2016

Model Selection via Racing

Tiantian Zhang
University of Central Florida

MODEL SELECTION VIA RACING

by

TIANTIAN ZHANG
B.S. Northwestern Polytechnical University, 2009
M.S. University of Central Florida, 2014

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2016

Major Professor: Michael Georgiopoulos, Georgios C. Anagnostopoulos

ABSTRACT

Model Selection (MS) is an important aspect of machine learning, as necessitated by the No Free Lunch theorem. Briefly speaking, the task of MS is to identify a subset of models that are optimal in terms of pre-selected optimization criteria. There are many practical applications of MS, such as model parameter tuning, personalized recommendations, A/B testing, etc. Lately, some MS research has focused on trading off exactness of the optimization with somewhat alleviating the computational burden entailed. Recent attempts along this line include metaheuristics optimization, local search-based approaches, sequential model-based methods, portfolio algorithm approaches, and multi-armed bandits.

Racing Algorithms (RAs) are an active research area in MS, which trade off some computational cost for a reduced, but acceptable likelihood that the models returned are indeed optimal among the given ensemble of models. All existing RAs in the literature are designed as Single-Objective Racing Algorithm (SORA) for Single-Objective Model Selection (SOMS), where a single optimization criterion is considered for measuring the goodness of models. Moreover, they are offline algorithms in which MS occurs before model deployment and the selected models are optimal in terms of their overall average performances on a validation set of problem instances.

This work aims to investigate racing approaches along two distinct directions: Extreme Model Selection (EMS) and Multi-Objective Model Selection (MOMS).

In EMS, given a problem instance and a limited computational budget shared among all the candidate models, one is interested in maximizing the final solution quality. In such a setting, MS occurs during model comparison in terms of maximum performance and involves no model validation. EMS is a natural framework for many applications. However, EMS problems remain unaddressed by current racing approaches. In this work, the first RA for EMS, named Max-Race,

is developed, so that it optimizes the extreme solution quality by automatically allocating the computational resources among an ensemble of problem solvers for a given problem instance. In Max-Race, significant difference between the extreme performances of any pair of models is statistically inferred via a parametric hypothesis test under the Generalized Pareto Distribution (GPD) assumption. Experimental results have confirmed that Max-Race is capable of identifying the best extreme model with high accuracy and low computational cost.

Furthermore, in machine learning, as well as in many real-world applications, a variety of MS problems are multi-objective in nature. MS which simultaneously considers multiple optimization criteria is referred to as MOMS. Under this scheme, a set of Pareto optimal models is sought that reflect a variety of compromises between optimization objectives. So far, MOMS problems have received little attention in the relevant literature. Therefore, this work also develops the first Multi-Objective Racing Algorithm (MORA) for a fixed-budget setting, namely S-Race. S-Race addresses MOMS in the proper sense of Pareto optimality. Its key decision mechanism is the non-parametric sign test, which is employed for inferring pairwise dominance relationships. Moreover, S-Race is able to strictly control the overall probability of falsely eliminating any non-dominated models at a user-specified significance level. Additionally, SPRINT-Race, the first MORA for a fixed-confidence setting, is also developed. In SPRINT-Race, pairwise dominance and non-dominance relationships are established via the Sequential Probability Ratio Test with an Indifference zone. Moreover, the overall probability of falsely eliminating any non-dominated models or mistakenly retaining any dominated models is controlled at a prescribed significance level. Extensive experimental analysis has demonstrated the efficiency and advantages of both S-Race and SPRINT-Race in MOMS.

To my beloved mother, you have always been my strength.

ACKNOWLEDGMENTS

This work was supported in part by National Science Foundation (NSF) grants No. 1200566 and No. 0525429. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the NSF.

Also, this work would have been simply impossible without the support of Dr. Michael Georgiopoulos and Dr. Georgios C. Anagnostopoulos, who have supervised my research activities since 2010. They have been extremely supportive and their advice on all aspects of my research and academic activities has been highly invaluable. I would like to thank them for their guidance, encouragement and friendship throughout my Ph.D. life. In addition, I also would like to thank my committee members, Dr. Annie S. Wu, Dr. Haiyan Hu, and Dr. David M. Nickerson, for their time and effort devoted to assessing my research work and for providing me with their helpful comments. Finally, I wish to thank all Machine Learning Lab members, Cong Li, Yinjie Huang, Niloofar Yousefi and Mahlagha Sedghi, who have shared sweat, tears, joy, and food with me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Problem Statement and Motivation

Model Selection (MS) is an important aspect in machine learning [56]. Typically, the performance of a machine learning model is heavily dependent on its parameters, which necessitates parameter tuning. Moreover, the No Free Lunch (NFL) theorem for optimization [138] supports the importance of MS: generally there is no algorithm that performs uniformly better than any other algorithm over all possible optimization problems. Consequently, the problem of choosing an appropriate algorithm to effectively solve the given type of optimization problem is and will remain an open problem. The volume of machine learning algorithms is overwhelming. Therefore, the practitioner in the field faces an important problem, as he/she is confronted with a plethora of machine learning algorithms to choose from in solving for a specified optimization problem of interest.

The task of MS is to identify a subset of models that are optimal in terms of selected optimization criteria. Typically, the MS problems fall into four categories [56]: feature selection, preprocessing approach selection, learning methodology selection and hyper-parameter selection. A common practice in MS is the Brute Force Approach (BFA), which selects models according to their performances on the same validation set of problem instances. Much research in MS focuses on alleviating the computational burden of this selection process. Recent attempts along this line include metaheuristics optimization, local search-based approaches, sequential model-based methods, portfolio algorithm approaches, and multi-armed bandits.

Racing Algorithms (RAs) are an active research area in MS. RAs refer to a series of algorithms that perform MS via statistical analysis of accumulated historical observations. RA is an iterative

procedure which starts off with an initial ensemble of competing models. The performances of the models are evaluated on a sequence of problem instances and the under-performing ones are eliminated once sufficient statistical evidence has been collected to demonstrate their inferiority. Eliminations occur periodically during the racing process and the optimal models will be retained at the end of racing. This process terminates if only one candidate model is retained, or when problem instances are exhausted. Due to the timely elimination of under-performing models, the computational resources are automatically allocated and more computational effort will be concentrated on fine exploitation of near-optimal models. Compared with the BFA, RAs save the computational resources wasted on evaluating unpromising models. In other words, RAs trade off some computational complexity for a reduced but acceptable likelihood that the models returned are indeed optimal among the given ensemble of candidate models.

The crux of designing a RA is how to discover inferior models as soon as possible based on accumulated statistical evidence. Several primary RAs have been put forward in the literature: Hoeffding Race (HR) [90] based on Hoeffding's inequality, Block Race (BRACE) [94] based on Bayesian statistics, Bernstein Race (BR) [93, 85] based on Bernstein's inequality, and F-Race [21, 22] based on Friedman test. Other instances of RA include A-Race based on ANOVA analysis, tn-Race based on paired t-test without any correction for multiple-test, tb-Race based on paired t-test with Bonferroni correction, racing based on Welch's t-test, etc. Ever since the first RA was proposed, RAs have been widely used in a variety of applications, such as algorithm configuration and hybridization [86, 60], robot control [57], and industrial applications [17, 31].

The MS problems have received great attention in the literature by different research communities, and various MS problems have been introduced in recent years. In this research, two key ideas are identified: Extreme Model Selection (EMS) and Multi-Objective Model Selection (MOMS).

Currently, all the RAs are offline per-set based algorithms where MS occurs before model deployment and models are compared in terms of their overall average performances on the common validation set of problem instances. Considering a different MS scenario of solving a given problem instance with limited computational budget, one wants to maximize the final solution quality achieved at the end by optimally distributing the overall computational resources among a set of problem solvers. Such problems are referred to as Extreme Model Selection (EMS), in which, model selection occurs during problem solving and no separate validation set is needed. Moreover, the models are compared in terms of their extreme/maximum performances.

Depending on the number of optimization criteria selected for measuring the goodness of models, MS problems can be single-objective, or multi-objective. Heretofore, all existing RAs were designed as Single-Objective Racing Algorithms (SORAs) for Single-Objective Model Selection (SOMS) which takes only one optimization criterion into consideration. For instance, HR was used to select optimal supervised learning model only in terms of their prediction accuracy. BRACE was adopted to minimize the Leave-One-Out Cross Validation (LOOCV) error of regression models with different subsets of features. However, in machine learning, as well as many real-world applications, many of the MS problems are multi-objective in nature. In supervised learning, for example, a good model needs to maintain a balance between prediction accuracy and model complexity. In multi-task learning, multiple related tasks are learned at the same time with a shared representation and hence it is expected to optimize the objectives of all constituent tasks simultaneously. We refer to MS with multiple optimization objectives as Multi-Objective Model Selection (MOMS) where a set of Pareto optimal models is expected to be returned with compromised optimization objectives to various extents.

However, neither EMS nor MOMS have received significant attention in the literature of RAs. Such limitations inspired us to design RAs for EMS and MOMS.

Contributions

This work aims at investigating racing approaches along two distinct directions: EMS and MOMS. More Specifically, the contributions of this research are summarized as follows:

- In this research, we proposed the first RA for EMS, named Max-Race, which optimizes the final solution quality by automatically allocating the computational resources among an ensemble of problem solvers when solving a particular problem instance. In Max-Race, the tail distribution of a model's performance is approximated via the Point Over Threshold (POT) approach rooted in Extreme Value Theory (EVT). Moreover, a parametric hypothesis test based on the Generalized Pareto Distribution (GPD) assumption is developed to identify significant difference between the extreme performances of a pair of models.

- In this work, we put forward the first Multi-Objective Racing Algorithm (MORA) in a fixed-budge setting, namely S-Race. In S-Race, the MOMS problems are properly addressed in the sense of Pareto-optimality. In the fixed-budget setting, the total number of available problem instances for validation is fixed and is known by the forecaster. It is most common in medical trials, for example, where the length of the test phase is determined beforehand (*e.g.* enrolls a fixed number of patients). In S-Race, the non-parametric pairwise sign test is utilized for pairwise dominance relationship inference. Besides, the overall probability of making any Type I errors is strictly controlled at a user specified level via a discrete Holm's Step-Down Family-Wise Error Rate (FWER) control method.

- A novel MORA based on Sequential Probability Ratio Test (SPRT) with an Indifference zone was developed, called SPRINT-Race. SPRINT-Race is the first MORA in the fixed-confidence setting, which is applicable for situations where the samples arrive sequentially in an online fashion. In the fixed-confidence setting, the goal of the forecaster is to min-

imize the number of validation instances required to achieve a fixed confidence about the optimality of the returned models. In online controlled experiments at large scale for data-driven decisions, for instance, selected beta users are enrolled sequentially in the test until it is, *e.g.* 95%, confident that the best test object(s) is(are) identified. In SPRINT-Race, a non-parametric ternary-decision dual-SPRT is employed to establish pairwise dominance and non-dominance relationships. In addition, a sequential Holm's Step-Down procedure is employed to control the overall probability of making any Type I or Type II errors at a user specified level.

Organization of the Dissertation

The rest of this dissertation is organized as follows: Chapter 2 provides an brief overview of MS and the state-of-the-art MS algorithms. Moreover, a detailed review of RAs is given along with discussions of their current limitations in Chapter 3. Chapter 4 introduces the first RA for EMS, named Max-Race, which features a parametric hypothesis testing based on EVT and GPD modeling via POT. The problem of MOMS are described in Chapter 5. Chapter 5 also presents the first MORA in a fixed-budget setting, namely S-Race, with details of the Sign Test and the Holm's Step-down Procedure. Chapter 6 discusses the details of a fixed-confidence MORA, namely SPRINT-Race, including the background knowledge of SPRT, the dual-SPRT for statistical inference of pairwise dominance and non-dominance relationships, and the sequential Holm's procedure. Finally, concluding remarks are provided in Chapter 7.

CHAPTER 2: BACKGROUND AND STATE-OF-THE-ART IN MODEL SELECTION

Model Selection Problem

Model Selection (MS) (also known as model configuration, algorithm selection, algorithm configuration, or parameter tuning) is a sequential decision making problem. The task of MS is to identify the optimal problem solvers for specific problem instance(s). MS problems have been studied extensively in machine learning. The No Free Lunch (NFL) theorem for optimization [138] demonstrates that there is no algorithm that performs uniformly better than any other algorithm over all possible optimization problems. Moreover, most machine learning algorithms possess a certain number of parameters, either categorical or numeric, which have significant impact on their performances. As a result, MS problems have gained much attention in recently years, and now there exists plenty of literature presenting varieties of MS approaches.

The MS problem is first formalized as the *algorithm selection* problem [109, 110], in which an algorithm serves as a selection procedure itself to select free parameters for a specific class of algorithms so as to fulfill predefined selection objectives. Such algorithm selection problem features three characteristics: problem space, algorithm space and performance measurement.

A more recent formal description of MS problem is provided in [21, 20]: the solution of a MS problem is to identify the optimal configuration parameters, from a set of candidate configurations, which optimizes the desirability of the parameters in terms of a typically infinite set of problem instances. The author emphasized that the difficulty might arise from the black-box features of the models.

Formally speaking, the MS problem can be stated as follows:

Given

- a model configuration space $\mathcal{M}$ (*e.g.* parameter settings, etc), in which each model $M \in \mathcal{M}$'s behavior on a given problem instance is totally defined.
- a problem instance space $\mathcal{I}$, where the problem instances are sampled.
- a performance metric c, which measures the performance of any $M \in \mathcal{M}$ on a subset of $\mathcal{I}$.

find a model $M \in \mathcal{M}$ that optimizes c.

MS encompasses a wide variety of problems [56], such as feature selection, data pre-processing method selection (*e.g.* normalization, dimension reduction), learning strategy selection (*e.g.* neural network, kernel method, nearest neighbor) and hyper-parameter selection. Examples of real-world applications can be found in computer vision, pattern recognition, industrial engineering, etc. A list of such applications can be found in [95].

Leading MS Methods

Although there are various MS methods, many of them can be considered as extensions or variations of several major ideas. In *model-based selection*, an explicit predictive model is built based on previously accumulated knowledge of the MS problem, which will be used later to generate new models. On the contrary, no model is built to map the relationship between model parameters and model performance in *model-free selection*. Another independent distinction is made between *static selection*, in which the pool of candidate models remains fixed after initialization; and *dynamic selection*, in which the ensemble of candidate models is updated periodically during

the entire selection process. Moreover, if selection happens during problem solving, it is *online selection*; and if a separate validation phase is required for selection, it is *offline selection*. In the latter type of selection, the chosen model is optimal for a specific class of problem instances. Hence, it offers the most value when the problem instances are similar. Online selection, on the contrary, offers the most value when the problem instances are diverse and different. Furthermore, if the selected model is optimal for just a specific problem instance, the MS problem is referred to as *per-instance selection* [67] or *specialist selection* [43]. On the other hand, if a set of problem instances is targeted, the corresponding MS problem is referred to as *per-set selection* or *generalist*.

The classification of existing state-of-the-art MS approaches will help understand how they are developed and when to apply each. In this research, we identify two key ideas, *static selection* and *dynamic selection*, to organize the current literature accordingly. Each division is further sub-divided into *model-free selection* and *model-based selection* according to whether an explicit model is built to guide the selection for optimal models. Figure 2.1 presents the proposed categorization of several major MS approaches which will be discussed in more detail in the following subsections.

In the remainder of this section, five classes of MS approaches are discussed. *Search-based Selection* utilizes powerful stochastic local search methods to search for good parameter configurations within potentially vast model space; *Metaheuristic Optimization* uses population-based algorithms as higher-level optimizers to select lower-level optimization algorithms; *Sequential Model-based Selection* builds a regression model to learn a mapping from the model space to the objective space based on which promising configurations are selected for further investigation; *Portfolio-based Selection* combines several algorithms, which run independently or are interleaved following a pre-determined policy, into a portfolio; and *Multi-Armed Bandit* features different model selection strategies and stopping conditions for sequential decision making problems defined by a set of actions.

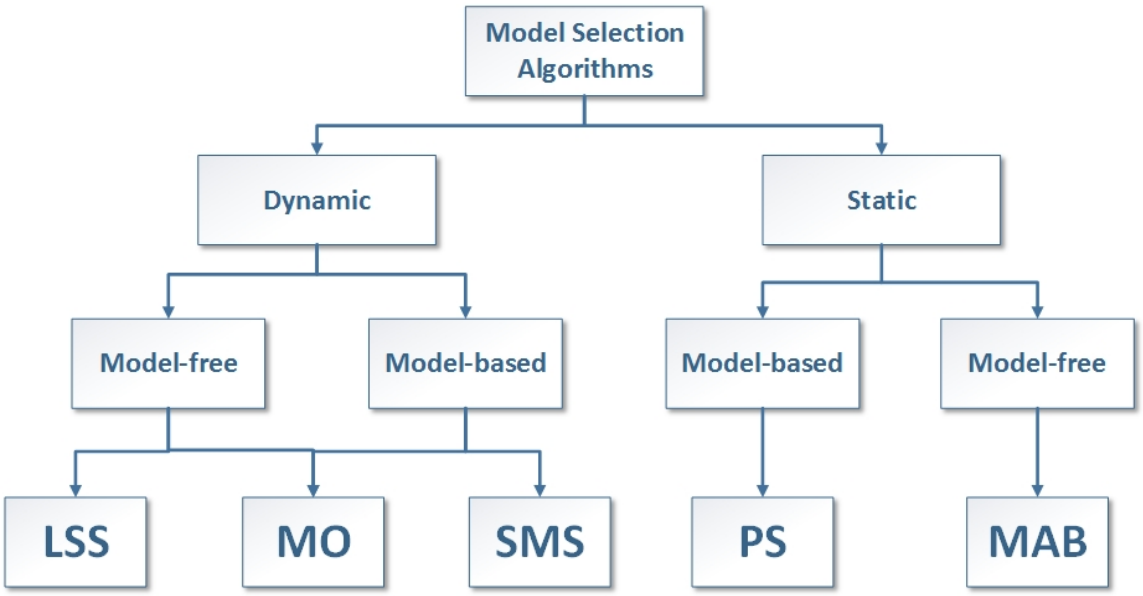


Figure 2.1: Classification of existing state-of-the-art MS algorithms, *i.e.* Local Search-based Selection (LSS), Metaheuristic Optimization (MO), Sequential Model-based Selection (SMS), Portfolio-based Selection (PS), and Multi-Armed Bandit (MAB).

Local Search-based Selection

The idea of performing local search in solving configuration problems is similar to searching for the optimal configuration in the configuration space manually. Given a randomly chosen configuration to start off with as an incumbent configuration, the user tries to improve its performance by modifying one parameter at a time. If no improvement is detected, the previous modification is rejected; otherwise, the new configuration is accepted as the new incumbent configuration. Similar to such iterative first-improvement search process, the key idea behind ParamILS [69, 72] is to utilize Iterative Local Search (ILS) methods [88] to search for good parameter configurations. ILS methods consist of iterating calls to build a sequence of locally optimal solutions either by perturbing the current optimal solution, or by applying local search or hill climbing methods starting from a modified solution. In ParamILS, a number of initial candidate solutions are evaluated

and the best of them is selected as the starting point for the ILS process. The ILS method used in ParamILS is based on one-exchange neighborhood in which only one parameter is modified at a time. After a certain number of permutations, the new configuration is accepted as the new starting point with prescribed probability, so as to maintain diversity to some degree.

The most straightforward variant of ParamILS is BasicILS [69, 72] in which all candidate permutations are evaluated by equal number of trials. When the number of benchmark instances is large, however, this approach is less than satisfactory since computational resources are wasted on underperforming configurations. One way of addressing such limitation is to perform multiple BasicILS simultaneously in parallel. FocusedILS [69, 72], another variant of ParamILS, addresses the same problem by assigning more computational resources to promising configurations. In FocusedILS, a small number of runs are performed for the initial ensemble of configurations. The candidate configurations are compared with each other on a common set of problem instances, and inferior ones are removed from further evaluation. Additional runs, namely *bonus runs*, are performed for the retained configurations. Moreover, the Probably Approximately Correct (PAC) property of FocusedILS is provided in [69]: the probability of finding the global optimal configuration in FocusedILS asymptotically approaches 1 for increasing number of iterations. Obviously, FocusedILS is more efficient than BasicILS, especially when the configuration space is large. To further speed up ParamILS, the Trajectory-preserving Adaptive Capping (TP capping) and Aggressive Adaptive Capping (Aggr capping) strategies were introduced [72] which adaptively determine the cutoff time of each trial by comparing the lower bound of the performance of one configuration and the upper bound of another comparison configuration's performance.

Metaheuristic Optimization

In metaheuristic optimization, a metaheuristic refers to a higher-level optimization algorithm that is used to select a lower-level optimization algorithm according to pre-selected optimization criteria. Metaheuristic algorithms mostly involve evolutionary computing techniques, such as evolutionary algorithms, ant colony optimization, particle swarm optimization, differential evolution, etc. The advantage of using metaheuristics is that few assumptions are required about the optimization problem to be solved.

As a metaheuristic, Genetic Algorithms (GAs) were employed in [83] for finding the optimal hyper-parameters of a kernel-based Support Vector Machine (SVM) model. In GA, a linear genotype encoding of the SVM's hyper-parameters is defined, including kernel exponents, kernel combination operators, kernel parameters and the regularization parameter. The individual quality is measured in terms of ten-fold cross validation error of some classification tasks.

In [54], a Particle Swarm Optimization (PSO) algorithm served as a search algorithm to explore the discrete parameter space of SVM configurations with given initial starting points. In their PSO implementation, each particle represents a parameter configuration of a SVM model. For regression problems, the fitness function is defined as the normalized mean square error of a ten-fold cross validation experiment.

In [114], the author took advantage of the Ant Colony Optimization (ACO) algorithm to select time delays for different dimensions in non-uniform embedding. For a given embedding problem, each individual ant represents a solution. Three fitness functions are designed for different optimization criteria (*i.e.* the minimal neighborhood distance criterion, the minimal false nearest neighborhood criterion, and the minimum description length criterion).

The Relevance Estimation and Value Calibration (REVAC) algorithm [96] was proposed to tackle the problem of choosing the best mutation operators and parameters of an Evolutionary Algorithm (EA). In REVAC, the joint probability distribution over all possible vectors is maintained. Shannon entropy is used to measure parameter relevance. Hence, the parameters that maximize Shannon entropy are always preferred. The objective of the REVAC algorithm is to maximize the expected performance of the specific EA.

In [117], a new Multi-objective Evolutionary Algorithm (MOEA), namely Multi-Function Evolutionary Tuning Algorithm (M-FETA), was proposed to select the optimal parameters of EAs on a collection of functions. Due to the stochastic nature of EAs, the performance of each candidate EA is assessed by looking at the performances of its nearest neighbors. To be more specific, the dominance relationship between a pair of models is inferred via a t-test in an objective-wise fashion based on the performance vectors of 2 sets of nearest neighbors. The experimental results illustrated that Pareto front solutions provide strong insight into the optimization problem itself.

Feature selection is a MS problem in which a small subset of relevant features is expected to be identified so that the same or better prediction accuracy is achieved. In [140], a multi-objective PSO approach was developed for feature selection in classification. Each particle is a d-dimensional binary vector where d is the number of total available features. Two conflicting optimization objectives are considered: minimizing the number of features and maximizing the classification accuracy. The obtained Pareto front can be further refined by the users according to their specific needs.

Sequential Model-based Selection

Sequential model-based selection is also widely known as Sequential Model-based Optimization (SMBO) in which a regression model is built explicitly to learn a mapping from the model space to

the objective space. Typically in model-based selection, the regression model is initialized based on the performances of the candidate models in preliminary runs. A set of potential configurations is sampled according to the predictions made by the regression model, and the optimal ones are retained which in turn help update the regression model to improve its prediction accuracy. The problems that SMBO addresses can be regarded as black-box optimization [63]: Given an unknown objective function f, either deterministic or stochastic, and a search space X, the goal is to find an input $x \in X$ that optimizes f by using surrogate models. The surrogate models in SMBO are used to approximate the actual performances of any models based on which promising configurations are selected for further investigation.

In [37], good parameter settings are selected via design of experiments. More specifically, a two-level (*i.e.* low level, high level) full factorial design method is adopted to construct a linear response surface. Gradient descent is applied afterwards, so that the gradient of the linear model guides the search direction for better parameter settings. However, the proposed method may not work well when the class of problems being studied is too broad for one set of parameter settings. Therefore, it is important to classify the problems based on their characteristics.

CALIBRA was proposed in [2] which utilizes Taguchi's fractional factorial experimental designs [125] coupled with a local search procedure. A non-linear response surface is built and updated occasionally. CALIBRA performs well when the number of parameters is small (*e.g.* ≤ 5) and the correlation among parameters is negligible. As emphasized by the author, CALIBRA is more beneficial in situations where the parameter values have significant impact on the performance of the algorithms being tuned.

The Efficient Global Optimization (EGO) algorithm proposed in [73] fits a response surface model by modeling the objective and constraint functions via noise-free Gaussian process, which is known as the "Design and Analysis of Computer Experiments (DACE) stochastic process model".

The EGO algorithm starts off with a set of initial points specified by a space-filling experimental design to fit a DACE model using maximum likelihood estimation. Then, cross-validated standardized residuals are examined to determine whether the model is satisfactory, and whether a log or inverse transformation is needed to refit the model. By maximizing the expected improvement using a branch-and-bound algorithm, new samples are generated and the DACE model is updated iteratively. However, the weakness of the DACE model is that it assumes that the parameters of the Gaussian process are known, which is generally not true in practice. Therefore, utilizing estimated parameters will result in a slight underestimation of the prediction error when the sample size is small.

The Sequential Kriging Optimization (SKO) method [64] was proposed as an extension of the EGO algorithm to deal with stochastic black-box optimization problems. In SKO, an initial kriging meta-model is built for the objective functions in which the response is assumed to be the sum of linear models with random errors. Cross validation is adopted to ensure the prediction accuracy of the kriging model. New samples are selected from the model using the Nelder-Mead Simplex algorithm [98], so as to maximize the augmented expected improvement. The advantage of the kriging model is that it is able to approximate a wide variety of objective functions via the noisy Gaussian process model. However, its approximation is poor, no better than a purely random search, if the sample size is small or the objective function is not smooth and behaves irregularly, since the assumption that the noise is normally distributed is violated. Moreover, the computation cost of fitting a kriging model is high when the dimensionality if too high (> 10).

Similar to SKO, the response is modeled by a stochastic regression model in Sequential Parameter Optimization (SPO) [16]. To be more specific, a Gaussian correlation function and a regression model with polynomial of order 2 have been adopted in the noise-free Gaussian process model. The Latin Hypercube Sampling (LHS) method [92] is used to determine an initial set of design points, while additional such points are sampled in order to maximize the generalized expected

improvement. As the author mentioned, the selection of an appropriate number of design points requires a balance between exploration and exploitation.

In [71], the two most influential methods for model-based optimization of noisy objective functions, namely SKO and SPO, were compared and the experimental evidence indicated that SPO is more robust than SKO. The impact of four key design factors of SPO were investigated, which were the initial design, data transformation, intensification mechanism and expected improvement criterion. According to the experimental findings, a variant of SPO was proposed, named SPO$^+$, which extends SPO with a Latin hypercube design, log-transformed response values, expected improvement criterion and a new intensification mechanism. The experimental results demonstrated that SPO$^+$ achieves state-of-art performance, and it is better than SPO on CMA-ES configuration optimization problems.

Another variant of SPO was developed in [68], named Time-Bounded Sequential Parameter Optimization (TB-SPO). It replaces the time-consuming LHS with interleaving random sampling. Moreover, a time-bounded intensification mechanism is employed which ensures that the cost of running the target algorithms does not exceed a pre-set limit. In addition, an approximate Gaussian process model, named the projected process approximation, is used to gain substantial improvements in reducing computational complexity. The author demonstrated that TB-SPO leads to significant improvement over SPO in terms of computational efficiency when applied on optimizing local search algorithms for SAT problems.

The intensification mechanism is the key component for any SMBO algorithms, which determines how many runs to allocate to each candidate configuration. Consequently, a simple instantiation of the general SMBO algorithm, namely Randomized Online Approximate Racing (ROAR), was designed with four design components including initialization, model fitting, configuration selection and intensification. ROAR is model-free since it performs random sampling and no model

is ever used. Accordingly, the Sequential Model-based Algorithm Configuration (SMAC) [70] method was introduced which utilizes ROAR's intensification mechanism. In SMAC, each candidate configuration is evaluated if and only if sufficient evidence has been collected to demonstrate its competitiveness. In order to support both numerical and categorical parameters, SMAC employs a weighted Hamming distance kernel function for Gaussian Process (GP) model and random forests as the regression model. SMAC also supports model selection for multiple problem instances with large mixed categorical and numerical parameters. GP is a common tool for modeling a learning algorithm's generalization performance on optimizing expensive functions, especially in Bayesian optimization [118]. In [70], a fully Bayesian optimization approach, which aims at maximizing the expected improvement over the current best value, was developed for hyper-parameter selection.

Portfolio-based Selection

The idea of Algorithm Portfolio (AP) is to combine several algorithms into a portfolio which will be running independently or be interleaved following a pre-determined policy. The need of AP is motivated by the NFL theory: since there is no unique optimal problem solver that is best for all the problems, combining different algorithms into a portfolio is able to improve the overall general performance (*e.g.* computation time, solution quality, etc).

Generally speaking, there are two aspects of AP design [28, 123]:

- selection - it starts by comparing the candidate algorithms following some simple rules based on preliminary runs and then the optimal one(s) is(are) selected to run for the remainder of the time.

- scheduling - the computational resources are allocated, statically or dynamically, among all candidate algorithms. The candidate algorithm run either in parallel, or alternatively following a switching paradigm.

In [28], both selection-type and switching-type AP approaches are studied by applying machine learning techniques to low-knowledge algorithm control. In the selection-type AP, a Bayesian classifier is trained to predict the algorithm that has the best performance on the new problem instances to be solved. In the switching-type AP, a reinforcement learning approach is used to allocate more runtime to promising algorithms.

Machine learning plays an important role in predicting algorithm performance. In [139], a per-instance AP for solving Satisfiability Problems (SATs), namely SATzilla, was developed that employs empirical hardness models as runtime predictors. By modeling several algorithms' runtime based on their historical performances and the characteristics of the given problem, the algorithm with the best predicted performance will be selected as the problem solver for the new problem instance(s) to be solved.

Other similar work in the field of Constraint Programming (CP) were *CPHydra* [100] and *claspfolio* [50]. In *CPHydra*, the performance of each algorithm is learned by solving a large validation set of problem instances. When a new problem instance is set to be solved, k-Nearest Neighbor is used to find similar validation problem instances according to their features. Then the algorithm with the best overall performance, with respect to the set of similar validation instances, will be selected to solve the new problem instance. *claspfolio* also takes advantage of instance features for algorithm selection. The idea of *claspfolio* is to train classifiers on features of benchmark instances in order to predict the best solver from a pre-selected algorithm portfolio for the new problem instances.

