FINAL INTERNSHIP REPORT

MULTI-OBJECTIVE OPTIMIZATION OF COMPUTATIONALLY EXPENSIVE MODELS BY PROGRESSIVE IMPROVEMENT OF SURROGATE MACHINE LEARNING MODELS

Author

Huanyu ZHAI

Engineering Student at University of Nice-Sophia Antipolis

Mentors

Dimitri SOLOMATINE

Professor of Hydroinformatics, Head of the IWSG Department at IHE DELFT

Juan Carlos CHACON-HURTADO

PhD Fellow in Hydroinformatics, IWSG Department, IHE DELFT

September 2018

United Nations
Educational, Scientific and
Cultural Organization

Institute for
Water Education
in partnership with UNESCO

ECOLE POLYTECHNIQUE DE L'UNIVERSITE DE NICE-SOPHIA ANTIPOLIS

Département Hydroinformatique et Ingénierie de L'Eau

1645, Route des Lucioles - 06410 Biot - France

Tél +33 (0)4 92 38 85 50 ● Fax +33 (0)4 92 38 85 02 ● hydro@polytech.unice.fr ● www.polytech.unice.fr

Contents

ABSTRACT

Process-based models (PMs) are computationally-expensive to optimize, since one probably need to run the model iteratively to find desired model parameters. In this report, we propose a multi-objective optimization algorithm MOPRISM to tackle the problem. MOPRISM is boosted by computationally-cheaper machine learning surrogate models that learn the behavior of the PM and use the surrogate in the optimization problem instead of the PM. This surrogate-assisted optimization approach exists for more than 30 years, during which many new techniques have been developed. Our MOPRISM is proposed because it shows better results than some other algothms when dealing problems with many local minima. It has a special "deadlock" detection mechanism to decide wether to udergo a "progressive improvement" step to possibly overcome the sub-optimality during the iteration. In this report, MOPRISM is tested with 11 benchmarking problems and a water distribution network design problem. A comparative study between MOPRISM and another algorithm GOMORS is present.

Keywords: Multi-objective optimization, Surrogate modelling, Machine learning models, Evolutionary algorithm

RESUME

Les modèles basés sur les processus sont chers à optimiser, car l'un a besoin d'exécuter le modèle itérativement jusqu'à les paramètres optimaux soivent trouvés. Nous proposons, dans le rapport, un nouvel algorithme, MOPRISM, réalisé pour ce problème. Les modèles d'apprentissage automatique sont présents dans MOPRISM pour que l'optimisateur évalue ces modèles au lieu des modèles chers. Ce technique existe depuis plus de 30 ans, cependant, beaucoup des études sont développées pour améliorer les algorithmes de cette famille. Notre MOPRISM est proposé pour la raison qu'il est plus performant que beaucoup d'autres algorithmes sur les problèmes avec de nombreux optima locaux. Dans ce rapport, nous allons comparer MOPRISM avec un autre algorithme GOMORS. Ils sont testés sur 11 problèmes classiques dans le domaine d'optimisation globale, et un problème de la réalisation d'un réseau hydraulique urbain.

Mots Clés: Optimisation multi-objectif, Modélisation de substitution, Apprentissage automatique, Algorithme évolutif

I INTRODUCTION

I.1 Motivation

Engineering design problems usually require simulations to evaluate design objective and constraint functions as function of design variables. In the field of water resources engineering, to design urban water distribution systems or urban drainage systems can be classified as those problems. Engineers may evaluate their trial designs in numerical models and finally choose the desired one among those choices. Each execution of such numerical models for the above-stated problems can take from seconds to hours, usually depending on the inner mechanisms of the model and the complexity of the problem. One way in practice to find the optimal solutions to such problems is to take the model as a black-box of which only the input and the output variables are known, and use this black-box model to evaluate proposals generated by a heuristic optimizer. However, heuristic methods usually involves large-scale sampling from the decision space. If each sample is evaluated via the above-constructed black-box model, the whole optimization problem can become extremely computationally-expensive and sometimes unfeasible for models which require hours to run, even in parallel.

Many studies have been carried out to alleviate this burden of computationality. Some authors focus on improving the effeciency of the heuristic algorithms (we will only discuss evolutionary algorithms as heuristics in this report), so that they propose evolutionary algorithms (EA) with different techniques to boost the search procedure. In recent years, researchers have also developed auto-adaptive EAs which showed promising results in some problems to find optimal solutions with limited function evaluations. Another popular approach applied to reduce the total number of expensive evaluations is building computationally cheaper but less accurate surrogate models and use these surrogates in the iterative evolutionary computations instead of the original expensive model. However, such surrogate-assisted optimization approcahes often behave sub-optimally due to the poor accuracy of the surrogates. The essence of using such method is to balance the trade-off between the quality of the solutions and your computer resources. There also present many studies dedicated to improve the effeciency of the surrogate-assisted optimization algorithms and to improve the accuracy of the surrogates. A more detailed review of the existing methods is present in the section I.3, where we will discuss the advantages and the limitations of several different algorithms.

In this report, we propose an algorithm framework, MOPRISM (**M**ulti-objective **O**ptimization by **PR**ogressive **I**mprovement of **S**urrogate **M**odels), in which surrogate models are used to replace the expensive model, and techniques such as adaptive sampling and local search are also present in order to minify the number of evaluations of the expensive model. Another key feature of MOPRISM is the "Progressive Improvement" step that could possibly avoid bad samples after several consecutive iterations without improvement in the performance evaluation metrics. This kind of "no improvement" behavior is referred as "dead-lock" in this report and can be frequently observed in surrogate-assisted optimization algotithms with sampler when the accuracy of the surrogates is poor.

I.2 Objectives

Our primary objective 1) is to introduce MOPRISM as a problem-independent algorithmic framework, so that one can choose, according to the problem, the type of the surrogate, the family of the embedded EA and if necessary, even the steps used in the computation. A pseudo-code is present to show the framework's outline. The components of MOPRISM are also explained in detail. But before we introduce MOPRISM, it is also essential 2) to review the recent progresses in developing surrogate-assisted optimization techniques, and 3) to target the state-of-the-art problems and limitations encountered in recent related works. In order to 4) test the robustness of MOPRISM, we run the algorithm to solve 15 classic global optimization benchmarking problems, and a real-world urban drainage system design problem. 5) A comparison study is present to show that MOPRISM outperforms some popular surrogate-free MOEAs and is comparable to some related algorithms, GOMORS for example. The GOMORS algorithm is proposed in Akhtar et Shoemaker 2016 [2], which shows promising results in both benchmarking problems and a real-world ground water remediation test problem, with very limited number of function evaluations. Finally, we will 6) analyse the limitations of MOPRISM and give use recommodations.

Besides this report, we would also like to make our software and its source code fully accessible to the public. One is able to reproduce any of the results showed in this report with the corresponding random seed. Those seed numbers are also available online for the purpose of testing and reviewing. For more information, please visit https://github.com/NikoZHAI/samolib.

I.3 Literature Review

Three substaintial literature reviews by Nicklow et al.[19], Razavi et al. [20] and Maier et al. [14] summarize both the breakthroughs in developing new robust algorithms and the applications of surrogate modelling in water-resource problems. In this report, we will only focus on some critical works on improving surrogate-assisted global optimization algorithms, which form the fundation of the proposed MOPRISM.

Jones et al., 1998 [9] proposed the Efficient Global Optimization (EGO) using Kriging for response surface modelling. Regis et Shoemaker 2005 [23] used radial basis functions (RBFs) to improve the efficiency of an evolutionary algorithm. Knowles proposed a hybrid methode parEGO [10] to solve MO optimization problems by decomposing the MO problem into several single objective problems. Recently in 2016, Akhtar et Shoemaker proposed GOMORS [2], in which the surrogate is still RBFs but a "Gap Optimization" technique was applied.

All of the above surface modelling techniques use either Kriging or RBFs as their surrogates. There are also other robust function appoximators like artificial neural network (ANN) or support vector machine (SVM). Yan et Minsker proposed the Trust Region based Adaptive Meta-model Genetic Algorithm (TRAMGA) in 2005 [30], then the Adaptive Neural Network Genetic Algorithm (ANGA) in 2006 [31], and its variate Noisy-ANGA in 2010 [32]. All these three frameworks are originally for single-objective problems. Xu et al. 2010 implemented a multi-objective version of ANGA and compared the MO ANGA with MOPRISM [29]. Martinez et al. 2010 [15] and Rosales-Pérez et al. 2013 [24] showed their results with SVM surrogates. But optimal SVM parameter values are difficult to interpret and relate to characteristics of the response surface [20], and the results in [24] are considered less satisfying than the MOPRISM results on same problems in Section IV. Various authors have indicated that RBFs could be more effective than other approximation methods in MO optimization of PMs with more than 15 decision variables [2, 4, 24, 16].

The accuracy of the surrogate substaintially influences the performance of the surrogate-assisted MOEA. But it is difficult to train an ML surrogate to be accurate with a limited number of evaluated samples. So various authors showed the significance of cooperating with other techniques such as adaptive sampling [2, 31, 32], and local search [2, 27, 28, 30].

II BACKGROUND & NOTATION

II.1 Background

Before we look into both the algorithm and the first results, we would like to elaborate on some notions used in this report for those who may not have much prior knowledge in global optimization and surrogate modelling. Figure II-1 shows an introductory example of the optimization results of an MO problem (MOP) to help understanding of the following explanations.

