville Reservoir

the proposed strategy. In Stage-IV of HAMS, we employ a decision support system to assist in manually choosing one calibration alternative. Our decision support system visually depicts the hydrograph and various other statistics for all calibration alternatives identified by the first three stages of HAMS. Figure A.11 depicts the best calibration obtained via HAMS for CVILLE2 and also shows the other alternatives, identified after Stage-III.

Application of HAMS to CVILLE3 results in 10 alternatives, after Stage-III of the proposed strategy. A single calibration can be identified after analyzing each of the 10 alternatives through the decision support system of Stage-IV. Figure A.11 depicts the best calibration obtained via HAMS for CVILLE3 and also shows the other alternatives, identified after Stage-III.

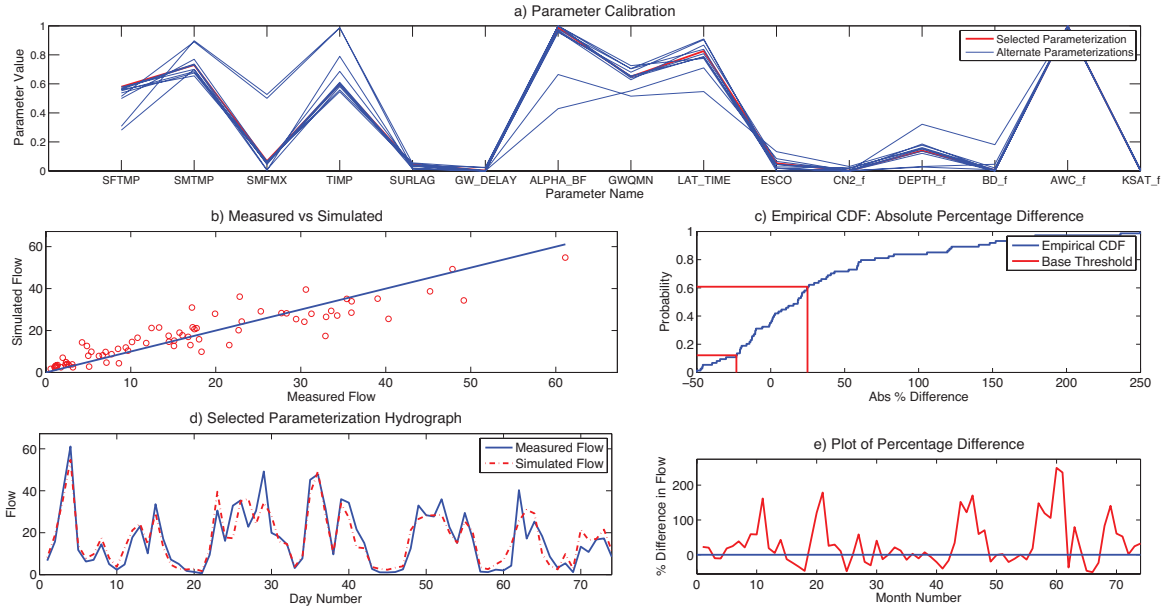

Figure A.11: Best Solution Selected after manual evaluation: CVILLE3 - Cannonsville Reservoir

A.9 Validation

In order to analyze the effectiveness of the HAMS strategy, we employed the best calibration obtained via manual analysis in Stage-IV during a validation period in order to visualize the robustness of the calibration. Additionally, we employ the reduced solutions obtained via Stage-III of HAMS in order to assess the ambiguity in model prediction resulting from choosing any calibration from those identified in Stage-III. This analysis was carried out for the CVILLE2 flow formulation, i.e., the threshold based formulation.

Figure A.12 provides a visual summary of simulation forecasts obtained from the seven reduced calibrations deduced by HAMS for CVILLE2, forecasted for a 2-year period. Figure A.12 (b) provides a visualization of the hydrographs obtained through the 7 reduced solutions, and a hydrograph visualization of the calibration deduced by the Dynamically Dimesnioned Search (DDS) algorithm

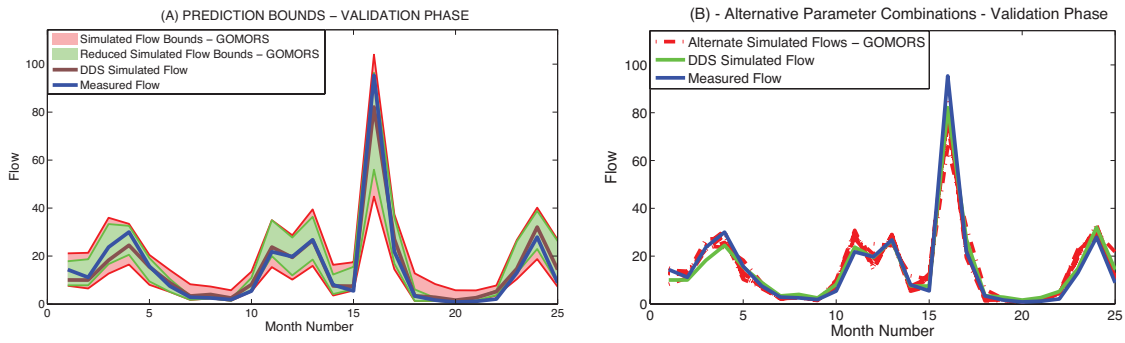

Figure A.12: Validation Phase: Best Solution Selected after manual evaluation: Formulation 2 - Cannonsville Reservoir

proposed by Tolson and Shoemaker (2007) [42]. Prediction bounds [43] [3] can also be obtained for the flow time series. They are determined by independently deducing the maximum and minimum flow values predicted by the alternative HAMS calibrations at each time step (daily). Figure A.12 (a) provides a visualization of the prediction bounds obtained through the non-dominated solutions of Stage-II (pink shaded region) and the 7 reduced solutions of Stage-III (green shaded region). The automatic calibration via DDS minimizes the Sum of Squared Errors to deduce a single calibration via single objective optimization, and the result depicted in the figure is obtained after 1000 simulation runs with DDS. The validation analysis depicted in Figure A.12 sheds light on the added information obtained via a multi-objective calibration analysis, highlighting the ambiguities within the model through prediction bounds.

Furthermore, we can analyze model ambiguity by observing the variations in predicted values of key statistics, for instance, relative bias, and mean percentage error, as depicted by Figure A.13. This information can assist decision makers in understanding model ambiguity, and consequently help in the management decision making process.

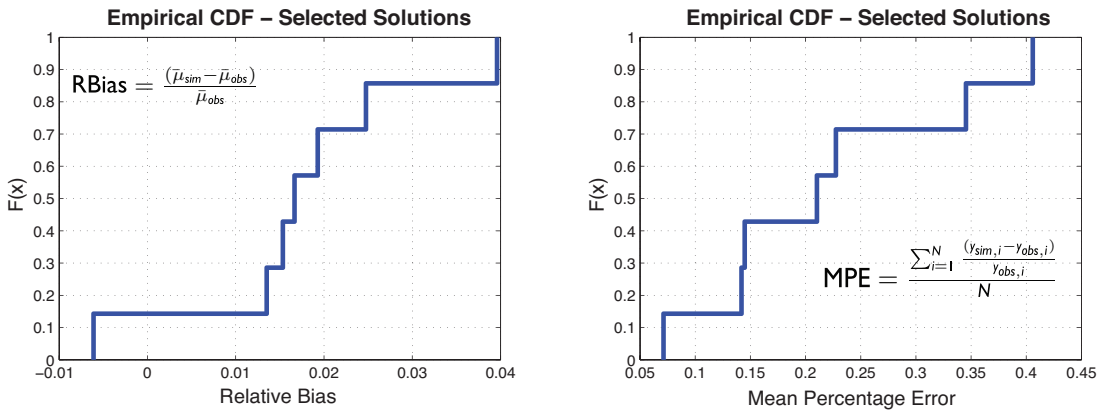

Figure A.13: Validation Phase: Empirical CDF Plots with 7 alternate calibrations identified by HAMS: Formulation 2 - Cannonsville Reservoir

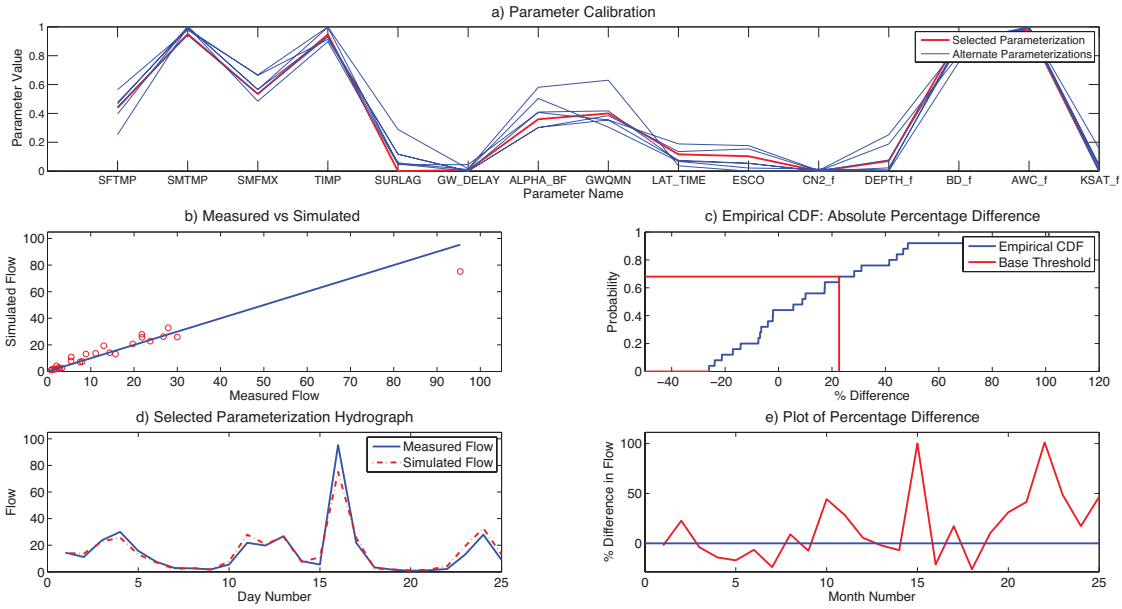

Figure A.14: Validation Phase: Best Solution Selected after manual evaluation: Formulation 2 - Cannonsville Reservoir

A.10 Conclusion

Figure A.14 provides a summary of the simulation forecast obtained via the calibration deduced by HAMS, for a 2-year period. Our results indicate a good

visual fit, which in turn depicts that the calibration obtained via multi-objective analysis is comparable to the automatic calibration obtained via DDS. Added information is provided as well, within the same simulation evaluation budget, and hydrologists can actively participate in the calibration process to deduce suitable calibrations, in accordance with the proposed use of the model. Hence, the HAMS strategy is a promising methodology which can be employed to bridge the gap between manual and automatic calibration methodologies.