Distinguished from previous work, Modular Architecture for Probabilistic Portfolios (MAPP) [116] assumes that the runtime of each individual solver on a given problem instance follows a fixed but unknown distribution. It first estimates the Run Time Distributions (RTDs) by building a generative model. Then it incorporates feature information to predict problem instances on which the solvers exhibit similar performances. Finally, an execution schedule is provided for an ensemble of problem solvers via dynamic programming approximation.

Multi-Armed Bandit

Multi-Armed Bandit (MAB) problems refer to a wide variety of sequential allocation problems with an exploration-exploitation tradeoff. Depending on the assumptions of the payoff/reward process, there are three fundamental formalizations of MAB problems: stochastic, adversarial, and Markovian. In our research, we are only interested in stochastic MABs.

The stochastic MAB problems can be formalized as follows: Given K arms with K unknown payoff probability distributions $p_1, p_2, \cdots, p_K$ respectively, at each time t, the forecaster chooses an arm $I_t \in \{1, \cdots, K\}$ to pull once, and obtains an observation of the identically and independently distributed (i.i.d.) payoff, denoted as $X_{I_t} \sim p_{I_t}$. If μ_i denotes the means of p_i, for $i = 1, \cdots, K$, the index of the arm with the best expected payoff k^* and the best expected payoff μ^* are defined as

$$k^* \triangleq \underset{k \in 1, \cdots, K}{\operatorname{argmax}} \mu_k \ , \ \mu^* \triangleq \mu_{k^*} \tag{2.1}$$

A wide variety of stochastic MAB algorithms are investigated in the literature, which are distinguished by their optimization objectives.

Exploitation versus Exploration Dilemma

A common criterion for measuring the performance of a MAB algorithm is the expected cumulative regret, defined in Equation (2.2). It refers to the expected value of the difference between the cumulative rewards of an ideal MAB strategy and the rewards of the proposed MAB algorithm.

$$\begin{aligned}\mathbb{E}\left[R_T\right] &\triangleq \max_{i=1,\cdots,K} \mathbb{E}\left[\sum_{t=1}^{T} X_{i,t} - \sum_{t=1}^{T} X_{I_t,t}\right] \\ &= T\mu^* - \sum_{t=1}^{T} \mathbb{E}\left[\mu_{I_t}\right]\end{aligned} \tag{2.2}$$

The fundamental dilemma in MAB is the *exploitation-exploration trade-off*. Exploitation refers to the tendency of pulling the current best arm, while exploration denotes the behavior of exploring other arms to gain new knowledge in order to increase the probability of identifying the actual best arm. A good strategy for tackling MAB problems, aiming at minimizing $\mathbb{E}\left[R_T\right]$ in Equation (2.2), is all about maintaining a balance between exploitation and exploration. Such exploitation versus exploration dilemma is also faced in Reinforcement Learning (RL) [124].

A well-known result of Lai and Robbins [79] shows that the expected cumulative regret of the optimal MAB algorithm grows at least logarithmically, that is $\mathbb{E}\left[R_T\right] = \mathcal{O}(\log T)$ for large T. In the MAB literature, theoretical regret bounds have been established for various bandit algorithms.

The most popular principle of many stochastic sequential decision-making problems is the so-called *optimism in face of uncertainty* [79]. In face of uncertainty, the forecaster accumulates observations from previous actions to estimate the desirableness of a set of potential actions. Almost all existing MAB algorithms aiming at minimizing the expected cumulative regret are based on this simple heuristic principle.

The Upper Confidence Bound (UCB) family of bandit algorithms [10] is a classical implementation of the *optimism in face of uncertainty* principle for bounded rewards. At each iteration of the simplest UCB-1 algorithm, the arm with the best upper confidence bound of the expected reward is always pulled. To be more specific, at time t, the UCB-1 strategy always plays the arm with index I_t, such that

$$I_t \triangleq \underset{i\in 1,\cdots,K}{\operatorname{argmax}} \left\{ \hat{\mu}_{i,t-1} + \sqrt{\frac{2\log t}{T_i(t-1)}} \right\} \tag{2.3}$$

where $\hat{\mu}_{i,t-1} = \frac{1}{T_i(t-1)} \sum_{s=1}^{T_i(t-1)} X_{i,s}$, in which $X_{i,s} \in [0,1]$ is an i.i.d. sample of distribution p_i, and $T_i(t-1)$ is the number of times that the i^{th} arm has been played up to time $t-1$.

In [10], it was shown that, via Hoeffding's inequality, at any time t, the expected regret of UCB-1 is bounded by:

$$R_t \leq 8 \sum_{i:\mu_i<\mu^*} \left(\frac{\ln t}{\Delta_i}\right) + \left(1+\frac{\pi^2}{3}\right) \sum_{i=1}^{K} \Delta_i \tag{2.4}$$

Hence it is said that the UCB-1 algorithm solves the MAB problem because it achieves the the logarithmic lower regret bound of Lai and Robbins [79], up to a multiplicative constant.

Intuitively, tighter confidence bounds imply smaller regret. The regret bound of UCB-1 can be improved in many ways. The UCB-V algorithm [9] utilizes Bernstein's inequality, which considers the variance of the distributions as well as the expected value, to achieve a smaller regret. A similar idea was developed in [91]. Moreover, the KL-UCB algorithm was presented in [48] based on Kullback-Leibler divergence. At each iteration of KL-UCB, only the arm with the maximum

upper bound of the deviations of the exponential moments is pulled. It was shown that the KL-UCB algorithm achieves a uniformly better regret bound than UCB and its variants.

Unlike optimistic algorithms based on concentration inequalities, Thompson Sampling (TS), also called Bayesian Bandits, uses Bayesian sampling in which any arm is sampled according to the posterior probabilities that it maximizes the expected reward. In [30], some empirical results were provided to demonstrate the superiority of TS over UCB on bandit problems. The basic idea of TS is similar to probability matching in which the probability of each arm being drawn is proportional to its probability of being the optimal arm. While this heuristic, originally appeared in [128], has been used for many years as bandit problem solvers, it was not until recently that theoretical analysis of its performance was achieved for Bernoulli bandits [4]. More specifically, the expected cumulative regret at time T is $\mathcal{O}\left(\frac{\ln T}{\Delta}+\frac{1}{\Delta^3}\right)$ for the 2-armed bandit, where Δ defined as $\mu_1-\mu_2$ $(\mu_1>\mu_2)$, and it is $\mathcal{O}\left(\left[\left(\sum_{i=2}^{K}\frac{1}{\Delta_i^2}\right)^2\ln T\right]\right)$ for aMAB, where $\Delta_i=\mu^*-\mu_i$. There are some discrepancies between analyzing TS-based MAB and UCB-based MAB, but it was shown in [30] that the logarithmic bounds provided above are optimal in terms of Δ_i with some constant factors.

In [78], the authors proposed a TS algorithm for exponential family bandits using Jeffreys prior, which is an asymptotically optimal algorithm for bandit models and is a competitive alternative to KL-UCB for exponential family bandits. The cornerstone of their proof is a finite time concentration bound for posterior distributions in exponential families.

In [5], the author provides two examples of TS: one with Bernoulli distribution and Beta priors, and the other with Normal distribution and Normal priors. The authors used novel martingale-based analysis techniques to find the optimal problem-dependent bound of $\mathbb{E}\left[R_T\right]$ and the first near-optimal problem-independent bound, which are $(1+\epsilon)\sum_i\frac{\ln T}{\Delta_i}+\mathcal{O}\left(\frac{K}{\epsilon^2}\right)$ and $\mathcal{O}\left(\sqrt{KT\ln T}\right)$ respectively.

Due to its outstanding performance, TS is applied later on solving complex bandit problems [55] in which multiple arms are pulled simultaneously each time. The authors are able to show that the regret bound scales logarithmically with time.

There are two distinct settings of stochastic MAB in the literature. In the frequentist setting, the reward distributions of the arms are determined by unknown but deterministic quantities, and the goal of the MAB algorithm is to achieve the best parameter-dependent performance. On the contrary, in Bayesian approaches, each arm is characterized by a parameter which is endowed with a prior distribution. Consequently, the Bayesian approaches are evaluated by their average performances over all possible problem instances weighted by the prior information. In [77], Bayesian theory is integrated into the UCB family, termed Bayes-UCB. It is similar to TS which draws samples from the payoff distributions so as to select the arms with probability proportional to their posterior probability of being optimal. The authors showed that the Bayes-UCB algorithm is optimal in the sense that it achieves the logarithmic lower bound of Lai and Robbins [79]. However, finite-time regret bounds and asymptotic optimality of Bayes-UCB have only been provided for binary MABs.

In [41], the author studied how to extend UCB to solve multi-objective bandit problems, either by adopting a linear or non-linear scalarization function, or using the concept of Pareto dominance. The upper bound of the expected Pareto regret of the proposed Pareto UCB-1 is obtained directly following the analysis of UCB-1 and is given as $\sum_{i \notin A^*} \frac{8 \log\left(n \sqrt[4]{D|A^*|}\right)}{\Delta_i} + \left(1 + \frac{\pi^2}{3}\right) \sum_{i \notin A^*} \Delta_i$, where D is the number of objectives and A^* is the set of Pareto optimal arms in terms of their expected payoffs.

Pure Exploration Problems

The Best Arm Identification (BAI) problem is a variant of the MAB problem. Instead of minimizing the cumulative regret over the course of problem solving, the goal of BAI is to find i^*, the index of the optimal arm. Let us denote the index of the arm returned by the forecaster after T iterations as J_T. A simple regret of measuring the performance of BAI algorithm is

$$r_T = \mu^* - \mu_{J_T} \tag{2.5}$$

Moreover, the error probability as defined in Equation (2.6) and denoted as e_T, is also a commonly used performance metric for BAI, which refers to the probability that a non-optimal arm is returned.

$$e_T = \mathbb{P}\left(J_T \neq i^*\right) \tag{2.6}$$

If MAB problems for cumulative regret minimization are regarded as online MS problems which require a balance between exploration and exploitation, the BAI problems can be regarded as offline MS problems which emphasize more on exploration.

In [45], Action Elimination (AE) was proposed which eliminates sub-optimal arms as early as possible to reduce the total number of samples. Three (ϵ, δ)-PAC style AE algorithms are designed for BAI: a naive algorithm, Successive Elimination (SE), and Median Elimination (ME) with sample complexities $\mathcal{O}\left(\frac{K}{\epsilon^2}\log\frac{K}{\delta}\right)$, $\mathcal{O}\left(\log\frac{K}{\delta}\sum_{i=2}^{K}\frac{1}{\Delta_i^2}\right)$ and $\mathcal{O}\left(\sum_{i=2}^{K}\frac{\ln\left(\frac{K}{\delta\Delta_i}\right)}{\Delta_i^2}\right)$ respectively. In SE, the arms with minimal average rewards will be eliminated. On the other hand, in ME, any arms with average rewards below the media are removed.

In [7], two exploration strategies in the fixed budget setting were suggested: a highly exploring UCB policy with error probability upper bounded by $2TK\exp\left(-\frac{2a}{25}\right)$, and a parameter-free Successive Rejects (SR) algorithm with error probability no larger than $\frac{K(K-1)}{2}\exp\left(-\frac{T-K}{\overline{\log}(K)H_2}\right)$. Moreover, the hybridization of them leads to a new algorithm, called adaptive UCB-E. Although analytical results of its optimality are not provided, the experimental results demonstrated its superiority.

In [47], a unified approach for BAI is proposed, namely Unified Gap-based Exploration (UGapE) algorithm, which uses a unique arm-selection strategy for both fixed budget setting and fixed confidence setting. They only differ in the way of computing the confidence interval. Analytical proof about the upper bound of the simple regret, success probability and sample complexity are provided for both versions of the unifying UGapE algorithm.

In [27], a novel Successive Accepts and Rejects (SAR) algorithm was proposed for top-m arms identification as an extension of SR in which the best m arms are expected to be returned. Let us assume that $\mu_1 > \cdots > \mu_K$ for K arms, and the arms with indices $J_1, \cdots, J_m$ are returned. In top-m arms identification, the probability of misidentification is defined as follows:

$$e_T = \mathbb{P}\left(\{J_1, \cdots, J_m\} \neq \{1, \cdots, m\}\right) \tag{2.7}$$

In SAR, each arm could be accepted as a top-m candidate or be excluded from being a top-m candidate if sufficient evidence is collected. The evidence is dependent on the empirical means of the payoffs. It was shown in the paper that the error probability of SAR for top-m arms identifica-

tion is upper bounded by $2K^2 \exp\left(-\frac{T-K}{8\overline{\log}(K)H_2^{\langle m\rangle}}\right)$, where $H_2^{\langle m\rangle}$ is a commonly used complexity measure of the multiple identification bandit problem.

$$H_2^{\langle m\rangle} \triangleq \max_{i\in\{1,\cdots,K\}} i\left(\Delta_i^{\langle m\rangle}\right)^{-2},\ \Delta_i^{\langle m\rangle} \triangleq \begin{cases} \mu_i - \mu_{m+1} & \text{if } i \le m \\ \mu_m - \mu i & \text{if } i > m \end{cases} \tag{2.8}$$

In [74], a PAC-style m-best arms identification, namely Explore-m, is studied and three algorithms were proposed. In their formulation, the objective is to minimize the sample complexity necessary to reliably identify (with probability of at least $1-\delta$) m ϵ-optimal arms. An arm is called ϵ-optimal if its expected reward is within ϵ of μ^*. The corresponding algorithms are called (ϵ, m, δ)-optimal algorithms. Generally, the sample complexity is a function of K, m, ϵ and δ. These algorithms are called Fixed-Sample-Complexity algorithms, since they guarantee to achieve the predefined probability of correct identification in the worst case when the differences between $\mu_i, i \in \{1, \cdots, K\}$ are small. The sampling efficiency can be improved in practice by adapting the relative spacing between arms and their variances.

Distinguished from their previous research which is restricted to the worst-case sample complexity, Kalyanakrishnan & Stone developed a set of PAC-based Variable-Sample-Complexity algorithms for Explore-m bandit problems [75]. The proposed algorithms distinguish between the stopping rule and the sampling strategy. The accuracy and efficiency of the algorithms only depend on the proposed stopping rule which could be coupled with any sampling strategy. It meets the PAC correctness requirement with the best-known worst-case sample complexity, and its expected sample complexity on a bandit instance, which can be substantially lower than the worst-case sample complexity, depends on the difficulty of that instance. Based on different confidence intervals, the Lower Upper Confidence Bound (LUCB)-1 and LUCB-2 algorithms were developed in [75], both of which are similar to the UCB algorithm with a greedy sampling strategy.

Max-K Armed Bandit

The Max K-Armed Bandit problem was first introduced in [33]. The objective of Max K-Armed Bandit is to allocate trials among the arms so as to maximize the best single reward received at the end. The objective could be formally stated as $\max \max_{i=1,\cdots,K} \max_{j=1,\cdots,T_i} R_j(p_i)$, where $R_j(p_i)$ refers to the reward of the j^{th} pull of arm i with underlying reward distribution p_i. The Double Exponential Sampling strategy was proposed in which the number of pulls of the observed best arm would grow double exponentially with respect to the number of pulls of the second best. The proposed strategy is based on the assumption that $p_i, i = 1, \cdots, K$ follows a Gumbel distribution, one of three Generalized Extreme Value (GEV) distributions [34].

In [120], a PAC-style Max K-Armed Bandit algorithm was proposed which follows an *explore-and-exploit* pattern. Assuming that each arm, when pulled, returns a random reward sample drawn from a GEV distribution, the proposed approach first pulls each arm $\mathcal{O}\left(\ln \frac{1}{\delta} \frac{\ln(T)^2}{\epsilon^2}\right)$ times to estimate its expectation. Then the arm with the maximum estimated expectation value will be pulled for the remaining trials. The author is able to show that, with probability at least $1 - \delta$, the expected maximum payoff received over a series of n trials is within ϵ of the real optimal.

All previous approaches dealing with Max K-Armed Bandit are parametric approaches with GEV distribution assumptions about the rewards. If such assumptions are invalid, however, parametric approaches cannot work well. Therefore, non-parametric approaches are developed [121, 122]. The motivation behind these approaches is straightforward: maximizing the expected maximum reward obtained at the end is equivalent to maximizing the probability of obtaining a solution that is $> t$ where t is a prescribed threshold.

In [121], an interval estimation algorithm was proposed based on Chernoff's inequality. When applied as a thresholded Max K-Armed Bandit algorithm, it has additive regret of $\mathcal{O}\left(Tp^*k \ln T\right)$

where $p^* \triangleq \max_i p_i$ with p_i referring to the probability that the reward of the i^{th} arm exceeds t and T is the total number of trials. Similarly, a Chernoff Interval Estimation algorithm was proposed in [122] which is applied on Max K-Armed Bandit as Threshold Ascent algorithm.

A semi-parametric Max K-Armed Bandit algorithm was developed by Carpentier *et al.* [29], named ExtremeHunter. It assumes that the reward distribution is heavy-tailed and only the second order Pareto assumption [58] is made. Moreover, ExtremeHunter selects models based on an *optimism in face of uncertainty* principle. The author is able to show that the proposed strategy performs almost as well as the ideal strategy and is able to detect the arm with the heaviest tail with high precision.

CHAPTER 3: RACING ALGORITHM

In this chapter, we discuss a class of algorithms, namely Racing Algorithms (RAs), which are popular Model Selection (MS) algorithms in machine learning. Simply speaking, RAs can be regarded as elimination-type Multi-Armed Bandit (MAB) algorithms. To be more precise, the task of RAs is to identify the best model(s) with respect to specific optimization criteria from an ensemble of models, while guaranteeing certain level of confidence with minimal computational cost. This family of approaches is also the main focus of this dissertation.

The Racing Approach

As their name suggests, RAs are iterative procedures for MS in which the under-performing models are automatically excluded from further consideration as early as possible during racing. RAs start off with an ensemble of candidate models which are repeatedly evaluated during racing on a given validation set of problem instances. Under-performing models will be eliminated immediately once sufficient statistical evidence has been collected to support their inferiority. When only one model is remained in racing, or when the total computational resources are exhausted, racing stops and all the remaining models are returned as optimal ones in terms of pre-selected optimization criteria. The general framework of a RA is depicted in Table 3.1.

The goal of racing is to allocate computational resources optimally among the candidate models, so that more computational efforts will be concentrated on fine-tuning the superior models. The overall computational effort is reduced when compared with the Brute Force Approach (BFA) which distributes the computational resources evenly among all competing models. A visual representation of the computational efforts needed by a RA and the BFA is given in Figure 3.1.

Table 3.1: General Framework of Racing Algorithm

Step 1	Randomly select problem instance(s) to evaluate the performances of candidate models.
Step 2	Identify statistically significant difference between candidate models.
Step 3	Remove all inferior models from race.
Step 4	Go to Step 1, if available problem instances are not exhausted, or more than one model is left.

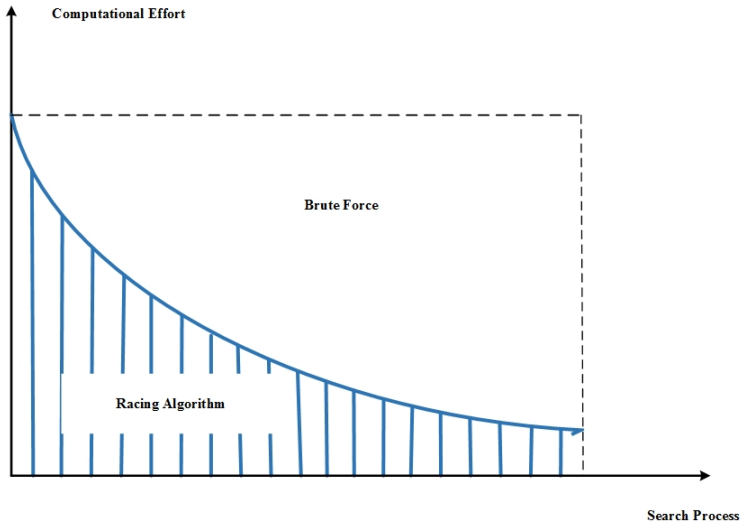


Figure 3.1: A visual representation of computational effort needed by RA and BFA.

A racing strategy is characterized by its way of statistically inferring significant differences in models' performances with respect to pre-selected optimization criteria for particular applications. Several leading RAs in the literature are summarized in Table 3.2

Leading Racing Algorithms

Hoeffding Race (HR) [90] is the first racing algorithm which is applied on selecting the optimal memory-based learning models in terms of their prediction accuracy (*i.e.* Leave-One-Out Cross

Table 3.2: Leading RAs in the Literature

Name	Year	Statistical Test	Criterion of Goodness	Application
HR	1994	Hoeffding's Inequality	Prediction Accuracy	Classification & Function Approximation
BRACE	1994	Bayesian Statistics	Cross-validation Error	Classification
F-Race	2002	Friedman Test	Function Optimization	Parameter tuning of Max-Min-Ant-System
BR	2008	Bernstein's Inequality	Function Optimization	Policy Selection in CMA-ES

Validation (LOOCV) error). The proposed HR algorithm combines the robustness of BFA and the computational efficiency of gradient descent. HR derives its name from Hoeffding's inequality [61]. It assumes that the LOOCV errors $x_i \in [a, b], i = 1, \cdots, n$ of a model are identically and independently distributed (i.i.d.) random variables (RVs) with true mean E_{true}. Given its empirical mean $E_{emp} = \frac{1}{n}\sum_{i=1}^{n} x_i$, the probability that the difference between E_{emp} and E_{true} is within a distance of ϵ is bounded by

$$Pr\left\{|E_{emp} - E_{true}| > \epsilon\right\} < 2e^{-\frac{2n\epsilon^2}{(b-a)^2}} \tag{3.1}$$

Therefore, the $1-\delta$ confidence bound of E_{true} is easily obtained as $[E_{emp} - \epsilon, E_{emp} + \epsilon]$ with $\epsilon \triangleq \sqrt{\frac{(b-a)^2 \log(2/\delta)}{2n}}$. When x_i refers to the LOOCV error, the model with minimum E_{true} is expected to be returned as the optimal model. As a result, a model whose lower bound of E_{true} is greater than the upper bound of any other competing model will be removed from racing. The algorithm repeats until only one model is left, or until we run out of problem instances. At the end, it returns a set of models whose error rates are close to each other (within ϵ). Given m initial

models and a validation set of size n, the author is able to show that when $\delta = \frac{\Delta}{nm}$, the probability that HR eliminates the optimal model is $\leq \Delta$, which is referred to as the correctness of HR.

A parametric analogue of HR was proposed in [94], known as Block Race (BRACE), which is also applied on selecting memory-based (or instance-based) models based on LOOCV. One key characteristic of BRACE is the use of blocking to quickly eliminate near-identical models. Moreover, as a parametric RA, it assumes the LOOCV errors of the i^{th} model over the validation data follows a Gaussian distribution with an unknown mean e_i^* and variance σ_i^*. Given empirical mean $\hat{e}_i^*$ and empirical variance $\hat{\sigma}_i^*$, a posterior distribution is estimated via Bayesian statistics (*e.g.* the Welch approximation to the Behrens-Fisher problem [136]). In BRACE, the i^{th} model is eliminated if and only if

$$\exists j(i \neq j),\ s.t.\ Pr\left\{e_i^* < e_j^* - \gamma | e_i(1), \cdots, e_i(T), e_j(1), \cdots, e_j(T)\right\} < \delta \tag{3.2}$$

To be more specific, a new RV $h_{ij}^* \triangleq e_i^* - e_j^*$ is introduced by a statistical technique called blocking [24]. The probability that $h_{ij}^* < -\gamma$ is calculated based on the following statistics:

$$\hat{\mu}_{ij}^h(n) \triangleq \frac{1}{n}\sum_{k=1}^{n}\left(e_i(k) - e_j(k)\right),\ \ \hat{\sigma}_{ij}^h(n) = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}\left[e_i(k) - e_j(k) - \hat{\mu}_{ij}^h\right]^2} \tag{3.3}$$

If $\mathbb{P}\left(h_{ij}^* < -\gamma\right) < \delta$ where δ is a user-specified significance level, the j^{th} model will be removed from racing.

When the candidate models demonstrate similar performances, it is hard for HR to eliminate relatively inferior models efficiently. To overcome the limitation of HR, Bernstein Race (BR) [93, 91]

was developed based on the empirical Bernstein bound [8]. The empirical Bernstein bound, which states that, with probability at least $1-\delta$, we have

$$|E_{emp} - E_{true}| \leq \bar{\sigma}_n \sqrt{\frac{2\log\left(\frac{3}{\delta}\right)}{n}} + \frac{3R\log\left(\frac{3}{\delta}\right)}{n} \tag{3.4}$$

where $R \triangleq b-a$, $\bar{\sigma}_n \triangleq \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}_n)^2$ refers to the empirical standard deviation of $\{x_i\}_{i=1,\cdots,n}$. Since R is decreasing at a rate of n^{-1}, this bound will mainly depend on the first term when $\bar{\sigma}_n \ll R$, implying that the empirical Bernstein bound will quickly become much tighter than the Hoeffding bound.

A generalization of BR was later proposed in [85] which uses a jackknife estimate to replace the unknown variance. The proposed BR is more widely applicable than HR, BRACE and BR when the statistics of interest are not simple empirical averages of independent observations (*e.g.* U-statistics, discrete entropy, cross-validation, etc). The proposed method constructs empirical Bernstein bound based on a jackknife estimates of the variances of the statistics of interest. Empirical simulation results in [85] have demonstrated that the asymptotic Bernstein bound results in significant speedups.

In [21], the most popular racing procedure was proposed, namely F-Race, which has the advantages of both HR (*i.e.* non-parametric) and BRACE (*i.e.* block design). The author first provides a formal definition of the metaheuristic configuration problem: Given the set of candidate configurations Θ, the set of problem instances I and its probability measure P_I, function t measuring the computation time of each problem instance, the cost space C for all possible combinations of $\theta \in \Theta$ and $i \in I$, and its probability measure P_C, the objective of the configuration problem is to find θ^* such that

$$\theta^* \triangleq \text{argmin}_\theta \, C(\theta) \tag{3.5}$$

where $C(\theta) = C\left(\theta|\Theta, I, P_I, P_C, t\right)$ is the cost with respect to θ.

Next, F-Race was developed based on the Friedman test [35] for multiple comparisons which is also known as Friedman two-way analysis of variance by ranks. In F-Race, the i^{th} model will be assigned a rank R_{ij} based on its performance on the j^{th} problem instance. Then the Friedman test detects if all rankings of the candidates are equally likely or not using the following test statistic:

$$T = \frac{(n-1)\sum_{i=1}^{n}\left(R_i - \frac{k(n+1)}{2}\right)^2}{\sum_{j=1}^{k}\sum_{i=1}^{n} R_{ij}^2 - \frac{kn(n+1)^2}{4}} \tag{3.6}$$

where n is the number of current models, k is the number of problem instances seen so far, and $R_i = \sum_{j=1}^{k} R_{ij}$.

If the null hypothesis is rejected, implying that at least one model is superior to at least one other, a pairwise t-test is employed to identify the under-performing models to eliminate. The i^{th} and h^{th} model are statistically different if

$$\frac{|R_i - R_h|}{\sqrt{\frac{2k\left(1-\frac{T}{k(n-1)}\right)\left(\sum_{j=1}^{k}\sum_{i=1}^{n} R_{ij}^2 - \frac{kn(n+1)^2}{4}\right)}{(k-1)(n-1)}}} > t_{1-\frac{\alpha}{2}} \tag{3.7}$$

where $t_{1-\frac{\alpha}{2}}$ is the $1-\frac{\alpha}{2}$ quantile of the Student's t distribution.

Originally in F-Race, full factorial design was adopted to initialize the candidate configurations. However, the drawbacks of a full factorial design are: i) the levels of each parameter need to be explicitly determined by the experimenter; and ii) the number of initial configurations grows exponentially with the number of parameters. To overcome these drawbacks, a F-Race based on random sampling design was proposed in [12]. Moreover, an Iterative F-Race (I/F-Race) was developed

which incorporates the idea of model-based search. In I/F-Race, a simple probabilistic model biasing the next sampling towards better configurations is maintained and updated periodically during racing. Whenever inferior models are removed, they are replaced with newly sampled ones.

Other instances of RAs include A-Race which is based ANOVA analysis [143], tn-Race which is based on t-tests without multiple tests correction, tb-Race which is based on t-tests with Bonferroni correction [21], and a racing algorithm which is based on Welch's t-test [80], etc.

Limitations of Existing Racing Algorithms

Ever since the first RA was proposed, RAs have received significant attention and have been applied in various contexts, such as algorithm configuration and hybridization [86, 60], robot control [57], and industrial applications [17, 31].

As discussed above, all existing RAs are designed as Single-Objective Racing Algorithms (SORAs) for Single-Objective Model Selection (SOMS) in which only one optimization criterion is considered for measuring the goodness of models. Moreover, they are offline MS algorithms and the returned models are optimal in terms of their overall average performance on a validation set of problem instances. However, adaptations of MS to several other important problems/settings remain unaddressed by existing racing approaches, or have, so far, received little attention in the literature. This work aims to investigate racing approaches along two distinct directions: Extreme Model Selection (EMS) and Multi-Objective Model Selection (MOMS). EMS and MOMS are important problems in machine learning, and they are also natural frameworks of many real-world applications.

Extreme Model Selection

In extreme model selection, the goal is to detect outstanding events or estimate extreme values. To be more specific, given a particular problem instance and fixed computation budget, one may want to maximize the final solution quality that is achieved by automatically allocating the computation budget among different problem solvers.

Let us assume that we have K candidate solvers and the i^{th} model returns a solution $x_{i,t} \sim P_i$ for the t^{th} iteration. As stated in [29], the objective of EMS is to minimize the extreme regret $\mathbb{E}[r_T]$ which is defined as follows:

$$\mathbb{E}[r_T] = \max_{i \in K} \mathbb{E}\left[\max_{t \leq T} x_{i,t}\right] - \mathbb{E}\left[\max_{t \leq T} x_{I_t,t}\right] \tag{3.8}$$

where I_t refers to the index of the model sampled at the t^{th} iteration. In other words, we want to allocate most of the pulls to the model with the highest expected maximum value $\mathbb{E}\left[\max_{t \leq T} x_t\right]$.

EMS is a natural framework of many applications. In many domains, such as outlier detection [1], security control [105], and disease surveillance [97], outstanding events or extreme values require special attention. For example in [29], the author is interested in detecting anomalies from different sources in network intrusion detection. The need for EMS is also motivated in practice when tackling combinatorial optimization problems (*e.g.* vehicle routing, Knapsack problems, Traveling Salesman problems) using several number of stochastic search heuristics [120].