II.1.1 Multi-objective optimization

Multi-objective optimization is a type of decision-making problems in which multiple criteria are involved to determine the goodness of decision choices. In this paper, we only focus on some MOPs where more than one objective functions are taken into account to be optimized simultaneously.

II.1.2 Domination & Non-domination

Domination In the introductory example of a minimization problem presented in Figure II-1-a, $\boldsymbol{S_m}$ is an ensemble of found optimal solutions. For any other solution $\boldsymbol{x_1}, \boldsymbol{x_2} \in \Omega$ but $\boldsymbol{x_1} \notin \boldsymbol{S_m}$, we have $\exists \boldsymbol{x_2} \in \boldsymbol{P_m}$, which satisfies $\boldsymbol{f_1(x_2)} < \boldsymbol{f_2(x_1)}$ *and* $\boldsymbol{f_2(x_2)} < \boldsymbol{f_2(x_1)}$. Then solution $\boldsymbol{x_1}$ is called dominated by solution $\boldsymbol{x_2}$, noted as $\boldsymbol{x_2} \prec \boldsymbol{x_1}$ ($\boldsymbol{x_2}$ dominates $\boldsymbol{x_1}$).

Non-domination With $\boldsymbol{S_m}, \boldsymbol{x_1}$, and $\boldsymbol{x_2}$ defined above, $\forall \boldsymbol{x_i}, \boldsymbol{x_j} \in \boldsymbol{S_m}, \boldsymbol{x_i} \nprec \boldsymbol{x_j}$ *and* $\boldsymbol{x_j} \nprec \boldsymbol{x_i}$. We then say that $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are non-dominated to each other.

II.1.3 Pareto optimal & Pareto front

In multi-objective optimization, when the different objectives are contradictory, an optimal solution is said Pareto optimal when it is not possible to improve an objective without degrading the others [12]. In our introductory example in Figure II-1-a, if all the solutions found in $\boldsymbol{S_m}$ are global optima, they are then **Pareto optimal** because they will not be dominated by another solution. In the objective space, one can find a set of global optimal solutions $\boldsymbol{P_m}$ that satisifies $\forall \boldsymbol{x_i}, \boldsymbol{x_j} \in \boldsymbol{S_m}, \boldsymbol{x_i} \nprec \boldsymbol{x_j}$ *and* $\boldsymbol{x_j} \nprec \boldsymbol{x_i}$, the solution set $\boldsymbol{S_m}$ is then called **Pareto front** or **Pareto frontier**.

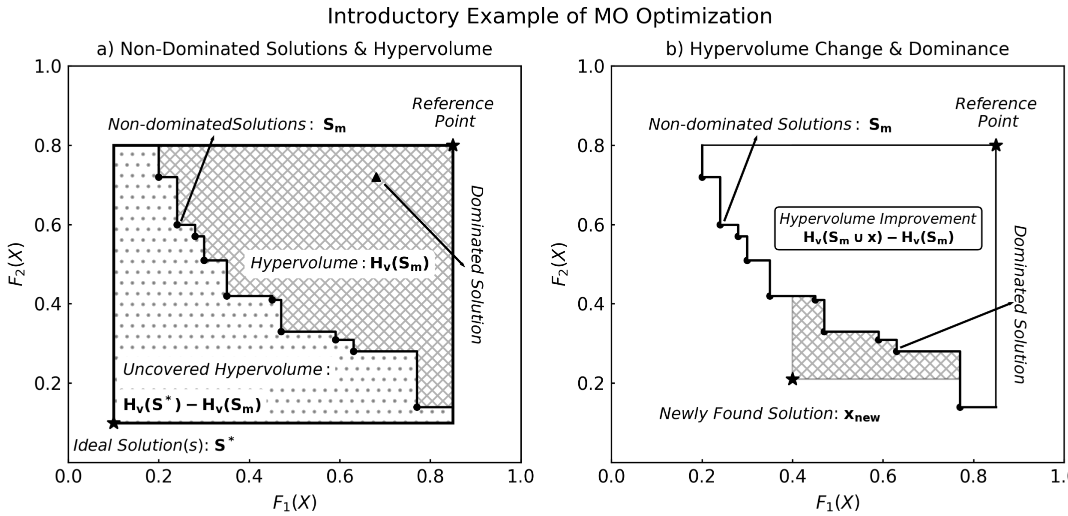

Figure II-1 *An introductory example of dominance, Pareto optimality, and hypervolume metrics in multi-objective optimization problems*

II.1.4 Evolutionary algorithm (EA)

The detail of an evolutionary algorithm is complicated and it is not our interest to discuss it here. We will quote a brief definition of an EA in [26]: *"An EA uses mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection. Candidate solutions to the optimization problem play the role of individuals in a population, and the fitness function determines the quality of the solutions. Evolution of the population then takes place after the repeated application of the above operators."*

In an multi-objective evolutionary algorithm (MOEA), individuals have multiple fitnesses with respect to multiple objective functions. An individual $\boldsymbol{A}$ is considered strictly better than another individual $\boldsymbol{B}$ if and only if $\boldsymbol{A} \prec \boldsymbol{B}$ $(\boldsymbol{A}\ \boldsymbol{dominates}\ \boldsymbol{B})$.

II.1.5 Surrogate modelling

A surrogate model is used to mimic the behaviour of another model whose outcome of interest cannot be easily directly measured. In most cases, a surrogate is computationally cheap(er) to evaluate than the original model. For example, a conceptual hydrological model is an emmulation of the physic processes, so it can be regarded as a surrogate. In our study, we will focus on machine learning models as surrogates.

II.1.6 Mechine learning models

Like EA, an ML model is also a complex concept so that we would like to quote its definition by Thomas Mitchell in his book [17]: *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."*

In this research, we will mainly discuss about three types of ML approaches: 1) Artificial Neural Net (ANN), 2) Radial Basis Function Network (RBF Net), and 3) Kriging (also known as Gaussian Process). We will not elaborate on the detailed mechanisms of these algorithms in this report.

II.2 Notations

The Table II-1 shows notations that are applicable to the rest of the report.

Table II-1 Definitions of Notations

Notation	Definition
$N \in \mathbb{N}^+$	Number of decision variables of the MOP
$M \in \mathbb{N}^+$	Number of objective functions of the MOP
$k \in \mathbb{N}$	Number of episode
Ω^N	The decision space of the MOP
$\boldsymbol{x} = [x_1, ..., x_N]$	A vector of decision variables, a decision vector
$F(\boldsymbol{x}) = [f_1(\boldsymbol{x}), ..., f_M(\boldsymbol{x})]$	Objective functions of the MOP
$\hat{F}_k(\boldsymbol{x}) = [\hat{f}_{1,k}(\boldsymbol{x}), ..., \hat{f}_{M,k}(\boldsymbol{x})]$	Objective functions emmulated by the surrogate at episode k
$X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$	A set of decision vectors of size n, X_k means that X is obtained in episode k
$S_{ideal} = \{\boldsymbol{x}_{ideal,1}, ..., \boldsymbol{x}_{ideal,n}\}$	The Pareto Set (PS) of the MOP
$S_k = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$	A set of non-dominated solutions obtained in episode k by evaluating F
$\hat{S}_k = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$	A set of non-dominated solutions obtained in episode k by evaluating $\hat{F}_k$
$P_k = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$	A population of decision vectors in MOEA in episode k
$C_k = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$	Candidates to be expensively evaluated in episode k
$A_k = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$	An archive of all of the expensively evaluated desicion vectors till episode k
μ_k	The value of metrics at episode k, i.e. the hypervolume $\mu_k = H_v(S_k)$
^{local}	Stands for local search, i.e. P_k^{local} means a local search population at episode k

III METHODOLOGY

Our surrogate-assisted MOEA aims to optimize a multi-objective function by solving a sub-problem learned by a metamodel. Our approach outperforms a surrogate-free MOEA with respect to the number of expensive model evaluation. This boosted MOEA consists of a sampler, a local search step and one or more metamodels. In addition to the MOEA over the surrogate, a progressive, step-by-step, evolutionary computation over the expensive model is performed once the algorithm traps in sub-optimal solutions. This technique is referred as "progressive improvement" in this report. The peudo-code of the MOPRISM algorithm framework is present in the section III.2.