A.11 References

[1] J. G. Arnold and N. Fohrer. SWAT2000: current capabilities and research opportunities in applied watershed modelling. *Hydrological Processes*, 19(3):563–572, February 2005.

[2] E Bekele and J Nicklow. Multi-objective automatic calibration of SWAT using NSGA-II. *Journal of Hydrology*, 341(3-4):165–176, August 2007.

[3] Keith Beven and Andrew Binley. The future of distributed models: Model calibration and uncertainty prediction, July 1992.

[4] Douglas P. Boyle, Hoshin V. Gupta, and Soroosh Sorooshian. Toward improved calibration of hydrologic models: Combining the strengths of manual and automatic methods. *Water Resources Research*, 36(12):null+, 2000.

[5] M D Buhmann. Radial basis functions. *Acta Numerica*, pages 1–38, 2000.

[6] Martin D Buhmann. *Radial Basis Functions*. Cambridge University Press, New York, NY, USA, 2003.

[7] C.A. Coello Coello, G.L. Lamont, and D.A. van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation. Springer, Berlin, Heidelberg, 2nd edition, 2007.

[8] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, PPSN VI, pages 849–858, London, UK, UK, 2000. Springer-Verlag.

[9] Qingyun Duan. *Calibration of Watershed Models*. AGU, Washington D.C., USA, 2003.

[10] Gift Dumedah, Aaron a. Berg, Mark Wineberg, and Robert Collier. Selecting Model Parameter Sets from a Trade-off Surface Generated from the Non-Dominated Sorting Genetic Algorithm-II. *Water Resources Management*, 24(15):4469–4489, May 2010.

[11] Fabrizio Fenicia, Hubert H. G. Savenije, Patrick Matgen, and Laurent Pfister. A comparison of alternative multiobjective calibration strategies for hydrological modeling. *Water Resources Research*, 43(3):1–16, March 2007.

[12] Ryan C. Fleming and Christine Shoemaker. Sensitivity Analysis of Model Predictions for Phosphorus Transport in Watersheds. *World Environmental and Water Resources Congress 2007*, pages 316–316, 2007.

[13] Margaret W. Gitau, Tamie L. Veith, William J. Gburek, and Albert R. Jarrett. Watershed Level Best Management Practice Selection and Placement in the Town Brook Watershed, New York 1. *Journal of the American Water Resources Association*, 42(6):1565–1581, December 2006.

[14] V. Guinot, B. Cappelaere, C. Delenne, and D. Ruelland. Towards improved criteria for hydrological model calibration: theoretical analysis of distance- and weak form-based functions. *Journal of Hydrology*, 401(1-2):1–13, April 2011.

[15] Hoshin V Gupta, Keith J Beven, and Thorsten Wagener. 142 : Model Calibration and Uncertainty ST THE NATURE OF RAINFALL-RUNOFF. *Time*, 2005.

[16] Hoshin V Gupta, Harald Kling, Koray K Yilmaz, and Guillermo F Martinez. Decomposition of the mean squared error and NSE performance criteria : Implications for improving hydrological modelling. *Journal of Hydrology*, 377(1-2):80–91, 2009.

[17] Hoshin Vijai Gupta, Soroosh Sorooshian, and Patrice Ogou Yapo. Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information. *Water Resources Research*, 34(4):751, 1998.

[18] Hoshin Vijai Gupta, Soroosh Sorooshian, and Patrice Ogou Yapo. Status of Automatic Calibration for Hydrologic Models: Comparison with Multilevel Expert Calibration. *Journal of Hydrologic Engineering*, 4(2):135, 1999.

[19] H Gutmann. A Radial Basis Function Method for Global Optimization. *Journal of Global Optimization*, pages 201–227, 2001.

[20] T Hogue, H Gupta, and S Sorooshian. A User-Friendly approach to parameter estimation in hydrologic models. *Journal of Hydrology*, 320(1-2):202–217, March 2006.

[21] Yongtai Huang and Lei Liu. Multiobjective Water Quality Model Calibration Using a Hybrid Genetic Algorithm and Neural NetworkBased Approach. *Journal of Environmental Engineering*, 136(10):1020, 2010.

[22] N Kajami, H Gupta, T Wagener, and S Sorooshian. Calibration of a semi-distributed hydrologic model for streamflow estimation along a river system. *Journal of Hydrology*, 298(1-4):112–135, October 2004.

[23] Soon-thiam Khu and Henrik Madsen. Advances in Water Resources Incorporating multiple observations for distributed hydrologic model cali-

bration : An approach using a multi-objective evolutionary algorithm and clustering. *Advances in Water Resources*, 31:1387–1398, 2008.

[24] Sang Min Kim, Brian L. Benham, Kevin M. Brannan, Rebecca W. Zeckoski, and John Doherty. Comparison of hydrologic calibration of HSPF using automatic and manual methods. *Water Resources Research*, 43(1):1–12, January 2007.

[25] Daniel P. Loucks, Eelco Van Beek, Jery R. Stedinger, Jozef P M Dijkman, and Monique. *Water Resources Systems Planning and Management : an Introduction to Methods, Models and Applications*. UNESCO, 2005.

[26] H Madsen. Automatic calibration of a conceptual rainfallrunoff model using multiple objectives. *Journal of Hydrology*, 235(3-4):276–288, August 2000.

[27] H Madsen. Parameter estimation in distributed hydrological catchment modelling using automatic calibration with multiple objectives. *Advances in Water Resources*, 26(2):205–216, February 2003.

[28] Henrik Madsen, Geoffrey Wilson, and Hans Christian Ammentorp. Comparison of different automated strategies for calibration of rainfall-runoff models. *Journal of Hydrology*, 261:48–59, 2002.

[29] Pradeep Mugunthan and Christine A Shoemaker. Assessing the impacts of parameter uncertainty for computationally expensive groundwater models. *Water Resources*, 42:1–15, 2006.

[30] M Muleta and J Nicklow. Sensitivity and uncertainty analysis coupled with automatic calibration for a distributed watershed model. *Journal of Hydrology*, 306(1-4):127–145, May 2005.

[31] J.E. Nash and J.V. Sutcliffe. River flow forecasting through conceptual models part i a discussion of principles. *Journal of Hydrology*, 10(3):282 – 290, 1970.

[32] S.L. Neitsch, J.G. Arnold, J.R. Kiniry, and J.R. Williams. *Soil and Water Assessment Tool Input/Output File version 2005*. US Department of Agriculture Agricultural Research Service, September 2004.

[33] S.L. Neitsch, J.G. Arnold, J.R. Kiniry, and J.R. Williams. *Soil and Water Assessment Tool theoretical documentation version 2005*. US Department of Agriculture Agricultural Research Service, January 2005.

[34] Rajesh Raj Shrestha and Michael Rode. Multi-objective calibration and fuzzy preference selection of a distributed hydrological model. *Environmental Modelling & Software*, 23(12):1384–1395, December 2008.

[35] Jens C. Refsgaard. Parameterisation, calibration and validation of distributed hydrological models. *Journal of Hydrology*, 198:69–97, November 1997.

[36] R Regis, C Shoemaker, and A. stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J. of Computing*, 19:497–509, 2007.

[37] Scientific Investigations Report. Hydrologic and Water-Quality Characterization and Modeling of the Onondaga Lake Basin , Onondaga County , New York Scientific Investigations Report 2008 5013. *New York*, 2008.

[38] M. Shafii and F. De Smedt. Multi-objective calibration of a distributed hydrological model (WetSpa) using a genetic algorithm. *Hydrology and Earth System Sciences Discussions*, 6(1):243–271, January 2009.

[39] Y. Tang, P. Reed, and T. Wagener. How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration? *Hydrology and Earth System Sciences*, 10(2):289–307, May 2006.

[40] B Tolson and C Shoemaker. Cannonsville reservoir watershed swat2000 model development, calibration and validation, J. *of Hydrology*, 337:68–86, 2007.

[41] Bryan A Tolson and Christine A Shoemaker. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. *Water Resources*, 43:1–16, 2007.

[42] Bryan A Tolson and Christine A Shoemaker. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. *Water Resources*, 43:1–16, 2007.

[43] Bryan a. Tolson and Christine a. Shoemaker. Efficient prediction uncertainty approximation in the calibration of environmental simulation models. *Water Resources Research*, 44(4):1–19, April 2008.

[44] Jasper a. Vrugt. Improved treatment of uncertainty in hydrologic modeling: Combining the strengths of global optimization and data assimilation. *Water Resources Research*, 41(1):1–17, 2005.

[45] Jasper a Vrugt and Bruce a Robinson. Improved evolutionary optimization from genetically adaptive multimethod search. *Proceedings of the National Academy of Sciences of the United States of America*, 104(3):708–11, January 2007.