However, EMS problems remain unaddressed by current racing approaches. In this research, we propose the first RA for EMS which is capable of allocating the overall computational resources optimally among multiple candidate models so as to optimize the extreme solution achieved at the end. RA for EMS is the subject of Chapter 4 which details the analysis of such racing procedure

and provides experimental results to demonstrate its efficiency. Distinct from existing RAs, RAs for EMS are online MS approaches in which model comparison and selection occur during problem solving and therefore requires no validation set.

Multi-Objective Model Selection

In machine learning, as well as many real-world applications, a great variety of MS problems are multi-objective in nature. In supervised learning, for instance, a good supervised learning model requires a tradeoff between prediction accuracy and model complexity. Complex models usually result in high prediction accuracy but require large computational resources. Simple models are sometimes preferred because they are easy to implement and have better generalized performance. In multi-task learning, for example, multiple related tasks are learned at the same time with a shared representation. The goal of multi-task learning is to optimize the objective functions of all learning tasks simultaneously. Moreover, Recommender System (RS) construction for Top-N recommendation is also a multi-objective optimization problem, regarding accuracy, novelty and diversity [130, 107] as the measurements of model's goodness.

MS with multiple optimization criteria is referred to as MOMS. In MOMS, a set of Pareto optimal models is expected to be returned with compromised optimization objectives to various extents. So far, MOMS problems have received little attention in the relevant literature. In this work, we propose the first two Multi-Objective Racing Algorithms (MORAs) which address MOMS problems in the proper sense of Pareto optimality. The details of the proposed MORAs are given along with experimental analysis in Chapter 5 and Chapter 6. In the proposed MORAs, the pairwise dominance and non-dominance relationships are established statistically according to historically observed performance vectors.

The objective of MORAs is to minimize the overall computational cost needed to ensure a pre-defined upper bound of the probability of falsely retaining any dominated models or mistakenly removing any non-dominated models. More formally speaking, the objective of MORAs can be defined as

$$\begin{aligned} &\min \mathbb{E}\left[s\left(e, \mathcal{P}\right)\right] \\ &s.t.\ \ e \leq C \end{aligned} \tag{3.9}$$

where s refers to the overall sample complexity of a complete run of a MORA, C is a prescribed significance level, and $e \triangleq \mathbb{P}\left(\mathcal{P}_R \neq \mathcal{P}_{PF}\right)$ where $\mathcal{P}_R$ denotes the ensemble of models returned by the racing procedure and $\mathcal{P}_{PF}$ denotes the true Pareto front models.

CHAPTER 4: EXTREME RACING BASED ON EXTREME VALUE THEORY

In this chapter, the first Racing Algorithm (RA) for Extreme Model Selection (EMS), named Max-Race [147], is proposed. Given a maximization problem, the goal of Max-Race is to allocate the computational resources optimally among a portfolio of problem solvers in order to maximize the quality of the final solution obtained. Max-Race is an online Single-Objective Racing Algorithm (SORA) because model selection occurs during model comparison in terms of extreme performance and does not involve model validation. In Max-Race, significant difference between the extreme performances of a pair of models is statistically inferred via a newly developed hypothesis test based on the Generalized Pareto Distribution (GPD) assumption. Finally in this research, Max-Race is applied directly on constructing a population-based Algorithm Portfolio (AP).

Motivation

The need for Max-Race is motivated by the problem of constructing a population-based AP. Population-based algorithms have become popular in recent years due to their good characteristics, such as their ability to optimize non-differentiable, non-linear, multi-modal functions, their inherent parallelizability, and ease of use, as well as their good convergence properties. Some well-known population-based algorithms are evolutionary algorithms, artificial bee colony algorithm, particle swarm optimization, differential evolution, etc. Each of these algorithms have advantages and disadvantages that makes the selection of an appropriate algorithm for solving a given optimization function an open problem. One attempt to address this issue is using APs where several algorithms are combined and share the total available computational resources to create a superior algorithm that does better than its constituent algorithms on average. Population-

based algorithms are iterative algorithms. Therefore, given an objective function, the ultimate goal of population-based AP approaches is to wisely allocate the predefined maximum number of iterations among the constituent algorithms to maximize the optimum solution achieved. In [131], Vrugt, et al. put forward A Multi-algorithm Genetically Adaptive Method for Single Objective Optimization (AMALGAM-SO) which utilizes a self-adaptive learning strategy to automatically assign the number of offsprings to the constituent algorithms. In [102], a Population-based Algorithm Portfolio (PAP) approach was proposed with fixed shares of the overall computational resources among the constituent algorithms. Another AP approach, named Multiple Evolutionary Algorithm (MultiEA), was proposed in [144] where a regression model is built based on historical optimum values obtained from each individual algorithm, and only the algorithm with the best predicted performance in the nearest future is allowed to run for the next iteration.

However, existing AP approaches have certain limitations. First of all, how to design the hybridization scheme of synchronizing different algorithms to improve the overall efficiency requires much effort. Secondly, their success is largely dependent on a good synchronization of the constituent algorithms, and inappropriate selection of the algorithms may impede the optimization process and result in a worse final solution. Last but not least, a great amount of computational resources may be wasted on the constituent algorithm with inferior performance on the given optimization problem.

Ideally, a Winner-Take-All (WTA) methodology based AP approach is more preferable in practice. In particular, we are proposing to run the algorithms in the suite concurrently and independently, and decide from time to time, using their intermediate performances, which ones of these algorithms will continue running and which ones, deemed as algorithms that will eventually produce inferior solutions, will be aborted. Such a population-based AP approach frees the designers from the sometimes heavy demand of understanding the constituent algorithms well enough to coordi-

nate them. Moreover, it saves unnecessary computational efforts on algorithms with inferior final solutions.

Extreme Value Theory

The outcome of a stochastic optimizer, which can be regarded as an estimator of the optimum value of the objective function, is a random variable (RV) following an unknown distribution. We often are interested in the best possible outcome that the model could obtain which can be viewed as the extreme performance of the model. Extreme Value Theory (EVT) is a branch of statistics dealing with extreme or rare events [34]. Therefore, EVT is ideal for modeling the extreme outcomes/performances of the candidate models. In the literature, extreme value analysis via EVT has been widely used to study the underlying distribution of the outcomes of some stochastic optimizers [66, 32, 33].

The univariate EVT states that the distribution of the maximum of a sequence of identically and independently distributed (i.i.d.) normalized samples converges to a Generalized Extreme Value (GEV) distribution when the sample size goes to infinity. Mathematically speaking, let $X_i, i = 1, 2, \cdots, n$ denote a series of i.i.d. RVs following an unknown Cumulative Distribution Function (CDF) H, and let $Y \triangleq \max_{i=1,2,\cdots,n}(X_i)$. As stated in [34], we have

$$\lim_{n\to\infty} \mathbb{P}\left(\frac{Y - b_n}{a_n} \leq y\right) = \lim_{n\to\infty} H^n(a_n y + b_n) = G(y) \tag{4.1}$$

where $a_n > 0$ and b_n are normalization coefficients, and $G(y)$ represents the CDF of the GEV distribution with the following standard form

$$G(y) \triangleq \exp\left\{-\left[1+\tilde{\xi}\left(\frac{y-\tilde{\mu}}{\tilde{\sigma}}\right)\right]^{-1/\tilde{\xi}}\right\} \tag{4.2}$$

There are three parameters defining a GEV distribution: the location parameter $\tilde{\mu}$, the scale parameter $\tilde{\sigma} > 0$ and the shape parameter $\tilde{\xi}$. When normalized, $G(y)$ belongs to one of three non-degenerate distribution families: Frechét, Weibull and Gumbel.

Moreover, we can derive from Equation (4.1) that the tail distribution of X_i approximates the GPD in the limit

$$\lim_{u \to u^*} \mathbb{P}\left(X > x | X > u\right) = F_{\xi,\mu,\sigma}(x) \tag{4.3}$$

where u^* is the endpoint of distribution H, and $F_{\xi,\mu,\sigma}(x)$ has the following standard form

$$F_{\xi,\mu,\sigma}(x) \triangleq \begin{cases} 1-\left(1+\frac{\xi(x-\mu)}{\sigma}\right)^{-1/\xi} & for\ \xi \neq 0 \\ 1-\exp\left(-\frac{x-\mu}{\sigma}\right) & for\ \xi = 0 \end{cases} \tag{4.4}$$

with $\sigma > 0$, and $x \geq \mu$ when $\xi \geq 0$, and $\mu \leq x \leq \mu - \sigma/\xi$, if otherwise. The corresponding Probability Density Function (PDF) is derived as

$$f_{\xi,\mu,\sigma}(x) = \frac{1}{\sigma}\left(1+\frac{\xi(x-\mu)}{\sigma}\right)^{-\frac{1}{\xi}-1} \tag{4.5}$$

Similar to the GEV distribution, a GPD is also determined by three parameters: the shape parameter ξ, the location parameter μ, and the scale parameter σ, which are closely related to $\tilde{\xi}$, $\tilde{\mu}$ and $\tilde{\sigma}$ of the associated GEV distribution [34]. Correspondingly, there are three GPD families. If $\xi > 0$, the GPD is equivalent to a Pareto distribution; if $\xi = 0$, the GPD is equivalent to an exponential distribution; and if $\xi < 0$, we have a short-tailed Pareto II distribution.

There exist two approaches for practical extreme value analysis in EVT, which are Block Maxima (BM) for GEV distribution modeling and Point Over Threshold (POT) for GPD modeling, respectively. In BM, the underlying distribution of a series of independently sampled block maximum is approximated via a GEV distribution according to the results of the *Fisher-Tippett-Gendenko theorem*. In practice, it requires to collect a very large number of observations of i.i.d. RVs from the same distribution over a long period of time to guarantee modeling accuracy. However, only the block maximum is utilized and the rest are wasted. Distinct from the BM method, POT has a more efficient use of the collected data. It relies on all the peak values, from a continuous record, that exceed a pre-selected threshold value. A tail-fitting of the GPD is then employed for these exceedances according to the *Pickands-Balkema-de Haan theorem* [13, 103]. Moreover, POT is more practical and powerful for extreme value analysis due to its "threshold stability" property[1]. In the literature, the POT method is widely used in distribution modeling and analysis in the areas of finance, insurance and environmental studies.

Modeling Model Performance via GPD

In this work, the models considered are Evolutionary Computation (EC) algorithms. Given an optimization problem, the population of each model consists of sampled solutions in the objective

[1]If the conditional distribution of $X > u$ follows a GPD, the conditional distribution of $X > v$ given $v > u$ also follows a GPD with the same ξ but a shifted σ

space which are updated every iteration/generation via a guided random search. Without loss of generality, only maximization problems are considered in this work. Due to the stochastic nature of EC algorithms, the sampled solutions can be regarded as RV from an unknown distribution that gradually congregate close to local maxima. We are only interested in the extreme outcome (*i.e.* the final global/local maximum achieved), and the intermediate outcomes can be considered as realizations of RVs from the right tail of the underlying distribution. Therefore, the POT approach is a good choice for extreme value analysis in which the right tail of the unknown distribution is approximated. The underlying distribution of the outcomes during the course of optimization is approximated by a GPD, and the accuracy of modeling increases as the optimization continues because the underlying distribution becomes more stable as more samples concentrate around the global/local maximum in EC algorithms.

Let us use $gbest_t$ to denote the best objective value obtained so far at the t^{th} generation of any EC algorithm. It is obvious that $gbest_t, t = 1, 2, \cdots$ is a series of RVs of a discrete time Markov chain since $gbest_t \triangleq \max\{\max(pop_t), gbest_{t-1}\}$ where pop_t refers to the population at the t^{th} generation. Let us assume that the Markov chain enters a state of equilibrium at some mixing time t_0 where the collected $\{gbest_t\}_{t \geq t_0}$ can be regarded as i.i.d. samples from a stationary distribution. Therefore, in this work, we assume that the sub-sequence $\{gbest_t\}_{t=t_0}^{\infty}$ of the stochastic sequence $\{gbest_t\}_{t=1}^{\infty}$ for very large t_0 contains i.i.d. samples. Due the this i.i.d. assumption, the upper order statistics of the collected $\{gbest_t\}_{t=t_0}^{\infty}$ are modeled by a GPD with $\xi < 0$ and a right endpoint $\mu - \frac{\sigma}{\xi}$ for maximization problems.

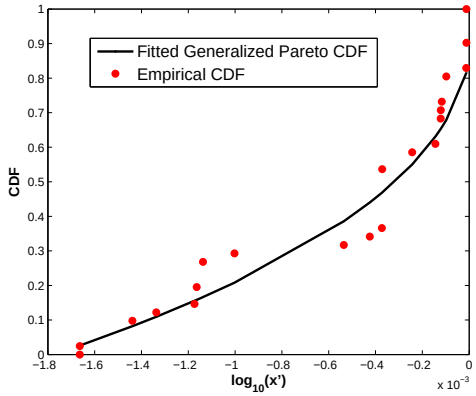
The first step of the POT method is the selection of a threshold u. There are several threshold selection methods proposed in the literature [112]. In this work, we simply choose the q-quantile of the sample as the threshold, which is a simple but popular rule. During the evolutionary process, the populations will converge to the maximum. In other words, the number of samples that are close to the endpoint is increasing proportionally to the growth of the sample size. Hence, the q

value is designed to be decreasing exponentially with the sample size s as $q = q_0 e^{-\lambda s}$. q_0 and λ are determined based on the experimental data; in the conducted experiments, the values $q_{100} = 0.7$ and $q_{5000} = 0.01$ were used. After the value of u is properly selected, we resort to the Method of Moments (MOM) [40] for ξ, μ and σ estimation which is computationally simple and yields consistent estimators under very weak assumptions.

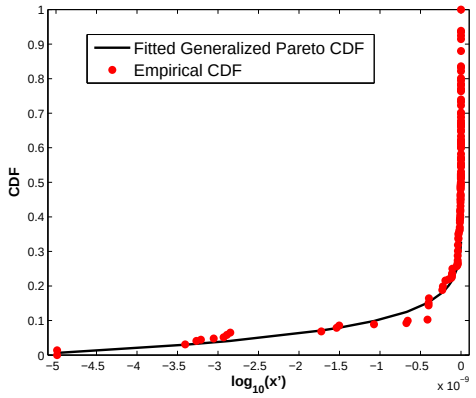
In Figure 4.1, we show an example of GPD fitting for $gbest_t$ via MOM. The sample data was collected from one run of a CPSO algorithm on a multi-model numerical function. First of all, it can be seen that threshold selection plays an important role in GPD fitting. The estimated CDF curve in Figure 4.1b has a much better fit to the empirical CDF than the one in Figure 4.1c. It is, because the selected q-quantile is too large for $s = 500$ in Figure 4.1c and the retained samples are insufficient for GPD fitting. Moreover, comparing the fitted CDF curves between Figure 4.1a and Figure 4.1b, it can be seen that the GPD provides more accurate modeling of the empirical distribution as more samples are collected during the evolutionary process.

Statistical Inference of End-Point Equivalence in Max-Race

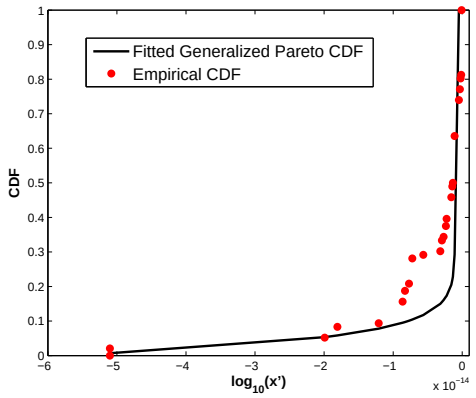
As we discussed in the previous section, when comparing two models in terms of their extreme performances, we are actually comparing the right endpoints of the associated GPDs. In order to make statistical inference of pairwise equivalence of right endpoints, a parametric pairwise hypothesis test is required in which the sampled data are assumed to be i.i.d. RVs from a three-parameter GPD with a finite right endpoint. Unfortunately, there exists no statistical test for such equivalence in the literature. Therefore, we develop a hypothesis test to identify pairwise equivalence relation of the right endpoints of two GPDs.



(a) $s = 100, q = 0.7$



(b) $s = 500, q = 0.5$



(c) $s = 500, q = 0.9$

Figure 4.1: MOM fitting of GPD for $gbest_t$ collected from one run of CPSO with a multi-modal optimization function where x' denotes the normalized x value.

Let $\{X_i\}_{i=1}^n$ denote a set of observations from model A, where $X_i \overset{i.i.d.}{\sim} F_{\xi_A,\mu_A,\sigma_A}(x)$, $\xi_A < 0$. Let $\{Y_i\}_{i=1}^m$ denote a set of observations from model B, where $Y_i \overset{i.i.d.}{\sim} F_{\xi_B,\mu_B,\sigma_B}(x)$, $\xi_B < 0$. The null hypothesis is $H_0 : \mu_A - \frac{\sigma_A}{\xi_A} = \mu_B - \frac{\sigma_B}{\xi_B}$, which implies the equivalence of two endpoints. Correspondingly, the alternative hypothesis is $H_a : \mu_A - \frac{\sigma_A}{\xi_A} > \mu_B - \frac{\sigma_B}{\xi_B}$, indicating that model A will achieve a better extreme performance than model B and it is certainly preferable to B. The sample maximum is commonly used as an estimation of the real endpoints. Given $U \triangleq \max\{X_1, \cdots, X_n\}, U \in [\mu_A, \mu_A - \frac{\sigma_A}{\xi_A}]$ and $V \triangleq \max\{Y_1, \cdots, Y_m\}, V \in [\mu_B, \mu_B - \frac{\sigma_B}{\xi_B}]$, the PDFs of U and V are given as follows:

$$f_U(u) = n f_{\xi_A,\mu_A,\sigma_A}(u) F^{n-1}_{\xi_A,\mu_A,\sigma_A}(u) \tag{4.6}$$

and

$$f_V(v) = m f_{\xi_B,\mu_B,\sigma_B}(v) F^{m-1}_{\xi_B,\mu_B,\sigma_B}(v) \tag{4.7}$$

Let $\Delta \triangleq U - V, \Delta \in [\mu_A - \mu_B + \frac{\sigma_B}{\xi_B}, \mu_A - \mu_B - \frac{\sigma_A}{\xi_A}]$ denote the difference between two sample maxima. Considering Δ as the test statistic, its PDF is

$$f_\Delta(\delta) = \int_{\mu_A}^{\mu_A - \frac{\sigma_A}{\xi_A}} f_U(u) f_V(u - \delta) du \tag{4.8}$$

Hence, under H_0, we have $\Delta \in [\frac{\sigma_A}{\xi_A}, -\frac{\sigma_B}{\xi_B}]$. Therefore, the probability that $\Delta \leq d$ under H_0 is given as

$$\begin{aligned} P(\Delta \leq d) &= \int_{\frac{\sigma_A}{\xi_A}}^{d} f_\Delta(\delta) d\delta \\ &= \int_0^1 F^m_{\xi_B,\mu_B,\sigma_B} \left\{ \frac{\sigma_A \left[\left(1 - P^{\frac{1}{n}}\right)^{-\xi_A} - 1 \right]}{\xi_A} + \mu_A - \frac{\sigma_A}{\xi_A} \right\} dP \\ &- \int_0^1 F^m_{\xi_B,\mu_B,\sigma_B} \left\{ \frac{\sigma_A \left[\left(1 - P^{\frac{1}{n}}\right)^{-\xi_A} - 1 \right]}{\xi_A} + \mu_A - d \right\} dP \end{aligned} \tag{4.9}$$

where $P = F^n_{\xi_A,\mu_A,\sigma_A}(u)$. Subsequently, the p-value of the proposed test is computed as

$$\pi(d; A, B) = P(\Delta > d) = 1 - P(\Delta \leq d) \tag{4.10}$$

where d is the observed value of Δ.

If $\pi(d; A, B) < \alpha$ where α is the predefined maximum Type I error rate, H_0 will be rejected and it is concluded that A has better extreme performance than B and thus B will be eliminated from racing. Note that there is no closed form for the probability in Equation (4.9) and, hence, numerical integration method (*e.g.* adaptive Gaussian quadrature method) must be used to approximate $\pi(d; A, B)$.

The other parameters involved in Equation (4.9) are estimated from the observed data. μ_A and μ_B are first estimated via MOM [40]. The data is then shifted according to the estimated μ_A and μ_B values, and the remaining parameters under H_0, including σ_A, ξ_A, σ_B, and ξ_B, are estimated via a hybrid method consisting of Maximum Likelihood Estimation and Maximum-Goodness-of-Fit

Estimation [134]. The interested reader may refer to [134] for a detailed description of the adopted parameter estimation method.

Max-Race Specifics

The pseudo-code of Max-Race is provided in Algorithm 1. Given the stochastic objective function f, the goodness of each candidate model $M_i, i = 1, 2, \cdots, K$ is measured as $f(M_i)$. In Max-Race, $f(M_i)$ is assumed to be a RV following a GPD. At each step of Max-Race, we run each remaining model for an additional trial and obtain a new $f(M_i)$ observation. The statistical test discussed in Section 4 is employed to infer pairwise equivalence of extreme performances based on historical observations. If sufficient statistical evidence has been collected to demonstrate that M_i has inferior extreme performance when compared to any other model, M_i will be eliminated from racing immediately and the saved computational resources will be allocated to models with promising extreme performances. Max-Race ends when the available computational resources are exhausted.

Since there are multiple hypotheses being tested simultaneously, the Benjamini-Hochberg-Yekutieli procedure [19] is employed in Max-Race to control the overall False Discovery Rate (FDR), which is denoted by $FDR(\cdot, \cdot)$. FDR is one way of conceptualizing the overall rate of making any Type I errors in a set of hypothesis tests. By strictly controlling the FDR, the risks of falsely eliminating any models with best extreme performances are reduced and manageable. FDR control is a recommended alternative to Family-Wise Error Rate (FWER) control, and it provides greater power at the cost of increased probability of Type I errors. Given a set of p-values P_{values} and a prescribed significance level α, FDR(P_{values}, α) returns the indices of the models to be removed from racing due to the rejection of the corresponding null hypotheses. In this work, the α value is chosen to increase exponentially with the sample size n since the GPD fit gets more accurate as n increases.

To be more specific, we have $\alpha_n = \alpha_0 e^{-b(n_{max}-n)}$ where n is the current sample size, and n_{max} is the maximum possible sample size. The α_0 and b values are determined experimentally so that $\alpha_{n_0} = 10^{-30}$ and $\alpha_{n_{max}} = 0.01$.

Algorithm 1: Max-Race Pseudo-code

input : K initial models $\mathcal{M} \leftarrow \{M_1, M_2, \cdots, M_K\}$ and an objective function f
output: the best objective value obtained

while *stopping criteria not met* **do**
  foreach *model* $M_i \in \mathcal{M}$ **do**
    run another trial and obtain a new sample $f(M_i)$
  end
  $P_{values} \leftarrow \varnothing$
  foreach *pair of models* $M_i, M_j \in \mathcal{M}$ **do**
    $P_{values} \leftarrow P_{values} \cup \pi(d; M_i, M_j)$
  end
  $\mathcal{M} \leftarrow \mathcal{M} \setminus \{M_k\}$, where $k \in \text{FDR}(P_{values}, \alpha)$
end

In this work, we introduce a novel population-based AP hinging on Max-Race (see Algorithm 2). K sub-populations are randomly initialized corresponding to K candidate EC algorithms. At each step of Max-Race, each remaining sub-population is evolved for another N_{sc} generations and the $gbest$ values after every generation are recorded as historical observations. Once a candidate EC algorithm is identified as under-performing in term of the extreme performance, it will be eliminated from racing and its population will be stopped from further evolution. The portfolio stops when a predefined maximum Number of Function Evaluations (NFEs) is reached.

Experiments and Applications

In this section, we assess the performance of Max-Race by comparing it with the Brute Force Approach (BFA), two baseline algorithms (*i.e.* BestEC, RandEC) and three popular population-based APs (*i.e.* AMALGAM-SO, PAP, MultiEA). 6 popular EC algorithms are selected to form

Algorithm 2: Max-Race Algorithm Portfolio Pseudo-code

input : K candidate EC algorithms $\mathcal{A} \leftarrow \{A_1, A_2, \cdots, A_K\}$ and an objective function f
output: the best objective value obtained

Initialize and evaluate K sub-populations $\{pop_1, pop_2, \cdots, pop_K\}, (K \geq 2)$
For each sub-population, record the best objective value as $gbest_0$
while *the maximal NFEs is not reached* **do**
  foreach *algorithm* $A_i \in \mathcal{A}$ **do**
    Evolve pop_i for another N_{sc} generations
    Record the $gbest_t$ values of every generation
  end
  $P_{values} \leftarrow \varnothing$
  foreach *pair of algorithms* $A_i, A_j \in \mathcal{A}$ **do**
    $P_{values} \leftarrow P_{values} \cup \pi(d; A_i, A_j)$
  end
  $\mathcal{A} \leftarrow \mathcal{A} \setminus \{A_k\}$, where $k \in \text{FDR}(P_{values}, \alpha)$
end

the initial ensemble of models, which are PSO [149], CPSO [129], jDE [25], JADE [145], SaDE [106] and CMA-EA [59]. We use D-dimensional multi-modal functions generated via a Gaussian landscape generator [142] with bounded search space $[-100, 100]^D$ as the objective functions.

Comparisons of Max-Race to the Baseline Algorithms

A common practice for Model Selection (MS) is the BFA which assigns the same amount of computational resources to each model. We first conducted a set of simple experiments to compare the performances of Max-Race and the BFA. 30 benchmark functions were generated with $D = 10$ and $D = 30$, respectively. In the BFA, each EC algorithm runs for $5000 \times 5D$ NFEs which is large enough to guarantee convergence. The intermediate observations of $gbest_t$ collected from the BFA were used as pseudo-samples for simulating a run of Max-Race. In Max-Race, the statistical comparisons occurred every 10 generations ($N_{sc} = 10$). Each experiment was repeated for 25 trials.

Two performance metrics were selected, which were prediction accuracy PA and running time ratio RTR. PA denotes the selection accuracy of Max-Race. $PA = 1$ if Max-Race retains the EC algorithm with the true best extreme performance at the end of racing, and 0 otherwise. RTR refers to the ratio of the NFEs consumed by Max-Race over the NFEs spent by the BFA, and it reflects the computational savings of Max-Race with respect to the BFA. Obviously, it is desirable to have RTR close to 0.

In Table 4.1, we report the relevant statistics of the average PA and RTR values. In addition, we show the average values of the number of models with the best extreme performances, denoted by N_b, and the number of models retained by Max-Race at the end, denoted by N_r. Regarding the PA values in Table 4.1, their median values are larger than 0.80, implying that Max-Race is able to retain the models with best extreme performances in most cases for both $D = 10$ and $D = 30$. Compared to $D = 10$, Max-Race performs slightly worse for $D = 30$. This is because, for high-dimensional multi-modal functions, it is hard for EC algorithms to reach a good objective value and, thus, all the candidate models show similar performances. Hence, it is highly likely that Max-Race will retain sub-optimal models since it trades off some computational cost for a reduced likelihood of identifying the true optimal models.

Regarding the RTR values in Table 4.1, they are no larger than 0.30 for $D = 10$ and 0.46 for $D = 30$, indicating that the computational advantage of Max-Race over the BFA is substantial. Max-Race always saves more than 50% of the NFEs spent by the BFA.

The ultimate goal of Max-Race is to maximize the final objective value obtained with a limited computational budget as shown in Algorithm 1 and Algorithm 2. Therefore, we compare the performance of Max-Race to two baseline algorithms which are BestEC and RandEC. In BestEC, we only select the EC algorithm with the best extreme performance to run, which is the ideal strategy. In RandEC, we randomly pick an EC algorithm to run until the maximal NFEs are

Table 4.1: Statistics of Average PA, RTR, N_b and N_r Values of Max-Race

	D	max	median	min	std
PA	10	1.00	0.91	0.80	0.07
	30	1.00	0.83	0.36	0.13
RTR	10	0.30	0.23	0.22	0.02
	30	0.46	0.41	0.38	0.02
$\mathbf{N_b}$	10	3.92	1.42	1.04	0.60
	30	2.44	1.28	1.00	0.38
$\mathbf{N_r}$	10	1.24	1.00	1.00	0.06
	30	2.20	2.00	1.92	0.05

reached. We still used the pseudo-samples generated in the previous experiments with $D = 30$ and the maximum NFEs for Max-Race was set to $5000 \times 5D$.

The average objective values obtained by each approach, as well as the true global maximum of the objective function, are shown in Table 4.2. In addition, we adopted a paired Wilcoxon signed-rank test with significance level 0.01 to identify significant performance differences. In Table 4.2, if BestEC or RandEC outperforms Max-Race significantly, the corresponding entry is bold-faced. If any of them is significantly outperformed by Max-Race, the corresponding entry is underlined. As observed from Table 4.2, Max-Race demonstrates inferior performance to BestEC in only 13.33% of the cases, while Max-Race demonstrates much better performance than RandEC in all cases. The experimental results confirm that Max-Race performs almost as well as BestEC, the ideal EMS strategy.

In Figure 4.2, we provide a visual representation of the difference between the performances of Max-Race, BestEC and RandEC. We estimated the distributions of their average final outcomes from bootstrapped samples using a normal kernel function. The estimated density for f_5 with $D = 30$ is depicted in Figure 4.2. It can be seen that, Max-Race performs equally well as BestEC,

and both of them outperform RandEC significantly in terms of the maximum objective values achieved at the end.

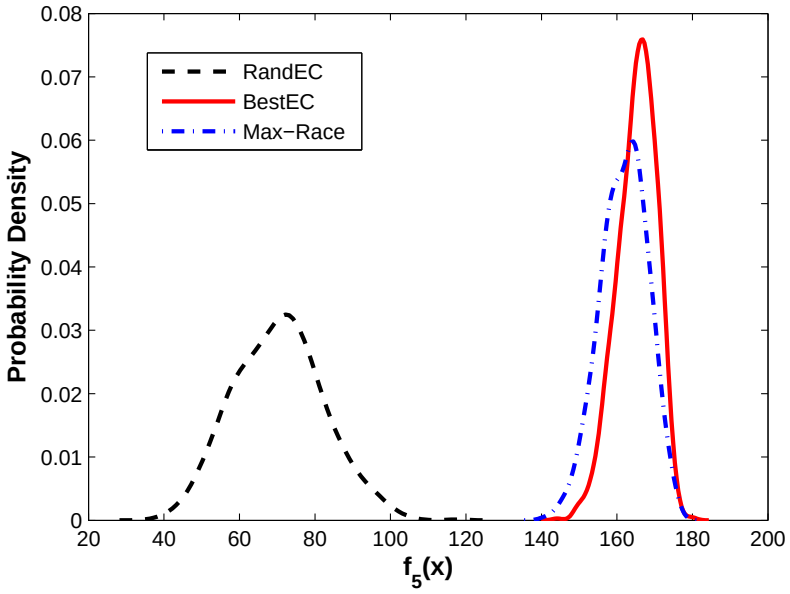


Figure 4.2: Bootstrap distributions of the means of final best solutions achieved by Max-Race, BestEC and RandEC for f_5 with $D = 30$.