III.1 Problem Formulation

We consider a general optimization problem, minimizing a N-dimensional function F with M objectives, subject to the decision space Ω:

$$x = \underset{x}{\operatorname{argmin}}\, F(x)\ ,\ F(x) = [f_1(x), ..., f_M(x)]\ ,\ x \in \Omega^{\mathbb{N}} \tag{3.1.1}$$

Note that in a classic, surrogate-free MOEA, the problem is optmized by iteratively running evolutionary computations over $F(x)$ directly (eq 3.1.1). In MOPRISM, a sub-problem $\hat{F}(x)$ is constructed by metalearning. The metamodel $\hat{F}$ is considered iteratively improved over episodes, so $\hat{F}_k$ represents the metamodel at episode k. Let θ be the parameters of the metamodel, so $\hat{F}_k(x) \equiv \hat{F}(x, \theta_k)$. With the loss function L of the metamodel, the metalearning problem in MOPRISM is defined:

$$\hat{\theta}_k = \underset{\theta}{\operatorname{argmin}}\, L(\theta, F, \hat{F}, x, k)\ ,\ x \in \Omega^{\mathbb{N}},\ k \in \mathbb{N} \tag{3.1.2}$$

At each episode k, once the training of the surrogate (eq 3.1.2) is finished, the sub-problem $\hat{F}_k(x)$ is optmized in the MOEA instead of the $F(x)$ (eq 3.1.3):

$$\hat{S}_k = \underset{x}{\operatorname{argmin}}\, \hat{F}_k(x)\ ,\ \hat{F}_k(x) = [\hat{f}_{1,k}(x), ..., \hat{f}_{M,k}(x)]\ ,\ x \in \Omega^{\mathbb{N}},\ k \in \mathbb{N} \tag{3.1.3}$$

At each episode k, we sample n candidates $\hat{C}_k$ from $\hat{S}_k$ and $\hat{P}_k$ to be expensively evaluated, and thus obtain C_k. Update the archiver A_k at each episode k: $A_k \leftarrow A_{k-1} \cup C_k$. After termination, the final episode $k = K$, compute all the non-dominated solutions S_K from A_K. S_K is considered our set of solutions to the problem (eq 3.1.1), by using MOPRISM.

III.2 Algorithm Outline

Algorithm 1: General Outline of Multi-Objective PRogressive Improvement Surrogate Model

Global Setup: Problem definition, MOEA and surrogate configuration.

Initialization: Sample population P_1 and evaluate fitness with F: $P_1 \leftarrow ExpensiveEval(P_1)$

$S_1 \leftarrow ComputeFront(P_1), Update\ A_1 \leftarrow P_1, \hat{F}_1 \leftarrow TrainSurrogate(A_1)$

begin [MOPRISM]

  while $Not\ StopMOPRISM()$ **do**

    [Optional] $P_k \longleftarrow CrossOver(S_k)$

    k ++, (re)set MOEA generation counter to 1

    begin [Evolutionary Computation]

      $\hat{P}_k \leftarrow MOEA(P_{k-1})$, optimize: $\hat{F}_{k-1}(x) = [\hat{f}_{1,k-1}(x), ..., \hat{f}_{M,k-1}(x)], x \in \Omega$

      if $StopMOEA()$ **then** Break;

    end

    begin [Local Search]

      Sample $X_k^{local} = \{x_1, ..., x_n \mid x_i \in \hat{S}_k\ , n \in \mathbb{N}\}$ for local search

      $\hat{S}_k^{local} \leftarrow LocalSearch(X_k^{local})$, optimize: $\hat{F}_{k-1}(x) = [\hat{f}_{1,k-1}(x), ..., \hat{f}_{M,k-1}(x)], x \in \Omega_k^{local}$

    end

    [Sampling] Sample candidates $\hat{C}_k = Sampler(\hat{P}_k, \hat{S}_k, \hat{S}_{local,k}, \Omega)$ for expensive evaluation

    $C_k \leftarrow ExpensiveEval(\hat{C}_k)\ ,\ Update\ archiver\ A_k \leftarrow A_{k-1} \cup C_k$

    $S_k \leftarrow ComputeFront(S_{k-1} \cup C_k)$

    $\mu_k \leftarrow ComputeMetrics(S_k,\ S_{k-1})$

    if $IsDeadlock(\mu_k)$ **then**

      $P_k \leftarrow ExpensiveEval(\hat{P}_k)\ or\ ExpensiveEval(\ CrossOver(\hat{S}_k)\ )$

      $S_k \leftarrow ComputeFront(S_k \cup P_k)$

      $Update\ archiver\ A_k \leftarrow A_k \cup P_k$

    end

    [Retraining] $\hat{F}_k \leftarrow TrainSurrogate(A_k)\ or\ TrainSurrogate(A_k - A_{k-1})$

  end

end

III.3 Initiatives

The MOPRISM algorithm framework was first proposed in [29]. The novelty of the framework is that when the surrogate-assisted MOEA trapped into local optimal solutions MOPRISM will choose to expensively evaluate one entire set of samples ($\hat{P}_k$ or $\hat{S}_k$, etc.) in the next one or more episodes and retrain the surrogate with these new samples. We name this operation the ”progressive improvement” step, and the behavior that the algorithm confused by sub-optimality a“dead-lock”. During the “progressive improvement” phase, MOPRSIM can be viewed as a normal MOEA. Thanks to this technique, MOPRISM is considered capable to solve any problem that is tractable to a normal, surrogate-free MOEA.

The previous version of MOPRISM lacks effecient sampling techniques to select samples for expensive evaluation. Imagine that we have 100 candidates in the current iteration, and only 5 in 100 are true non-dominated solutions. If we expensively evaluate all the 100 candidates, 95 simulations will be ineffective. This is a waste of our computer resources. Now we would like to apply some robust sampling techniques to select, before knowing their true fitness values, 5 in the 100 candidates to be expensively evaluated. If only one in the five sampled candidates is among the five true non-dominated solutions, our effective sampling rate will be augmented by four times, from 5% to 20%. So, we add a sampling step in the new MOPRISM to possibly boost its performance on the problems of which the total number of evaluation is limited. The new MOPRISM also has a local search phase. During the local search step, an evolutionary computation is performed to optimize a sub-problem with objective functions: $\hat{F}_{k-1}(x) = [\hat{f}_{1,k-1}(x), ..., \hat{f}_{M,k-1}(x)]$, subject to a shrinked decision space: $x \in \Omega_k^{local}$. With performant surrogates, a local search can be used for refined optimization in one or more local modals to provide candidates for expensive evaluation.

In general, the new MOPRISM takes full advantage of its sampler to reduce the number of expensive evaluation, and is able to jump out of the sub-optimal solutions by performing “progressive improvements”. Besides, in our implementation of MOPRISM, several minor techniques such as caching, paralellization, and vecorized operations, are applied but not showed in Algo.1. They are considered mandatory in order to minify unnecessary performance losses.

Surrogates are the essence to a surrogate-assisted MOEA. If a surrogate emmulates well the expensive model, the problem will be half-way solved. But to choose and train a ML surrogate is problem-dependent. So, it is beyond our interests to discuss such pronblem in this report. We have run tests using artificial neural nets (ANN), radial basis functions (RBFs), and gaussian process regression (GPR). Details on the test results are present in the section IV.

III.4 Components of MOPRISM

MOPRISM contains five main components, the MOEA, the surrogate, the sampler, the local search operator, and the "deadlock-progressive improvement" technique. Since, the MOEA and the surrogate have their own configuration routines, we will only discuss the rest three components.

The sampler is desired to sample possible solutions from the cheaply-evaluated decision vectors. A good sampler can dramatically improve the effective sampling rate. One can choose multiple sampling rules to construct a sampler. Akhtar and Shoemaker (2016) [2] propose 5-rule sampling in GOMORS: 1) random sampling in Ω, sampling in $\hat{S}_k$ by maximizing the Euclidean distance, in 2) the desicion space and in 3) the objective space, between the candidates and the points in S_k, sampling in 4) $\hat{S}_k$ and 5) $\hat{S}_k^l ocal$ y maximizing hypervolume improvement. The tests shows in Section **??** used the above-stated sampling rules. Other problem-specific rules can also be applied.

The local search operator constructs one or more sub-problems to optimize. By choosing several sub-search domains, the operator renders a series of locally-optimized sets to be sampled from for expensive evaluations.

Porgressive Improvement is performed after n consecutive steps without improvement (a "deadlock"). In this step, a set of individuals evolve for one more generation to form the population of the next MOPRISM episode. This step gives MOPRISM the momentum to jump out of local optimal solutions and inaccurate solutions aprroximated by the surrogate. It is also useful to accelerate the optimization procedure when the algorithm converges very fast to global optimal solutions before the stopping criteria satisfied, but continue to sample very few points for expensive evaluation at each episode.

IV EXPERIMENTS

We applied our MOPRISM algorithm to several classic benchmarking problems for global optimization and a real-world water distribution network design problem. These test problems are chosen beacause they have very noisy objective functions with many local optima, and they require the algorithm to identify the Pareto solutions in high-dimensional desicion spaces (from 8 to 34 dimensions). A comparative study between MOPRISM and another algorithm GOMORS (see [2]) is present on each benchmarking problem. On most of these problems, MOPRISM shows acceptable results and good capabilities to jump out of the local optima then converge to global solutions. However, our implementation of the GOMORS algorithm tends to "stuck" in local solutions and fails to solve some problems globally within the given number of expensive evaluations.

IV.1 Benchmarking Problems

Many benchmarking problems have been developed to test the performance of an optimizer. In this report, we applied both MOPRISM and GOMORS on 11 test problems, the Kursawe function (KURS) [11], 5 functions from the ZDT family [33] and 5 functions from the LZ family [13]. These functions are cheap to compute, but to derive their Pareto solutions is complicated. All of the 11 problems are multi-modal. The second objective function of the Kursawe problem is very noisy, and the problem has a disconnected, irregular Pareto front. The ZDT functions have relatively high-dimensional desicion space (from 10 to 30 dimensions). The pareto solutions of the LZ functions are complex distributed in the desicion space. The KURS function and the ZDT functions are described in Table IV-1. The LZ functions was first proposed in Li and Zhang (2009) to carry out a comparative study between MOEA/D and NSGA-II [13]. There are in total 9 functions in the family. In this report, we choose the LZ function No.1 to No.5 for testing, they are described in Table IV-2.