[46] Kathryn Van Werkhoven, Thorsten Wagener, Patrick Reed, and Yong Tang. Advances in Water Resources Sensitivity-guided reduction of parametric

dimensionality for multi-objective calibration of watershed models. *Advances in Water Resources*, 32(8):1154–1169, 2009.

[47] P Yapo, H Gupta, and S Sorooshian. Multi-objective global optimization for hydrologic models. *Journal of Hydrology*, 204(1-4):83–97, January 1998.

APPENDIX B

SUPPLEMENTARY MATERIAL FOR MULTI OBJECTIVE OPTIMIZATION OF COMPUTATIONALLY EXPENSIVE MULTI-MODAL FUNCTIONS WITH RBF SURROGATES AND MULTI-RULE SELECTION[1]

B.1 Introduction

This appendix is a supplement to Chapter 2 of this document (Chapter 2 will be referred as the main paper in subsequent discussions). The primary purpose of the supplementary material within this appendix is to provide an elaborate discussion on the use of 1) Alternative algorithms for embedded evolutionary optimization in Steps 2.2 and 2.3 of the GOMORS Algorithm Framework described in Section 2.3.1 of the main paper, and 2) Multiple Selection rules employed within the GOMORS algorithm for selection of points for expensive evaluations. Furthermore, this supplement also provides a description of the Test Problems used for comparison of GOMORS with ParEGO and NSGA-II, and a description of the algorithm parameters employed for GOMORS, ParEGO and NSGA-II.

Section B.3 provides a summary of computer experiments performed with different evolutionary algorithms for solving the embedded global optimization problems in Steps 2.2 and 2.3 of the Algorithm (defined in Section 2.3.1 of the main paper). Section B.4 provides a summary of the results obtained from computer experiments performed for analyzing the comparative advantage of using multiple selection rules for selection of points for expensive evaluation (especially, if they are used to select multiple points in a single algorithm iteration for evaluation in parallel). Section B.2 provides a description of the Synthetic Test Problems used for comparing algorithms, as well as an elaborate description

[1]This Appendix has been published as an 'Online Supplement' to [1].

of the Hypothetical Groundwater Remediation Problem used in the computer experiments. Section B.5 provides the parameters used for all algorithms compared in this study, which include NSGA-II, ParEGO and GOMORS. Section B.6 provides additional figures for comparative analysis of GOMORS and ParEGO.

B.2 Test Problems

B.2.1 Synthetic Test Suite

The performance of GOMORS was tested via computational experiments performed on eleven continuous test functions (collectively referred as the Synthetic Test Suite). Five of the test functions employed in the experiments are part of the ZDT test suite [17], while the remaining six are derived from the work of Li and Zhang [8]. The problems incorporated in the synthetic test suite are not expensive to evaluate. However, they incorporate various optimization challenges (for instance multi modality, non-convexity of the Pareto Front, complicated Pareto Sets etc) inherent in real world multi objective optimization problems.

The ZDT test problems proposed by Zitzler et. al [17] include five continuous multi-objective optimization test problems all of which propose different challenges in converging to the Pareto optimal front. These challenges, along with mathematical descriptions of the ZDT optimization problems, are provided in Table B.1.The Pareto fronts for the five test problems are known. Meaningful comparisons of relative performance of different algorithms on these test problems can be based on how the quality of optimization solutions increases as the number of function evaluations increase, rather than on the actual running time of the algorithms (since purpose of comparison is evaluation of algorithm

Table B.1: Description of the ZDT Test Functions

Name	Mathematical Description	Problem Challenge
ZDT1	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$ where $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$ $x \in [0,1]^n$ n may vary between 2 and 30	
ZDT2	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ where $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$ $x \in [0,1]^n$ n may vary between 2 and 30	The Pareto Front is non-convex.
ZDT3	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)} - \frac{f_1(x)}{g(x)} \sin(10\pi x_1)]$ where $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$ $x \in [0,1]^n$ n may vary between 2 and 30	The Pareto front is disconnected and has several noncontiguous convex parts.
ZDT4	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$ where $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}[x_i^2 - 10\cos(4\pi x_i)]$ $x_1 \in [0,1], x_i \in [-5,5], \forall i \in \{2,\ldots,n\}$ n may vary between 2 and 30	21^9 locally optimal fronts exist.
ZDT6	$f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ where $g(x) = 1 + 9[(\sum_{i=2}^{n} x_i)/(n-1)]^{0.25}$ $x \in [0,1]^n$ n may vary between 2 and 30	The search space is non-uniform, and distribution of solutions on the Pareto front is non-uniform.

performance for computationally expensive problems).

Prior literature on the design of suitable test problems indicate that complicated shapes of Pareto Fronts add to the challenges of solving multi objective optimization problems via heuristics [8]. The ZDT test functions incorporate some of these challenges, including non-convex Pareto Fronts, disconnected Pareto fronts etc. However, the shapes of Pareto Sets for the ZDT functions are very simple. The Pareto Set of a problem refers to the set of Pareto optimal solutions of the problem in the decision space that corresponds to the Pareto Front. Pareto

Table B.2: Description of the LZF Test Functions

Name	Mathematical Description	Problem Challenge
LZF1	$f_1(x) = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}[x_j - \sin(6\pi x_1 + \frac{j\pi}{n})]^2$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}[x_j - \sin(6\pi x_1 + \frac{j\pi}{n})]^2$, where $J_1 = \{j \mid j$ is odd and $2 \le j \le n\}$, $J_2 = \{j \mid j$ is even and $2 \le j \le n\}$, and $x \in [0,1]$ x $[-1,1]^{n-1}$	The Pareto Set is a non-linear curve. Problem is also referred as F2 in [8].
LZF2	$f_1(x) = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} y_j^2$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} z_j^2$, where $J_1 = \{j \mid j$ is odd and $2 \le j \le n\}$, $J_2 = \{j \mid j$ is even and $2 \le j \le n\}$, $y_j = x_j - [0.3x_1^2\cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1]\cos(6\pi x_1 + \frac{j\pi}{n})$, $z_j = x_j - [0.3x_1^2\cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1]\sin 6\pi x_1 + \frac{j\pi}{n})$, and $x \in [0,1]$ x $[-1,1]^{n-1}$	The Pareto Set is a non-linear curve. Problem is also referred as F5 in [8].
LZF3	$f_1(x) = x_1 + \frac{2}{\lvert J_1\rvert}[4\sum_{j\in J_1} y_j^2 - 2\prod_{j\in J_1}\cos(\frac{20y_j\pi}{\sqrt{j}}) + 2]$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{\lvert J_1\rvert}[4\sum_{j\in J_1} y_j^2 - 2\prod_{j\in J_1}\cos(\frac{20y_j\pi}{\sqrt{j}}) + 2]$ where J_1 and J_2 are the same as in LZF1, $y_j = x_j - x_1^{0.5(1+\frac{3(j-2)}{n-2})}$, and $x \in [0,1]^n$	Many locally optimal fronts exist. Problem is also referred as F8 in [8].
LZF4 .	$f_1(x) = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} h(y_j)$ $f_2(x) = 1 - x_1^2 + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} h(y_j)$, where $J_1 = \{j \mid j$ is odd and $2 \le j \le n\}$, $J_2 = \{j \mid j$ is even and $2 \le j \le n\}$, $y_j = \sin(6\pi x_1 + \frac{j\pi}{n})$, $h(t) = \lvert t\rvert/(1 + 2e^{2\lvert t\rvert})$, and $x \in [0,1]$ x $[-2,2]^{n-1}$	Pareto Front is non-convex and Pareto Set is non-linear. The Problem is referred as UF4 in .
LZF5	$f_1(x) = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}[x_j - 0.8x_1\cos(6\pi x_1 + \frac{j\pi}{n})]^2$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}[x_j - 0.8x_1\sin(6\pi x_1 + \frac{j\pi}{n})]^2$ $J_1 = \{j \mid j$ is odd and $2 \le j \le n\}$ $J_2 = \{j \mid j$ is even and $2 \le j \le n\}$ $x \in [0,1]$ x $[-1,1]^{n-1}$	The Pareto Set is a non-linear curve. Problem is also referred as F3 in [8].
LZF6	$f_1(x) = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}[x_j - 0.8x_1\cos(\frac{6\pi x_1 + \frac{j\pi}{n}}{3})]^2$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}[x_j - 0.8x_1\sin(6\pi x_1 + \frac{j\pi}{n})]^2$ $J_1 = \{j \mid j$ is odd and $2 \le j \le n\}$ $J_2 = \{j \mid j$ is even and $2 \le j \le n\}$ $x \in [0,1]$ x $[-1,1]^{n-1}$	The Pareto Set is a non-linear curve. Problem is also referred as F4 in [8].

Sets for the ZDT problems are line segments. It is highly unlikely for real world problems to have such simple Pareto Sets.

Li and Zhang [8] introduced a general class of continuous multi objective optimization test problems where Pareto Sets could be defined arbitrarily. Furthermore, Li and Zhang [8] proposed nine specific test problems from the general class of problems introduced. We have incorporated five of these test in our synthetic test suite. We will refer these test problems as the LZF test functions for future discussion reference. The LZF test suite also includes a test problem which was proposed in the CEC 2009 MOEA competition [14]. This test function (LZF4) is also from the general class of optimization problems introduced by Li and Zhang. The LZF test problems propose different optimization challenges. These challenges, along with mathematical descriptions of the LZF optimization problems, are provided in Table B.2.

The ZDT and LZF functions will collectively be referred as the Synthetic Test Suite (eleven test problems) in future discussions. While Synthetic Test Suite identified for performance analysis addresses most of the optimization challenges discussed, some additional optimization challenges have been left unexplored, for instance, constraint handling, and testing on problems with more than two objectives.