Comparisons of Several APs

As we mentioned in Section 4, the need for Max-Race is motivated by the problem of constructing population-based AP. In order to demonstrate that Max-Race is a promising online MS algorithm, we compare the performance of a population-based AP hinging on Max-Race with three other popular APs, including AMALGAM-SO [131], PAP [102] and MultiEA [144]. In this set of experiments, 20 test functions were generated for $D = 30$. The initial population size of each sub-population was 500 and the maximum NFEs was 3×10^5. Other AP parameters were set as suggested in the respective original papers.

The final outcomes, including the mean and standard deviation values over 25 runs, of Max-Race, AMALGAM-SO, PAP and MultiEA are reported in Table 4.3. A paired Wilcoxon signed-rank test with significance level 0.01 is employed to identify significant performance differences. In Table 4.3, if Max-Race is significantly outperformed by an algorithm, the corresponding entry of that algorithm is bold-faced. If an algorithm is significantly outperformed by Max-Race, its corresponding entry is underlined. As observed from Table 4.3, Max-Race significantly outperforms AMALGAM-SO, PAP and MultiEA in 65%, 55% and 65% of all the cases, respectively. Meanwhile, it presents significantly inferior performance in only 10%, 5% and 10% of the cases.

If we take a closer look at the results of the other three APs, AMALGAM-SO shows the worst performance. The success of AMALGAM-SO is largely related to a good synchronization of its constituent algorithms via population sharing, elitism selection, diversity control, premature convergence detection, etc. Hence, the optimization process may be hindered unexpectedly due to inappropriate selection of the constituent algorithms. Thus, a simpler AP with less interaction among the candidate models is preferred. PAP, for instance, achieves better performance than AMALGAM-SO in most cases because its sub-populations only interleaved with each other by occasional population migration. MultiEA does not require any population interaction and each sub-population evolves independently. However, its selection of the best model is based on the predicted performance in the nearest common future point via a linear regression model of the evolutionary curve. Therefore, it is highly likely that MultiEA wastes too many computational resources on algorithms with a fast convergence to a local maximum.

The experimental results confirm the outstanding performance of Max-Race based AP over the other APs. Moreover, it has been demonstrated that, by accurately identifying the optimal models with the best extreme performances and eliminating the under-performing models as early as possible, Max-Race is able to maximize the quality of the final solution by concentrating most of the computational resources on the superior problem solvers.

Table 4.2: Averaged Final Objective Values Obtained by Max-Race, BestEC, RandEC and the True Global Maximum

Prob.	f_1	f_2	f_3	f_4	f_5
Max-Race	168.39	275.45	108.04	12.76	161.79
BestEC	171.09	275.79	114.57	**13.61**	165.23
RandEC	107.01	170.20	81.88	11.43	69.91
Global	389.68	341.94	193.99	29.16	193.99
Prob.	f_6	f_7	f_8	f_9	f_{10}
Max-Race	103.62	41.19	157.77	7.76	34.47
BestEC	105.39	41.20	157.77	8.46	34.76
RandEC	83.65	36.25	94.40	5.99	24.68
Global	125.27	287.79	470.55	9.93	41.61
Prob.	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
Max-Race	14.22	187.78	256.18	50.42	78.82
BestEC	14.24	**224.36**	269.29	50.55	89.61
RandEC	9.51	83.48	144.61	31.93	48.50
Global	39.93	275.21	415.96	327.66	465.00
Prob.	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}
Max-Race	62.32	274.44	30.03	13.43	44.19
BestEC	68.56	274.44	35.38	13.59	58.38
RandEC	639.70	193.69	16.73	9.56	17.23
Global	400.16	473.35	49.48	70.55	49.48
Prob.	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}
Max-Race	102.14	257.90	155.67	6.65	34.47
BestEC	104.98	262.40	157.51	6.91	35.04
RandEC	69.58	131.12	103.49	3.91	26.61
Global	175.20	275.21	183.65	316.59	41.61
Prob.	f_{26}	f_{27}	f_{28}	f_{29}	f_{30}
Max-Race	13.98	178.67	1.38	233.32	30.75
BestEC	14.20	**220.17**	1.39	**265.19**	32.21
RandEC	9.37	61.14	1.06	168.22	19.86
Global	39.93	341.94	10.63	446.68	73.57

Table 4.3: Averaged Final Objective Values Obtained by Max-Race, AMALGAM-SO, PAP and MultiEA

Prob.		f_1	f_2	f_3	f_4	f_5
Max-Race	mean	53.77	332.34	439.52	278.74	158.16
	std	2.27	29.42	1.59e-13	26.38	11.10
AMALGAM-SO	mean	43.00	318.42	438.52	237.56	148.52
	std	4.21	30.87	1.01e-05	22.04	13.50
PAP	mean	52.05	268.23	264.42	259.70	148.52
	std	4.41	35.42	18.87	18.24	11.26
MultiEA	mean	42.20	241.01	164.21	206.20	133.07
	std	3.76	57.40	15.57	35.71	18.17
Prob.		f_6	f_7	f_8	f_9	f_{10}
Max-Race	mean	336.00	9.16	48.66	13.07	111.91
	std	22.35	1.39	2.42	1.62	11.94
AMALGAM-SO	mean	307.96	8.04	45.24	12.42	**119.44**
	std	27.63	4.84	7.87	1.27	**8.15**
PAP	mean	324.77	9.41	46.27	12.84	**113.46**
	std	19.34	5.61	5.70	1.71	**14.87**
MultiEA	mean	309.15	**10.29**	38.75	**15.64**	110.60
	std	26.64	**3.91**	10.63	**2.90**	2.23
Prob.		f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
Max-Race	mean	53.95	99.50	129.64	102.98	9.94
	std	2.11	6.49	4.06e-14	6.09	0.58
AMALGAM-SO	mean	46.40	85.89	117.20	72.97	**8.47**
	std	1.13	5.53	15.10	9.02	**1.17**
PAP	mean	51.13	99.62	121.67	96.67	9.10
	std	6.12	4.79	12.28	3.77	1.65
MultiEA	mean	45.91	87.09	94.70	91.69	8.20
	std	3.66	6.51	7.35	6.43	1.51
Prob.		f_{16}	f_{17}	f_{18}	f_{19}	f_{20}
Max-Race	mean	90.15	52.78	51.64	1.51	48.48
	std	7.58	2.54	8.37	1.31	0.10
AMALGAM-SO	mean	86.58	**61.23**	49.34	1.11	48.23
	std	7.40	**4.17**	7.03	1.16	0.83
PAP	mean	76.50	43.59	42.64	1.23	39.52
	std	3.21	4.65	7.59	1.16	3.03
MultiEA	mean	61.23	50.05	48.85	0.83	45.85
	std	6.19	3.52	9.90	0.28	2.81

CHAPTER 5: MULTI-OBJECTIVE RACING BASED ON SIGN TEST

In this chapter, we propose the first Multi-Objective Racing Algorithm (MORA) in a fixed-budget setting, namely S-Race, which addresses the problem of Multi-Objective Model Selection (MOMS) in the proper sense of Pareto optimality. In S-Race, an ensemble of Pareto optimal models is returned with multiple conflicting objectives optimized simultaneously. In S-Race, the pairwise dominance relationship is statistically inferred via a non-parametric Sign Test (ST) based on historical performance vectors. Moreover, a discrete Holm's step-down Family-Wise Error Rate (FWER) control method is leveraged to control the overall probability of making any Type I errors (*false discoveries*). In S-Race, the size of the validation set of problem instances is predefined and fixed. Given the lower bound of the total probability of making no Type I error in S-Race, denoted as Δ, the α values assigned to each family at each step of S-Race is adjusted adaptively according to the number of remaining steps and the number of retained models.

Multi-Objective Model Selection

In multi-objective optimization, multiple conflicting objectives are expected to be optimized simultaneously. As a result, there is no single unique optimal solution. The concept of Pareto optimality plays an important role in multi-objective optimization. We assume without loss of generality that all the objectives are to be maximized. Let us assume that the performance of each model is measured in terms of D stochastic real-valued objective functions $\{f_i\}_{i=1,\cdots,D}$. In other words, the performance each model M is measured as a *performance vector*, denoted as

$\mathbf{f}(M) \triangleq \{f_i(M)\}_{i=1,\cdots,D}$. When comparing model M and M' in terms of their performance vectors, $\mathbf{f}(M)$ dominates $\mathbf{f}(M')$, denoted as $\mathbf{f}(M) \succ \mathbf{f}(M')$, if and only if

$$\begin{aligned} & f_i(M) \geqslant f_i(M') \ \ \forall\, i \in \{1, 2, \cdots, D\} \\ & \text{and } \exists\, j \in \{1, 2, \cdots, D\} \mid f_j(M) > f_j(M') \end{aligned} \tag{5.1}$$

Assume that the performance vector of each model is a random variable (RV) following an unknown distribution. It is not reliable to determine the dominance and non-dominance relationship between a pair of models based on only one observation. Therefore, in this research, the concept of *probabilistic dominance*[1] is utilized. To be more specific, model M is said to dominate model M', denoted as $M \succ M'$, if and only if $\mathbb{P}\left(\mathbf{f}(M) \succ \mathbf{f}(M')\right) > \mathbb{P}\left(\mathbf{f}(M) \prec \mathbf{f}(M')\right)$. On the contrary, M is said to be dominated by M', denoted as $M \prec M'$, if and only if $\mathbb{P}\left(\mathbf{f}(M) \prec \mathbf{f}(M')\right) > \mathbb{P}\left(\mathbf{f}(M) \succ \mathbf{f}(M')\right)$. Otherwise, they are non-dominated to each other, denoted as $M \sim M'$. In MOMS, all the non-dominated models consist of the Pareto front which are expected to be returned as Pareto optimal models.

Statistical Inference of Dominance in S-Race

Sign Test

The pairwise Pareto dominance relationship cannot be established with absolute certainty based on limited number of observations when the objective functions are stochastic in nature. Therefore, a hypothesis test is employed to infer the dominance relation with confidence. First of all, a non-parametric hypothesis test is preferred over a parametric one. This is because in non-parametric

[1]The notions laid out here are related to the concept of probabilistic dominance presented in [126, 65].

statistical analysis, no assumption of the underlying distributions of the performance vectors is required, neither marginal nor joint. Moreover, it is desirable to adopt a pairwise test in which the samples of the test are dependent on paired observations only. This is also known as *block design* in Block Race (BRACE) and F-Race. It excludes the risks caused by the variations due to different problem instances or different execution environments.

In S-Race, ST is employed to infer pairwise dominance relationship between models. Given two models M_i and M_j whose performances have been evaluated on batches of problem instances at the t^{th} step of S-Race, it is observed that there are n_{ij} times that $\mathbf{f}(M_i) \succ \mathbf{f}(M_j)$ and n_{ji} times that $\mathbf{f}(M_j) \succ \mathbf{f}(M_i)$. Let us use N_{ij} and N_{ji} to denote the corresponding RVs of n_{ij} and n_{ji}. Let $S \triangleq N_{ij} + N_{ji}$ with observed value $s \triangleq n_{ij} + n_{ji}$, and $p_{ij} \triangleq \mathbb{P}(N_{ij}|S = 1)$. For the ST, the null hypothesis H_0 is $p_{ij} = \frac{1}{2}$ and the alternative H_1 is $p_{ij} > \frac{1}{2}$. Under H_0, we have $N_{ij}|\{S = s\} \sim \text{Binomial}\left(s, \frac{1}{2}\right)$. Therefore, the test's p-value is computed as

$$\pi(n_{ij}, n_{ji}) = \frac{1}{2^s} \sum_{k=n_{ij}}^{s} \binom{s}{k} \tag{5.2}$$

Given α as the desired significance level which refers to the maximum probability of making any Type I errors allowed, H_0 will be rejected if $\pi(n_{ij}, n_{ji}) < \alpha$. Subsequently, M_j is eliminated from racing. The larger the difference between n_{ij} and n_{ji}, the stronger the support that one dominates the other.

Discrete Holm's Procedure

Assume that the initial ensemble of candidate models is of size K, the total number of pairwise STs is about $\binom{K}{2}$. When K is large, which is usually the case of Model Selection (MS) in machine learning, the number of pairwise comparisons is huge. As a result, controlling only the confidence

level of individual pairwise STs will lead to increased overall probability of making any false rejections (Type I errors). In other words, the probability of mistakenly removing any non-dominated models in S-Race will be unexpectedly large. This is a typical multiple comparison problem in statistical analysis. Therefore, a FWER control method is used to control the overall probability of making any Type I errors within a family of hypotheses, denoted as *family-wise error*. In other words, given the prescribed FWER significance level α, the overall probability of making any Type I errors within each family of hypotheses is strictly controlled at level α by adopting a FWER control approach.

In this work, the discrete Holm's step-down procedure [137] is used in S-Race for FWER control, which is an extension of the Holm's step-down procedure [62].

The original Holm's step-down procedure [62], as described in Table 5.1, is the step-down version of the Bonferroni approach. It controls the FWER in a strong sense, indicating that the FWER is always controlled at a user-specified level without any restrictions about the joint distribution of the test statistics involved. The Holm's procedure is more powerful than to the Bonferroni approach, meaning that it achieves smaller probability of making any Type II errors.

Table 5.1: Holm's Step-Down Procedure

	Given a maximum FWER α, a family of m tests of hypothesis, and their corresponding p-values
Step 1	Rank the m p-values in ascending order as $\pi_{(1)} \leq \pi_{(2)} \leq \cdots \leq \pi_{(m)}$
Step 2	Find the smallest $k = k_o$ such that $(m + 1 - k_o)\pi_{(k_o)} > \alpha$
Step 3	Reject the null hypotheses of the tests with p-values $\pi_{(k)}, k < k_o$; if $k_o = 1$, reject none; if no such k_o, reject all.

The discrete Holm's step-down procedure [137] is proposed to improve the power of Holm's procedure by utilizing the discreteness of the binomial distribution. Since in the ST the underlying distribution of the test statistics is binomial in nature, the discrete Holm's procedure is employed

in S-Race for improved efficiency. In the discrete Holm's procedure, the intersection hypotheses of each successive subsets $\mathcal{I}$ of the given family of null hypotheses $\mathcal{F}$ is tested sequentially. The actual smallest p-value in $\mathcal{I}$ is denoted as $T_{\mathcal{I}}\left(\boldsymbol{n}_{\mathcal{I}}^{+}\right) \triangleq \min_{\ell \in \mathcal{I}} \pi\left(n_{ij}^{\ell}, n_{ji}^{\ell}\right)$. The corresponding adjusted p-value of $\mathcal{I}$ is computed as

$$\tilde{p}_{\mathcal{I}}(\boldsymbol{n}_{\mathcal{I}}^{+}) \triangleq \sum_{\ell \in \mathcal{I}} P_{H_\ell}\left\{\pi\left(N_{ij}^{(\ell)}, N_{ji}^{(\ell)}\right) \leq T_{\mathcal{I}}\left(\boldsymbol{n}_{\mathcal{I}}^{+}\right) \middle| S^{\ell}=s^{\ell}\right\} \tag{5.3}$$

where $\boldsymbol{N}_{\mathcal{I}}^{+} \triangleq \left\{\left(N_{ij}^{\ell}, S^{\ell}\right) \middle| \ell \in \mathcal{I}\right\}$, and N_{ij}^{ℓ}, S^{ℓ} are the RVs associated with the ℓ^{th} hypothesis. Correspondingly, $\boldsymbol{n}_{\mathcal{I}}^{+} \triangleq \left\{\left(n_{ij}^{\ell}, s^{\ell}\right) \middle| \ell \in \mathcal{I}\right\}$ denotes an observed value of $\boldsymbol{N}_{\mathcal{I}}^{+}$.

Therefore, if $\tilde{p}_{\mathcal{I}}(\boldsymbol{n}_{\mathcal{I}}^{+}) < \alpha$, the intersection hypothesis $H_{\mathcal{I}}$ and all the $H_\ell, \ell \in \mathcal{I}$ will be rejected. The discrete version of the Holm's procedure, as depicted in Table 5.2, is uniformly more powerful than the original method due to the discreteness of the binomial distribution.

Table 5.2: Discrete Holm's Procedure

	Given a maximum FWER α, a family of m tests of hypothesis and their corresponding p-values
Step 1	Order the m p-values in ascending order as $\pi_{(1)} \leq \pi_{(2)} \leq \cdots \leq \pi_{(m)}$. Assume $r_1, r_2, \cdots, r_m$ are the corresponding indices of the sorted p-values
Step 2	Find the smallest $k = k_o$ such that $\tilde{p}_{R_k}(\boldsymbol{n}_{R_k}^{+}) > \alpha$ where $R_k = \{r_k, \cdots, r_m\}$.
Step 3	Reject the null hypotheses of the tests with p-values $\pi_{(k)}, k < k_o$; if $k_o = 1$, reject none; if no such k_o, reject all.

S-Race Specifics

The framework of S-Race is provided in Algorithm 3. Basically speaking, there are three major phases of S-Race which are *evaluation* (lines 3 - 4), *inference* (lines 7 - 12) and *elimination* (line 13). Given an initial ensemble of K models, a batch of problem instances is sampled without

replacement to evaluate the performances of all the remaining models during *evaluation*. The corresponding performance vectors are recorded as historical data that are used later for statistical inference of dominance. During *inference*, the ST is employed to infer dominance relation between each pair of models according to historical pairwise observations. The individual p-value is adjusted by the discrete Holm's procedure, denoted as DisHolms(P_{values}, α_t), with a significance level α_t at the t^{th} step of S-Race. At each step of S-Race, there are multiple independent families of hypotheses. A *family* in S-Race refers to all pairwise comparisons between the i^{th} model M_i and all the other remaining models. DisHolms(P_{values}, α_t) returns the indices of the models to be removed from racing due to the rejection of the corresponding null hypotheses. These models are believed to be dominated by their competitors, and thus are eliminated during *elimination*. The racing process stops when the validation set of problem instances is exhausted, or only one model is retained.

Algorithm 3: S-Race Pseudo-code

input : $Pool \leftarrow \{M_1, M_2, \cdots, M_K\}\ (K \geq 2)$
output: $Pool$

Initialize $t = 1$
repeat
  Randomly sample a batch from the validation set
  Evaluate all the remaining models on the batch
  foreach *model* $M_i \in Pool$ **do**
    $P_{values} \leftarrow \varnothing$
    foreach *model* $M_j \in Pool \setminus \{M_j\}$ **do**
      Update n_{ij} and n_{ji}
      if $n_{ij} > n_{ji}$ **then**
        $P_{values} \leftarrow P_{values} \cup \{\pi(n_{ij}, n_{ji})\}$
      end
    end
    $Pool \leftarrow Pool \setminus \{M_k\}$, where $k \in$ DisHolms(P_{values}, α_t)
  end
  $t = t + 1$
until *all validation batches are exhausted or* $|Pool| = 1$

It is very important to control the overall probability of making any false discoveries (Type I errors) in S-Race. If any of the Pareto optimal models is eliminated by mistake, especially in the early stages of the race, some of the dominated models will get the chance to survive due to lack of competitors.

Let us denote the accuracy of S-Race by Δ, which we define as the probability of no Type I errors committed during the entire racing process. Therefore, we are able to guarantee a minimum Δ by controlling individual FWER at level α. According to the Bonferroni inequality: $\mathbb{P}(\mathcal{A} \cup \mathcal{B}) \leq \mathbb{P}(\mathcal{A}) + \mathbb{P}(\mathcal{B})$ for any events $\mathcal{A}$ and $\mathcal{B}$, the relationship between Δ and α is established as follows.

As we know that the overall probability of making any Type I errors within each family is strictly controlled at level α by the discrete Holm's procedure.

$$\mathbb{P}(\text{at least one Type I error within a family}) \leq \alpha \tag{5.4}$$

Therefore, at the t^{th} step of S-Race, we have

$$\begin{aligned} &\mathbb{P}(\text{at least one Type I error at a step}) \\ &\leq F_t \alpha \\ &\leq (K_{t-1} - 1)\,\alpha \\ &\leq (K - 1)\,\alpha \end{aligned} \tag{5.5}$$

where F_t and K_t denote the number of families and the number of retained models at the t^{th} step, and K is the initial ensemble size. Since the validation set and batch size is predefined in advance,

the maximum number of steps T is available. Therefore, the overall probability of making any Type I errors in S-Race is

$$\begin{aligned}
&\mathbb{P}\left(\text{at least one Type I error during race}\right) \\
&\leq \sum_{t=1}^{T} F_t \alpha \\
&\leq \sum_{t=1}^{T} \left(K_{t-1} - 1\right) \alpha \\
&\leq T\left(K-1\right)\alpha
\end{aligned} \tag{5.6}$$

In other words, we have

$$\begin{aligned}
\Delta &\equiv \mathbb{P}\left(\text{no Type I error during race}\right) \\
&\geq 1 - T(K-1)\alpha
\end{aligned} \tag{5.7}$$

and subsequently

$$\alpha \leq \frac{1-\Delta}{T(K-1)} \tag{5.8}$$

Therefore, given Δ, the α value assigned to all families is easily computed via Equation (5.8), which is dependent on the initial ensemble size K and the total number of steps in racing T.

In reality, however, the number of models and the number of families is decreasing during racing. The bound provided in Equation (5.7) is very loose, unless no model is eliminated during the entire racing process. Such conservative control over α is helpful in controlling the overall probability of making any Type I errors at the cost of increasing the probability of making any Type II errors.

In other words, the risk of falsely retaining any dominated model is increased. In order to improve the power of S-Race, the α values assigned to the families at each step are adjusted adaptively according to the number of models retained.

Theorem 1. *In S-Race, let the significance level α_t at step t be given as*

$$\alpha_t = \frac{1 - \Delta - \sum_{i=1}^{t-1} \alpha_i F_i}{(T - t + 1)(K_{t-1} - 1)} \qquad \textit{if } 2 \leq t \leq T \tag{5.9}$$

with

$$\alpha_1 = \frac{1 - \Delta}{T(K - 1)} \tag{5.10}$$

Then, the overall probability of making any Type I errors in S-Race is no larger than $1 - \Delta$.

Proof. Based on Equation (5.9), we have

$$\begin{aligned} 0 < \alpha_t &= \frac{1 - \Delta - \sum_{i=1}^{t-1} \alpha_i F_i}{(T - t + 1)(K_{t-1} - 1)} \\ &= \frac{1 - \Delta - \sum_{i=1}^{t-1} \alpha_i F_i - \alpha_t (K_{t-1} - 1)}{(T - t)(K_{t-1} - 1)} \\ &\leq \frac{1 - \Delta - \sum_{i=1}^{t-1} \alpha_i F_i - \alpha_t F_t}{(T - t)(K_{t-1} - 1)} \\ &\leq \frac{1 - \Delta - \sum_{i=1}^{t} \alpha_i F_i}{(T - t)(K_t - 1)} = \alpha_{t+1} \end{aligned} \tag{5.11}$$

Hence, $\alpha_{t+1} \geq \alpha_t$ for $1 \leq t \leq T-1$. Then

$$\sum_{i=1}^{T-1} \alpha_i F_i + \alpha_T F_T = 1 - \Delta - \alpha_T (K_{T-1} - 1) + \alpha_T F_T \tag{5.12}$$

and

$$\sum_{i=1}^{T} \alpha_i F_i = 1 - \Delta + \alpha_T [F_T - (K_{T-1} - 1)] \tag{5.13}$$

Since $F_T \leq K_T - 1 \leq K_{T-1} - 1$, $F_T - (K_{T-1} - 1) \leq 0$, and $\alpha_T \geq \alpha_1 > 0$ (due to Equation (5.11)), we have

$$\sum_{i=1}^{T} \alpha_i F_i = 1 - \Delta + \alpha_T [F_T - (K_{T-1} - 1)] \leq 1 - \Delta \tag{5.14}$$

□

Obviously, the adaptive α scheme introduced in Equation (5.9) improves the power of S-Race, compared to the fixed α scheme in Equation (5.8). This is because a larger α allows for more discoveries of pairwise dominance relationships. As shown in the proof, the α value increases with growing t, implying that S-Race behaves more conservative in the early stages of racing. Later on, when more performance vectors are observed, it is more reliable to establish pairwise dominance relationships and eliminate the dominated models. In other words, the overall probability of making any Type II errors will be reduced by utilizing the adaptive α scheme. Meanwhile, the overall accuracy Δ is still strictly controlled beyond a certain level.

Δ is an important parameter in S-Race since it maintains a balance between the accuracy of S-Race in returning all Pareto front models and the computational cost. To be more specific, higher Δ leads to smaller α values, representing a lower probability of making any false rejections and falsely removing any non-dominated models. Correspondingly, high Δ results in a more conservative racing procedure that is more cautious in model elimination. Consequently, it requires more computational effort. In the extreme case when $\Delta = 1$ and $\alpha = 0$, no model will be removed and S-Race will behave like the Brute Force Approach (BFA). To sum up, the selection of a proper Δ reflects the user's preference over the final quality of the ensemble of models returned by S-Race and its computational cost.

Experiments and Applications

In this section, we apply S-Race on three MOMS problems to demonstrate its efficiency, which are choosing optimal parameters for Support Vector Machines (SVMs) for classification and Artificial Bee Colony (ABC) algorithms for numerical optimization, and hybrid Recommender Systems (RSs) construction for movie recommendation.

Performance Metrics

The ultimate goal of S-Race is to identify the ensemble of Pareto front models exactly as the ones returned by the multi-objective BFA which allocates the total computational resources uniformly among all candidate models. Therefore, we compare the performance of S-Race to the BFA to demonstrate its efficiency in MOMS. In this work, the BFA is equivalent to running S-Race with a single step using all of the available validation samples.

Assume that $\mathcal{P}_S$ and $\mathcal{P}_{BFA}$ denote the final set of models returned by S-Race and BFA respectively, the selected performance metrics *retention* and *excess*, denoted by R and E, are defined as follows:

$$R \triangleq \frac{|\mathcal{P}_S \cap \mathcal{P}_{BFA}|}{|\mathcal{P}_{BFA}|} \tag{5.15}$$

$$E \triangleq \frac{|\mathcal{P}_S \setminus \mathcal{P}_{BFA}|}{|\mathcal{P}_S|} = 1 - \frac{|\mathcal{P}_S \cap \mathcal{P}_{BFA}|}{|\mathcal{P}_S|} \tag{5.16}$$

As their name suggests, R measures S-Race's ability of retaining the ensemble of non-dominated models returned by the BFA, and E measures its ability of eliminating the dominated models identified by the BFA. R is directly regulated by the predefined Δ value. The overall probability of committing any Type II errors is, however, beyond the control of S-Race, which implies that E is not strictly controlled in S-Race. Ideally, it is desirable to have $R = 1$ and $E = 0$, indicating that $\mathcal{P}_S = \mathcal{P}_{BFA}$. As a matter of fact, high R and low E are acceptable. Note that in information retrieval contexts, R and $1 - E$ are also referred to as *precision* and *recall*, respectively.

In addition to the accuracy of S-Race, the time efficiency of S-Race is also of great interest. Therefore, a third performance metric T is adopted which denotes the ratio between the computation time of S-Race and the BFA.

Theoretically, the computational complexity of S-Race is $O(DV\overline{K} + V\overline{K}^2 \log \overline{K} + V^2\overline{K}^2)$ where D is the number of objectives, V is the total number of validation samples and $\overline{K}$ is the average number of models during the entire racing process. Similarly, the computational complexity of the BFA is $O(DVK + VK^2 \log K + V^2K^2)$. In S-Race, since the under-performing models are gradually eliminated, it is expected to have $\overline{K} < K$ and $T < 1$.

Support Vector Machine for Classification

In this section, S-Race was applied for tuning the parameters of SVM [36] for binary- and ternary-class classification problems. For each classification problem, the classification accuracies, measured as Percent of Correctly Classified (PCC), of every class are of equal importance. Therefore, the performance vector of each SVM consists of the PCCs of every class. In other words, parameter tuning for binary-classification SVM is a bi-objective maximization problem, and parameter tuning for ternary-classification SVM is a tri-objective maximization problem.

A total of 12 datasets were selected [84, 42, 104, 135] (see Table 5.3). Each dataset was randomly split into a training set and a validation set. The initial ensemble of models contained 50 SVMs with different kernel types, kernel parameters and C values (see Table 5.4). In each experiment, all the candidate SVMs were trained on the common training set before S-Race started. At each step of racing, the performances of the remaining models are assessed on a batch of validation samples which was randomly selected without replacement from the validation set. The batch size varied from dataset to dataset in order to ensure the maximum number of steps is 100. S-Race stopped if only one model was left or the maximum number of steps was reached. In order to demonstrate the influence of Δ on S-Race's performance, its value was selected from the discrete set $\{0.7, 0.8, 0.9, 1 - 0.1^9\}$.

The observed retention (R), excess (E), time ratio (T) values, and the number of Pareto front models identified by the BFA ($|\mathcal{P}_{BFA}|$), averaged over 30 runs, are reported in Table 5.5 and Table 5.6. On one hand, it is observed that all the R values are greater than 0.90. To be more specific, in 87.5% cases we have $R \geq 0.95$, and in 32.3% cases we have $R = 1$. The high values of R demonstrate that S-Race is capable of identifying almost the exact ensemble of Pareto optimal models that selected by the BFA. Since the observed R values are always above the predefined Δ values, it implies that the overall probability of making any false discoveries in S-Race is strictly

Table 5.3: Selected Datasets for Classification

name	**Training #**	**Validation Samples #**	**Class #**	**Attribute #**
a2a	2265	10098	2	123
acoustic(1)	2872	3969	2	50
connect-4(1)	6401	13676	2	126
MiniBooNE	2815	4961	2	50
mushrooms	1626	1620	2	112
w1a	2477	15757	2	300
acoustic(2)	2464	2464	3	50
combined	2464	2464	3	100
protein	1915	2209	3	357
seismic	2465	2465	3	50
connect-4(2)	5632	5632	3	126
dna	600	1186	3	180

Table 5.4: SVM Parameter Description

parameter	**range**
C	$[0.001, 50]$
kernel type	$K_{poly}(x_i, x_j) = (a(x_i \cdot x_j) + b)^c$ $K_{rad}(x_i, x_j) = exp(-\gamma \|\|x_i - x_j\|\|^2)$ $K_{sig}(x_i, x_j) = tanh(\gamma(x_i \cdot x_j) + d)$
a	$[1, 10]$
b	$[0, 5]$
c	$[1, 5]$
γ	$\gamma = q^t$, where $q \in (1, 9), t \in (-9, 9)$
d	$[-0.1, 0]$

controlled by the discrete Holm's procedure. Generally, R values increase with growing Δ values. This is because, as Δ approaches to 1, α at every step of S-Race will approach to 0 and it will be less likely to eliminate any non-dominated models by mistake. However, since R is calculated as the ratio in Equation (5.15), it may decrease as Δ increases if $|\mathcal{P}_{BFA}|$ changes more than $|\mathcal{P}_S \cap \mathcal{P}_{BFA}|$. On the other hand, it is observed that about 87.5% E values are below 0.18, and 69.1% of them are even smaller than 0.10. Smaller E values demonstrate that S-Race is capable of eliminating almost

all the dominated models which are the ones identified by the BFA. High Δ values generally result in low E values. Additionally, the standard deviation of the resulting R and E values are 0.01 and 0.20. Since R is strongly related to the overall Type I error probability of S-Race which is strictly controlled at a predefined level, the degree of variation of R is significantly smaller than that of E.