Experiment Setup

[Dimension] The dimensions of the three test sets are theoritically customizable. In this study, we choose the classic setups for ZDT and KURS problems see Table IV-1. For the LZ problems we set N to 8 for all the five problems, for variable boundaries see Table IV-2. All the 11 test problems have 2 objective functions for minimization.

[MOEA] In our experiments, we use NSGA-II and MOEA/D as MOEAs embedded in MOPRISM and GOMORS. The population size is set to 100, iterate over 50 generations in each episode. Both algorithm applies *Simulated Binary Crossover (SBX)* [1] and *Polynomial Mutation* [3]. Crossover probability is set to 0.9, mutation rate 0.1 for NSGA-II and $\frac{1}{N}$ for MOEA/D. Results obtained with MOEA/D are shown in Section V.1 and those with NSGA-II are shown in the Appendix-B.

[MOPRISM and GOMORS] For the purpose of comparison, we apply *Radial Basis Functions (RBFs)* surrogate in both of the two algorithms and apply the sampling and local search strategies of GOMORS proposed in [2] to MOPRISM. This configuration allows us to tell the importance of the "progrssive improvement" step in MOPRISM. The improvement tolerance in MOPRISM is set to 1% in three consecutive episodes. The numbers of total expensive evaluation are shown in Section V.1.

Table IV-1 Kursawe and ZDT Test Functions Used for MOPRISM Benchmarking

Problem	N^*	Variable Range	Objective Functions	Analytical Solutions	Comments
KURS	3	$[-5, 5]$	$f_1(x) = \sum_{i=1}^{2} \left[-10exp\left(-0.2\sqrt{(x_i^2 + x_{i+1}^2)}\right)\right]$ $f_2(x) = \sum_{i=1}^{3} \left[\lvert x_i\rvert^{0.8} + 5sin(x_i^3)\right]$	See [11]	Convex
ZDT-1	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x)h(f_1(x), g(x))$ $g(x) = 1 + \sum_{i=2}^{30} x_i$ $h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}}$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, ..., N^*$ See [33]	Convex
ZDT-2	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x)h(f_1(x), g(x))$ $g(x) = 1 + \sum_{i=2}^{30} x_i$ $h(f_1(x), g(x)) = 1 - \left(\frac{f_1(x)}{g(x)}\right)^2$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, ..., N^*$ See [33]	Non-convex
ZDT-3	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x)h(f_1(x), g(x))$ $g(x) = 1 + \sum_{i=2}^{30} x_i$ $h(x) = 1 - \sqrt{\frac{f_1(x)}{g(x)}} - \left(\frac{f_1(x)}{g(x)}\right) sin(10\pi f_1(x))$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, ..., N^*$ See [33]	Non-convex Disconnected
ZDT-4	10	$x_1 \in [0, 1]$ $x_i \in [-5, 5]$ $i \in \{2, ..., 10\}$	$f_1(x) = x_1$ $f_2(x) = g(x)h(f_1(x), g(x))$ $g(x) = 91 + \sum_{i=2}^{10}(x_i^2 - 10cos(4\pi x_i))$ $h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}}$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, ..., N^*$ See [33]	Non-convex
ZDT-6	10	$[0, 1]$	$f_1(x) = 1 - exp(-4x_1)sin^6(6\pi x_1)$ $f_2(x) = g(x)h(f_1(x), g(x))$ $g(x) = 1 + 9\left[\frac{\sum_{i=2}^{10} x_i}{9}\right]^{0.25}$ $h(f_1(x), g(x)) = 1 - \left(\frac{f_1(x)}{g(x)}\right)^2$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, ..., N^*$ See [33]	Non-convex Non-uniform spaced

N^* is the dimensionality of the problem's decision space.

Table IV-2 LZ Test Functions Used for MOPRISM Benchmarking

Problem	Variable Range	Objective Functions and Pareto Solutions (PS)
LZ-1	$[0,1]^n$	$f_1(x) = x_1 + \frac{2}{J_1}\sum_{j\in J_1}\left(x_j - x_1^{0.5+\frac{1.5(j-2)}{n-2}}\right)^2$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{J_2}\sum_{j\in J_2}\left(x_j - x_1^{0.5+\frac{1.5(j-2)}{n-2}}\right)^2$ where $J_1 = \{j \mid j$ is odd and $2 \leqslant j \leqslant n\}$ and $J_2 = \{j \mid j$ is even and $2 \leqslant j \leqslant n\}$ Its PS is $x_j = x_1^{0.5+\frac{1.5(j-2)}{n-2}}$, $j = 2, ..., n$.
LZ-2	$[0,1] \times [-1,1]^{n-1}$	$f_1(x) = x_1 + \frac{2}{J_1}\sum_{j\in J_1}\left(x_j - sin(6\pi x_1 + \frac{j\pi}{n})\right)^2$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{J_2}\sum_{j\in J_2}\left(x_j - sin(6\pi x_1 + \frac{j\pi}{n})\right)^2$ where J_1 and J_2 are the same as those of LZ-1 Its PS is $x_j = sin(6\pi x_1 + \frac{j\pi}{n})$, $j = 2, ..., n$.
LZ-3	$[0,1] \times [-1,1]^{n-1}$	$f_1(x) = x_1 + \frac{2}{J_1}\sum_{j\in J_1}\left(x_j - 0.8x_1cos(6\pi x_1 + \frac{j\pi}{n})\right)^2$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{J_2}\sum_{j\in J_2}\left(x_j - 0.8x_1sin(6\pi x_1 + \frac{j\pi}{n})\right)^2$ where J_1 and J_2 are the same as those of LZ-1 Its PS is $x_{j\in J_1} = 0.8x_1cos(6\pi x_1 + \frac{j\pi}{n})$, $x_{j\in J_2} = 0.8x_1sin(6\pi x_1 + \frac{j\pi}{n})$
LZ-4	$[0,1] \times [-1,1]^{n-1}$	$f_1(x) = x_1 + \frac{2}{J_1}\sum_{j\in J_1}\left(x_j - 0.8x_1cos(2\pi x_1 + \frac{j\pi}{3n})\right)^2$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{J_2}\sum_{j\in J_2}\left(x_j - 0.8x_1sin(6\pi x_1 + \frac{j\pi}{n})\right)^2$ where J_1 and J_2 are the same as those of LZ-1 Its PS is $x_{j\in J_1} = 0.8x_1cos(2\pi x_1 + \frac{j\pi}{3n})$, $x_{j\in J_2} = 0.8x_1sin(6\pi x_1 + \frac{j\pi}{n})$
LZ-5	$[0,1] \times [-1,1]^{n-1}$	$f_1(x) = x_1 + \frac{2}{J_1}\sum_{j\in J_1}\{x_j - [0.3x_1^2cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1]cos(6\pi x_1 + \frac{j\pi}{n})\}^2$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{J_2}\sum_{j\in J_2}\{x_j - [0.3x_1^2cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1]sin(6\pi x_1 + \frac{j\pi}{n})\}^2$ where J_1 and J_2 are the same as those of LZ-1 Its PS is $x_{j\in J_1} = [0.3x_1^2cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1]cos(6\pi x_1 + \frac{j\pi}{n})$, $x_{j\in J_2} = [0.3x_1^2cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1]sin(6\pi x_1 + \frac{j\pi}{n})$

IV.2 Water Distribution Problem

We also applied our algorithm on the water distribution network design problem of the city of Hanoi, Vietnam. The problem was first introduced in Fujiwara and Khang (1990)[6], in which the trunk layout of the water distribution network was to be realized. The network consists of 34 pipes, 31 nodes and one reservoir, organized in three loops (see Figure IV-1). In this design, pumping facilities are not required. All the flows are distributed by gravity. The reservoir is considered to have a constant head of 100m. The data of the network can be found in Appendix-A.