B.2.2 Groundwater Remediation Design Problem

The test problems employed in our computer experiments also include a complex model describing the detoxification of contaminated groundwater using aerobic bioremediation [9]. Aerobic bioremediation is a decontamination process that involves the injection of electron acceptors (e.g. oxygen) or nutrients (e.g. nitrogen and phosphorus) into the groundwater to promote the growth of

microorganisms that can transform the contaminant into a harmless substance. Cleaning up contaminated groundwater is financially expensive and the cost could run into tens of millions of dollars. Hence, the use of a complex model describing the decontamination process, coupled with an efficient global optimization algorithm could assist in devising cleanup strategies that may yield huge savings for environmental cleanup.

The groundwater remediation model used in our experiments considers a hypothetical contaminated aquifer with characteristics that are symmetric about a horizontal axis. We assume that there are 6 injection wells in the contaminated aquifer, the locations of which have been fixed. 84 monitoring wells are also assumed which measure the concentration of the contaminant at specified time periods. The aquifer is discretized using a two-dimensional finite element mesh. It is assumed that the injection wells and monitoring wells are located at the nodes of the mesh and these wells are also symmetric about the horizontal axis. A two-dimensional finite element simulation model called Bio2D [11], is used to describe groundwater flow and the changes in the concentrations of the contaminant, oxygen, and biomass. The model equations are nonlinear because the growth equations for the microorganisms represent Monod kinetics.

The optimization process considers the problem of deciding the pumping rates for injection wells, which are used to supply oxygen into the groundwater. Due to the symmetry of the contaminated area, we only need to make pumping decisions for the 3 injection wells on one side of the axis of symmetry. The planning time period is divided into multiple management periods, and pumping rates remain constant in each management period. We use 2, 4 and 8 management periods in our computer experiments which correspond to 6, 12 and 24 decision variables, respectively. Because the pumping of oxygenated water is

expensive, our goal is to determine the pumping rates for each injection well at the beginning of each management period that minimize the total pumping cost and the contaminant concentration at the monitoring wells. Yoon and Shoemaker [13] incorporated the minimization of contaminant concentration, into the total pumping cost objective by means of a penalty term. This resulted in the formulation of the groundwater remediation design problem as a single objective global optimization problem.

We visualize the groundwater remediation design problem as a multi objective optimization problem where the conflicting objectives of the problem are 1) minimization of remediation cost and 2) minimization of contaminant concentration. The decision variables of the problem are the injection rates at 3 different well locations during each of m management periods. The number of decision variables vary between 6 and 24 (as discussed above), depending upon the number of management periods incorporated in the numerical computation model. The groundwater problem can be formulated as two separate multi-objective optimization problems. In the first formulation, the first objective is minimization of the cost of remediation, and the second objective is minimization of the maximum contaminant concentration recorded at the 84 monitoring wells at the end of the planning time. This formulation is referred to as the GWDM problem in the main paper and in this document. In the second formulation, the first objective is minimization of the cost of remediation and the second objective is minimization of the average contaminant concentration recorded at the 84 monitoring wells at the end of the planning time. This formulation is referred to as the GWDA problem in the main paper and in this document. Mathematical formulations of the two multi objective variants of the groundwater remediation design problem are provided in Table 2.2 of the main

paper.

The simulation times for groundwater remediation problems can vary between minutes and hours depending on the complexity of the model and the size of the modeled region. This example uses a relatively coarse grid and requires only about 0.5 sec. to run each simulation on a 1.3 GHz Intel Core i5 machine. However, it is representative of the type of multi objective optimization formulations used in more complex and computationally expensive groundwater bioremediation problems.

B.3 Alternative Methods for Embedded Evolutionary Optimization

Section 2.3.1 of the main paper defines the general framework of the GOMORS algorithm where surrogate optimization problems defined in Steps 2.2 and 2.3b of the algorithm framework are solved within each algorithm iteration. The primary purpose of the surrogate optimizations is to find near optimal fronts given that the expensive functions, $F(x)$, are replaced by the corresponding response surface model approximations. Evaluation points are subsequently chosen from the solutions obtained via the surrogate optimizations (also referred as candidate solutions or candidate populations) during each algorithm iteration.

Step 2.2 performs a global surrogate assisted search and hence, solves a surrogate multi objective optimization problem over the whole decision space, while Step 2.3b performs a local surrogate assisted search and hence, solves a surrogate optimization problem within a localized neighborhood of the least crowded solution on the expensively evaluated non-dominated front (see Step 2.3a of the algorithm framework defined in Section 2.3.1 of the main paper).

While solving them may be relatively inexpensive, the surrogate optimization problems are not trivial (since surrogates can be multi modal).

The optimization literature indicates that Multi Objective Evolutionary Algorithms (MOEAs) are a revolutionary heuristic approach for solving non trivial and multi modal multi objective optimization problems. Deb at al. [6] and Coello et al. [4] suggest that use of MOEAs can be highly beneficial, since the population based structure of an evolutionary algorithm can be exploited to simultaneously converge towards the Pareto front, and maintain a diverse set of trade-offs. Since the surrogate optimization problems defined in Steps 2.2 and 2.3 of the algorithm framework could potentially have non-linear and multi-modal objectives, and our aim is to find good approximate solutions of the surrogate problems (both in terms of convergence and diversity), we make use of evolutionary optimization for solving the surrogate problems within each iteration of GOMORS. In this regard, we performed computer experiments with either NSGA-II [5], MOEA/D [15], or AMALGAM [12] as evolutionary algorithms for solving the surrogate optimization problems within each iteration of the GOMORS algorithm framework. The purpose of these experiments was to identify one of these algorithm as most suited for surrogate optimization within GOMORS. Please note that surrogate optimization with reference to GOMORS is also referred as embedded evolutionary optimization in subsequent discussions. The evolutionary algorithms mentioned above are described briefly in sub sections B.3.1 - B.3.3. The experimental setup employed for comparison of performance of these algorithms within the GOMORS framework, and pertaining results are discussed in sub sections B.3.4 and B.3.5.

B.3.1 Non Dominated Sorting Genetic Algorithm - II (NSGA-II)

NSGA-II, proposed by Deb et al. [5] is a revolutionary and extremely popular multi-objective (MOEA), and has been applied to various applied engineering problems across numerous disciplines [4] [3]. NSGA-II handles the evolutionary search optimization process by ranking and archiving parent and child populations according to a non-domination sorting (a measure of convergence), and crowding distance, which is a measure of diversity of a solution, on a particular front. The NSGA-II aims at using Pareto-dominance to move the the non-dominated front towards convergence, during the search process, and can handle highly non-linear and multi modal objectives.

B.3.2 Multi Objective Evolutionary Algorithm - Decomposition (MOEA/D)

Aggregate functions can also be used in the multi-objective optimization process in order to convert a multi-objective problem into many single- objective problems. The aggregate methodology falls under a separate class of MOEAs within Coello et al.'s [4] classification, where the aim is to solve many single objective optimization problems, with different aggregation priorities, to converge to the Pareto front, and maintain divergence. The MOEA/D [15] uses aggregate functions, and simultaneously solves many single-objective Tchebycheff decompositions (Tchebycheff decomposition is a preference based aggregation method for converting a vector of objectives into a single objective) of multi-objective problems in a single run.The MOEA/D is an established benchmark algorithm and has won the 2009 IEEE Congress on Evolutionary Computation

(CEC 2009) competition.

B.3.3 AMALGAM - A Multi Algorithm Genetically Adaptive Multi-Objective Optimization Method

AMALGAM [12] is a multi-method evolutionary algorithm, which incorporates search mechanics of various algorithms. The key feature of AMALGAM is simultaneous multi-method search and the search mechanism selection is self-adaptive. AMALGAM incorporates four candidate MOEAs, NSGA-II, PSO, DE, and AMS. Recent literature has seen various applications of AMALGAM including Zhang et al.'s work [16], which includes application of AMALGAM for multi-site calibration, with comparison against SPEA2 and NSGA-II.

B.3.4 Experimental Setup

The relative performance of NSGA-II, MOEAD and AMALGAM as embedded MOEAs within the GOMORS algorithm framework was assessed on the synthetic test suite. Number of decision variables for all test problems were fixed at eight. All parameter settings of GOMORS were kept constant. The parameter settings of GOMORS used in the comparative analysis of the embedded MOEAs are defined in Table B.6 and Section B.5.3. The only change in GOMORS was the use of either of the three embedded MOEAs for solving the embedded optimization problems defined in Steps 2.2 and 2.3b of the algorithm framework (see Section 2.3.1 of the main paper).Ten optimization experiments were performed for each of the three modified versions of GOMORS, on each test problem and function evaluations were limited to 200. The same Latin hypercube samples were used for corresponding trials of each version of GOMORS, on each test

problem.