Table 5.5: S-Race on 2-Objective SVM Selection: Average R, E, T and $|\mathcal{P}_{BFA}|$ Values for Varying Δ Values

Δ		Dataset					
		a2a	acoustic(1)	connect-4(1)	MiniBooNe	mushrooms	w1a
0.7	R	0.99	0.99	0.96	0.97	0.94	0.95
	E	0.13	0.06	0.06	0.14	0.03	0.04
	T	0.51	0.50	0.59	0.38	0.26	0.39
	$\lvert\mathcal{P}_{BFA}\rvert$	8.60	4.27	9.00	5.37	2.30	3.67
0.8	R	1.00	0.97	0.97	0.97	0.96	0.95
	E	0.11	0.04	0.04	0.14	0.01	0.08
	T	0.51	0.52	0.61	0.39	0.27	0.41
	$\lvert\mathcal{P}_{BFA}\rvert$	8.83	4.47	9.27	5.67	2.33	3.80
0.9	R	1.00	0.98	0.98	0.99	0.93	0.95
	E	0.09	0.03	0.03	0.17	0.04	0.06
	T	0.59	0.61	0.62	0.88	0.28	0.43
	$\lvert\mathcal{P}_{BFA}\rvert$	9.13	4.57	9.43	6.00	2.50	3.90
$1-0.1^9$	R	1.00	0.98	0.99	1.00	0.99	1.00
	E	0.04	0.07	0.07	0.05	0.02	0.01
	T	0.82	0.86	0.88	0.82	0.61	0.85
	$\lvert\mathcal{P}_{BFA}\rvert$	12.17	14.73	22.30	23.70	5.00	14.27

Regarding the T values, it is always smaller than 1 even when $\Delta = 1 - 0.1^9$. When Δ is not close to 1, S-Race saves about 50% of the computation time taken by the BFA to finish on average. Generally, it is observed that higher Δ values lead to lower computational savings. Because when $\Delta \approx 1$, $\alpha \approx 0$ and it is less likely reject the null hypothesis. And thus more models will be identified as non-dominated models and are retained to the end of racing. Moreover, as the experimental results demonstrated [146], smaller batch size generally results in larger computational savings.

Table 5.6: S-Race on 3-Objective SVM Selection: Average R, E, T and $|\mathcal{P}_{BFA}|$ Values for Varying Δ Values

Δ		Dataset					
		acoustic(2)	combined	protein	seismic	connect-4(2)	dna
0.7	R	0.95	0.97	1.00	1.00	0.90	1.00
	E	0.22	0.09	0.07	0.24	0.05	0.17
	T	0.68	0.49	0.68	0.50	0.46	0.41
	$\lvert\mathcal{P}_{BFA}\rvert$	4.97	3.67	19.80	6.17	3.07	7.90
0.8	R	0.96	0.98	1.00	0.99	0.92	1.00
	E	0.22	0.10	0.07	0.23	0.08	0.15
	T	0.71	0.50	0.70	0.51	0.47	0.41
	$\lvert\mathcal{P}_{BFA}\rvert$	5.47	3.93	20.20	6.73	3.10	8.23
0.9	R	0.91	0.98	1.00	0.99	0.90	1.00
	E	0.29	0.11	0.07	0.22	0.05	0.12
	T	0.72	0.51	0.71	0.53	0.49	0.43
	$\lvert\mathcal{P}_{BFA}\rvert$	6.30	4.37	20.57	7.57	3.53	8.73
$1-0.1^9$	R	1.00	0.97	1.00	0.99	0.96	1.00
	E	0.06	0.09	0.06	0.04	0.04	0.05
	T	0.90	0.89	0.90	0.91	0.78	0.69
	$\lvert\mathcal{P}_{BFA}\rvert$	26.80	23.53	39.47	29.47	9.63	15.47

In Figure 5.1 and Figure 5.2, we provide a visual representation of how R, E, T and P changes during the racing process. The depicted results are average values over 30 runs of S-Race with the *MiniBooNE* dataset. P refers to the proportion of initial candidate models that are retained by S-Race, which starts from 1 and decreases step by step as under-performing models are eliminated gradually. Correspondingly, T values slightly increase as S-Race proceeds but remain uniformly below 1. Moreover, as observed from Figure 5.1 and Figure 5.2, R values stay close to 1 throughout racing for all Δ values, while E values reduce significantly from a value close to 1 to a value close to 0. This implies that S-Race is able to distinguish non-dominated models from dominated ones with high confidence. When $\Delta \in \{0.7, 0.8, 0.9\}$, the behaviors of the average R, E, T and P values are similar to each other. Except when Δ is close to 1, as shown in Figure 5.2b, the computational savings achieved by S-Race, as compared to the BFA, has reduced from about 60% to 20%.

Additionally, we compared the performances of S-Race without and with the adaptive α scheme, to demonstrate the effect of the adaptive α scheme. For S-Race without the adaptive α scheme, the α values assigned for all the families are fixed as α_1 in Equation (5.10). The comparison results are shown in Table 5.7 in which we report the differences between their average E and T values. To be more specific, we compute the differences of the average R, E and T values achieved by S-Race with and without the adaptive α scheme over all 12 datasets. Relative statistics, including the *average*, the *maximum* and the *minimum*, are computed. The differences in R are insignificant, indicating that the overall probability of making any Type I errors is always strictly controlled at level $1 - \Delta$. E and T values are, however, significantly reduced as a result of employing the adaptive α scheme as shown in Table 5.7. The comparison results demonstrate that the adaptive α scheme enhances the power of S-Race in eliminating dominated models. S-Race with the adaptive α scheme is able to further reduce computational effort.

Table 5.7: The Differences of Average E and T Values of S-Race Without and With the Adaptive α Scheme on SVM Selection

Δ	E			T		
	Avg	Max	Min	Avg	Max	Min
0.7	0.135	0.309	0.035	0.027	0.047	0.012
0.8	0.135	0.302	0.029	0.022	0.050	0.008
0.9	0.142	0.301	0.031	0.024	0.057	0.009
$1 - 0.1^9$	0.035	0.117	0.007	0.006	0.017	0.001

Artificial Bee Colony Algorithm for Numerical Optimization

As one of the most popular swarm intelligence algorithms, the ABC algorithm [76] has been widely applied to a variety of real-world problems in industrial engineering, mechanical engineering, software engineering, etc. In this section, S-Race is used to select the optimal parameters of the ABC algorithm for numerical optimization problems. Given a numerical optimization function, the best

objective value obtained and the actual computation time required to achieve that objective value are two performance metrics of ABC algorithms that we are interested in. Therefore, the ABC parameter selection problem is a binary-objective model selection problem. More specifically, in the experiments conducted in this section, each candidate model is an ABC algorithm and its performance vector, corresponding to a given optimization function, consists of the best objective value achieved and the computation time it takes to finish.

In this set of experiments, S-Race starts off with a set of 54 ABC algorithms with different parameter settings described in Table 5.8, including the population size (NP), the initialization scheme, the selection scheme and the mutation scheme. For each ABC algorithm, the number of food sources and the limit were set as $0.5 \times NP$ and $NS \times D$ where $D \in \{2, 5, 30, 50, 100\}$ refers to the problem dimension, respectively. The numerical optimization functions used in the experiments were generated via a Gaussian landscape generator [142]. The maximal number of steps in S-Race was set to 50 and, at each step, a different objective function was randomly generated to assess the performances of candidate models. The ABC algorithm stopped if it reached the maximal number of iterations (*i.e.* 500 for $D = 2, 5$, and 2000 for $D = 30, 50, 100$), or it converged. In this set of experiments, the Δ value was selected from the discrete set $\{0.7, 0.8, 0.9, 1 - 0.1^9\}$.

Average R, E, T and $|\mathcal{P}_{BFA}|$ values over 30 trials are reported in Table 5.9. First of all, it is observed that all R values are above 96%, demonstrating that S-Race has successfully returned the set of Pareto front models identified by the BFA at the prescribed confidence level. Regarding the E values, 85% of them are no greater than 15%, implying that S-Race is able to identify and eliminate most of the dominated models. When $D = 30$, however, E values are relatively high on average. Generally speaking, it is hard to find an optimal solution for high-dimensional and multi-model functions. Therefore, the candidate ABC algorithms demonstrate similar performances in the experiments with $D = 30$. In this case, however, it is recommended to have more test instances for further comparisons. In addition, it is obvious that the T values increase with increasing Δ.

Table 5.8: ABC Parameter Description

parameter	**choices**
NP	50, 150
initialization scheme	random sampling Latin square sampling opposition-based sampling [44]
selection scheme [101]	proportional selection disruptive selection rank selection
mutation scheme [3]	original mutation best-based mutation dist-based mutation

This is because, when Δ is large, it is harder for S-Race to make a rejection of the null hypothesis and more of the dominated models are identified at a later stage of the racing process. All T values are, however, strictly smaller than 1, even when $\Delta = 1 - 0.1^9$.

The relevant statistics of the differences between average E and T values of S-Race without and with the adaptive α scheme are shown in Table 5.10. As demonstrated by the experimental results, updating α adaptively during racing will improve the power and computational efficiency of S-Race, especially when the performances of the candidates are very similar to each other (*e.g.* when $D = 30$).

Based on the ABC algorithms returned as Pareto optimal, some interesting facts can be determined about the parameters. Firstly, all of the Pareto optimal ABC algorithms have $NP = 50$ for all studied D values. The experimental results demonstrate that, for the considered optimization problems, population size of 50 provides enough diversity and good convergence behavior. Small population size is good enough to balance the explorative and exploitative behavior of the ABC algorithms in order to obtain good results in reasonable time frame. Secondly, the three initialization schemes provide different but well-balanced trade-off between the quality of the best objective

Table 5.9: S-Race on 2-Objective ABC Selection: Average R, E, T and $|\mathcal{P}_{BFA}|$ Values for Varying Δ Values

Δ		D				
		2	5	30	50	100
0.7	R	0.98	0.99	0.98	1.00	0.99
	E	0.06	0.12	0.22	0.11	0.07
	T	0.36	0.52	0.64	0.50	0.42
	$\lvert\mathcal{P}_{BFA}\rvert$	3.70	11.50	15.63	14.83	14.93
0.8	R	0.98	0.99	0.96	0.99	0.98
	E	0.08	0.15	0.21	0.09	0.06
	T	0.38	0.54	0.67	0.52	0.44
	$\lvert\mathcal{P}_{BFA}\rvert$	3.87	11.87	16.83	15.33	15.50
0.9	R	0.97	0.99	0.97	0.99	0.99
	E	0.08	0.14	0.20	0.08	0.04
	T	0.40	0.57	0.70	0.55	0.45
	$\lvert\mathcal{P}_{BFA}\rvert$	4.37	13.07	18.13	15.97	16.13
$1 - 0.1^9$	R	1.00	1.00	1.00	1.00	1.00
	E	0.07	0.15	0.04	0.13	0.00
	T	0.91	0.99	0.99	0.98	0.88
	$\lvert\mathcal{P}_{BFA}\rvert$	21.23	43.20	51.90	44.77	18.00

value obtained and the computational time required. Random sampling is easy to implement and the resulting population converges to a local optimum quickly. On the contrary, opposition-based sampling affords more diversity and better results. Latin Square sampling falls somewhere in between. Thirdly, proportional selection and disruptive selection are recommended. The former one has the advantage of being fast in convergence, and the latter one possesses more diversity. Fourthly, the original mutation operator and the best-based mutation operator are preferred over the dist-based mutation operator. This is because the original mutation operator and the best-based mutation operator demonstrate better trade-off between exploration and exploitation in searching for an optimum objective value.

Table 5.10: The Differences of Average E and T Values of S-Race Without and With the Adaptive α Scheme on ABC Selection

Δ	E			T		
	Avg	Max	Min	Avg	Max	Min
0.7	0.105	0.265	0.040	0.016	0.021	0.006
0.8	0.100	0.249	0.031	0.016	0.021	0.002
0.9	0.091	0.256	0.030	0.015	0.020	0.002
$1 - 0.1^9$	0.002	0.015	0.006	0.001	0.001	0.001

Hybrid Recommendation System for Recommendation Tasks

One common practice in RS research is the hybrid RS approach which combines different recommendation techniques to overcome common shortcomings (*e.g.* cold start [113], sparsity [82]), and thus achieves improved performance [18, 14, 107]. A variety of basic RS techniques have been proposed in the literature: Collaborative Filtering (CF), content-based, knowledge-based and demographic techniques [108]. Netflix, for instance, is a good example of hybrid RS which hybridizes CF and content-based systems. A popular hybridization technique is weighted hybridization in which the final predicted rating is a linear combination of the ratings returned by the component RSs. The performance of such hybrid RS is determined by the assigned weights of its component RSs. Finding the optimal set of weights is actually a multi-objective optimization problem in practice. There are several criteria for measuring the goodness of a RS, such as prediction accuracy, ranking accuracy, diversity, and utility score, etc. Therefore, S-Race is appropriate for finding the optimal weighted hybrid RSs with respect to multiple optimization criteria.

In this section, we applied S-Race on Pareto optimal hybrid RSs selection to illustrate its performance. In this set of experiments, the candidate models were weighted hybridization of 7 CF RSs available in the `PREA` package [81]: User-based CF, Item-based CF, Slope One, Regularized Singular Value Decomposition, Probabilistic Matrix Factorization, Singleton Global Local Low-Rank

Matrix Approximation, and Rank-based Recommenders. Moreover, three RS performance metrics provided in the `PREA` package [81] were selected, which are Asymmetric measures ($Asym$), Half-life Utility (HLU) and Kendall's Tau (KT). They are used to measure the prediction accuracy, utility score and ranking accuracy, respectively. For consistency and simplicity, the performance vector of each candidate RS contains $Asym$, $1 - HLU$ and KT. Therefore, the resulting hybrid RS selection problem is a ternary-objective minimization problem. Two movie recommendation dataset were used: MovieLens and Netflix. In the MovieLens dataset, there are a total of 6039 users, 3883 items, and 1000209 ratings. In the Netflix dataset, there are a total of 8662 users, 3000 items and 293299 ratings. Each dataset was randomly split into two parts: 75% of the ratings were used as a training set for model selection, and the remaining 25% of the ratings were used as a testing set for evaluating the set of Pareto optimal models returned by S-Race. The training set was divided into 100 batches and, at each step of racing, one batch is randomly sampled without replacement to assess the performances of the remaining models. 100 hybrid RSs were initialized whose weights were generated via Latin Square sampling. Similarly, the Δ value was selected from the discrete set $\{0.7, 0.8, 0.9, 1 - 0.1^9\}$.

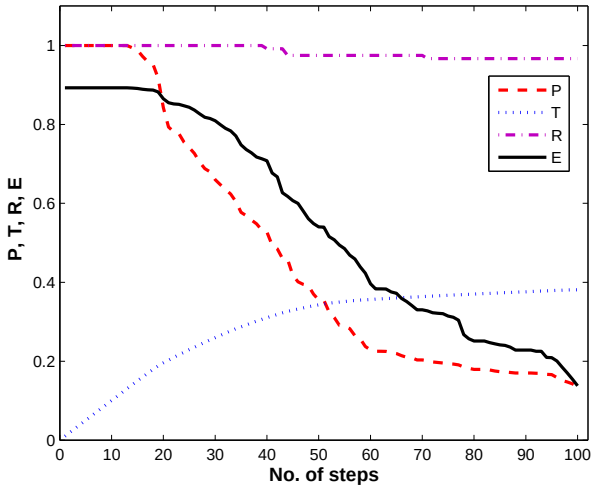
The resulting R, E, T and $|\mathcal{P}_{BFA}|$ values, averaged over 30 runs, are shown in Table 5.11. All the R values in the Movielens dataset are close to 1, indicating S-Race is able to return all of the Pareto front hybrid RSs as the BFA identifies. For the the Netflix movie recommendation task, the resulting R values are all above 0.92 and slightly increase with increasing Δ as expected. Meanwhile, most of the E values are below 0.10, indicating that S-Race fails at eliminating a small portion of dominated models. Moreover, S-Race consumes only half of the computational resources that BFA uses according to the T values reported, except when Δ is extremely close to 1.

To further illustrate the Pareto optimality of the hybrid RSs returned by S-Race, we compare the performances of dominated hybrid RSs and non-dominated hybrid RSs on a test set, as depicted in

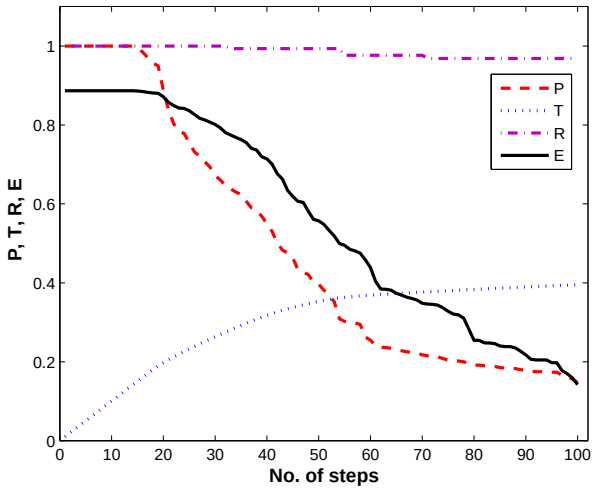
Table 5.11: S-Race on 3-Objective Hybrid RS Selection: Average R, E, T and $|\mathcal{P}_{BFA}|$ Values for Varying Δ Values

Δ		Dataset	
		MovieLens	Netflix
0.7	R	1.00	0.92
	E	0.06	0.11
	T	0.35	0.50
	$\lvert\mathcal{P}_{BFA}\rvert$	6.67	6.50
0.8	R	0.98	0.94
	E	0.07	0.10
	T	0.35	0.52
	$\lvert\mathcal{P}_{BFA}\rvert$	7.07	6.97
0.9	R	1.00	0.94
	E	0.07	0.10
	T	0.38	0.55
	$\lvert\mathcal{P}_{BFA}\rvert$	7.67	7.93
$1-0.1^9$	R	1.00	0.98
	E	0.06	0.04
	T	0.79	0.89
	$\lvert\mathcal{P}_{BFA}\rvert$	26.30	41.43

Figure 5.3 and Figure 5.4. For each dataset, we collect the results of a single trial of S-Race with $\Delta = 0.9$ and $\Delta = 1-0.1^9$. In Figure 5.3 and Figure 5.4, the performance vectors of the dominated models are denoted by black circles (○), while those of the non-dominated models are denoted by red-filled circles (●). First of all, it is observed that the non-dominated models identified by S-Race concentrated around (0.55, 0.305,0.335) in Figure 5.3 and (0.62, 0.174, 0.186) in Figure 5.4, where all three objectives are minimized simultaneously. Therefore, it can be concluded that the models that are Pareto optimal on the training set are also Pareto optimal on the test set as desired. Moreover, it is obvious that more models will be returned by S-Race as non-dominated models when Δ increases. This is because S-Race can hardly make any null hypothesis rejections when Δ approaches 1 and, correspondingly, when α approaches 0.

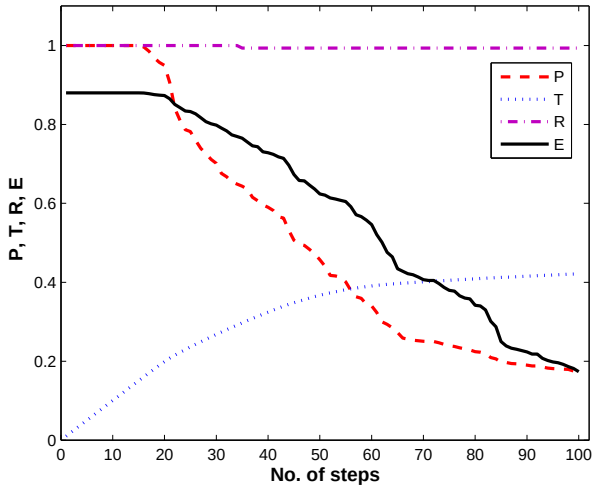


(a) $\Delta = 0.7$



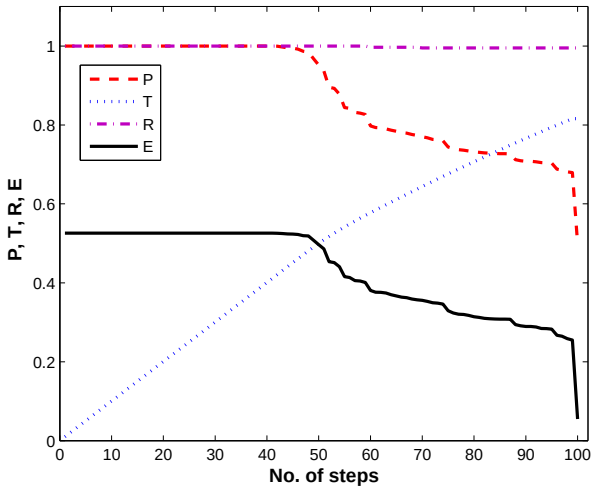
(b) $\Delta = 0.8$

Figure 5.1: Change of average R, E, T and P values in S-Race for MiniBooNE dataset for $\Delta = 0.7, 0.8$.



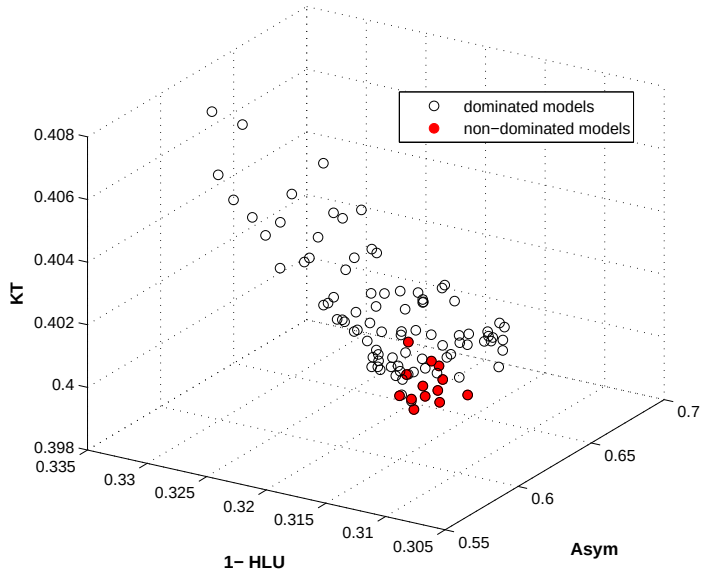
//

(a) $\Delta = 0.9$

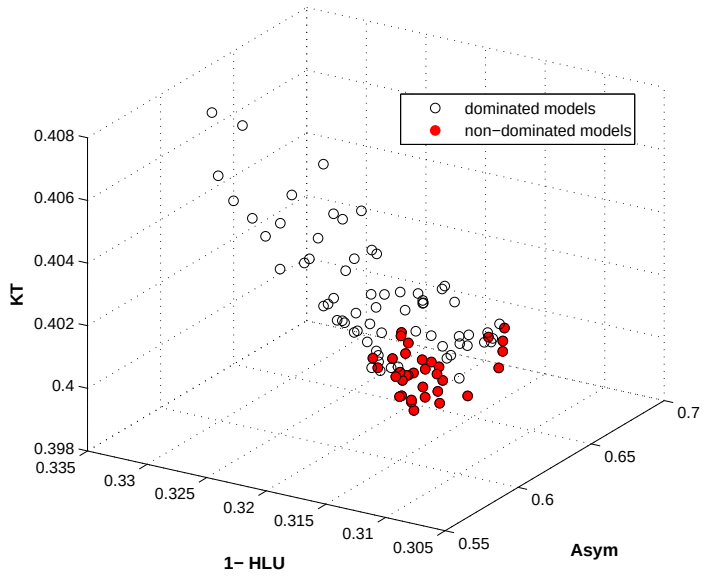


(b) $\Delta = 1 - 0.1^9$

Figure 5.2: Change of average R, E, T and P values in S-Race for MiniBooNE dataset for $\Delta = 0.9, 1 - 0.1^9$.

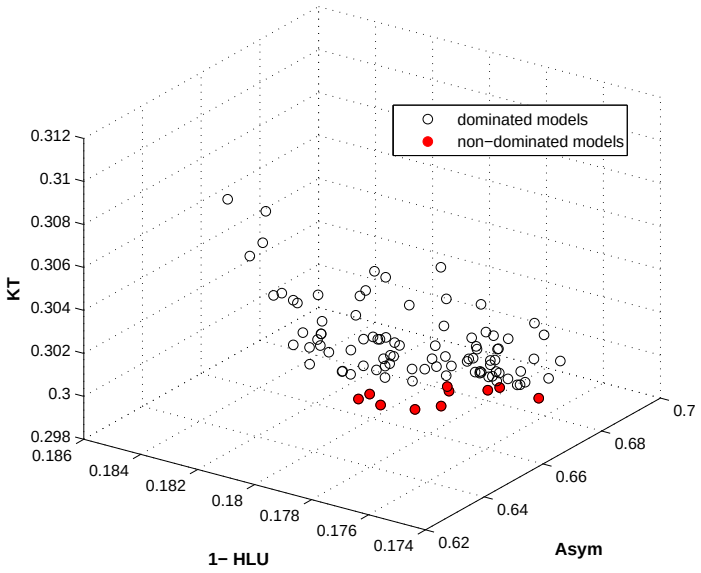


(a) MovieLens, $\Delta = 0.9$

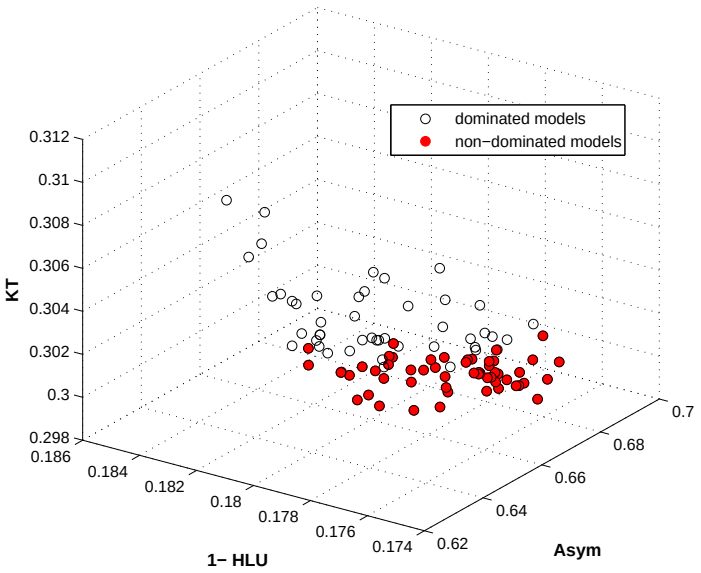


(b) MovieLens, $\Delta = 1 - 0.1^9$

Figure 5.3: Comparison of the performance of dominated and non-dominated hybrid RSs on the test set for MovieLens dataset.



(a) NetFlix, $\Delta = 0.9$



(b) NetFlix, $\Delta = 1 - 0.1^9$

Figure 5.4: Comparison of the performance of dominated and non-dominated hybrid RSs on the test set for NetFlix dataset.

CHAPTER 6: MULTI-OBJECTIVE RACING BASED ON SEQUENTIAL PROBABILITY RATIO TEST WITH INDIFFERENCE ZONE

In this chapter, we introduce a novel Multi-Objective Racing Algorithm (MORA) based on the Sequential Probability Ratio with INdifference zone Test, named SPRINT-Race. SPRINT-Race is the first MORA with a fixed confidence setting in which the size of the validation set for model selection is not available a priori and the number of validation samples could be possibly infinite. In the fixed-confidence setting, the goal of the forecaster is to minimize the number of validation instances required to achieve a fixed confidence about the optimality of the returned models. Distinct from S-Race introduced in Chapter 5, SPRINT-Race is applicable for situations where the samples arrive sequentially in an online fashion (*e.g.* online controlled experiments, etc.).

In SPRINT-Race, both the dominance and non-dominance relationship between a pair of models is statistically inferred via a ternary-decision non-parametric dual-SPRT with Trinomial distribution. The comparison between a pair of models will stop automatically when either dominance or non-dominance relation is established with prescribed confidence. The employment of Sequential Probability Ratio Test (SPRT) in SPRINT-Race only necessitates a near-minimal sample complexity. Moreover, the concept of indifference zone is utilized in SPRINT-Race, which allows near-optimal models to be returned with the benefit of reduced sample complexity. Using an indifference zone reduces the risk of loosing optimal models due to the existence of noise in performance measurements, as well as the computational efforts required for a thorough comparison. Additionally, the overall probability of making any Type I or Type II errors is strictly controlled via the sequential Holm's step-down Family-Wise Error Rate (FWER) control procedure. In SPRINT-Race, the maximum probability of falsely eliminating any non-dominated model or mistakenly returning any dominated model is predefined.

Sequential Probability Ratio Test

Given a sequence of identically and independently distributed (i.i.d.) random variables (RVs) $x_i, i = 1, \cdots, N$ following an unknown distribution $g_\theta(x)$ parameterized by θ only, a pair of simple hypotheses is made about the value of θ: $H_0 : \theta = \theta_0$ and $H_1 : \theta = \theta_1$. In a traditional fixed-sample test, a collection of samples is gathered in advance as evidence supporting either H_0 or H_1. A test statistic will be computed based on all the samples, and either H_0 or H_1 is accepted. Let us denote the maximum probability of a Type I error by $\alpha \in [0, 1]$ and the maximum probability of a Type II error by $\beta \in [0, 1]$ for any testing procedure. In a fixed-sample test, β is generally a function of α and the sample size N. Typically for fixed-sample test, the one that minimizes β is always preferred as *the most powerful test* with given α and N. As indicated by the Neyman-Pearson lemma [99], the most powerful fixed-sample test for H_0 and H_1 is the likelihood ratio test with the following test statistics

$$\lambda \triangleq \sum_{i=1}^{N} \left(\ln g_{\theta_1}(x_i) - \ln g_{\theta_0}(x_i) \right) \tag{6.1}$$

which is the natural log of the ratio of the likelihood of observed data under H_1 and H_0.