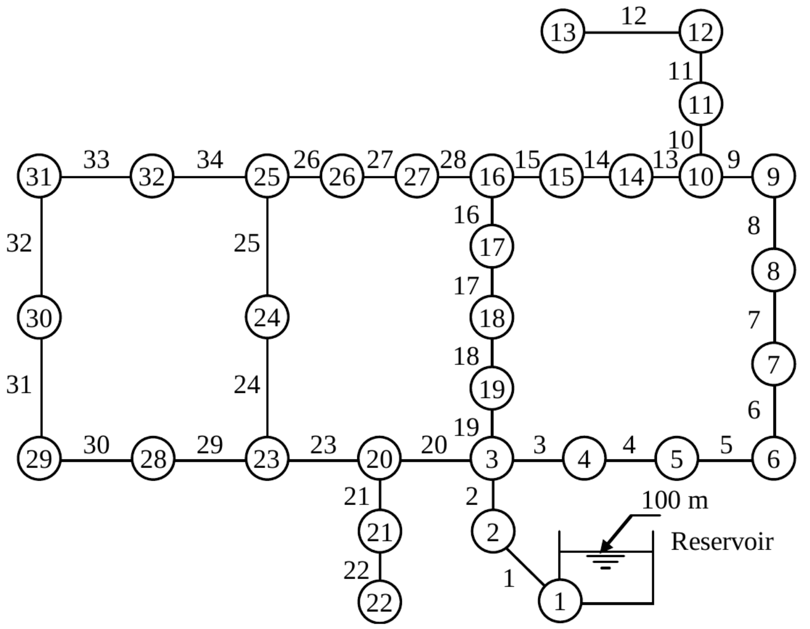

Figure IV-1 *Hanoi Network Layout, Source: [22]*

The main initiatives for choosing this problem are that 1) the problem has relatively high dimension (34 decision variables), 2) the objective functions are derived from a hydraulic solver that can be viewed as a black box model, and 3) many related studies [6, 7, 21, 22] exist to compare the results. In the Hanoi problem, we play with the diameters of the 34 pipes and put the modified network in a hydraulic solver, then find the optimal designs with low construction cost and high *weighted Demand Supply Ratio (wDSR)*. The two objective functions of the Hanoi problem can be thus defined as:

$$f_{cost} = \sum_{i=1}^{34} UnitPrice(D_i) \times L_i \ \rightarrow \min \tag{4.2.4}$$

$$f_{wDSR} = \frac{\sum_{j=1}^{32} Q_j^{act}}{\sum_{j=1}^{32} Q_j} \rightarrow \max \tag{4.2.5}$$

According to [6], pipes of 6 commercial diameters are available to choose and the unit-length price of the pipes are available in Appendix-A. In Eq.4.2.5, the actual simulated demand at node j, Q_j^{act} can be determined by the node's pressure head (see Eq.4.2.6 [22]):

$$Q_j^{act} = \begin{cases} 0 & p_j < p^{zero} \\ Q_j \sqrt{\dfrac{p_j - p^{zero}}{p^{req} - p^{zero}}} & p^{zero} \leqslant p_j \leqslant p^{req} \\ Q_j & p_j > p^{req} \end{cases} \tag{4.2.6}$$

The *wDSR* tells us to what extent the water demand at each node in the network is met. In Eq.4.2.6, p_j is the simulated pressure head at node j, p^{zero} is the minimum pressure head required to zero nodal water demand satisfaction, p^{req} is the required pressure head to complete nodal water demand satisfaction, and Q_j is the simulated demand at node j that is retrievable from the hydraulic solver. In some studies [6, 7, 21], the Hanoi problem is solved as a single-objective optimization problem to minimize the construction cost with a constraint, which punishes the objective values if the simulated pressure head is less than 30m:

$$f_{cost} = \sum_{i=1}^{34} [UnitPrice(D_i) \times L_i] + \sum_{i=1}^{32} [\max(30 - p_j, 0) \times Penalty] \to \min \tag{4.2.7}$$

The main difference between the two problem formulations Eq.4.2.7 and Eq.4.2.4, Eq.4.2.5 is that the former sets a strict working condition for the network and does not compromise with the solutions which have slightly poorer performances that are negligible . But the latter, multi-objective formulation, allows reasonable performance leaks depending on the configuration of p^{zero} and p^{req}. With this setup, it is possible for engineers to balance the trade-offs between the performance and the cost then find designs that work just slightly worse in extreme conditions (peak demands) but with more advantageous construction cost.

Experiment Setup

A great challenge in the Hanoi problem is that the decision variables can be categorical (pipe types) or discrete values (pipe diameters). This kind of problems are referred as *Integer Programming* (or *Mixed-Integer Programming (MIP)* if only a part of the decision variables are constrained to be integer values). According to many authors [18, 8, 5], MIPs are very difficult to solve because many combinations of specific integer values for the variables must be tested, and each combination requires the solution of a ”normal” linear or non-linear optimization problem. We implemented two sets of sub-experiments where 1) the desicion variables are pipe classes from 1 to 6 and 2) the desicion variables are first considered realed values ranged from 304.8mm (12in) to 1016mm (40in) thus the problem is considered continuous in the desicion space, then re-solve the problem by relaxing the real-number solutions to nearby integer solutions. We will discuss the Hanoi experiment realized with MOPRSIM only. Because our implementation of GOMORS fails to solve the Hanoi problem and no meaningful results are obtained.

[Dimension] The Hanoi network has 34 pipes and thus the problem is of 34-dimension in its desicion space. The inputs are the diameters of the pipes ranged from 304.8mm to 1016mm (6 classes, see Appendix-A). Two objectives are considered, the construction cost and the *wDSR*.

[MOEA] In Hanoi experiment, both NSGA-II and MOEA/D were first chosen as evolutionary strategy embedded in MOPRISM. However, experiments with embedded NSGA-II failed due to computer memory overflow. We will cosequently present only the test results of MOPRISM with embedded MOEA/D. The population size is set to 100, iterate over 50 generations in each episode. The MOEA/D applies *SBX* as crossover operator and *Polynomial Mutation* as mutation operator. Crossover probability is set to 0.9, mutation rate $\frac{1}{N}$.

[MOPRISM] We apply the same configuration of MOPRISM as introduced in IV.1, with the maximum number of expensive evaluation set to 3000.

V EXPERIMENT RESULTS

V.1 Benchmarking Results

In this section, we present the test results on the 11 benchmarking problems derived by both MOPRISM and GOMORS. The maximum number of expensive evaluation is set to 1000, 2000, and 1000 to KURS, ZDT and LZ respectively. The hypervolume coverage is applied as the metric measuring the results' goodness (higher is better, maximum 1). The previous version of MOPRISM proposed in [29] was also tested on the ZDT problems with 5000 expensive evaluations, but our new version performs better and with much less evaluations.

The results are shown in Figure V-1 to V-11. From the reslts, MOPRISM is considered to have solved the majority of the 11 problems except ZDT-4, which is extremely multi-modal, with 21^9 local Pareto fronts [33]. On the other hand, GOMORS failed on several problems which have either disconnected Pareto front or multi-modality. From the hypervolume coverage metric, GOMORS is often observed "stuck" over many iterations because the sampled candidates are found dominated after expensive evaluation. MOPRISM also encounters this situation, but thanks to the deadlock detection and the "progressive improvement" step, MOPRISM performs a normal evolutionary computation after having observed 3 consecutive steps with no improvement. So the "progressive improvement" is the essence of MOPRISM to make a difference in the benchmarking tests. Furthermore, since the "progressive improvement" technique is independent from other mechanisms in the algorithm, it is also possible to be integrated into other algorithms as a complement.

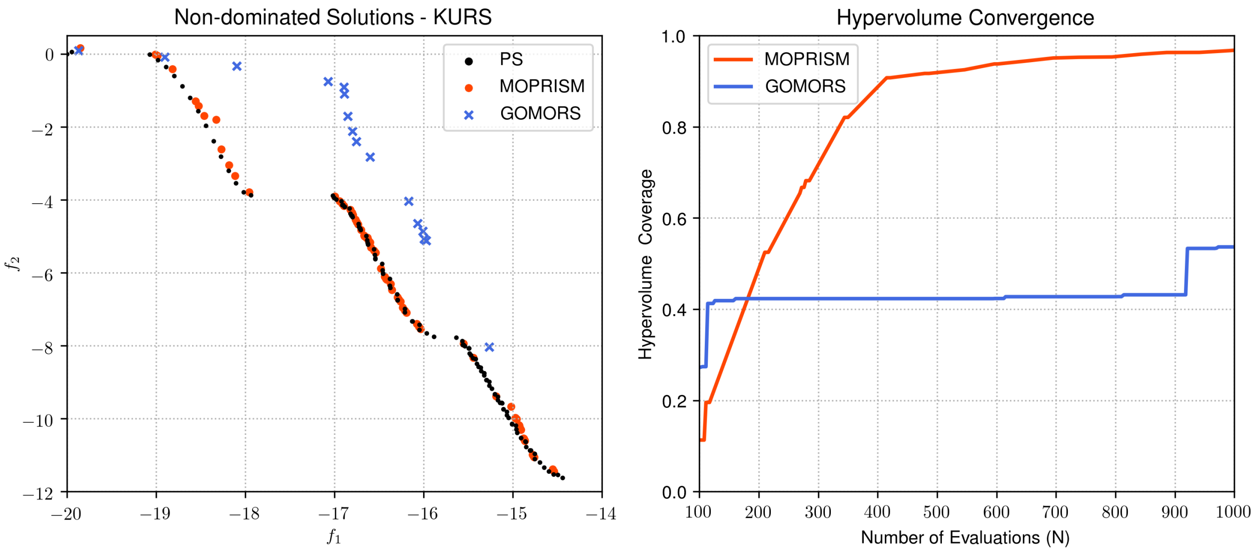

Figure V-1 *Compare MOPRISM and GOMORS results on KURS with 1000 expensive evaluations*