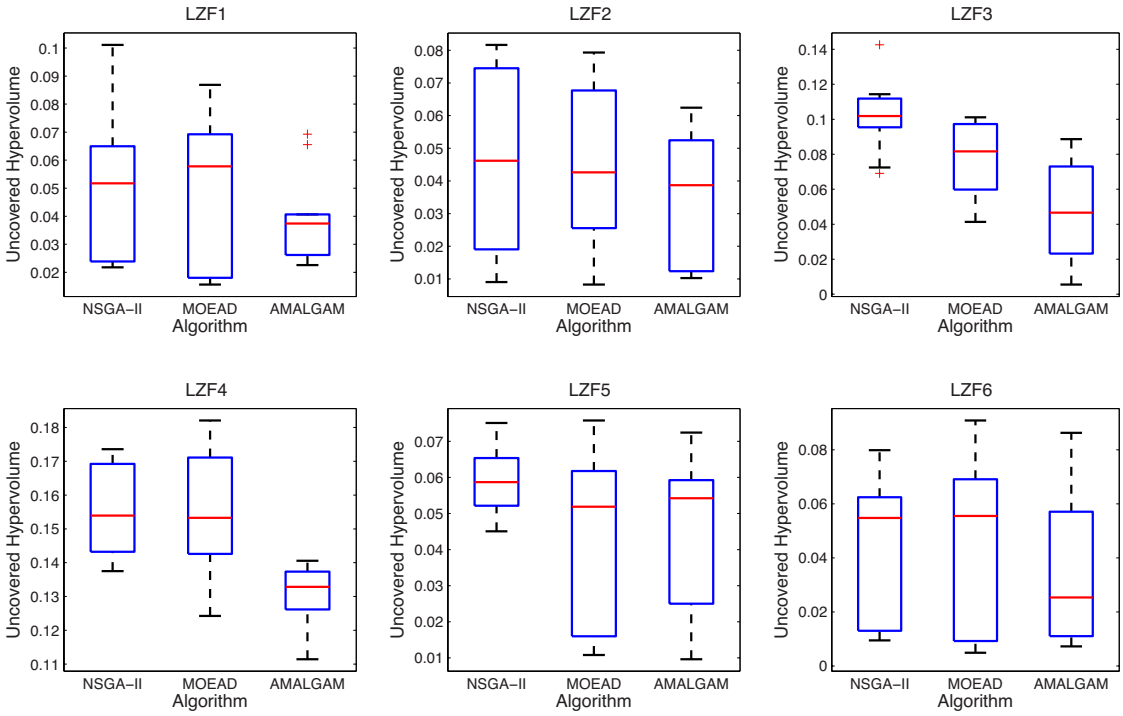

Figure B.1: Comparative analysis of evolutionary algorithms embedded into GOMORS - Box plots of uncovered hypervolume metric values obtained after 200 function evaluations with application of GOMORS using either of NSGA-II, MOEAD or AMALGAM as embedded evolutionary optimization algorithms . Results for the six LZF functions with 8 decision variables are shown in each subplot. In all cases lower values of uncovered hypervolume are best.

The uncovered hypervolume metric [2] was used to compare the performance of the three versions of GOMORS with different MOEAs. Uncovered hypervolume is the difference between the total feasible objective space (illustrated in Figure 2.1(a) of the main paper) and the objective space dominated by estimate of the Pareto front obtained by an algorithm. A lower value of the index indicates a better solution and the ideal value is zero. The next section discusses the results obtained from the experiments with different MOEAs.

B.3.5 Results and Analysis

The uncovered hypervolume metric values obtained after performing optimization experiments on the LZF problems with application of GOMORS with the three alternative embedded MOEAs, is depicted via box plots in Figure B.1. Each of the six sub plots of the figure display the results pertaining to each of the six LZF problems. Lower values of the metric signify superiority of performance and a lower spread within a box plot depicts definiteness of performance.

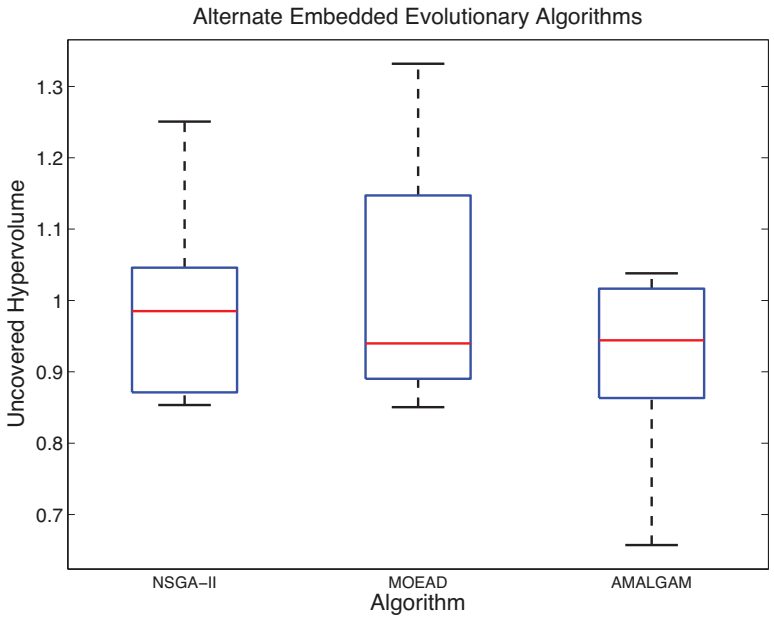

Figure B.2: Comparative analysis of evolutionary algorithms embedded into GOMORS - Box plots of uncovered hypervolume metric values summed over the six LZF problems, and obtained after 200 function evaluations with application of GOMORS using either of NSGA-II, MOEAD or AMALGAM as embedded evolutionary optimization algorithms. In all cases lower values of uncovered hypervolume are best.

The first conclusion that can be drawn from the results depicted in Figure B.1 is that it is hard to identify the difference between performance of all three embedded MOEAs. This is true especially in the case of LZF2, LZF5 and LZF6.

However, in the case of LZF1, LZF3, and LZF4, it can be seen that the overall performance of AMALGAM (as embedded MOEA of GOMORS) is better than NSGA-II and MOEAD, both in terms of attainment of lower median metric values and lower spread in the box plots. After an overall analysis of the box plots of Figure B.1, it can concluded that performance of AMALGAM is slightly better than NSGA-II and MOEAD as an embedded MOEA for GOMORS.

Results of all synthetic test problems were also compiled for analysis by summing the metric values of each of the ten optimization experiments performed on each test problem. The uncovered hypervolume metric value obtained from every trial of each individual test problem and algorithm combination could be considered as a random sample from an unknown distribution. Assuming that the unknown distributions of each individual test problem and algorithm combination are independent across a single algorithm, a sum of metric values of each trial across a single algorithm is another random sample from an unknown distribution which is a convolution of the independent unknown distributions. This convolution based metric summarizes overall performance of an algorithm on all synthetic test problems and is depicted in Figure B.2 for comparison of performance of embedded MOEAs across all synthetic test functions.

Figure B.2 also highlights that performance of all embedded MOEAs is comparable with reference to their application to the eleven synthetic test functions (and considering that results are shown as a convolution across all test problems). However, Figure B.2 also indicates that performance of AMALGAM is slightly better than NSGA-II and MOEAD. The worst attained convoluted metric value from AMALGAM is significantly lower than the worst attained convoluted metric values of NSGA-II and AMALGAM. Based on the results depicted in Figures B.1 and B.2, we used AMALGAM as the embedded MOEA

in GOMORS. However, it is worth mentioning that performances of the tested embedded MOEAs were close.

B.4 Experimental Assessment of Selection Rules

Section 2.3.4 of the main paper provides a discussion on the various selection rules employed within the GOMORS algorithm for selection of points for expensive evaluations within each algorithm iteration. Points for expensive evaluations are selected from the candidate points obtained after embedded optimizations of Steps 2.2 and 2.3 of the algorithm framework (defined in Section 2.3.1 of the main paper). The selected points are subsequently evaluated by the expensive objective functions, $F(x)$, within an algorithm iteration.

Selection of points for expensive evaluations is a critical step in the algorithm because evaluation of selected evaluations points is usually, by far, the most computationally expensive step of the algorithm, when applied to a computationally expensive problem. Furthermore, (as is discussed in Section 2.3.4 of the main paper) the strategy for selection of evaluation points is important for retraining of the surrogate functions in subsequent algorithm iterations, and can have a significant impact on the search dynamics of the algorithm. Major decisions pertaining to deduction of an appropriate selection strategy include the rule(s) for selection of evaluation point(s) and the number of points to be selected for expensive evaluation in each algorithm iteration.

Section 2.3.4 of the main paper proposes five selection rules which are employed to choose points for expensive evaluations in each iteration of GOMORS. In Section 2.3.4 of the main paper it is stated that GOMORS chooses one point for expensive evaluation in each algorithm iteration from four of the five rules stated (*Rules 1 to 4*), while one point for expensive evaluation is chosen from

Rule 0 with a probability of 0.1. Hence either 4 or 5 points are selected for expensive evaluations in each iteration of GOMORS. In order to deduce the above-mentioned strategy for selection of evaluation points, we performed various computer experiments with alternate evaluation point selection strategies. A total of six alternate selection strategies were compared in our analysis. The six selection strategies are discussed in Sections B.4.1 - B.4.6. The experimental setting for comparison of the selection strategies is discussed in Section B.4.7. Results obtained from the computer experiments are discussed in Section B.4.8.

B.4.1 Hypervolume Improvement

The first evaluation point selection strategy makes use of *Rule 1* (described on Section 2.3.4 of the main paper) to select one point for expensive evaluation in each iteration of GOMORS from the candidate solutions obtained from the surrogate optimization depicted in Step 2.2 of the algorithm framework (see Section 2.3.1 of the main paper for algorithm framework). *Rule 1* tends to choose a point from the candidate points obtained via global surrogate-assisted evolutionary optimization in Step 2.2, such that approximate improvement in hypervolume coverage is maximized. (see Figure 2.1 of the main paper for illustration of hypervolume improvement). This selection strategy, where one point is chosen for expensive evaluation via *Rule 1* in each iteration of GOMORS, is referred as the Hypervolume Improvement strategy in subsequent discussions. Please note that a second point for expensive evaluation may be selected in this selection strategy via random sampling (*Rule 0*), with a probability of 0.1.

B.4.2 Decision Space Euclidean Distance

The second point selection strategy analyzed in our computer experiments makes use of *Rule 2* (described in Section 2.3.4 of the main paper) to select one point for expensive evaluation in each iteration of GOMORS from the candidate solutions obtained from the surrogate optimization depicted in Step 2.2 of the algorithm framework (see Section 2.3.1 of the main paper for algorithm framework). *Rule 2* selects a point from the candidate points obtained via global surrogate-assisted evolutionary optimization in Step 2.2, such that the minimum euclidean distance (with respect to the decision space) from already evaluated points is maximized. This selection strategy, where one point is chosen for expensive evaluation via *Rule 2*, is referred to as the Decision Space Euclidean Distance strategy in subsequent discussions. Please note that a second point for expensive evaluation may be selected in this selection strategy via random sampling (*Rule 0*), with a probability of 0.1.