Distinguished from the fixed-sample test, the number of samples required by a sequential test procedure is a RV. Each individual sample arrives sequentially until a decision is made, and either H_0 or H_1 is accepted. At each step of a sequential test, after the test statistic is updated based on the newly sampled data, there are three possible actions to take: accept H_0, accept H_1, or continue sampling. Compared with the fixed-sample test, the sequential test takes advantage of the accumulated information during the data sampling process, and thus it is expected to require a smaller sample complexity. In this work, the SPRT [132], which is the most popular sequential testing procedure, is employed for pairwise dominance/non-dominance relationship inference. For

a pair of simple hypotheses H_0 and H_1, the SPRT utilizes a likelihood ratio test statistic with the following decision rules: assume that λ_n denotes the test statistic at the n^{th} step, if $\lambda_n \leq A$, H_0 is accepted; if $\lambda_n \geq B$, H_1 is accepted; otherwise, we continue sampling and $n = n + 1$. Generally, by setting $A = \ln \frac{\beta}{1-\alpha}$ and $B = \ln \frac{1-\beta}{\alpha}$, it is guaranteed that the maximum probability of Type I and Type II errors do not exceed α and β, respectively.

Statistical Inference of Dominance and Non-Dominance in SPRINT-Race

Dual-SPRT

Since we are dealing with stochastic performance vectors, the dominance and non-dominance relation between a pair of models should be established via formal tests of hypothesis. As we discussed in Section 5 of Chapter 5, a non-parametric pairwise test is preferred for its robustness.

According to the concept of probabilistic dominance, there are three outcomes of a pairwise comparison between the i^{th} model M_i and the j^{th} model M_j: $M_i \succ M_j$, if and only if $\mathbb{P}\left(\mathbf{f}(M_i) \succ \mathbf{f}(M_j)\right) > \mathbb{P}\left(\mathbf{f}(M_i) \prec \mathbf{f}(M_j)\right)$; $M_i \prec M_j$, if and only if $\mathbb{P}\left(\mathbf{f}(M_i) \succ \mathbf{f}(M_j)\right) < \mathbb{P}\left(\mathbf{f}(M_i) \prec \mathbf{f}(M_j)\right)$; and $M_i \sim M_j$, if and only if $\mathbb{P}\left(\mathbf{f}(M_i) \succ \mathbf{f}(M_j)\right) = \mathbb{P}\left(\mathbf{f}(M_i) \prec \mathbf{f}(M_j)\right)$. Let N_{ij} denote the RV which represents the number of times that $\mathbf{f}(M_i) \succ \mathbf{f}(M_j)$, with observed value n_{ij}. Additionally, let S denote the total number of comparisons between M_i and M_j with observed value s. Let us assume that $p_{ij} \triangleq \mathbb{P}\left(N_{ij} = 1 | S = 1\right)$ and $p_{ji} \triangleq \mathbb{P}\left(N_{ji} = 1 | S = 1\right)$. Then, $(N_{ij}, N_{ji}, S - N_{ij} - N_{ji}) \sim Trinomial\left(p_{ij}, p_{ji}, 1 - p_{ij} - p_{ji}\right)$ with the following Probability Mass Function (PMF) when $s = 1$

$$f\left(n_{ij}, n_{ji}; p_{ij}, p_{ji}\right) = \frac{1}{n_{ij}! n_{ji}! (1 - n_{ij} - n_{ji})} p_{ij}^{n_{ij}} p_{ji}^{n_{ji}} (1 - p_{ij} - p_{ji})^{1 - n_{ij} - n_{ji}} \tag{6.2}$$

where $n_{ij} \in \{0,1\}$, $n_{ji} \in \{0,1\}$, $n_{ij} + n_{ji} \leq 1$ and $0 \leq p_{ij} \leq 1 - p_{ji} \leq 1$.

In order to conclude any pairwise dominance and non-dominance relationship, we are only interested in the equivalence of p_{ij} and p_{ji}. Thus, the PMF can be rewritten as

$$f\left(n_{ij}, n_{ji}; \gamma, \eta\right) = \binom{1}{n_{ij}+n_{ji}} \gamma^{n_{ij}+n_{ji}} (1-\gamma)^{1-n_{ij}-n_{ji}} \binom{n_{ij}+n_{ji}}{n_{ij}} \eta^{n_{ij}} (1-\eta)^{n_{ji}} \tag{6.3}$$

where $\gamma \triangleq p_{ij} + p_{ji}$ and $\eta \triangleq \frac{p_{ij}}{p_{ij}+p_{ji}}$.

Therefore, the nuisance parameter γ will disappear in the test statistic shown in Equation (6.1), since $\sum_{n=1}^{N} \left(n_{ij}^n + n_{ji}^n\right)$ is a *Fraser-sufficient* statistic [46] of γ at the N^{th} step of the SPRT where n_{ij}^n and n_{ji}^n denote the n_{ij} and n_{ji} values at the n^{th} step. Moreover, due to the existence of the monotone likelihood ratio, the *locally most efficient* SPRT test procedure of testing $H_0 : \eta \leq \eta_0$ *vs.* $H_1 : \eta \geq \eta_1$ $(0 < \eta_0 < \eta_1 < 1)$ is equivalent to testing $H_0 : \eta = \eta_0$ *vs.* $H_1 : \eta = \eta_1$ using the following test statistic

$$\lambda_N = \ln \frac{\eta_1}{\eta_0} \sum_{n=1}^{N} n_{ij}^n + \ln \frac{1-\eta_1}{1-\eta_0} \sum_{n=1}^{N} n_{ji}^n \tag{6.4}$$

Regarding the three outcomes of a pairwise comparison between M_i and M_j, the problem of inferring pairwise dominance and non-dominance relationship is equivalent to making a decision among the following three mutually exclusive hypotheses

$$H_0 : \eta < \frac{1}{2}, \quad H_1 : \eta = \frac{1}{2}, \quad H_2 : \eta > \frac{1}{2} \tag{6.5}$$

If H_0 is accepted, it is inferred that M_i dominates M_j and M_j will be eliminated; if H_2 is accepted, it is inferred that M_j dominates M_i and M_i will be eliminated; and if H_1 is accepted, it is inferred that neither one dominates the other, and both of M_i and M_j need to remain in the race.

Moreover, when the concept of indifference zone is utilized, the three hypotheses in Equation (6.5) become

$$H_0 : \eta \leq \frac{1}{2} - \delta, \quad H_1 : \eta = \frac{1}{2}, \quad H_2 : \eta \geq \frac{1}{2} + \delta \tag{6.6}$$

The reason of introducing indifference zone is twofold: i) in practical application, the computational burden of a thorough investigation is high and the user is usually satisfied with near-optimal solutions; and ii) it reduces the risks of mistakenly eliminating non-dominated models due to inaccurate performance measurements (*e.g.* noise). The intervals $\left(\frac{1}{2} - \delta, \frac{1}{2}\right)$ and $\left(\frac{1}{2}, \frac{1}{2} + \delta\right)$ are called *indifference zones*. When $\eta \in \left(\frac{1}{2} - \delta, \frac{1}{2}\right)$, there is no difference if H_0 or H_1 is accepted, but the rejection of H_2 is strongly preferred. Similarly, if $\eta \in \left(\frac{1}{2}, \frac{1}{2} + \delta\right)$, no mistake is committed if either H_1 or H_2 is accepted. The selection of a proper δ value is not a statistical problem. It is totally dependent on practical concerns of the users.

The above described ternary-decision test procedure is called a dual-SPRT and is tackled by the Sobel-Wald test procedure [119]. In the dual-SPRT, two binary-decision SPRTs are combined as follows

$$\begin{array}{lll} \mathbf{SPRT^1} & H_0^1 : \eta \leq \frac{1}{2} - \delta & H_1^1 : \eta \geq \frac{1}{2} \\ \mathbf{SPRT^2} & H_0^2 : \eta \leq \frac{1}{2} & H_1^2 : \eta \geq \frac{1}{2} + \delta \end{array} \tag{6.7}$$

As a result, the decision rules of the dual-SPRT is summarized in Table 6.1.

Table 6.1: Decision Rules of Dual-SPRT

$SPRT^1$ accepts	$SPRT^2$ accepts	dual-SPRT accepts
$H_0^1 : \eta \leq \frac{1}{2} - \delta$	$H_0^2 : \eta \leq \frac{1}{2}$	$H_0 : \eta \leq \frac{1}{2} - \delta$
$H_1^1 : \eta \geq \frac{1}{2}$	$H_0^2 : \eta \leq \frac{1}{2}$	$H_1 : \eta = \frac{1}{2}$
$H_1^1 : \eta \geq \frac{1}{2}$	$H_1^2 : \eta \geq \frac{1}{2} + \delta$	$H_2 : \eta \geq \frac{1}{2} + \delta$

The decision rules are graphically presented in Figure 6.1. Starting from the origin, crossing EI and then AB leads to acceptance of H_0; crossing EI and then ID, or EI and then CI, or CI and then IF, or CI and then EI leads to acceptance of H_1; and crossing CI and then GH leads to acceptance of H_2. It is easy to show that H_0^1 and H_1^2 can not be accepted at the same time.

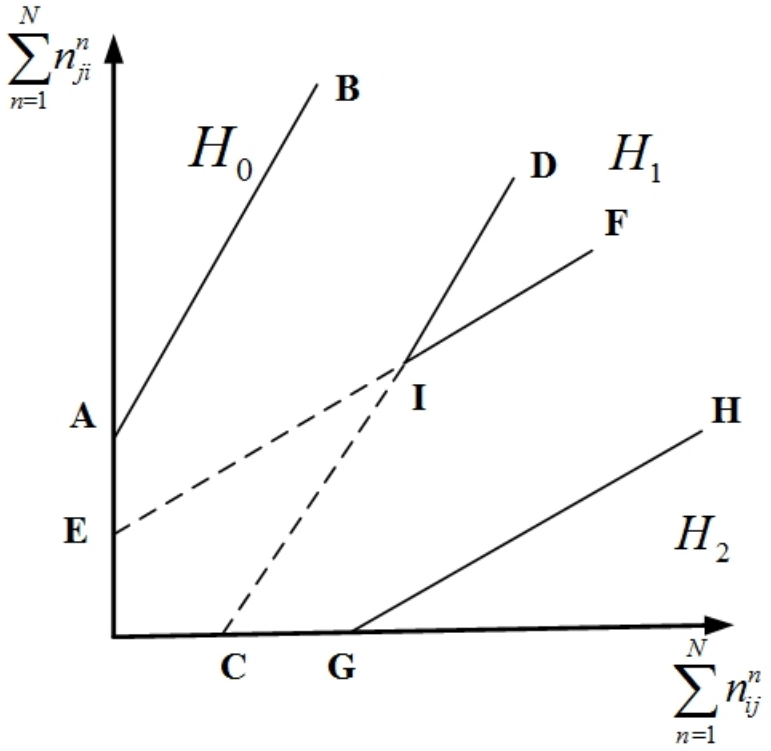


Figure 6.1: Decision rules of a ternary-decision Dual-SPRT

Sequential Holm's Step-Down Procedure

Assume that SPRINT-Race starts off with K models, there will be a total of $\binom{K}{2}$ dual-SPRTs and, correspondingly, $K(K+1)$ component SPRTs. As a typical multiple hypotheses problem, multiple comparison correction is required to control the overall probability of making any Type I or Type II errors. There are two types of FWER involved in multiple sequential test: Family-Wise Type I Error Rate (FWER-I) and Family-Wise Type II Error Rate (FWER-II) which are denoted by $FWER_I$ and $FWER_{II}$ respectively.

$$\begin{aligned} FWER_I &\triangleq \mathbb{P}\left(\bigcup_{i=1}^{K(K-1)} reject\ H_0^i \middle| H_0^i\right) \\ FWER_{II} &\triangleq \mathbb{P}\left(\bigcup_{i=1}^{K(K-1)} reject\ H_1^i \middle| H_1^i\right) \end{aligned} \tag{6.8}$$

As shown in Equation (6.8), FWER-I refers to the overall probability of making any Type I errors, and FWER-II denotes the overall probability of making any Type II errors. Distinct from the non-sequential FWER control methods, both FWER-I and FWER-II are expected to be controlled by the sequential FWER control approaches at predefined levels with low sampling complexity [38, 39, 15]. In SPRINT-Race, a sequential Holm's step-down procedure [15] is employed to control FWER-I at level α and FWER-II at level β. Similar to the non-sequential Holm's procedure, the sequential Holm's FWER control method is a closed testing procedure that controls both FWER-I and FWER-II in a strong sense without any assumptions of the interdependency among multiple hypotheses and their samples. Moreover, it was shown by the authors that the sequential Holm's procedure is more powerful than other sequential FWER control methods proposed in the literature, as well as its non-sequential analogues. A detailed description of the sequential Holm's procedure is provide in Table 6.2.

Table 6.2: Sequential Holm's Step-Down Procedure

	Given a predefined maximum FWER-I α, FWER-II β, and a family of k tests of hypothesis
Step 1	For $s = 1, 2, \cdots, k$ compute k adjusted confidence levels $\alpha_s = \frac{(k-s+1-\beta)\alpha}{(k-s+1)(k-\beta)}$, $\beta_s = \frac{(k-s+1-\alpha)\beta}{(k-s+1)(k-\alpha)}$
Step 2	Subsequently, for $s = 1, 2, \cdots, k$ compute k adjusted stopping boundaries $A_s = \ln \frac{\beta}{(k-s+1)(1-\alpha_s)}$, $B_s = \ln \frac{(k-s+1)(1-\beta_s)}{\alpha}$
Step 3	At n^{th} step of sequential testing, assume that a_n null hypotheses have been accepted, and r_n null hypotheses have been rejected. Accept the remaining null hypotheses of the tests with $\lambda_n \leq A_{a_n+1}$, and reject those with $\lambda_n \geq B_{r_n+1}$

SPRINT-Race Specifics

The framework of SPRINT-Race is provided in Algorithm 4. Initially, SPRINT-Race starts off with K candidate models. During racing, the problem instances are randomly sampled from the problem space and the remaining models are evaluated, and the test statistics for the corresponding dual-SPRTs are updated according to the resulting performance vectors. In each dual-SPRT of M_i and M_j, if H_0 is accepted, M_i will be inferred as being dominated by M_j and thus be eliminated from racing. Subsequently, all the dual-SPRTs involving M_i will be stopped. If H_2 is accepted, M_j will be removed and all the dual-SPRTs involving M_j will be ceased. If H_1 is accepted, however, it is determined that M_i and M_j are non-dominated to each other and thus only the current dual-SPRT involving both of them is terminated. The racing process repeats until all dual-SPRTs are terminated and no more pairwise comparison is needed. Note that the individual boundary values A and B of each SPRT are adjusted according to the sequential Holm's procedure SeqHolms(α, β) with given α and β to ensure that $FWER_I \leq \alpha$ and $FWER_{II} \leq \beta$.

As a fixed-confidence Multi-Objective Model Selection (MOMS) algorithm, the overall probability of making any false discoveries of SPRINT-Race is strictly controlled at a user-specified level via

Algorithm 4: SPRINT-Race Pseudo-code

input : α, β, δ, a stochastic multi-objective optimization function $\mathbf{f}$, a possibly infinite problem space $\mathcal{I}$, and $Pool \leftarrow \{M_1, M_2, \cdots, M_K\}$ $(K \geq 2)$

output: $Pool$

```
Initialize N = 1
repeat
    Randomly sample a problem instance I_N ∈ I
    foreach model M_i ∈ Pool do
        foreach each model M_j ∈ Pool  s.t.  i < j do
            if the corresponding dual-SPRT continues then
                Evaluate M_i and M_j on I_N
                Update the corresponding λ_N in Equation (6.4)
                Obtain A, B ∈ SeqHolms(α, β)
                if H_0 is accepted then
                    Pool ← Pool \ {M_i}
                    Stop all dual-SPRTs involving M_i
                else if H_2 is accepted then
                    Pool ← Pool \ {M_j}
                    Stop all dual-SPRTs involving M_j
                else if H_1 is accepted then
                    Stop the dual-SPRT involving M_i and M_j
            end
        end
    end
    N = N + 1
until All dual-SPRTs are terminated
```

the set of dual-SPRTs and the sequential Holm's step-down FWER control method. There are multiple possible false discoveries in a ternary-decision dual-SPRT when compared to a binary-decision SPRT with only Type I and Type II errors. Therefore, the probability of any erroneous decisions of a dual-SPRT, denoted by $\omega(\eta)$, is carefully analyzed in Table 6.3 [119, 23].

Table 6.3: Error Probability Analysis of Dual-SPRT

Interval	**Wrong Decisions**	$\boldsymbol{\omega(\eta)}$
$\eta \leq \frac{1}{2} - \delta$	accept H_1 or H_2	$\omega(\eta) = 1 - P(H_0^1 \vert \eta, \gamma) \leq \alpha_1$
$\frac{1}{2} - \delta < \eta < \frac{1}{2}$	accept H_2	$\omega(\eta) = P(H_1^2 \vert \eta, \gamma) < \alpha_2$
$\eta = \frac{1}{2}$	accept H_0 or H_2	$\omega(\eta) \leq P(H_0^1 \vert \eta, \gamma) + P(H_1^2 \vert \eta, \gamma) \leq \alpha_2 + \beta_1$
$\frac{1}{2} < \eta < \frac{1}{2} + \delta$	accept H_0	$\omega(\eta) = P(H_0^1 \vert \eta, \gamma) < \beta_1$
$\eta \geq \frac{1}{2} + \delta$	accept H_0 or H_1	$\omega(\eta) = 1 - P(H_1^2 \vert \eta, \gamma) \leq \beta_2$

The significance level of the dual-SPRT, denoted by ω^*, is measured as the maximum possible $\omega(\eta)$ with $\eta \in (0, 1)$. In other words, we have

$$\begin{aligned} \omega^* &\triangleq \max_{\eta \in [0,1]} \omega(\eta) \\ &= \max(\alpha_1, \alpha_2, \alpha_2 + \beta_1, \beta_1, \beta_2) \\ &\leq \alpha_1 + \beta_1 + \alpha_2 + \beta_2 \end{aligned} \tag{6.9}$$

Let Ω denote the overall probability of falsely eliminating any non-dominated models or returning any dominated models in SPRINT-Race. Obviously, Ω is dependent on the significance level of each individual dual-SPRT as well as the overall $FWER_I$ and $FWER_{II}$. According to the Bonferroni inequality, we have

$$\begin{aligned} \Omega &\triangleq \mathbb{P}\left\{\text{at least one Type I or Type II error in SPRINT-Race}\right\} \\ &= \mathbb{P}\left\{\text{at least one Type I or Type II error in any SPRTs}\right\} \\ &\leq FWER_I + FWER_{II} \\ &\leq \alpha + \beta \end{aligned} \tag{6.10}$$

Therefore, we can see that the overall probability of SPRINT-Race of failing to return exactly the true set of Pareto optimal models is strictly controlled at level $\alpha + \beta$.

Since the number of samples required by the dual-SPRT, denoted as N, is a RV, we are interested in its expected value, denoted as $\mathbb{E}(N)$, which implicitly reflects the overall sample complexity of SPRINT-Race. Since $N \triangleq \max(N_1, N_2)$ with N_1 and N_2 indicating the number of samples required by its component SPRTs, we have

$$\begin{aligned} \mathbb{E}(N) &= \mathbb{E}(\max(N_1, N_2)) \\ &= \mathbb{E}(N_1) + \mathbb{E}(N_2) - \mathbb{E}(\min(N_1, N_2)) \\ &\leq \mathbb{E}(N_1) + \mathbb{E}(N_2) \end{aligned} \tag{6.11}$$

Given $A_1 = \ln \frac{\beta_1}{1-\alpha_1}, B_1 = \ln \frac{1-\beta_1}{\alpha_1}$ for $SPRT^1$, and $A_2 = \ln \frac{\beta_2}{1-\alpha_2}, B_2 = \ln \frac{1-\beta_2}{\alpha_2}$ for $SPRT^2$, the well-known Wald approximations for $\mathbb{E}(N_1)$ and $\mathbb{E}(N_2)$ [133] is given as

$$\mathbb{E}(N_1) = \frac{Q_1(\eta)A_1 + (1 - Q_1(\eta))B_1}{\gamma\left(\eta \ln \frac{1}{1-2\delta} + (1-\eta)\ln \frac{1}{1+2\delta}\right)} \tag{6.12}$$

where $Q_1 = \frac{A_1^{h_1(\eta)} - 1}{B_1^{h_1(\eta)} - A_1^{h_1(\eta)}}$, and $h_1(\eta)$ is the unique non-zero solution of $(1-\eta)\left(\frac{1}{1+2\delta}\right)^{h_1} + \eta\left(\frac{1}{1-2\delta}\right)^{h_1} = 1$, and

$$\mathbb{E}(N_2) = \frac{Q_2(\eta)A_2 + (1 - Q_2(\eta))B_2}{\gamma\left(\eta \ln(1+2\delta) + (1-\eta)\ln(1-2\delta)\right)} \tag{6.13}$$

where $Q_2 = \frac{B_2^{h_2(\eta)} - 1}{B_2^{h_2(\eta)} - A_2^{h_2(\eta)}}$, and $h_2(\eta)$ is the unique non-zero solution of $(1-\eta)(1-2\delta)^{h_2} + \eta(1+2\delta)^{h_2} = 1$.

Based on Equation (6.11), Equation (6.12) and Equation (6.13), we now have an upper bound on the sample complexity of each dual-SPRT. Since there will be a total of $\binom{K}{2}$ dual-SPRTs in a SPRINT-Race with K initial models, the upper bound of the overall sample complexity of SPRINT-Race is available. Note that the exact value of $\mathbb{E}(N)$ can be numerically calculated via the direct method introduced in [23, 6]. Moreover, it was shown that $\mathbb{E}(N)$ has a local minimum at $\eta = \frac{1}{2}$, and two local maximum when $\eta \in \left(\frac{1}{2} - \delta, \frac{1}{2}\right)$ and $\eta \in \left(\frac{1}{2}, \frac{1}{2} + \delta\right)$.

Intuitively, $\mathbb{E}(N)$ is directly dependent on α and β. SPRINT-Race with smaller α and β has higher probability of returning the exact true Pareto front of the initial ensemble of models, but it necessitates larger $\mathbb{E}(N)$. On the contrary, larger α and β reduce the overall sample complexity but they result in higher probability of losing any non-dominated models or falsely returning any dominated models. To sum up, the selection of α and β requires a tradeoff between the accuracy of SPRINT-Race and its computational effort.

Experiments and Applications

In this section, we illustrate the performance of SPRINT-Race on three MOMS problems: an artificially constructed MOMS problem with known ground truth, a hybrid Recommender System (RS) construction problem for Top-S recommendation, and a multi-criteria stock selection problem.

Performance Metrics

Similar to S-Race, the goal of SPRINT-Race is to correctly identify the set of Pareto optimal models from a given ensemble of models. Therefore, in order to measure its selection accuracy, three metrics are used in this research which are False Positive Rate (see Equation (6.14)), False Negative Rate (see Equation (6.15)) and False Identification Rate (see Equation (6.16)).

$$FPR \triangleq \frac{|\mathcal{P}_{PF} \setminus \mathcal{P}_R|}{|\mathcal{P}_{PF}|} \tag{6.14}$$

$$FNR \triangleq \frac{|\mathcal{P}_R \setminus \mathcal{P}_{PF}|}{|\mathcal{P}_{PF}^C|} \tag{6.15}$$

$$FIR \triangleq 1 - \frac{|\mathcal{P}_R \cap \mathcal{P}_{PF}| + |\mathcal{P}_R^C \cap \mathcal{P}_{PF}^C|}{|\mathcal{P}|} \tag{6.16}$$

where $\mathcal{P}$ denotes the entire set of initial models, $\mathcal{P}_{PF} \subseteq \mathcal{P}$ refers to the true set of Pareto front models, $\mathcal{P}_R \subseteq \mathcal{P}$ refers to the set of models returned by SPRINT-Race, and $\mathcal{P}_R^C$ and $\mathcal{P}_{PF}^C$ are their complementary sets in $\mathcal{P}$, respectively.

Apparently, FPR measures the proportion of Pareto front models that are mistakenly eliminated by SPRINT-Race, FNR measures the proportion of dominated models that are falsely identified as Pareto optimal, and FIR measures the overall proportion models that are not correctly identified. Therefore, FIR is regulated by $\alpha + \beta$, the predefined upper bound of FWER-I and FWER-II. In ideal conditions, it is expected to have $FPR = 0$, $FNR = 0$ and $FIR = 0$, indicating that no Type I or Type II error is committed. In reality, however, low FPR, FNR and FIR values are acceptable as long as certain confidence level is guaranteed.

The sample complexity N is employed as an additional performance metric to assess the computational cost of SPRINT-Race. For a dual-SPRT, the sample complexity is calculated as the number of paired observations required to reach a conclusion. Accordingly, the sample complexity of SPRINT-Race is measured as the total number of samples consumed by all the dual-SPRTs.

Artificially Constructed Multi-Objective Model Selection Problems

In this section, SPRINT-Race is applied on an artificially constructed MOMS problem with known ground truth. To better illustrate its efficiency, SPRINT-Race is compared with a multi-objective Brute Force Approach (BFA) which allocates the overall computational resources uniformly among the candidate models and typically serves as a baseline algorithm in the Model Selection (MS) literature. Moreover, the influence of several parameters (*i.e.* the number of objectives D, the size of the initial ensemble K, significance levels α and β, and indifference zone size δ) are thoroughly studied.

For a D-objective MS problem with K models, $\binom{K}{2}$ Trinomial distributions are randomly generated to mimic the pairwise dominance and non-dominance relationships among the candidate models. All the distribution information is stored in a matrix, denoted as P. For any pair of models M_i and M_j, $P_{ij} > P_{ji}$ if $M_i \succ M_j$; $P_{ij} < P_{ji}$ if $M_i \prec M_j$; and $P_{ij} = P_{ji}$ if no one dominates the other. Therefore, $\mathcal{P}_{\mathcal{PF}} \triangleq \left\{M_i \mid P_{ij} \geq \frac{1}{2},\ i,j \in \{1,2,\cdots,K\}\right\}$. Every P presents a unique MOMS problem and, whenever a new sample is needed in SPRINT-Race, the new sample is randomly generated from the corresponding Trinomial distribution provided in P.

Impact of the Number of Objectives

In this set of experiments, we studied the impact of the number of objectives, denoted by D. We set $K = 50$, $\delta = 0.01$, $\alpha = 0.01$, $\beta = 0.01$, and $D \in \{2,3,\cdots,14\}$. 30 experiments were conducted for each D value.

In Table 6.4, we report the average error metrics FPR, FNR, and FIR values, as well as the average sample complexity N and the size of the true Pareto front $|\mathcal{P}_{PF}|$ values. It can be seen from Table 6.4 that, for varying D values, all of the FPR values are close to 0. In other words,

SPRINT-Race is always able to identify almost all the true Pareto optimal models. Meanwhile, all of the FNR values tend to 0, indicating that dominated models are correctly identified and thus are removed by SPRINT-Race. Therefore, the consequent FIR is close to 0 since $\mathcal{P}_R$ is almost the same as $\mathcal{P}_{PF}$. The experimental results demonstrate that SPRINT-Race is able to control the overall probability of making any Type I or Type II errors strictly below the predefined error bound $\alpha + \beta$. The resulting FIR is, however, far less than $\alpha + \beta$, which reveals the conservativeness of SPRINT-Race, meaning that the true probability of making any false decisions in SPRINT-Race is always smaller than the prescribed significance level. This conservativeness is partly because the upper bound of the error probability provided in Equation (6.10) is not tight enough. Another reason for such conservativeness is that the power of the sequential Holm's step-down procedure is somehow reduced. In the sequential Holm's procedure, the number of accepted null hypotheses a_n and the number of rejected null hypotheses r_n at the n-th step SPRINT-Race is needed to obtain the correct stopping boundaries. a_n and r_n are, however, not always timely updated in SPRINT-Race because some of the dual-SPRTs will be terminated before a final decision is reached (*i.e.* line 12 and 15 in Algorithm 4). How to improve the power of SPRINT-Race is worth further investigation. However, it needs to be emphasized that, compared to the Bonferroni procedure employed in [148], the sequential Holm's procedure is more powerful.

It is obvious that the size of the Pareto front of the given ensemble of models increases with growing D values. When $D = 14$, all the initial models become Pareto optimal. This explains why the FNR values are all 0 when $D \geq 7$. Moreover, it explains why the sample complexity N is monotonically increasing as the dimensionality ranges from $D = 2$ to $D = 14$, as reported in Table 6.4. This is because, generally, more samples are required to conclude a pairwise non-dominance relation than inferring a pairwise dominance relation. The growth of N is, however, not dramatic because the pairwise comparisons will cease automatically once a non-dominance relationship is identified.

Table 6.4: SPRINT-Race on D-Objective Model Selection: Average $FPR\%$, $FNR\%$, $FIR\%$, N and $|\mathcal{P}_{PF}|$ Values

D	$\mathbf{FPR}\%$	$\mathbf{FNR}\%$	$\mathbf{FIR}\%$	**N**	$\lvert\mathcal{P}_{\mathbf{PF}}\rvert$
2	0.00	0.00	0.00	1.76e6	4.43
3	0.00	0.18	0.13	7.32e6	11.70
4	0.00	0.12	0.07	2.01e7	20.00
5	0.09	0.78	0.33	3.77e7	27.57
6	0.00	0.32	0.13	5.97e7	35.30
7	0.00	1.00	0.27	7.63e7	39.97
8	0.08	0.00	0.07	8.98e7	43.80
9	0.14	0.00	0.13	1.04e8	47.20
10	0.14	0.00	0.13	1.10e8	48.43
11	0.20	0.00	0.20	1.11e8	48.67
12	0.07	0.00	0.07	1.15e8	49.50
13	0.02	0.00	0.02	1.15e8	49.73
14	0.12	0.00	0.12	1.17e8	50.00

To better demonstrate the efficiency of SPRINT-Race, we compare its performance with a multi-objective BFA which is commonly used as a baseline and allocates the same amount of computational resources for each candidate model. The BFA used in this work is also a fixed-confidence MS algorithm based on a fixed-sample test with strict control over the probability of making any Type I and Type II errors [51, p. 49]. In the BFA, the required sample size and the corresponding decision region are determined by the given values of $(\eta_0, \eta_1, \alpha, \beta)$. In Table 6.5, we provide the differences of the average $FPR\%$, $FNR\%$, $FIR\%$ values between SPRINT-Race and the BFA. In addition, we report the ratio of the sample complexity between SPRINT-Race and the BFA. It can be seen from Table 6.5 that SPRINT-Race almost always achieves smaller error rates (*i.e.* FPR, FNR, FIR) than the BFA. However, the gap is negligible. Both SPRINT-Race and the BFA have successfully identified the entire Pareto front with high accuracy. Regarding the ratios of the N values in Table 6.5, the numbers demonstrate that SPRINT-Race has significant advantages over

the BFA in terms of computational efficiency. However, the differences in N get smaller gradually with increasing D.