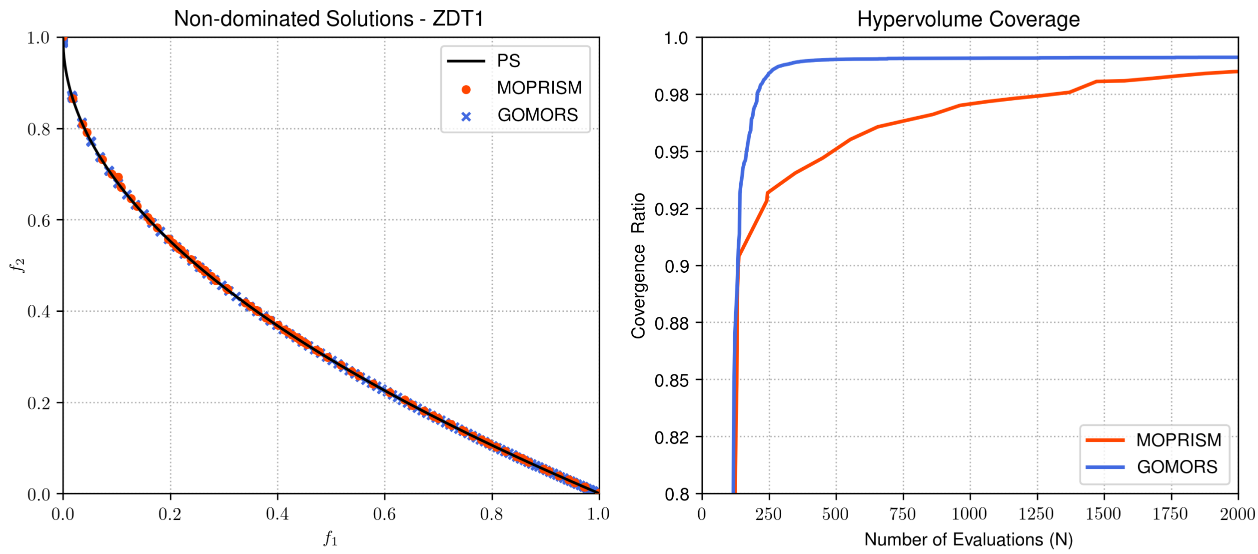

Figure V-2 *Compare MOPRISM and GOMORS results on ZDT-1 with 2000 expensive evaluations*

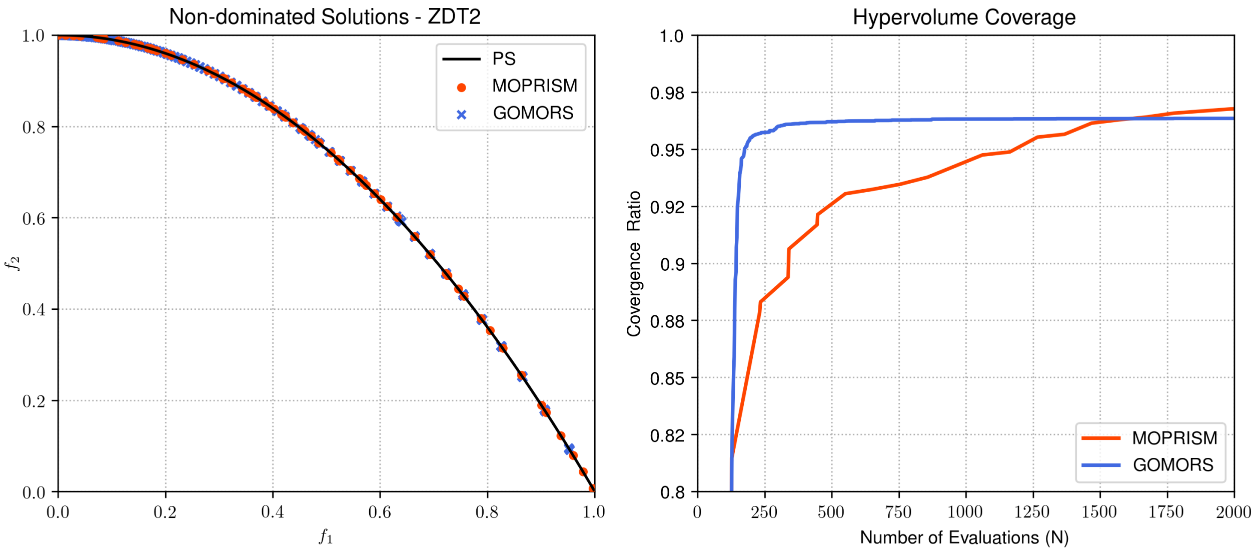

Figure V-3 *Compare MOPRISM and GOMORS results on ZDT-2 with 2000 expensive evaluations*

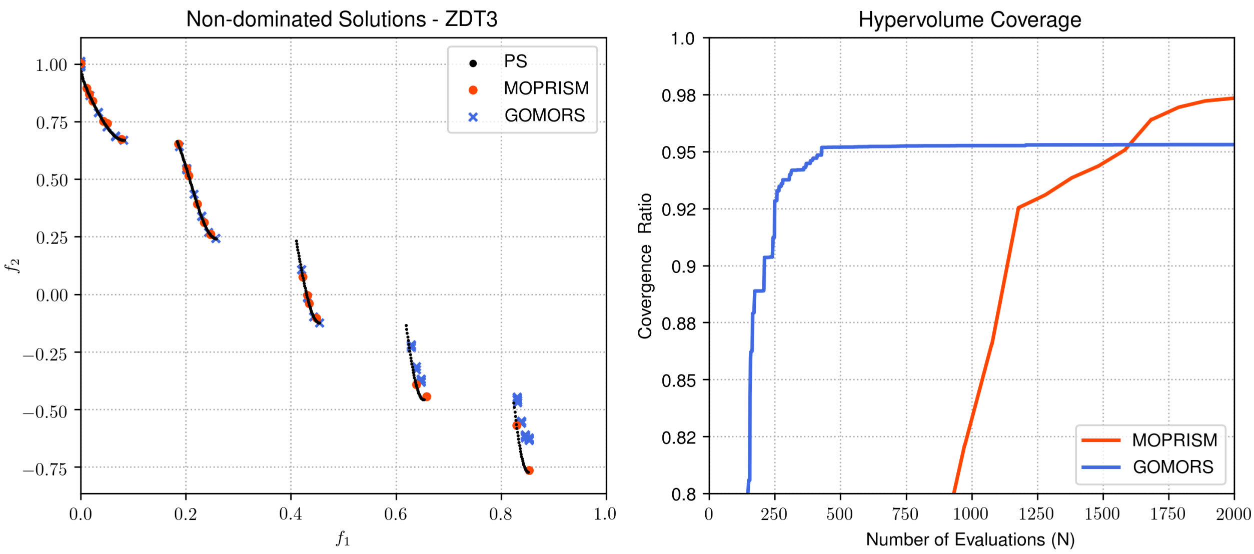

Figure V-4 *Compare MOPRISM and GOMORS results on ZDT-3 with 2000 expensive evaluations*

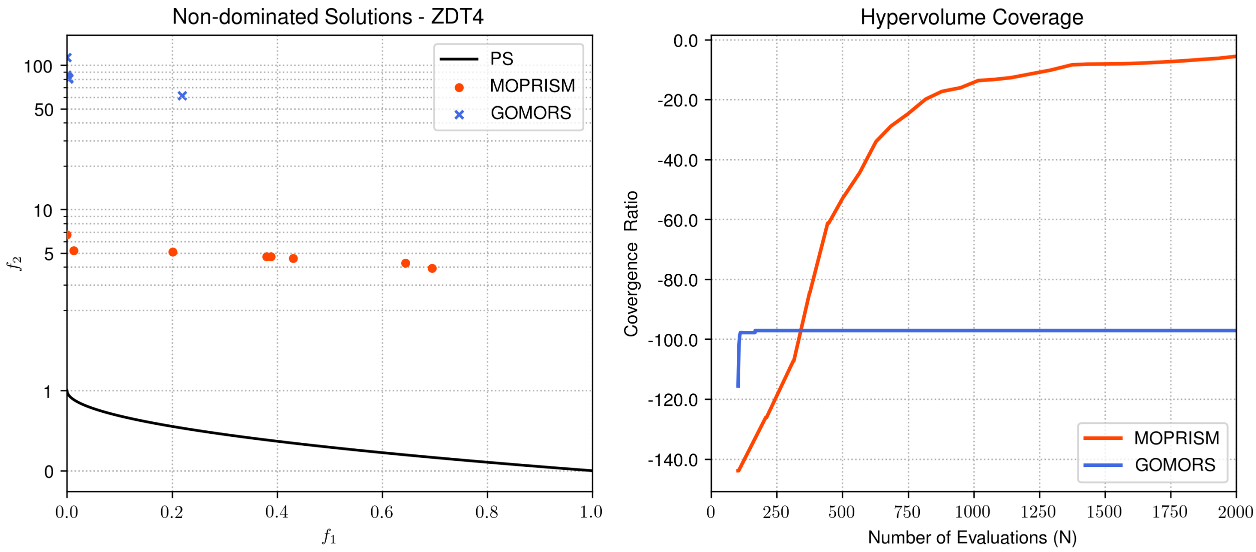

Figure V-5 *Compare MOPRISM and GOMORS results on ZDT-4 with 2000 expensive evaluations*