B.4.3 Objective Space Euclidean Distance

The third evaluation point selection strategy analyzed in our computer experiments makes use of *Rule 3* (described in Section 2.3.4 of the main paper) to select one point for expensive evaluation in each iteration of GOMORS from the candidate solutions obtained from the surrogate optimization depicted in Step 2.2 of the algorithm framework (see Section 2.3.1 of the main paper for an overview of the algorithm framework). *Rule 3* selects a point from the candidate points obtained via global surrogate-assisted evolutionary optimization in Step 2.2, such that the minimum euclidean distance (with respect to the objective function space) from already evaluated points is maximized. This selection strategy, where one point is chosen for expensive evaluation via *Rule 3*, is re-

ferred as the Objective Space Euclidean Distance strategy in subsequent discussions. Note that a second point for expensive evaluation may be selected in this selection strategy via random sampling (*Rule 0*), with a probability of 0.1.

B.4.4 Local Surrogate Assisted Search

The fourth evaluation point selection strategy analyzed in our computer experiments for comparing different selection strategies makes use of *Rule 4* (described in Section 2.3.4 of the main paper) to select one point for expensive evaluation in each iteration of GOMORS from the candidate solutions obtained from the surrogate optimization depicted in Step 2.3b of the algorithm framework (see Section 2.3.1 of the main paper for an overview of the algorithm framework). Evaluation point selection based on *Rule 4* is similar to *Rule 1*. However an evaluation point is selected from the candidate points obtained via local surrogate-assisted (or embedded) evolutionary optimization depicted in Step 2.3b. This selection strategy, where one point is chosen for expensive evaluation via *Rule 4* in each iteration of GOMORS, is referred to as the Local Surrogate Assisted Search strategy in subsequent discussions. Note that a second point for expensive evaluation may be selected in this selection strategy via random sampling (*Rule 0*), with a probability of 0.1.

B.4.5 Rule Cycling

The fifth evaluation point selection strategy selects one point for expensive evaluation in each iteration of GOMORS by cycling between the four point selection rules mentioned in the previous sections, i.e, hypervolume improvement, maximizing minimum euclidean distance in the decision space from previously

evaluated points, maximizing minimum objective space difference from previously evaluated points, and surrogate-assisted local search. This selection strategy, where only a single point is chosen for expensive evaluation via cycling between selection rules, is referred as the rule cycling selection strategy in subsequent discussions. Please note that a second point for expensive evaluation may be selected in this selection strategy via random sampling (*Rule 0*), with a probability of 0.1.

B.4.6 Parallel Selection

The sixth and final evaluation point selection strategy analyzed in our computer experiments selects one point each from Rules 1 to 4 (described on Section 2.3.4 of the main paper) for simultaneous expensive evaluations during each iteration of GOMORS. The primary purpose of investigating this strategy is to target an improvement in efficiency of GOMORS through parallel evaluation of expensive objectives in each algorithm iteration. This selection strategy, where four points are selected for expensive evaluation via Rules 1 to 4, is referred as the parallel selection strategy in subsequent discussions. Please note that a fifth point for expensive evaluation may be selected in this selection strategy during each algorithm iteration via random sampling (*Rule 0*), with a probability of 0.1.

B.4.7 Experimental Setup

The relative performance of the alternate evaluation point selection strategies of GOMORS was assessed on the synthetic test suite. Number of decision variables for all test problems were fixed at eight. All other parameter settings of the algorithm, for instance the choice of embedded MOEA, and related param-

Table B.3: The alphanumeric codes assigned to each selection strategy for Figures 3 and 4

Strategy Name	**Code**
Approximate Hypervolume Improvement	HYP
Decision Space Euclidean Distance	D-DS
Objective Space Euclidean Distance	D-OS
Surrogate-assisted Local Search	LS
Rule Cycling	CYC
Parallel Selection with 200 expensive evaluations	P200
Parallel Selection with 400 expensive evaluations	P400

eters, were the same in all experiments. The parameter settings are provided in Table B.6 and Section B.5.3. The only change in the algorithm was the use of the alternate evaluation point selection strategies.

Ten optimization experiments were performed for each strategy, on each test problem and the same Latin hypercube samples were used for corresponding trials of each strategy (in an effort to ensure that the analysis is as fair as possible). The function evaluations were limited to 200 for all strategies. However, for the parallel selection strategy, we also performed experiments with a limit of 400 evaluations, in order to analyze the advantage, if any, of simultaneous selection and evaluation of expensive objectives in each algorithm iteration, in terms of speed of convergence of the algorithm.

The uncovered hypervolume metric [2] was used to compare the performance of the alternate selection strategies. As discussed earlier, Uncovered hypervolume is the difference between the total feasible objective space (illustrated in Figure 2.1(a) of the main paper) and the objective space dominated by esti-

mate of the Pareto front obtained by an algorithm. A lower value of the index indicates a better solution and the ideal value is zero. Figures B.3 and B.4 summarize the results obtained from the experiments. These are discussed further in the next section. Please note that alphanumeric codes are used to identify the different selection strategies in Figures B.3 and B.4. Table B.3 provides the selection strategy names corresponding to the alphanumeric codes used in Figures B.3 and B.4. We refer the alphanumeric during our discussion or the results in the next section.

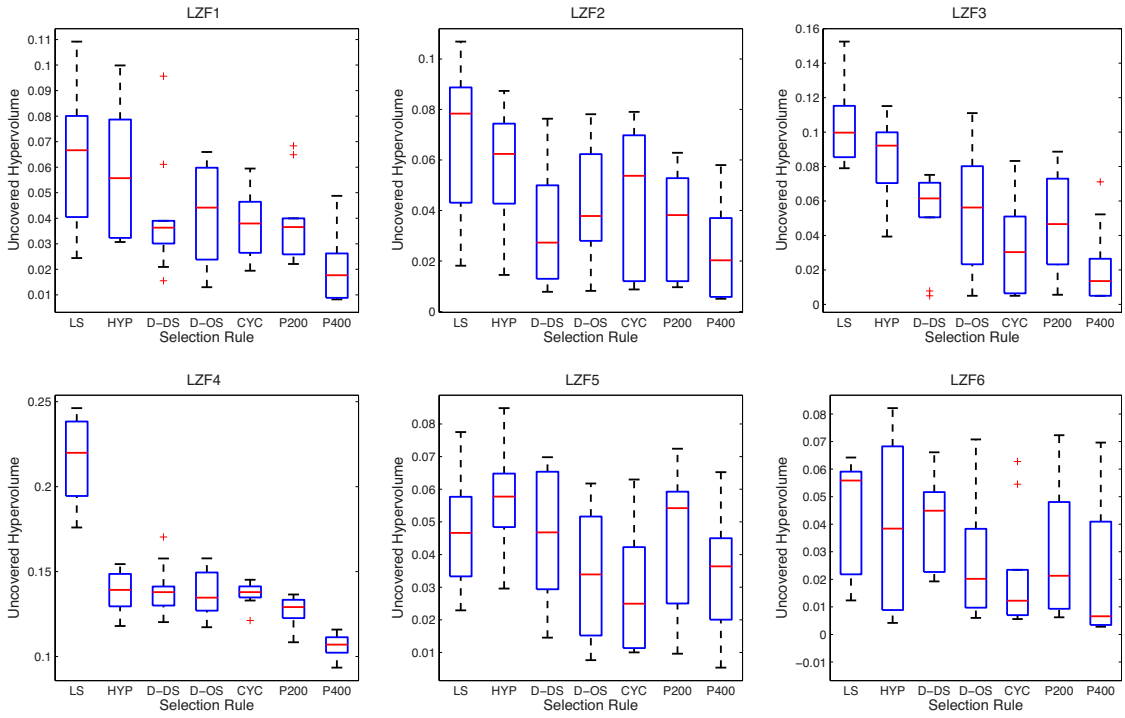

Figure B.3: Comparative analysis of selection strategies for LZF functions - Box plots of uncovered hypervolume metric values obtained after 200 function evaluations (400 in case of P400) through application of GOMORS with alternate evaluation point selection strategies. Results for the six LZF functions with 8 decision variables are shown in each subplot. In all cases lower values of uncovered hypervolume are best.

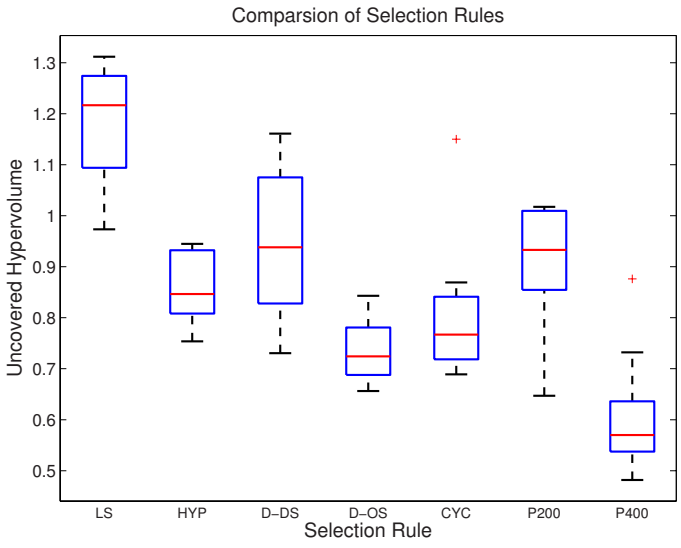

Figure B.4: Comparative analysis of election strategies - Box plots of uncovered hypervolume metric values summed over the eleven synthetic test functions, and obtained after 200 function evaluations (400 in case of P400) through application of GOMORS with alternate evaluation point selection strategies. In all cases lower values of uncovered hypervolume are best.