Table 6.5: Differences of Average $FPR\%$, $FNR\%$, $FIR\%$ Values, and Ratios of the Average N Values Between SPRINT-Race and the BFA

D	**Difference**			**Ratio**
	FPR$\%$	**FNR**$\%$	**FIR**$\%$	**N**
2	-0.00	-0.00	-0.00	0.009
3	-0.28	+0.10	-0.13	0.039
4	-0.18	-0.00	-0.07	0.107
5	-0.37	+0.10	-0.02	0.198
6	-0.11	+0.20	-0.00	0.314
7	-0.15	-0.00	-0.13	0.401
8	-0.07	-0.00	-0.07	0.473
9	-0.00	-0.00	-0.00	0.549
10	-0.21	-0.00	-0.20	0.578
11	-0.22	-0.00	-0.20	0.582
12	-0.13	-0.00	-0.13	0.603
13	-0.07	-0.00	-0.07	0.606
14	-0.24	-0.00	-0.24	0.683

To visualize how N changes during the racing process, we depicted the changes of N values at each step of SPRINT-Race. The step-wise N values of 4 runs of SPRINT-Race with $K = 50$, $D = 2$, $\alpha = 0.05$, $\beta = 0.05$, and $\delta = 0.01$ are drawn in Figure 6.2. Since SPRINT-Race discards under-performing models once sufficient statistical evidence has been collected, the N values drop gradually during racing. As shown from Figure 6.2, a sudden drop of the N value commonly happens around the 20%-th step, which illustrates that SPRINT-Race cuts down the computational cost significantly in an early stage of racing by quickly eliminating the dominated models. However, it takes more steps to infer pairwise non-dominance relationships.

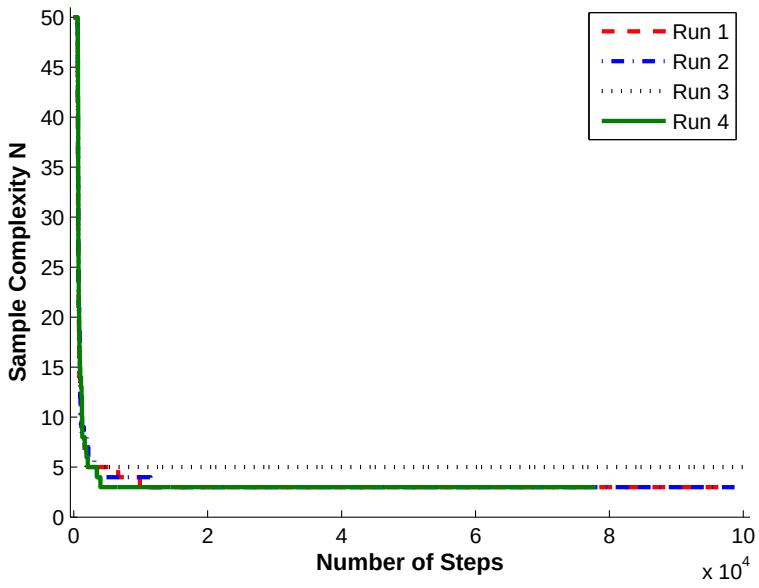


Figure 6.2: Changes of N values at each step of SPRINT-Race for $D = 2$

Impact of the Initial Ensemble Size

In this set of experiments, we study the impact of the number of initial models on the performance of SPRINT-Race. We allowed D to vary over $\{2, 3, 4, 5, 6\}$ and K ranged from 10 to 200 with a step size of 10. Regarding α, β, and δ, we held them fixed at 0.01. We report the average FPR, FNR, FIR, and $|\mathcal{P}_{PF}|$ values, as well as the average N values in Table 6.6. They are averaged over 30 trials. Additionally, the ratios of the sample complexity N between SPRINT-Race and the BFA are provided in Table 6.6.

First of all, it can been seen from Table 6.6 that all the FPR, FNR and FIR values are extremely close to 0 while the N ratios are no larger than 16% in all the cases, implying that SPRINT-Race returns almost exactly the true Pareto front but at a significantly reduced computational cost when compared to the BFA. The initial ensemble size K has little impact on the selection accuracy

of SPRINT-Race. However, the N values increase monotonically with growing K values. It is obvious that larger K results in larger N to establish pairwise dominance and non-dominance relationships. Moreover, the computational advantage of SPRINT-Race over the BFA becomes more pronounced with increasing K, as demonstrated by the experimental results. Similar conclusions could be drawn for $D \in \{2, 4, 5, 6\}$.

Table 6.6: SPRINT-Race on 3-Objective Model Selection: Average $FPR\%$, $FNR\%$, $FIR\%$, N with ratios, and $|\mathcal{P}_{PF}|$ Values for Varying K

K	**FPR**$\%$	**FNR**$\%$	**FIR**$\%$	**N**	**Ratio of N**	$\|\mathcal{P}_{\mathbf{PF}}\|$
10	0.00	0.00	0.00	6.42e5	0.152	4.83
20	0.00	0.00	0.00	1.63e6	0.071	6.37
30	0.00	0.37	0.33	3.73e6	0.063	9.03
40	0.00	0.32	0.25	6.90e6	0.060	11.73
50	0.00	0.26	0.20	7.36e6	0.039	11.27
60	0.00	0.23	0.17	1.01e7	0.035	13.10
70	0.67	0.00	0.25	1.27e7	0.031	14.80
80	0.00	0.15	0.13	1.14e7	0.021	12.93
90	0.00	0.00	0.00	1.50e7	0.021	15.03
100	0.35	0.00	0.30	1.78e7	0.020	15.57
110	0.00	0.00	0.00	1.80e7	0.016	15.80
120	0.00	0.09	0.08	1.85e7	0.014	14.70
130	0.00	0.00	0.00	2.16e7	0.013	16.37
140	0.29	0.27	0.29	1.78e7	0.013	13.27
150	0.07	0.00	0.07	2.82e7	0.013	17.80
160	0.00	0.00	0.00	3.42e7	0.012	19.97
170	0.06	0.00	0.06	3.21e7	0.011	18.43
180	0.36	0.24	0.25	2.92e7	0.009	15.80
190	0.00	0.12	0.11	3.35e7	0.009	16.83
200	0.00	0.20	0.15	3.63e7	0.009	17.90

The changes of the ratios between the average sample complexity of SPRINT-Race and the BFA with increasing K values are depicted in Figure 6.3 with $D \in \{2, 3, 4, 5, 6\}$. As Figure 6.3 shows, SPRINT-Race always requires a smaller sample size than the BFA. The computational advantage of SPRINT-Race over the BFA becomes more pronounced with increasing K. When $K = 200$,

SPRINT-Race saves more than 90% of the samples required by the BFA to achieve the same confidence level for MS. Another observation from Figure 6.3 is that the computational savings of SPRINT-Race with respect to the BFA get less significant with rising D values. When $D \in \{2, 3\}$, which is a common scenario in real-world problems, however, SPRINT-Race saves more than 80% of the samples taken by the BFA for varying K values. Even when $D = 6$, SPRINT-Race still saves more than 60% of the samples needed by the BFA, which is significant.

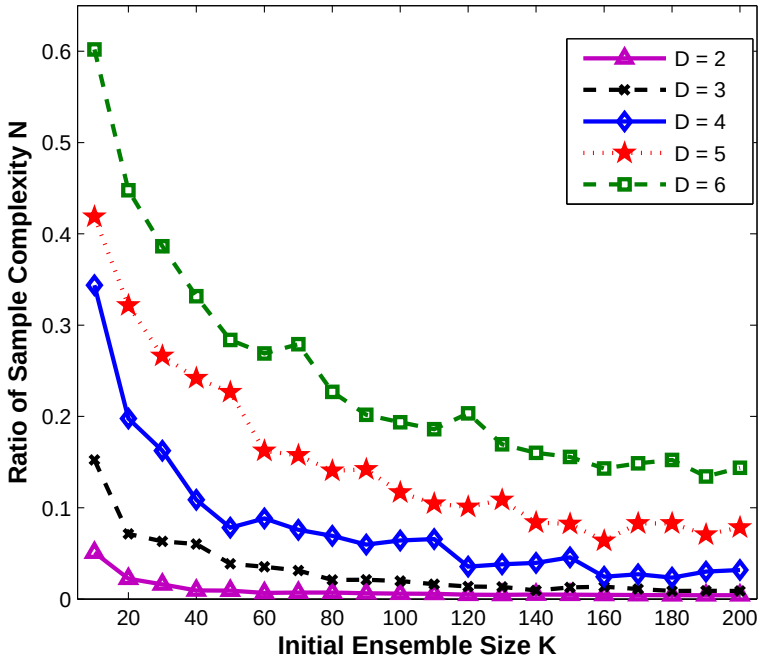


Figure 6.3: Changes of the ratios between the average N values of SPRINT-Race and the BFA with increasing K values for $D \in \{2, 3, 4, 5, 6\}$.

Impact of α and β Values

In order to study the impact of the upper bounds of the probability of making any Type I and Type II errors, we conducted a series of experiments with varying α and β values. In the first set of

experiments, we had $\alpha = 0.1$ and $\beta \in \{0.01, 0.03, \cdots, 0.15\}$, and in the second set of experiments, we had $\beta = 0.1$ and $\alpha \in \{0.01, 0.03, \cdots, 0.15\}$. The other parameters were held fixed at $K = 50$, $\delta = 0.01$ and $D \in \{2, 3\}$. The average FPR, FNR and FIR values over 30 trials with varying α are reported in Table 6.7. It can be seen from Table 6.7 that there is no significant correlation between α and the resulting error rates. The observed FPR, FNR and FIR values are always strictly below the prescribed significance level and vary within a reasonable range. This is another manifestation of SPRINT-Race's conservativeness. Similar observations are available for the set of experiments with fixed α and varying β values.

Table 6.7: SPRINT-Race on 2, 3-Objective Model Selection: Average $FPR\%$, $FNR\%$, $FIR\%$ Values for Varying α

α		0.01	0.03	0.05	0.07	0.09	0.11	0.13	0.15
$\mathbf{D = 2}$	$\mathbf{FPR}\%$	0.00	0.00	0.00	0.00	0.87	0.00	1.16	0.00
	$\mathbf{FNR}\%$	1.39	0.95	1.31	1.39	0.00	1.16	0.00	1.31
	$\mathbf{FIR}\%$	1.27	0.87	1.20	1.27	0.80	0.10	1.07	1.20
$\mathbf{D = 3}$	$\mathbf{FPR}\%$	0.00	0.00	0.00	2.19	0.00	0.00	1.95	0.00
	$\mathbf{FNR}\%$	1.81	1.79	2.32	0.00	1.42	1.33	0.00	2.96
	$\mathbf{FIR}\%$	1.47	1.40	1.87	1.67	1.13	1.07	1.47	2.27

The impact of the selection of α and β can be hardly identified via the error rates in Table 6.7. Their impact on the sample complexity of SPRINT-Race is, however, more evident. In Figure 6.4, we depicted the changes of the average N values with increasing α and β values. As you can see from Figure 6.4, the N values decrease with rising α or β values. This is reasonable because, when we raise the upper bounds of the probability of making any Type I and Type II errors, fewer samples are required to establish any pairwise dominance and non-dominance relationships via the dual-SPRT as a consequence.

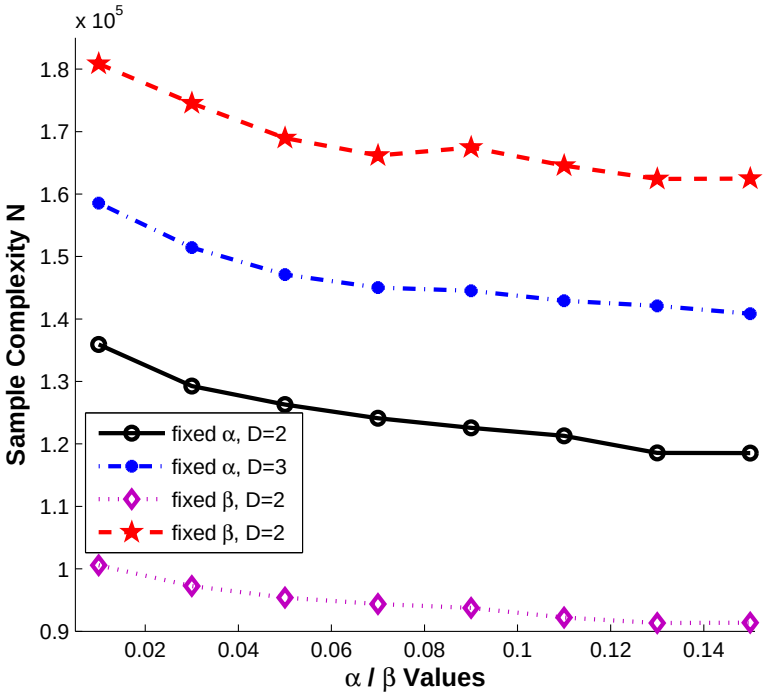


Figure 6.4: Changes of the average sample complexity N of SPRINT-Race with increasing α / β values for $D \in \{2, 3\}$.

Impact of δ Values

In this set of experiments, we studied the effect of δ on the performance of SPRINT-Race. The experimental setting was as follows: $K = 50$, $\alpha = 0.05$, $\beta = 0.05$, $D \in \{2, 3\}$ and δ is selected from the discrete set $\{0.02, 0.03, \cdots, 0.15\}$.

The changes of the average FIR and FNR values over 30 runs with varying δ values are depicted in Figure 6.5. The average FPR values are not reported because all of them are nearly 0. The uptrend of the error rates with increasing δ values can be observed from Figure 6.5. When the size of the indifference zone grows with increasing δ, it is highly likely that more dominated models that are located in the indifference zone will be returned as Pareto front models. Consequently,

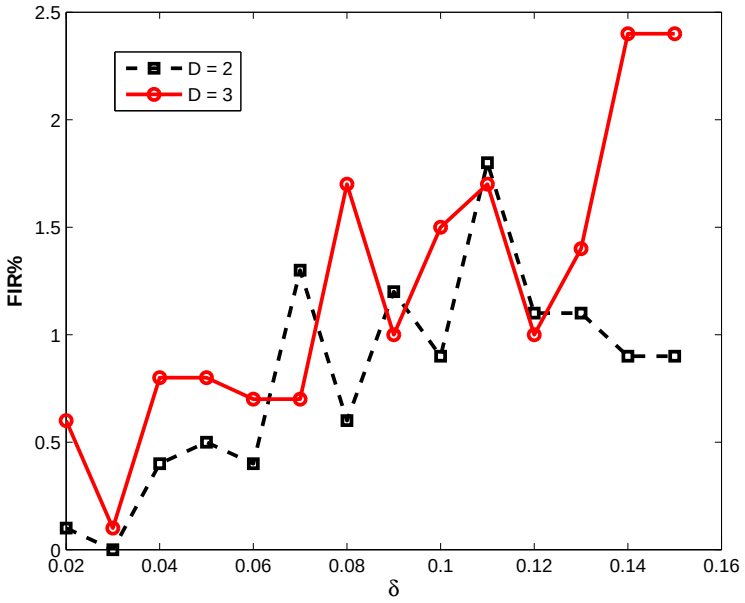
increasing δ results in higher FNR values since $|\mathcal{P}_R \setminus \mathcal{P}_{PF}|$ gets larger. As a result, FIR values increase as well. However, the observed probability of making any false eliminations of non-dominated models or any false acceptances of dominated models is always strictly controlled at the predefined significance level.

Moreover, the changes of the N values, as well as the ratios between the N values of SPRINT-Race and the BFA with varying δ values, are shown in Figure 6.6. It can been seen that N decreases greatly with increasing δ since it is easier for a dual-SPRT to reach a decision with a broader indifference zone. However, the computational savings of SPRINT-Race, compared to the BFA, are reduced since the sample complexity of the BFA decreases faster with growing δ. Even so, SPRINT-Race saves at least 85% of the samples when compared to the BFA.

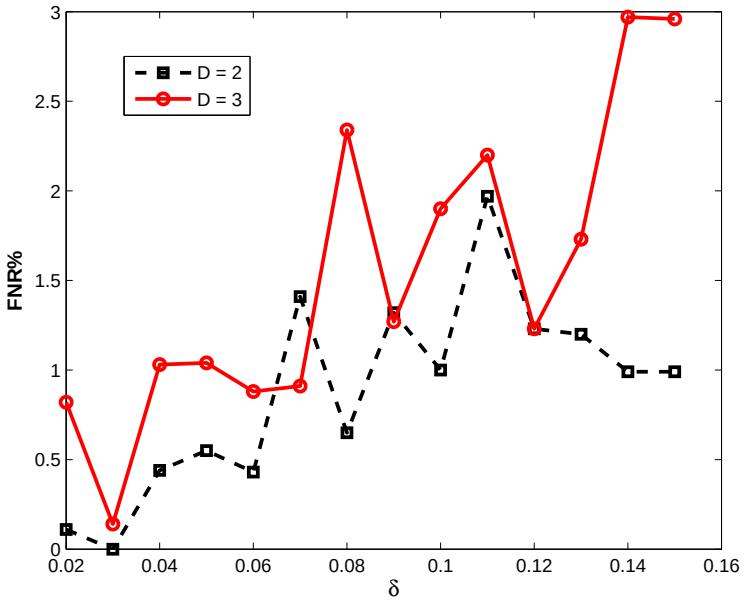
Hybrid Recommender System Construction for Top-S Recommendation

The motivation of the weighted hybridization of RSs has been emphasized in Section 16 of Chapter 5. In this section, we illustrate the performance of SPRINT-Race on Pareto optimal hybrid RSs selection for Top-S recommendation.

In Top-S recommendation, the task is to identify a list of S items as recommendations to a particular user. Top-S recommendation used to be solved as a single-objective problem in which only the prediction accuracy is considered. The other important aspects, such as novelty and diversity, are largely ignored until recently. The advantages of optimizing prediction accuracy, novelty and diversity simultaneously have been emphasized in a large amount of research [107, 49, 130]. Therefore, in this set of experiments, we applied SPRINT-Race on selecting the set of Pareto optimal weighted hybrid RSs. This is a ternary-objective maximization problem which optimizes prediction accuracy, novelty and diversity simultaneously.

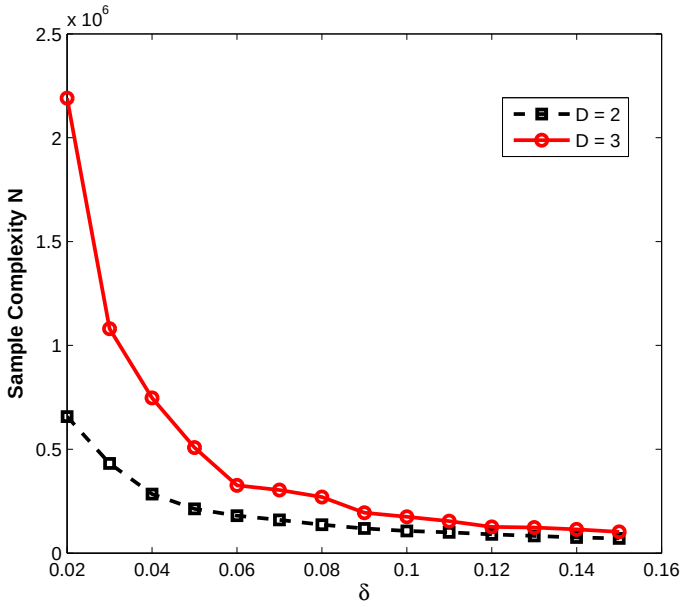


(a) Change of FIR with varying δ

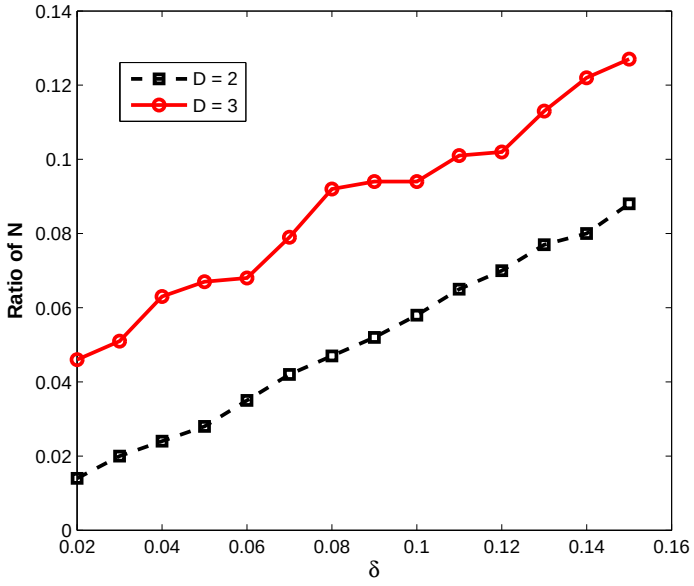


(b) Change of FNR with varying δ

Figure 6.5: Change of average FIR and FNR values with varying δ.



(a) Change of sample complexity N with varying δ

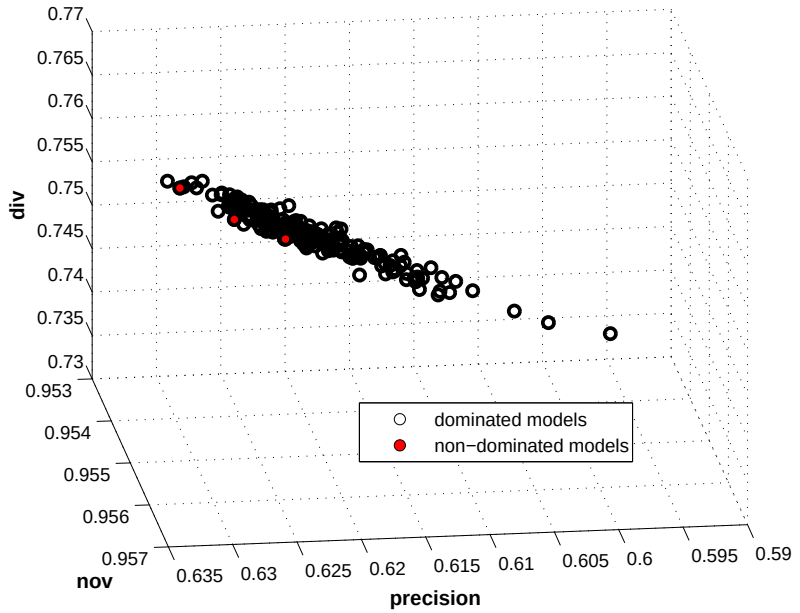


(b) Change of the ratios between sample complexity N of SPRINT-Race and the BFA with varying δ

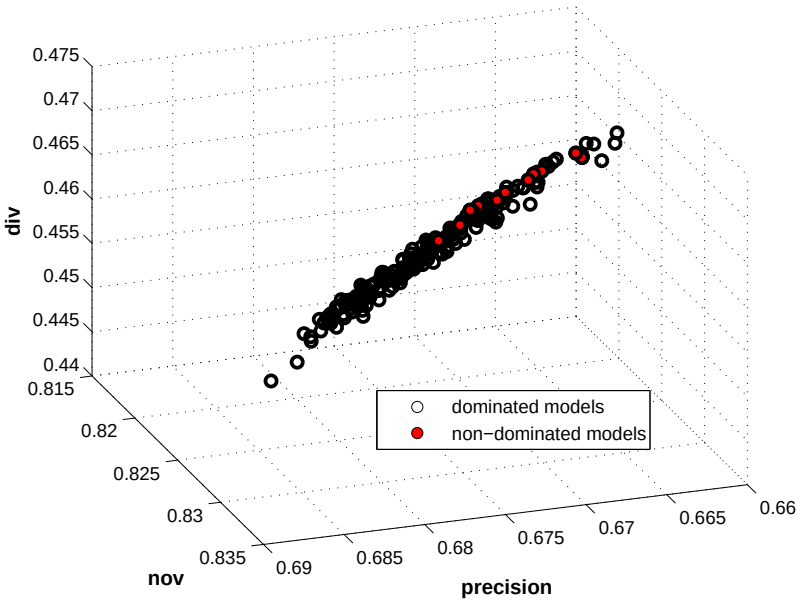
Figure 6.6: Change of average sample complexity N and the ratio between the N values of SPRINT-Race and the BFA with varying δ.

The candidate models were weighted hybridization of 8 RSs selected from the `PREA` package [81], including User-based Collaborative Filtering (CF), Item-based CF, Slope One, Regularized SVD, Non-negative Matrix Factorization, Probabilistic Matrix Factorization, Rank-based Recommender with asymmetric loss, and Singleton Global Local Low-Rank Matrix Approximation. The initial weights were generated via Latin Square sampling. Three performance metrics were employed for Top-S recommendation [107]: $precision(R(S))$ for prediction accuracy, $nov(R(S))$ for novelty and $div(R(S))$ for diversity, where $R(S)$ refers to the list of S items recommended. The MovieLens and Netflix dataset introduced in Section 16 of Chapter 5 were used and were randomly split into a training set and a test set 75% and 25% of the available data respectively. At each step of SPRINT-Race, a batch of 50 training samples was randomly chosen with replacement for model training and validation. The other parameters were set as $K = 200$, $\alpha = 0.05$, $\beta = 0.05$, $\delta = 0.05$, and each experiment was repeated for 45 trials.

The average N and $|\mathcal{P}_R|$ values for the MovieLens dataset are 7.92×10^7 and 2.73, while , for the Netflix dataset, the same values are 1.07×10^8 and 11.4. Since the true Pareto front is unknown, we did not compute FIR, FPR or FNR. Instead, we compared the performances of the resulting dominated hybrid RSs and non-dominated hybrid RSs on the test set. For each dataset, we depicted the results of a single trial of SPRINT-Race. In Figure 6.7, the performance vectors of the dominated models on the entire test set are denoted by black circles (○), while those of the non-dominated models are indicated by red-filled circles (●). As depicted in Figure 6.7, the non-dominated hybrid RSs mostly gather around (0.635, 0.957, 0.77) in MovieLens dataset and (0.69, 0.835, 0.475) in Netflix dataset, where all three objectives are maximized simultaneously. The experimental results demonstrate that the models that are Pareto optimal on the training set are also Pareto optimal on the test set.



(a) MovieLens dataset



(b) Netflix dataset

Figure 6.7: Comparison of the performances of dominated and non-dominated hybrid RSs on the test set.

Multi-Criteria Stock Selection

Playing the stock market is thrilling. Many stock-picking strategies have been proposed for finding good stocks according to a selected set of criteria. Based on the historically popular Mean-Variance paradigm [89], the objectives of stock selection is to maximize the average return and to minimize the risk. The stock with the best risk-return trade-off is more preferable. Therefore, stock selection is, typically, a binary-objective problem.

Let $x_i(t)$ denote the price of the i^{th} stock at time t. Then, its *log return* $r_i(t)$ is defined as

$$r_i(t) \triangleq \log \frac{x_i(t)}{x_i(t-1)} = \tilde{x}_i(t) - \tilde{x}_i(t-1) \tag{6.17}$$

where $\tilde{x}_i(t) \triangleq \log x_i(t)$ is referred to as the *log price*. Accordingly, the weekly return $\bar{r}_i$ and risk σ_i are defined as

$$\bar{r}_i \triangleq \frac{1}{7}\sum_{t=d}^{d+6} r_i(t), \quad \sigma_i \triangleq \sqrt{\frac{\sum_{t=d}^{d+6}\left(r_i(t)-\bar{r}_i\right)^2}{6}} \tag{6.18}$$

where d indexes starting days, and $\bar{r}_i$ and σ_i are calculated based on the daily stock prices over a week.

According to random walk theory, $r_t(t), t = 1, 2, \cdots$ could be regarded as i.i.d. samples from a heavy-tailed distribution [87, 53, 111]. Therefore, due to the stochasticity of the two optimization objectives (return and risk), SPRINT-Race is a proper tool for Pareto optimal stocks selection. In this binary-objective MS problem, each model is a stock and its performance vector consists of its weekly return and risk defined in Equation (6.18).

We selected 100 stocks and collected 4826 daily open price data for each stock from [141] between 08-29-1996 to 10-29-2015. Based on the collected data, we estimated the underlying Trinomial distribution of each pair of stocks via Monte Carlo estimation. The estimated distributions, representing pairwise dominance and non-dominance relationships, are used to identify the true Pareto front $\mathcal{P}_{PF}$. Given $\alpha = 0.05$, $\beta = 0.05$ and $\delta = 0.05$, the average FPR, FNR, FIR, N and $|\mathcal{P}_R|$ values of SPRINT-Race over 30 trials are 0, 6.4%, 6.3%, 1.82×10^6 and 1.63, respectively. Furthermore, the average runtime of SPRINT-Race is 5.91 minutes per run, using `MATLAB2013b` on two quad-core Intel Xeon E5-2609 with 32 GB of RAM.

When compared with the BFA on stock selection, SPRINT-Race achieves similar selection accuracy with respect to the FCR, FNR and FIR values, but with a significant reduced sample complexity. SPRINT-Race saves more than 90% of the samples needed by the BFA. The experimental results confirm that SPRINT-Race is capable of speeding up the binary-objective stock selection process and correctly identifying the set of Pareto optimal stocks with the best risk-return trade-offs.

To provide a better demonstration of SPRINT-Race's accuracy in Pareto optimal stock selection, we graph the minimum probability of dominance of each model, denoted as η_{min}, in Figure 6.8 across models. For the i^{th} model M_i, its minimum probability of dominance is computed as $\eta_{min} \triangleq \min_{i \neq j} \frac{p_{ij}}{p_{ij}+p_{ji}}$. Given $\delta = 0.05$, any model with $\eta_{min} \leq 0.45$ is expected to be eliminated as dominated model, and any model with $\eta_{min} \geq 0.50$ should be returned as a non-dominated model. Besides, any model with $\eta_{min} \in (0.45, 0.5)$ resides in the indifference zone of at least one dual-SPRT, and it can be identified as either dominated or non-dominated model with no error being committed. From Figure 6.8, it can be observed that SPRINT-Race returns all the Pareto front models accurately without falsely including any dominated models.

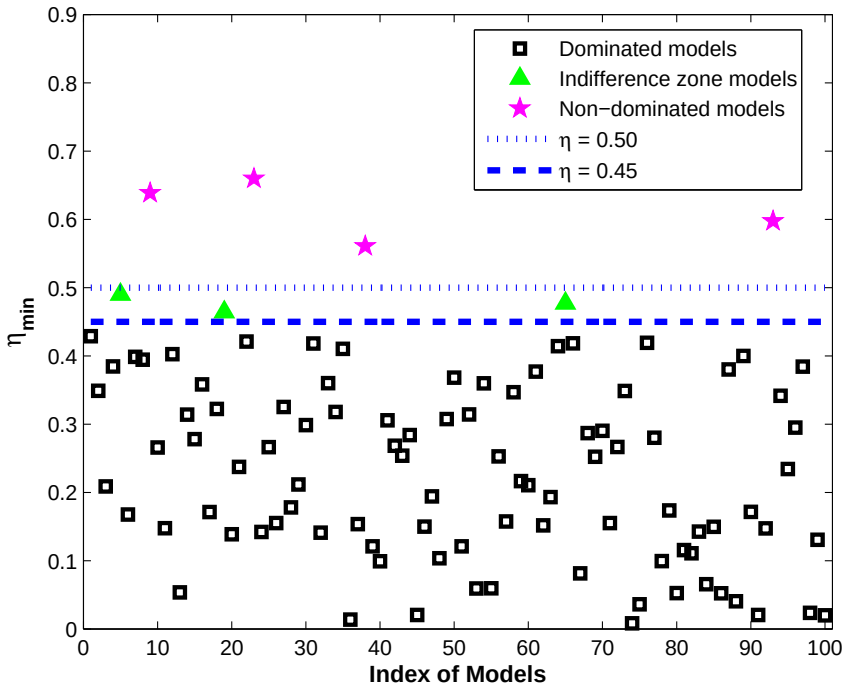


Figure 6.8: Models' minimum probability of dominance.