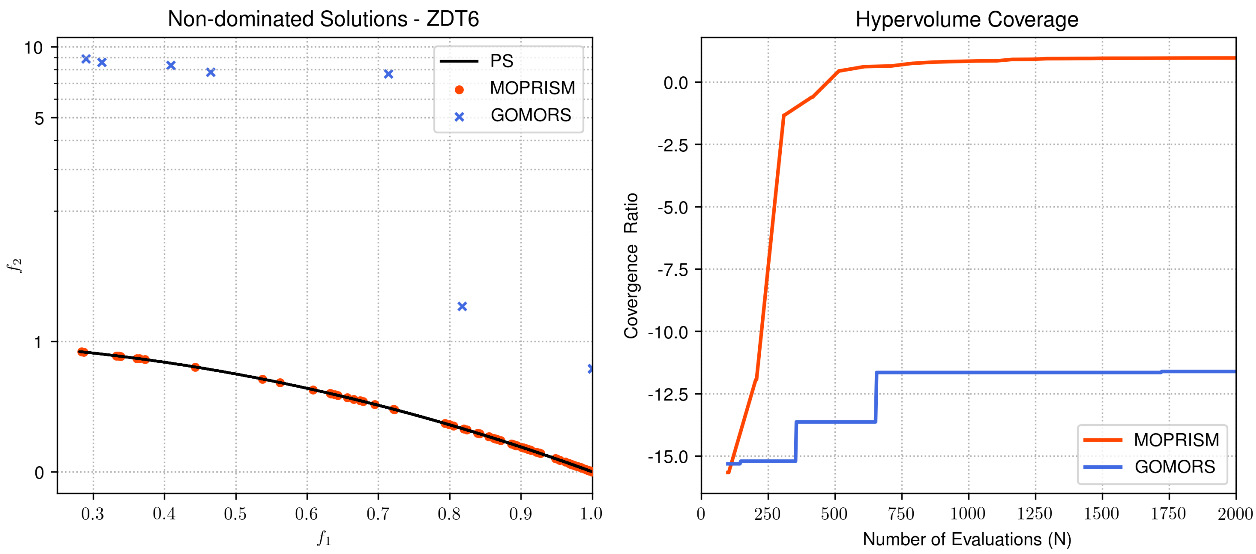

Figure V-6 *Compare MOPRISM and GOMORS results on ZDT-6 with 2000 expensive evaluations*

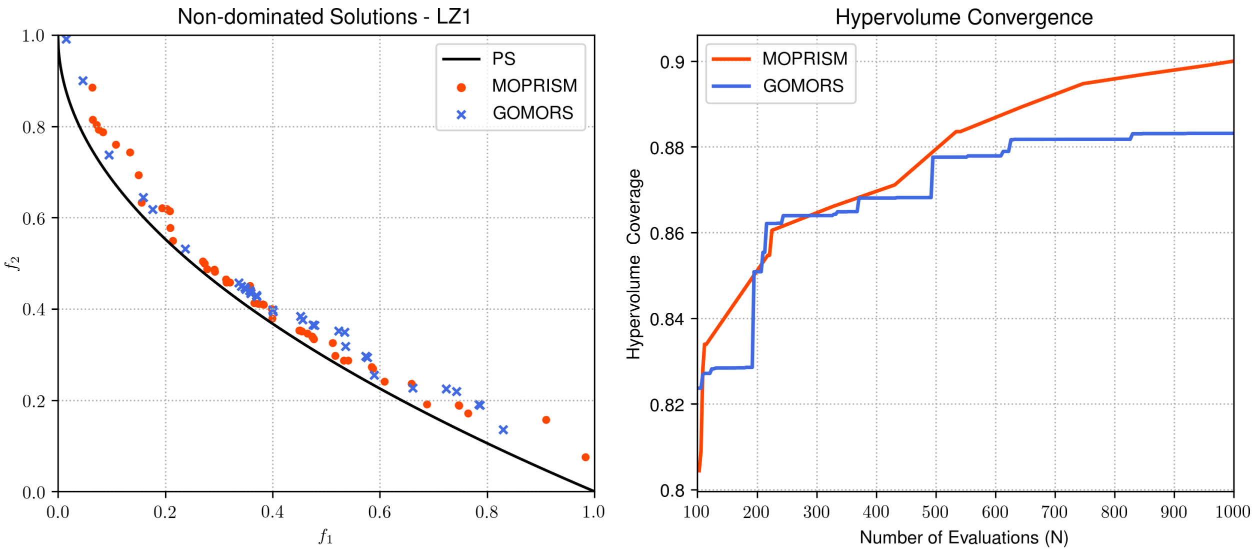

Figure V-7 *Compare MOPRISM and GOMORS results on LZ-1 with 1000 expensive evaluations*

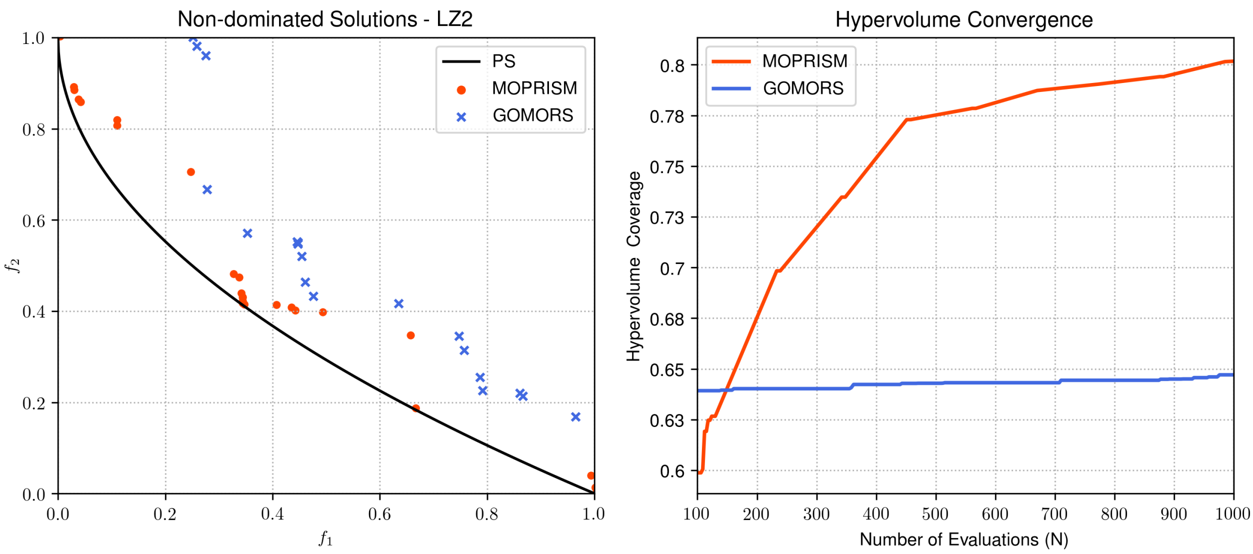

Figure V-8 *Compare MOPRISM and GOMORS results on LZ-2 with 1000 expensive evaluations*

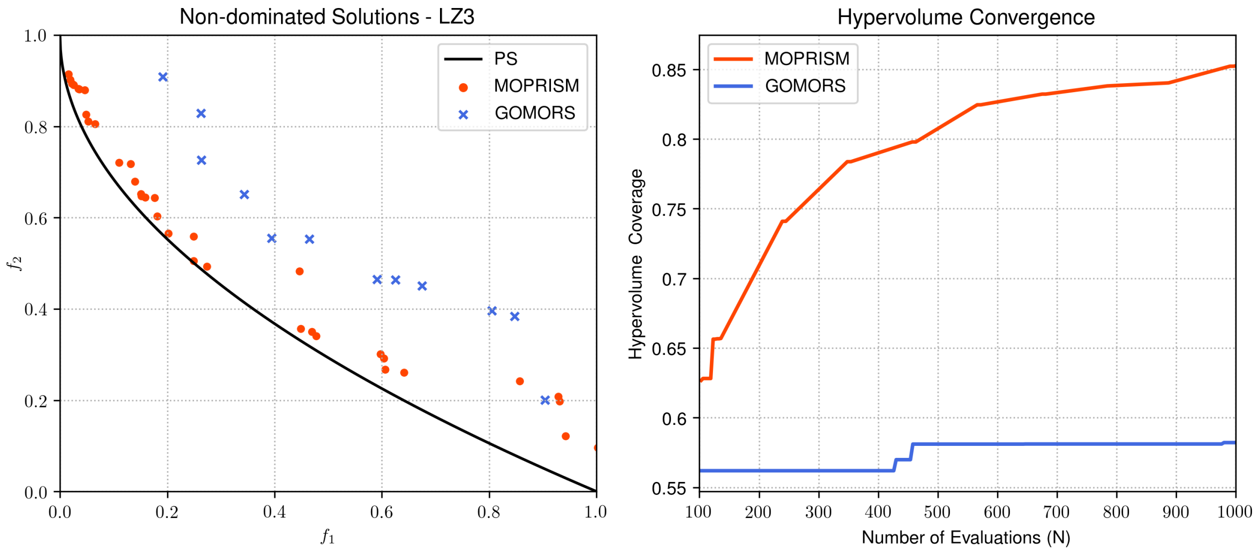

Figure V-9 *Compare MOPRISM and GOMORS results on LZ-3 with 1000 expensive evaluations*