B.4.8 Results and Discussion

The purpose of this section is to evaluate the performance of GOMORS with the different evaluation point selection strategies discussed in Sections B.4.1 to B.4.6. This is done to analyze the difference in performance of GOMORS when only one of four selection rules is employed for evaluation point selection, when cycling between the rules is employed for evaluation point selection and when all rules are employed simultaneously in each algorithm iteration to select multiple points for expensive evaluations. The purpose of the analysis is to understand the advantage (if any) of using multiple rules to select multiple points for simultaneous evaluations in each algorithm iteration.

The uncovered hypervolume metric values obtained after performing optimization experiments on the LZF problems with application of GOMORS with

the alternate selection strategies, are depicted via box plots in Figure B.3. The six sub plots in the figure display the results pertaining to each of the six LZF problems. Lower values of the metric signify superiority of performance and a lower spread within a box plot depicts definiteness of performance.

We first analyze the performance of the first four selection strategies, where a selection rules is exclusively used to select one point for expensive evaluation in each algorithm iteration. An interesting observation in this regard is that none of these four selection strategies (i.e, LS, HYP, D-DS and D-OS) clearly outperforms the others for all LZF test functions. However, it can be observed that performance of LS and HYP is relatively poor when considering the results on all LZF functions. The performance of LS is relatively poor for LZF1, LZF2, LZF3 and LZF4, and performance of HYP is relatively poor for LZF1, LZF2 , LZF3 and LZF5. The relatively poor performance of LS, where only a surrogate-assisted local search is performed to select an evaluation point in each algorithm iteration, is understandable. The relatively poor performance of HYP may be a consequence of errors in RBF approximation due to the existence of complicated Pareto sets in the LZF functions. Performance of D-DS and D-OS is comparable, and relatively better than the LS and HYP selection strategies. Since D-DS and D-OS select evaluation points based on maximization of minimum euclidean distance from already evaluated points (in the decision space and objective space respectively), they tend to maintain a balance between exploration and exploitation during the surrogate-assisted search. This might be the reason behind their relatively superior performance on the LZF functions where the Pareto sets are formed by complicated curves in the decision space.

The rule cycling (CYC) selection strategy also performs adequately on the LZF functions. The most interesting finding though is the performance of the

Parallel selection strategy (P200 and P400) where multiple points (one point each from Rules 1 to 4) are simultaneously selected for expensive evaluations in each algorithm iteration. Hence, four points are selected for simultaneous evaluations (a fifth point may be selected via random sampling with probability of 0.1). The primary motivation behind this strategy is to enable the GOMORS to evaluate expensive function in parallel during each algorithm iteration in order to improve performance in terms of speed of convergence.

Results for the Parallel selection strategy with a limit of 400 evaluations (P400) are particularly interesting in this regard. Results of the P400 selection strategy depicted in Figure B.3 indicate that performance of the P400 strategy is either comparable or better than the other selection strategies. If the value of parallelization is considered, the P400 selection strategy produces comparable results to the other selection strategies with approximately 100 effective expensive function evaluations since four points may be evaluated in parallel during each algorithm iteration (Our approximation of 100 effective evaluations is made with the assumption that the communication time overhead is inconsequential in comparison to the computational cost of an expensive function evaluation).

Results of all synthetic test problems (including the ZDT functions) were also compiled for analysis by summing the metric values of each of the ten optimization experiments performed on each test problem. The resulting box-plots are depicted in Figure B.4, and are consistent with the observations deduced from Figure B.3. Figure B.4 indicates that results from the parallel selection strategy after 400 function evaluations (P400) are considerably better than the other selection strategies after 200 function evaluations. If the value of parallelization is considered, Figure B.4 implies that the parallel selection strategy with 100

effective function evaluations (since 4 points may be evaluated in parallel during each algorithm iteration) outperforms the other selection strategies with 200 function evaluations.

Hence, in the main paper (Section 2.3.4) we recommend the use of a Multi-Rule selection strategy where selection rules 1 to 4 are employed to select four points for simultaneous evaluations in each iteration of GOMORS (A fifth point may be selected via Rule 0 with a probability of 0.1). If a serial implementation of GOMORS is employed, we recommend the use of either CYC or D-OS as the selection strategy, based on the results obtained from experiments performed on the synthetic test suite (see Figure B.4).

B.5 Parameter Settings

In order to evaluate and compare the performance of GOMORS, ParEGO and NSGA-II on the synthetic test suite and the groundwater remediation problems, ten optimization experiments were performed for each algorithm, on each test problem. As mentioned in Section 2.5.1 of the main paper, multiple experiments were performed on each test problem, since all algorithms are stochastic. Furthermore, the evaluation budget was limited to four hundred, since the purpose of the experiments was to evaluate algorithm performance within a limited evaluation budget. Parameters for the algorithms were set such that the comparison be as fair as possible.

Parameter settings for each algorithm are discussed in sections B.5.1 - B.5.3. The initial Latin hypercube sample sizes for GOMORS and ParEGO were fixed at $2d + 2$, where d is the number of decision variables. Since both GOMORS and ParEGO use Latin hypercube sampling for initiation, the fairest way to compare them is to have trial j for both algorithms have the same Latin hypercube for $j =$

Table B.4: Parameter Settings for NSGA-II

Parameter Name	**Setting**
Population Size	20
Maximum Generations	19
Crossover Probability	0.9
Real-value SBX parameter	20
Polynomial mutation parameter	20

$1, \ldots, N$ (when a particular problem is tested). Hence, the same Latin hypercube samples were used for corresponding trials of GOMORS and ParEGO on each test problem.

B.5.1 NSGA-II

We used a MATLAB implementation of the NSGA-II, coded by Aravind Seshadri. This implementation is available on the MATLAB Central File Exchange and is based on the algorithm description provided by Deb. et. al. [5]. Default settings of the algorithm parameters were used with the exception of the population size and the number of generations. We experimented with population sizes of 10, 20, 50 and 100, on the test functions, with an evaluation budget limit of four hundred. Results from the experiments indicated that a population size of 20 was most feasible in terms of algorithm performance. Hence, a population size of 20 was used and the number of generations deduced were 19. Table B.4 summarized the parameter settings used for NSGA-II.

Table B.5: Parameter Settings for ParEGO

Parameter Name	**Setting**
No. of initial Latin hypercube samples	$2d + 2$
No. of maximum evaluations	400
No. of scalarizing vectors	$11(k = 2)$
Scalarizing function type	Augmented Tchebycheff
Internal GA population size	20
Internal GA generations	10000
Crossover Probability	0.9
Real-value SBX parameter	10
Real-value mutation probability	$1/d$
Polynomial mutation parameter	50

B.5.2 ParEGO

The implementation code for ParEGO was obtained from the research web page of Joshua Knowles (first author of [7]). Default settings of the algorithm parameters, as prescribed in [7] were used with the exception of the number of initial Latin hypercube samples. Knowles [7] recommend $11d - 1$ as the number of initial Latin hypercube samples. However since our computer experiments incorporate problems with up to 24 decision variables with an evaluation budget of 400, an initial sample size of $11d - 1$ would result in a large initial sample relative to the evaluation budget. Consequently, we used an initial Latin hypercube sample size of $2d + 2$, as per the recommendations provided by Regis and Shoemaker [10]. Table B.5 summarized the parameter settings used for ParEGO (Please note that k is the number of objective functions).

B.5.3 GOMORS

GOMORS was implemented in MATLAB version 2012b. For the embedded optimization, MATLAB implementation codes for NSGA-II, MOEA/D and AMALGAM were obtained externally. The implementation for NSGA-II, coded by Aravind Seshadri, is available on the MATLAB Central File Exchange and is based on the algorithm description provided by Deb. et. al. [5]. The implementation for MOEA/D was obtained from Qingfu Zhang's research web page (Qingfu Zhang is the first author of MOEA/D [15]). The implementation code for AMALGAM was obtained from Jasper Vrugt through an email request (Jasper Vrugt is the first author of AMALGAM [12]). Default settings of the embedded algorithms were used in the analysis discussed in Section B.3. Subsequent to the analysis AMALGAM was highlighted as an appropriate embedded evolutionary algorithm for use within GOMORS. Default settings of AMALGAM were used within the embedded optimizations of Steps 2.2 and 2.3b of GOMORS (please refer to Section 2.3.1 of the main paper for reviewing the algorithm framework). Table B.6 summarized the parameter settings used for GOMORS.

Additionally, it should be noted that one point each is selected for expensive evaluation from rules no. 1 to 4, during each algorithm iteration (all selection rules are described in Section 2.3.4 of the main paper), in the final implementation of GOMORS. Furthermore, a point is selected via rule no. 0, during each algorithm iteration, with a probability of 0.1 (Please note that rule 0 is selection through random sampling). Hence, in this application of GOMORS only 4 or 5 points are selected for parallel evaluation in each algorithm iteration. However, more points could be selected from each rule during an algorithm iteration and more rules could be incorporated in the algorithm in future, resulting in an in-

Table B.6: Parameter Settings for GOMORS

Parameter Name	**Setting**
No. of initial Latin hypercube samples	$2d + 2$
No. of maximum evaluations	400
Local Search / Gap radius	0.1
Random selection probability	0.1
Embedded global MOEA population size	100
Embedded global MOEA generations	25
Embedded local MOEA population size	100
Embedded local MOEA population size	25

crease in the number of simultaneous evaluations in each algorithm iteration. The parallel properties of GOMORS are discussed briefly in Section B.4. The MATLAB implementation of GOMORS will soon be available on the research webpage of Christine Shoemaker.