CHAPTER 7: CONCLUSION

This research largely addresses the issues and practical applications of Racing Algorithms (RAs) for Extreme Model Selection (EMS) and Multi-Objective Model Selection (MOMS). This research is shown to be important since EMS and MOMS are important problems in machine learning, and they are also natural frameworks of many real-world applications. However, little work has been done in the literature of RAs for EMS and MOMS.

In EMS, the goal is to maximize the final objective value of the given problem instance by automatically allocating the overall computational resources among an ensemble of problem solvers. Ideally, we would allocate all computational resources to the model that will give us the overall best solution.

In this work, we propose the first RA for EMS, named Max-Race. Distinct from existing RAs, Max-Race performs Model Selection (MS) during model comparison in terms of extreme performance. Max-Race is an online per-instance based RA, aiming at maximizing the final outcome while solving a particular optimization problem. It achieves the goal by eliminating underperforming models as early as possible, and thus distributing the computational resources optimally among the competing models. In Max-Race, the underlying distribution of a model's performance is approximated via the Point Over Threshold (POT) approach in Extreme Value Theory (EVT) and, thus, its extreme performance is referred to as the right endpoint of the distribution. Additionally, a parametric hypothesis test under the Generalized Pareto Distribution (GPD) assumption is developed to infer significant difference between the extreme performances of a pair of models. The model with inferior extreme performance will be eliminated as early as possible in order to concentrate more computational resources on the optimal model(s). In this work, Max-Race is applied to constructing a population-based Algorithm Portfolio (AP) in which each competing model

is an Evolutionary Computation (EC) algorithm. We assess the performance of Max-Race by comparing it to the Brute Force Approach (BFA), two baseline algorithms (*i.e.* BestEC, RandEC), and three popular population-based APs (*i.e.* A Multi-algorithm Genetically Adaptive Method for Single Objective Optimization (AMALGAM-SO), Population-based Algorithm Portfolio (PAP), Multiple Evolutionary Algorithm (MultiEA)). The experimental results demonstrate that Max-Race is able to retain the optimal model with high precision and low computational overhead. By accurately identifying the optimal models with the best extreme performances and eliminating the under-performing ones as soon as possible, Max-Race is able to maximize the quality of the final solution obtained.

Another type of MS considered in this research is MOMS in which multiple conflicting optimization objectives are considered simultaneously during MS. In other words, more than one optimization criterion is used to measure the goodness of models. Consequently, in MOMS, the entire set of Pareto optimal models is expected to be returned without including any dominated models.

First of all, we put forward the first Multi-Objective Racing Algorithm (MORA) in a fixed-budget setting, called S-Race, which addresses the problem of MOMS in the proper sense of Pareto optimality. Given stochastic objectives, S-Race adopts the non-parametric pairwise sign test to establish pairwise dominance relationships. Moreover, the discrete Holm's Step-Down procedure for Family-Wise Error Rate (FWER) control is employed to control the overall probability of making any Type I errors. That is, the probability of falsely eliminating any non-dominated model in S-Race is strictly controlled at a user-specified level. The performance of S-Race is compared to the corresponding BFA on three MOMS problems which are hyper-parameter tuning for Support Vector Machines (SVMs) on binary- and ternary-classification tasks, optimal Artificial Bee Colony (ABC) selection for numerical optimization, and hybrid Recommender Systems (RSs) construction for movie recommendations. As the experimental results demonstrate, S-Race presents significant computational advantages over the BFA and high accuracy in identifying almost the

same ensemble of Pareto optimal models. First of all, we put forward the first MORA in a fixed-budget setting, called S-Race, which addresses the problem of MOMS in the proper sense of Pareto optimality. Given stochastic objectives, S-Race adopts the non-parametric pairwise sign test to establish pairwise dominance relationships. Moreover, the discrete Holm's Step-Down procedure for FWER control is employed to control the overall probability of making any Type I errors. That is, the probability of falsely eliminating any non-dominated model in S-Race is strictly controlled at a user-specified level. The performance of S-Race is compared to the corresponding BFA on three MOMS problems which are hyper-parameter tuning for SVMs on binary- and ternary-classification tasks, optimal ABC selection for numerical optimization, and hybrid RSs construction for movie recommendations. As the experimental results demonstrate, S-Race presents significant computational advantages over the BFA and high accuracy in identifying almost the same ensemble of Pareto optimal models.

Next, another novel MORA based on the Sequential Probability Ratio Test (SPRT) with Indifference zone, namely SPRINT-Race, is proposed with a fixed confidence setting. In SPRINT-Race, the dominance and non-dominance relationships between any pair of models are established via a non-parametric ternary-decision test with Trinomial distribution assumption, called dual-SPRT. SPRINT-Race offers strong control over the overall probability of making any Type I or Type II errors via the sequential Holm's step-down FWER control method. In other words, SPRINT-Race is able to strictly control the probability of mistakenly eliminating any non-dominated models or falsely retaining any dominated models at a prescribed significance level. The efficiency of SPRINT-Race is first studied on a set of artificially constructed MOMSs with known ground truth. Furthermore, we assess the performance of SPRINT-Race on two real-world problems: hybrid RSs construction for Top-S recommendation and multi-criteria stock selection. The experimental results confirm that SPRINT-Race is able to identify the entire Pareto front with high likelihood and significant computational savings when compared to a multi-objective BFA. Moreover, the Pareto

optimal models returned by SPRINT-Race exhibit good generalized performances on unseen problem instances. SPRINT-Race is a robust and efficient tool for MOMS.

The main limitation of the proposed RAs in this work is the assumption that the samples or observations are identically and independently distributed (i.i.d.) outcomes of stochastic optimizers, which may be invalid for some real-world applications. Hence, it could be beneficial to generalize our approaches to MS problems where no i.i.d. stochastic assumption is made (*i.e.* in Markovian settings [52, 127], or in adversarial settings [11, 26]). Moreover, another direction for future work is incorporating efficient probabilistic modelling techniques (*e.g.* Restricted Boltzmann Machines, Bayesian networks) into iterative RAs, as inspired by Sequential Model-based Optimization (SMBO) [63], Estimations of Distribution Algorithms [115] and I/F-Race [12]. Therefore, the search space of RAs will not be confined to the initial ensemble of models and will be significantly enlarged by in-race sampling of newly promising models from the continuously updated probabilistic model.

LIST OF REFERENCES

[1] N. Abe, B. Zadrozny, and J. Langford. Outlier detection by active learning. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 504–509, 2006.

[2] B. Adenso-Díaz and M. Laguna. Fine-tuning of algorithms using fractional experimental designs and local search. *Operations Research*, 54(1):99–114, 2006.

[3] A. Aderhold, K. Divold, A. Sheidler, and M. Middendorf. *NICSO 2010*, volume 284, chapter Artificial Bee Colony Optimization: A New Selection Scheme and Its Performance, pages 283–294. Springer Berlin Heidelberg, 2010.

[4] S. Agrawal and N. Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. *Journal of Machine Learning Research(JMLR): Workshop and Conference Proceedings*, 23:1–26, 2012.

[5] S. Agrawal and N. Goyal. Further optimal regret bounds for Thompson sampling. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS 2013)*, 2013.

[6] L. A. Aroian. Sequential analysis, direct method. *Technometrics*, 10:125–132, 1968.

[7] J. Audibert and S. Bubeck. Best arm identification in multi-armed bandits. In *Proceedings of the 23rd Conference on Learning Theory (COLT '10)*, 2010.

[8] J. Audibert, R. Munos, and C. Szepesvári. Tuning bandit algorithms in stochastic environments. In *Proceedings of the 18th International Conference on Algorithmic Learning Theory*, pages 150–165, 2007.

[9] J. Audibert, R. Munos, and C. Szepesvári. Exploration-exploitation trade-off using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410:1876–1902, 2009.

[10] P. Auer, N. Cesa-Bian, and P. Fischer. Finite-time analysis of the multi-armed bandit problem. *Machine Learning*, 47(2):235–256, 2002.

[11] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 322 – 331, 1995.

[12] P. Balaprakash, M. Birattari, and T. Stützle. *Hybrid Metaheuristics*, chapter Improvement Strategies for the F-Race Algorithm: Sampling Design and Iterative Refinement, pages 108–122. Springer Berlin Heidelberg, 2007.

[13] A. Balkema and L. de Haan. Residual life time at great age. *Annals of Probability*, 2:792–804, 1974.

[14] X. Bao, L. Bergman, and R. Thompson. Stacking recommendation engines with additional meta-features. In *Proceedings of the 3rd ACM Recommender System Conference (RecSys'09)*, pages 109 – 116, 2009.

[15] J. Bartroff and J. Song. Sequential tests of multiple hypotheses controlling type i and ii familywise error rates. *Journal of Statistical Planning and Inference*, 153:100–114, 2014.

[16] T. Bartz-Beielstein, C. Lasarczyk, and M. Preuss. Sequential parameter optimization. In *Proceedings of 2005 Congress on Evolutionary Computation (CEC'05)*, 2005.

[17] S. Becker, J. Gottlieb, and T. Stützle. *Artificial Evolution*, chapter Applications of Racing Algorithms: An Industrial Perspective, pages 271–283. Springer Berlin Heidelberg, 2006.

[18] R. M. Bell, Y. Koren, and C. Volinsky. The bellkor solution to the netflix prize. Technical report, AT&T Labs-Research, Middletown, NJ, USA, 2007.

[19] Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29:919–1188, 2001.

[20] M. Birattari. *The Problem of Tuning Metaheuristics as Seen From A Machine Learning Perspective*. PhD thesis, Université Librè de Bruxelles, Brussels, Belgium, 2004.

[21] M. Birattari, T. Stützle, L. Paquete, and K. Varrentrapp. A racing algorithm for configuring metaheuristics. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation (GECCO'02)*, pages 11–18, 2002.

[22] M. Birattari, Z. Yuan, P. Balaprakash, and T. Stützle. *Experimental Methods for the Analysis of Optimization Algorithms*, chapter F-Race and Iterated F-Race: An Overview, pages 311 – 336. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[23] J. De Boer. Sequential test with three possible decisions for testing an unknown probability. *Applied Science Research, Sec. B*, 3:249–259, 1954.

[24] George. E. P. Box, William. G. Hunter, and J. Stuart Hunter. *Statistics for Experimenters*. Wiley, 1978.

[25] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 10:646–657, 2006.

[26] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5:1–122, 2012.

[27] S. Bubeck, T. Wang, and N. Viswanathan. Multiple identification in multi-armed bandits. In *Proceedings of the 30th International Conference on Machine Learning (ICML'13)*, 2013.

[28] T. Carchrae and J. C. Beck. Applying machine learning to low-knowledge control of optimization algorithms. *Computational Intelligence*, 21(4):372–387, 2005.

[29] A. Carpentier and M. Valko. Extreme bandits. In *Advances in Neural Information Processing Systems 27*, pages 1089–1097, 2014.

[30] O. Chapelle and L. Li. An empirical evaluation of Thompson sampling. In *24th Advances in Neural Information Processing Systems (NIPS'11)*, 2011.

[31] M. Chiarandini, M. Birattari, K. Socha, and O. Rossi-Doria. An effective hybrid algorithm for university course timetabling. *Journal of Scheduling*, 9:403–432, 2006.

[32] V. A. Cicirello and S. F. Smith. *Principles and Practice of Constraint Programming - CP 2004*, volume 3258, chapter Heuristic Selection for Stochastic Search Optimization: Modeling Solution Quality by Extreme Value Theory, pages 197–211. Springer-Verlag Berlin Heidelberg, 2004.

[33] V A. Cicirello and S. F. Smith. The max K-Armed bandit: A new model of exploration applied to search heuristic selection. In *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 1355–1361, 2005.

[34] S. Coles. *An Introduction to Statistical Modeling of Extreme Values*. Springer, 2001.

[35] W. J. Conover. *Practical Nonparametric Statistics*. John Wiley and Sons, third edition edition, 1999.

[36] C. Cortes and V. N. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[37] S. P. Coy, B. L. Golden, G. C. Runger, and E. A. Wasil. Using experimental design to find effective parameter settings for heuristics. *Journal of Heuristics*, 7:77–97, 2001.

[38] S. K. De and M. Baron. Sequential bonferroni methods for multiple hypothesis testing with strong control of family-wise error rates i and ii. *Sequential Analysis*, 31:238–262, 2012.

[39] S. K. De and M. Baron. Step-up and step-down methods for testing multiple hypotheses in sequential experiments. *Journal of Statistical Planning and Inference*, 142:2059–2070, 2012.

[40] P. de Zea Bermudez and S. Kotz. Parameter estimation of the generalized pareto distribution - part i. *Journal of Statistical Planning and Inference*, 140:1353–1373, 2010.

[41] M. M. Drugan and A. Nowe. Designing multi-objective multi-armed bandits algorithms: a study. In *Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN'13)*, pages 1 – 8, 2013.

[42] M. F. Duarte and Y. H. Hu. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):826–838, 2004.

[43] A. E. Eiben and S. K. Smit. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation*, 1:19–31, 2011.

[44] M. El-Abd. Generalized opposition-based artificial bee colony algorithm. In *Proceedings of the 14th IEEE World Congress on Computational Intelligence (WCCI'12)*, pages 109–116, 2012.

[45] E. Even-Dar, S. Mannor, and Y. Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

[46] D. A. S. Fraser. Sufficient statistics with nuisance parameters. *The Annals of Mathematical Statistics*, 27:838–842, 1956.

[47] V. Gabillon, M. Ghavamzadeh, and A. Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *Proceedings of the 25th Neural Information Processing Systems (NIPS'12)*, 2012.

[48] A. Garivier and O. Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 359–376, 2011.

[49] M. Ge, C. Delgado-Battenfeld, and D. Jannach. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In *Proceedings of the 4th ACM Recommender System Conference (RecSys'10)*, pages 257–260, 2010.

[50] M. Gebser, R. Kaminski, B. Kaufmann, T. Schaub, M. Schneider, and S. Ziller. A portfolio solver for answer set programming: Preliminary report. In *Proceedings of the Eleventh International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'11)*, volume 6645, pages 352–357, 2011.

[51] B. K. Ghosh. *Sequential Tests of Statistical Hypotheses*. Addison-Wesley Publishing Company, Inc., 1970.

[52] J. Gittins, K. Glazebrook, and R. Weber. *Multi-Armed Bandit Allocation Indices*. John Wiley and Sons, 2011.

[53] P. Glasserman, P. Heidelberger, and P. Shahabuddin. Portfolio value-at-risk with heavy-tailed risk factors. *Mathematical Finance*, 12:239–269, 2002.

[54] T. A. F. Gomes, R. B. C. Prudêncio, C. Soares, A. L. D. Rossi, and A. Carvalho. Combining meta-learning and search techniques to select parameters for support vector machines. *Neurocomputing*, 75(1):3 – 13, 2012.

[55] A. Gopalan, S. Mannor, and Y. Mansour. Thompson sampling for complex online problems. *Journal of Machine Learning Research (JMLR): Workshop and Conference Proceedings*, 32:100–108, 2014.

[56] I. Guyon. A practical guide to model selection. In *Proceedings of the Machine Learning Summer School Springer Text in Statistics*, 2009.

[57] E. Haasdijk, A Atta ul Qayyum, and A. E. Eiben. Racing to improve on-line, on-board evolutionary robotics. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO'11)*, pages 187–194, 2011.

[58] P. Hall and A. H. Welsh. Best attainable rates of convergence for estimates of parameters of regular variation. *The Annals of Statistics*, 12(3):1079–1084, 1984.

[59] N. Hansen, S.D. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11:77–112, 2003.

[60] V. Heidrich-Meisner and C. Igel. Hoeffding and Bernstein races for selecting policies in evolutionary direct policy search. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML'09)*, pages 401–408, 2009.

[61] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Associations*, 58:13–30, 1963.

[62] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 2:65–70, 1979.

[63] H. H. Hoos. *Autonomous Search*, chapter Automated Algorithm Configuration and Parameter Tuning, pages 37–71. Springer Berlin Heidelberg, 2012.

[64] D. Huang, T. T. Allen, W. I. Notz, and N. Zeng. Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of Global Optimization*, 34:441–446, 2006.

[65] E. J. Hughes. Evolutionary multi-objective ranking with uncertainty and noise. *Evolutionary Multi-criterion Optimization*, 1999:329–343, 2001.

[66] J. Hüsler, P. Cruz, A. Hall, and C. M. Fonseca. On optimization and extreme value theory. *Methodology And Computing In Applied Probability*, 5:183–195, 2003.

[67] F. Hutter and Y. Hamadi. Parameter adjustment based on performance prediction: Towards an instance-aware problem solver. Technical report, Microsoft Research, Cambridge, UK, 2005.

[68] F. Hutter, H. Hoos, K. Leyton-Brown, and K. Murphy. Time-bounded sequential parameter optimization. In *Proceedings of the 7th International Conference on the Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR'10)*, volume 6073, pages 281–298. Springer-Verlag, LNCS, 2010.

[69] F. Hutter, H. Hoos, and T. Stützle. Automatic algorithm configuration based on local search. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, pages 1152 – 1157, 2007.

[70] F. Hutter, H. H. Hoos, and K. Leyton-Brown. *Learning and Intelligent Optimization*, chapter Sequential Model-based Optimization for General Algorithm Configuration, pages 507–523. Springer Berlin Heidelberg, 2011.

[71] F. Hutter, H. H. Hoos, K. Leyton-Brown, and K. P. Murphy. An experimental investigation of model-based parameter optimization: SPO and beyond. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO'09)*, pages 271–278, 2009.

[72] F. Hutter, H. H. Hoos, K. Leyton-Brown, and T. Stützle. ParamILS: An automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36(1):267–306, 2009.

[73] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black box functions. *Journal of Global Optimization*, 13:455 –492, 1998.

[74] S. Kalyanakrishnan and P. Stone. Efficient selection of multiple bandit arms: Theory and practice. In *Proceedings of the 27th International Conference on Machine Learning (ICML'10)*, pages 511–518, 2010.

[75] S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone. PAC subset selection in stochastic multi-armed bandits. In *Proceedings of the 29th International Conference on Machine Learning (ICML'12)*, pages 655–662, 2012.

[76] D. Karaboga. An idea based on honey bee swarm for numerical optimization. Technical report, Computer Engineering Department, Erciyes University, 2005.

[77] E. Kaufmann, A. Garivier, and T. Paristech. On bayesian upper confidence bounds for bandit problems. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS 2012)*, pages 592–600, 2012.

[78] N. Korda, E. Kaufmann, and R. Munos. Thompson sampling for 1-dimensional exponential family bandits. In *Advances in Neural Information Processing Systems (NIPS'13)*, pages 1448–1456, 2013.

[79] T. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 83(1):113–154, 1985.

[80] H. Leather, M. O. Boyle, and B. Worton. Raced profiles: Efficient selection of competing compiler optimization. In *Proceedings of the 2009 ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems*, pages 50–59, 2009.

[81] J. Lee, M. Sun, and G. Lebanon. PREA: Personalized recommendation algorithms toolkit. *Journal of Machine Learning Research*, 13:2699–2703, 2012.

[82] S. Lee, J. Yang, and S. Park. *Discovery Science*, chapter Discovery of Hidden Similarity on Collaborative Filtering to Overcome Sparsity Problem, pages 396–402. Springer Berlin Heidelberg, 2004.

[83] S. Lessmann, R. Stahlbock, and S. F. Crone. Genetic algorithms for support vector machine model selection. In *Proceedings of the 16th International Joint Conference on Neural Networks (IJCNN'06)*, pages 3063–3069, 2006.

[84] M. Lichman. UCI machine learning repository. `http://archive.ics.uci.edu/ml`, 2013. Irvine, CA: University of California, School of Information and Computer Science.

[85] P. Loh and S. Nowozin. Faster Hoeffding racing: Bernstein races via jackknife estimates. In *Proceedings of the 24th International Conference on Algorithmic Learning Theory (ALT'13)*, pages 203–217, 2013.

[86] M. Lopes-Ibanez and T. Stützle. Automatic configuration of multi-objective ACO algorithms. In *Proceedings of the 7th International Conference on Swarm Intelligence*, pages 95–106, 2010.

[87] J. A. Lopez and C. A. Walter. Evaluating covariance matrix forecasts in a value-at-risk framework. *FRB of San Francisco Working Paper No. 2000-21*, 2000.

[88] H. R. Lourenco, O. Martin, and T. Stützle. *Handbook of Metaheuristics*, chapter Iterative Local Search: Framework and Applications, pages 363–397. Kluwer Academic Publishers, 2010.

[89] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7:77–91, 1952.

[90] O. Maron and A. Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. *Advances in Neural Information Processing Systems 6 (NIPS'93)*, 6:59–66, 1994.

[91] A. Maurer and M. Pontil. Empirical bernstein bounds and sample-variance penalization. In *Proceedings of the 22nd Annual Conference on Learning Theory*, pages 115–124, 2009.

[92] 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*, 21:239–245, 1979.

[93] V. Mnih and C. Szepesvari. Empirical bernstein stopping. In *Proceedings of the 25th International Conference on Maching Learning (ICML'08)*, 2008.

[94] A. W. Moore. Efficient algorithms for minimizing cross validation error. In *Proceedings of the 11th International Conference on Maching Learning (ICML'94)*, pages 190–198, 1994.

[95] M. A. Muñoz, M. Kirley, and S. K. Halgamuge. *Computational Intelligence in Intelligent Data Analysis*, chapter The Algorithm Selection Problem on the Continuous Optimization Domain, pages 75–89. Springer-Verlag Berlin Heidelberg, 2013.

[96] V. Nannen and A. E. Eiben. Relevance estimation and value calibration of evolutionary algorithm parameters. In *International Joint Conferences on Artificial Intelligence*, pages 975–980, 2007.

[97] D. B. Neill and G. F. Cooper. A multivariate bayesian scan statistic for early event detection and characterization. *Machine Learning*, 79:261–282, 2010.

[98] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7:308–313, 1965.

[99] J. Neyman and E. S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231:289–337, 1933.

[100] E. O'Mahony, E. Hebrard, A. Holland, C. Nugent, and B. O'Sullivan. Using case-based reasoning in an algorithm portfolio for constraint solving. In *Proceedings of the 19th Irish Conference on Artificial Intelligence and Cognitive Science*, 2008.

[101] P. Pansuwan, N. Rukwong, and P. Pongcharoen. Identifying optimum artificial bee colony algorithm's parameter for scheduling the manufacture and assembly of complex products. In *Proceedings of the 2nd International Conference on Computer & Network Technology (ICCNT'10)*, pages 339–343, 2010.

[102] F. Peng, K. Tang, G. Chen, and X. Yao. Population-based algorithm portfolios for numerical optimization. *IEEE Transactions on Evolutionary Computation*, 14:782–800, 2010.

[103] J. Pickands. Statistical inference using extreme order statistics. *Annals of Statistics*, 3:119–131, 1975.

[104] J. C. Platt. *Advances in Kernel Methods - Support Vector Learning*, chapter Fast training of support vector machines using sequential minimal optimization. MIT Press, 1998.

[105] C. E. Priebe, J. M. Conroy, D. J. Marchette, and Y. Park. Scan statistics on enron graphs. *Computational and Mathematical Organization Theory*, 11:229–247, 2005.

[106] A. K. Qin, V. L. Huang, and P. N. Suganthan. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 3:398–417, 2009.

[107] M. T. Ribeiro, A. Lacerda, A. Veloso, and N. Ziviani. Pareto-efficient hybridization for multi-objective recommender systems. In *Proceedings of the 6th ACM Conference on Recommender Systems (RecSys'12)*, pages 19–26, 2012.

[108] F. Ricci, L. Rokach, and B. Shapira. *Recommender Systems Handbook*, chapter Introduction to Recommender Systems Handbook, pages 1 – 35. Springer, 2011.

[109] J. R. Rice. The algorithm selection problem. *Advances in Computers*, 1976.

[110] J. R. Rice. Methodology for the algorithm selection problem. In *Proceedings of the IFIP TC 2.5 Working Conference on Performance Evaluation of Numerical Software*, 1979.

[111] D. Ruppert. *Statistics and Data Analysis for Financial Engineering*. Springer-Verlag New York, 2011.

[112] C. Scarrott and A. MacDonald. A review of extreme value threshold estimation and uncertainty quantification. *Statistical Journal*, 10:33–60, 2012.

[113] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02)*, pages 253–260, 2002.

[114] M. Shen, W. Chen, J. Zhang, H. S. Chung, and O. Kaynak. Optimal selection of parameters for nonuniform embedding of chaotic time series using ant colony optimization. *IEEE Transactions on Cybernetics*, 43:790–802, 2013.

[115] V. A. Shim, K. C. Tan, J. Y. Chia, and A. A. Mamun. Multi-objective optimization with estimation of distribution algorithm in a noisy environment. *Evolutionary Computation*, 21:149–177, 2013.

[116] B. Silverthorn. *A Probabilistic Architecture for Algorithm Portfolios*. PhD thesis, Department of Computer Science, The University of Texas at Austin, 2012.

[117] S. K. Smit, A. E. Eiben, and Z. Szlvik. An MOEA-based method to tune ea parameters on multiple objective functions. In *Proceedings of the International Conference on Evolutionary Computation (ICEC'10)*, pages 261–268, 2010.

[118] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *25th Advances in Neural Information Processing Systems (NIPS'12)*, pages 1–9, 2012.

[119] M. Sobel and A. Wald. A sequential decision procedure for choosing one of three hypotheses concerning the unknown mean of a normal distribution. *The Annals of Mathematical Statistics*, 20:502–522, 1949.

[120] M. J. Streeter and S. F. Smith. An asymptotically optimal algorithm for the max k-armed bandit problem. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 135–142. AAAI Press, 2006.

[121] M. J. Streeter and S. F. Smith. Selecting among heuristics by solving thresholded k-armed bandit problems, 2006.

[122] M. J. Streeter and S. F. Smith. A simple distribution-free approach to the max k -armed bandit problem. In *Principles and Practice of Constraint Programming - CP 2006*, pages 560–574. Springer Berlin Heidelberg, 2006.

[123] M. J. Streeter and S. F. Smith. New techniques for algorithm portfolio design. In D. A. McAllester and P. Myllymäki, editors, *Proceedings of the Conference in Uncertainty in Artificial Intelligence (UAI)*, pages 519–527. AUAI Press, 2008.

[124] C. Szepesvári. *Algorithms for Reinforcement Learning*. Morgan & Claypool, 2009.

[125] G. Taguchi and Y. Yokoyama. *Taguchi Methods: Design of Experiments*. American Supplier Institute, Dearborn, MI, in conjunction with the Japanese Standards Association, 1994.

[126] J. Teich. Pareto-front exploration with uncertain objectives. *Evolutionary Multi-criterion Optimization*, 1999:314–328, 2001.

[127] C. Tekin and M. Liu. Online algorithms for the multi-armed bandit problem with markovian rewards. In *Proceedings of the 48th Annual Allerton Conference Communication, Control, and Computing*, pages 1675–1682, 2011.

[128] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25:285–294, 1933.

[129] F. van den Bergh and A. P. Engelbrecht. A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8:225– 239, 2004.

[130] S. Vargas and P. Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys'11)*, pages 109–116, 2011.

[131] J. Vrugt, B. Robinson, and J. Hyman. Self-adaptive multimethod search for global optimization in real-parameter spaces. *IEEE Transactions on Evolutionary Computation*, 13:243–259, 2009.

[132] A. Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16:117–186, 1945.

[133] A. Wald. *Sequential Analysis*. Dover, 1948.

[134] C. Wang. A new hybrid estimation method for the generalized pareto distribution. Master's thesis, University of Calgary, Alberta, Canada, 2011.

[135] J. Wang. Application of support vector machines in bioinformatics. Master's thesis, National Taiwan University, 2002.

[136] B. L. Welch. The significance of the difference between two means when the population variances are unequal. *Biometrika*, 29, 1937.

[137] P. H. Westfall, J. F. Troendle, and G. Pennello. Multiple mcnemar tests. *Biometrics*, 66:1185–1191, 2010.

[138] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1:67 – 82, 1997.

[139] L. Xu, F. Hutter, H. H. Hoos, and K. Leyton-Brown. SATzilla: Portfolio-based algorithm selection for sat. *Journal of Artificial Intelligence Research*, 32:565–606, 2008.

[140] B. Xue, M. Zhang, and W. N. Browne. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Transactions on Cybernetics*, 43:1656–1671, 2013.

[141] Yahoo. Yahoo! Finance, 2016.

[142] B. Yuan and M. Gallagher. On building a principled framework for evaluating and testing evolutionary algorithms: A continuous landscape generator. In *Proceedings of the 5th IEEE Congress on Evolutionary Computation (CEC'03)*, pages 451–458, 2003.

[143] B. Yuan and M. Gallagher. Statistical racing techniques for improved empirical evaluation of evolutionary algorithms. In *Parallel Problem Solving from Nature - PPSN VIII, 8th International Conference*, pages 175–184, 2004.

[144] S. Y. Yuen, C. K. Chow, and X. Zhang. Which algorithm should I choose at any point of the search: an evolutionary portfolio approach. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO'13)*, pages 567–574, 2013.

[145] J. Zhang and A. C. Sanderson. JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 13:945–958, 2009.

[146] T. Zhang, M. Georgiopoulos, and G. C. Anagnostopoulos. S-Race: A multi-objective racing algorithm. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO'13)*, pages 1565–1572, 2013.

[147] T. Zhang, M. Georgiopoulos, and G. C. Anagnostopoulos. Online model racing based on extreme performance. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation (GECCO'14)*, pages 1351–1358, 2014.

[148] T. Zhang, M. Georgiopoulos, and G. C. Anagnostopoulos. SPRINT multi-objective model racing. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation (GECCO'15)*, pages 1383–1390, 2015.

[149] Q. Zheng, Y. Fan, Z. Shi, and Y. Wang. Adaptive inertia weight particle swarm optimization. In *Proceedings of Artificial Intelligence and Soft Computing (ICAISC'06)*, volume 4029, pages 450–459, 2006.