B.6 Additional Results

Additional results pertaining to Non-Dominated Front visualizations for the two groundwater problems are provided in this section. There are two figures (B.5 and B.6) corresponding to the worst solutions (as per uncovered hypervolume) obtained from multiple experiments performed via GOMORS and ParEGO on the GWDA and GWDM groundwater problems. The red line within each sub-plot of the figures is an estimate of the Pareto front. This estimate was obtained (for both GWDM and GWDA) through a single trial of NSGA-II with 50000 function evaluations. The green dots within each sub-plot cor-

respond to the non-dominated solutions obtained via application of the algorithm referenced in the sub-plot. There are six sub-figures within each figure and two sub-plots within each sub-figure, depicting the difference in performance of GOMORS and ParEGO. Each sub-figure corresponds to the worst non-dominated fronts obtained from each algorithm for a different number of decision variables (6 and 24) and after a fixed number of function evaluations (100, 200 and 400). A bottom to top traversal of each figure provides a visualization of the change in performance of algorithms as function evaluations increase, and a left to right traversal assists in visualizing performance differences when decision variables increase.

Figures B.5 and B.6 are consistent with the conclusions drawn in Section 2.5 of the main paper, and depict that performance of GOMORS is superior to ParEGO even in the worst case scenario. GOMORS is able to produce diverse trade-offs in the worst case scenario for the 24 variable problems, while ParEGO fails to obtain diverse solutions. This notion is true for both GWDA and GWDM.

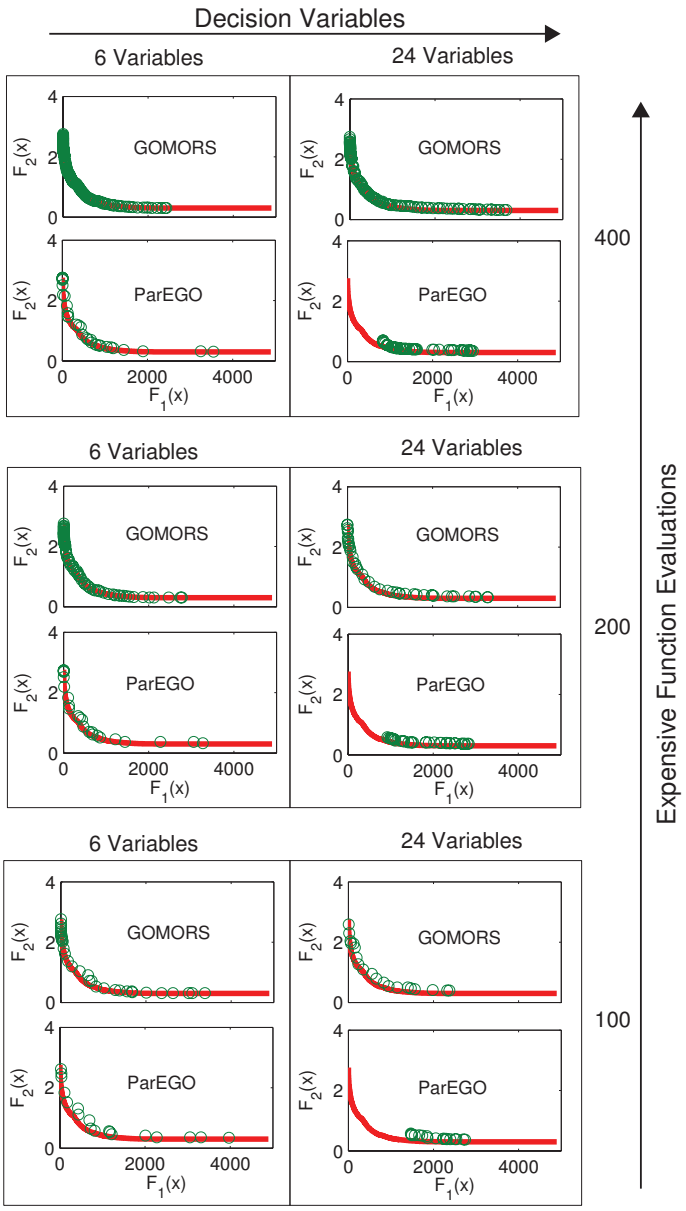

Figure B.5: Estimated Non-dominated Front for Groundwater Problem GWDM: Visual comparison of worst non-dominated fronts obtained from GOMORS and ParEGO for GWDM with 6 and 24 decisions, and after 100, 200 and 400 expensive function evaluations. Red line is result of 50,000 evaluations with NSGA-II. Green circles are non-dominated solutions from GOMORS or ParEGO.

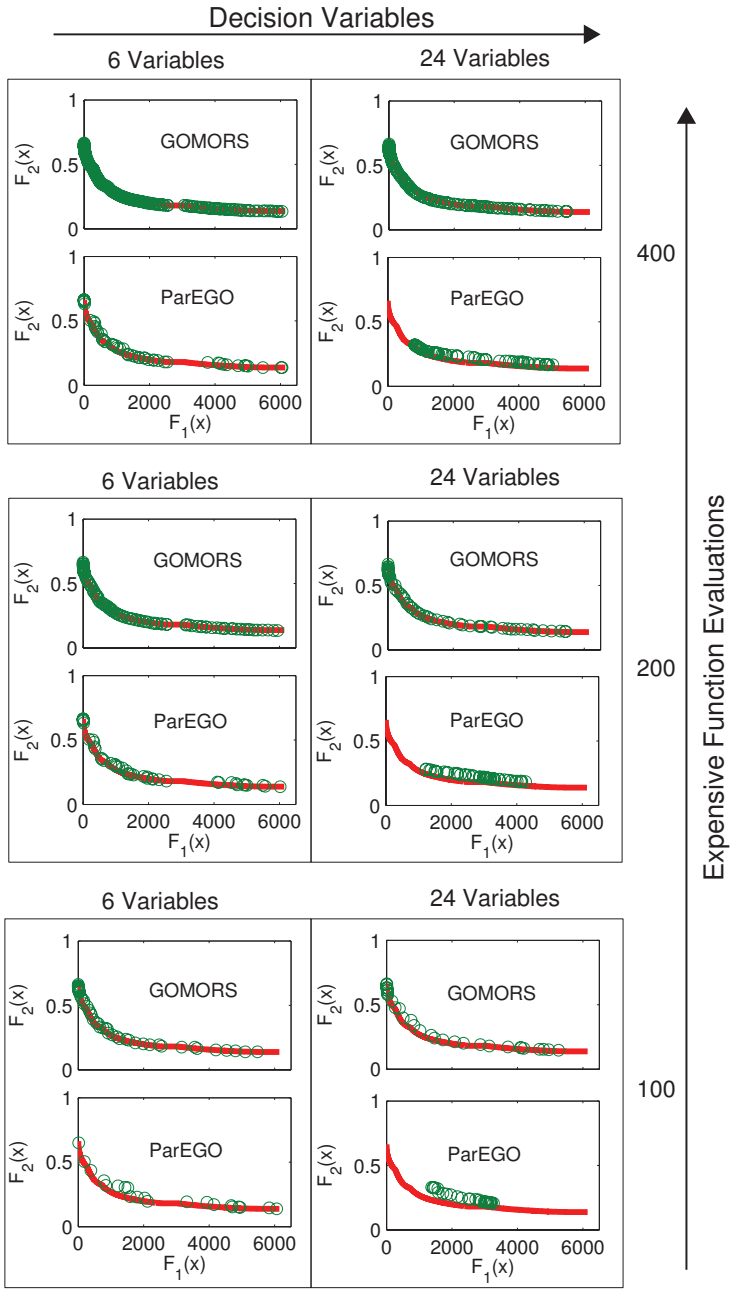

Figure B.6: Estimated Non-dominated Front for Groundwater Problem GWDA: Visual comparison of worst non-dominated fronts obtained from GOMORS and ParEGO for GWDA with 6 and 24 decisions, and after 100, 200 and 400 expensive function evaluations. Red line is result of 50,000 evaluations with NSGA-II. Green circles are non-dominated solutions from GOMORS or ParEGO.

B.7 References

[1] Taimoor Akhtar and Christine A. Shoemaker. Multi objective optimization of computationally expensive multi-modal functions with rbf surrogates and multi-rule selection. *Journal of Global Optimization*, pages 1–16, 2015.

[2] Johannes M. Bader. *Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods*. CreateSpace, Paramount, CA, 2010.

[3] E Bekele and J Nicklow. Multi-objective automatic calibration of SWAT using NSGA-II. *Journal of Hydrology*, 341(3-4):165–176, August 2007.

[4] C.A. Coello Coello, G.L. Lamont, and D.A. van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation. Springer, Berlin, Heidelberg, 2nd edition, 2007.

[5] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, PPSN VI, pages 849–858, London, UK, UK, 2000. Springer-Verlag.

[6] Kalyanmoy Deb and Deb Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 1 edition, June 2001.

[7] J. Knowles. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 8(5):1341–66, February 2006.

[8] Hui Li and Qingfu Zhang. Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, April 2009.

[9] Barbara S. Minsker and Christine A. Shoemaker. Differentiating a finite element biodegradation simulation model for optimal control. *Water Resources Research*, 32(1):187–192, 1996.

[10] R Regis, C Shoemaker, and A. stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J. of Computing*, 19:497–509, 2007.

[11] S. W. Taylor. Modeling enhanced in-situ biodegradation in groundwater: Model response to biological parameter uncertainty. In *1993 Groundwater Modeling Conference*, 1993.

[12] Jasper a Vrugt and Bruce a Robinson. Improved evolutionary optimization from genetically adaptive multimethod search. *Proceedings of the National Academy of Sciences of the United States of America*, 104(3):708–11, January 2007.

[13] J.H. Yoon, , and Christine A. Shoemaker. Comparison of optimization methods for ground-water bioremediation. *Journal of Water Resources Planning and Management*, 125:54–63, 1999.

[14] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari. Multiobjective optimization test instances for the cec 2009 special session and competition. Technical Report CES-487, University of Essex and Nanyang Technological University, 2008.

[15] Qingfu Zhang and Hui Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.

[16] Xuesong Zhang, Raghavan Srinivasan, and Michael Van Liew. On the use of multi-algorithm, genetically adaptive multi-objective method for multi-site calibration of the SWAT model. *Hydrological Processes*, 24(8):955–969, April 2010.

[17] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multi-objective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8:173–195, 2000.