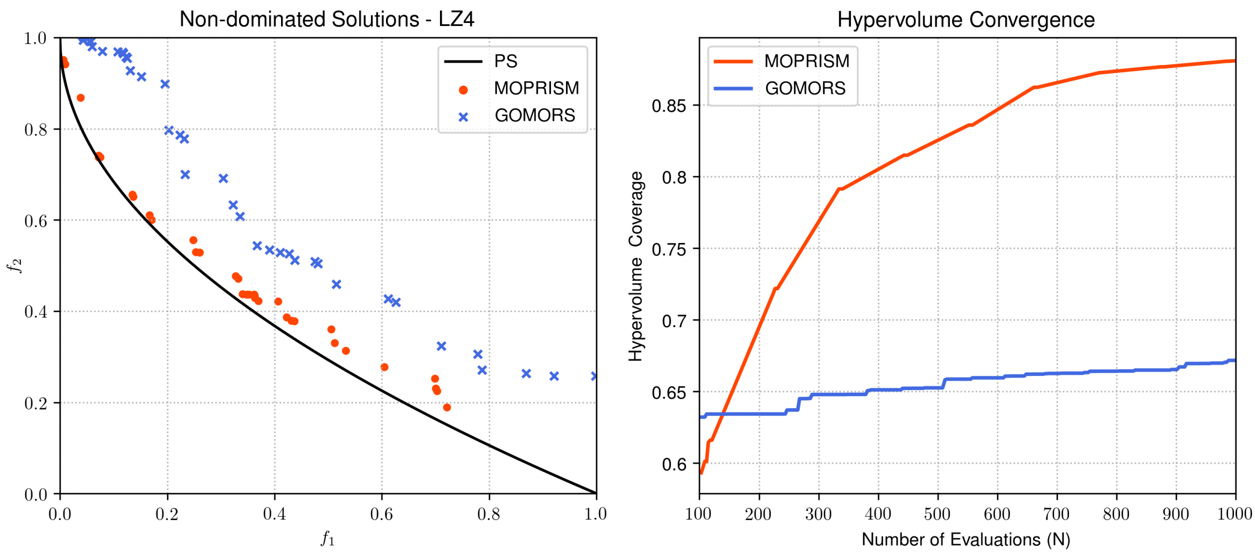

Figure V-10 *Compare MOPRISM and GOMORS results on LZ-4 with 1000 expensive evaluations*

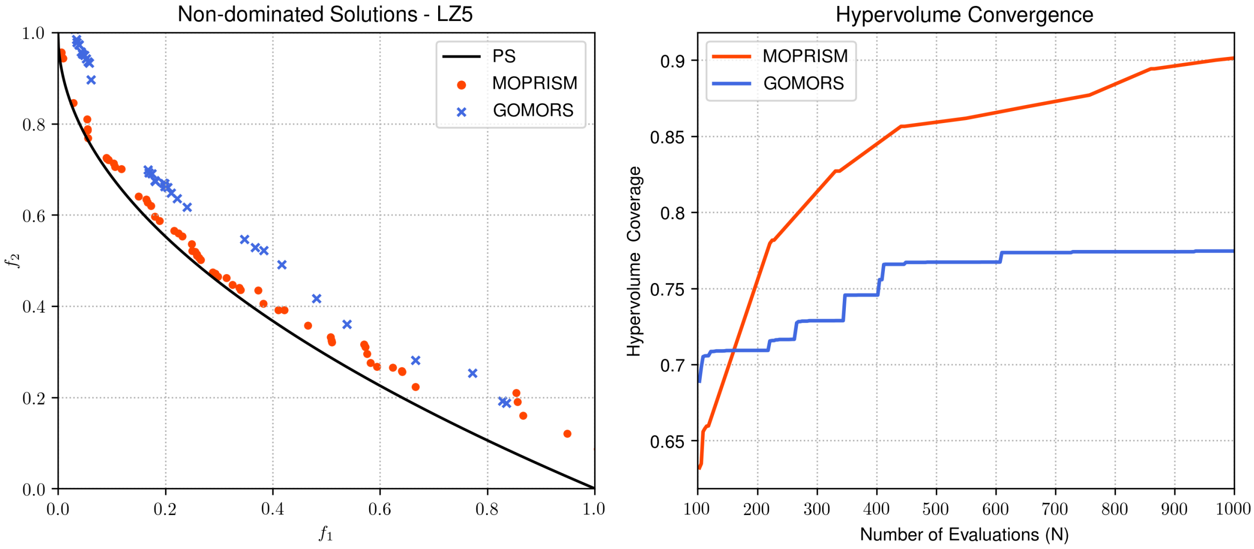

Figure V-11 *Compare MOPRISM and GOMORS results on LZ-5 with 1000 expensive evaluations*

V.2 Hanoi Results

Figure V-12 shows the optimization results of the Hanoi problem. The points in black are the solutions obtained by a naive MOEA using *IP* after 60,000 expensive evaluations. The design point on the top right of the non-dominated set indicates a construction cost of 6.13M$, with a *wDSR* of 1. This design is better than the optimal solution proposed in Fujiwara and Khang (1990) [6] with a construction cost of 6.32M$, but slightly worse than the one proposed by Savic and Walters (1997) [25] with aconstruction cost of 6.07M$. Consequently, we can assume that those solutions are Pareto optimal to the problem and then use them as a reference set to calculate the maximum possible hypervolume and benchmark MOPRISM in the Hanoi problem.

In the experiment, MOPRISM fails to solve the problem within 3000 expensive evaluations. When MOPRISM applies *IP*, the surrogate has difficulty to learn the model's behavior mapping from the discrete, categorical inputs to real-valued outputs. Although we tried, before feeding the input classes to the surrogate, to convert them into real-valued pipe diameters, the results turned out to be no better than using integer inputs directly. On the right in Figiure V-12, using real number variables seems to work slightly better than *IP*. But after relaxing those solutions to realistic, integer solutions, the problem remains unsolved.

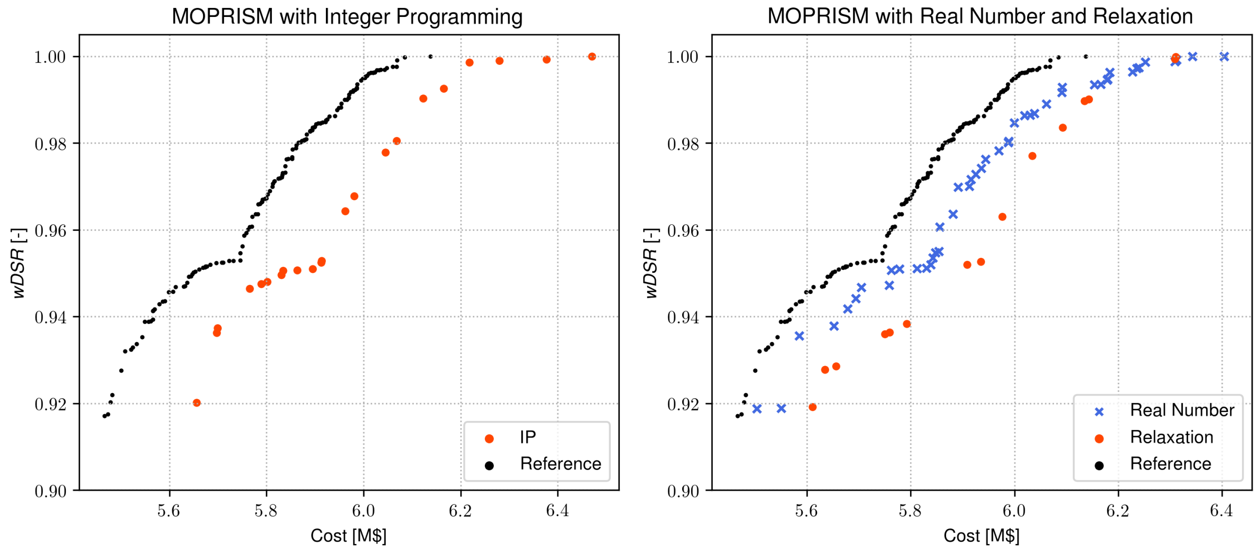

Figure V-12 *Non-dominated solutions to Hanoi problem obtained by MOPRISM with both Integer Programming (IP) and real number desicion variables combined with relaxation with 3000 expensive evaluations*

VI CONCLUSION & DISCUSSION

The new MOPRISM tends to be more promising to solve complex problems compared to GOMORS and possibly other surrogate-assisted MOEAs. As stated in section IV, the "deadlock" detection and "progressive improvement" techniques are the essence of MOPRISM. Without them, MOPRISM is just like some other urrogate-assisted MOEAs which have a sampler and local search. However, the "deadlock" and "progressive improvement" operate fully independently in MOPRISM. So, it is possible to integrate those techniques into other surrogate-assisted MOEAs as a complement to improve their robustness.

MOPRISM also has its own limitations. In section V, we see that MOPRISM fails to solve ZDT-4 and the Hanoi problem. The former is a extreme multi-modal problem and the latter uses categorical inputs. These two kinds of problems are still considered difficult to solve today in the field of global optimization. A surrogate-free, robust MOEA is capable to solve these problems with tens of thousands of evaluations. But to a surrogate-assisted MOEA, or other metamodel-embedded heuristic methods, is way too difficult. Better techniques remain to be explored.

The accuracy of the surrogate is also crucial to MOPRISM. In our study, we did not stress on the training of the surrogate, and instead of using a sophisticated surrogate that requires a lot of parameterizations (like an ANN), we prefer to use some "simple", fine-tuning-free, models such as RBFs and Gaussian Processes. Further more, limited number of expensive evaluations means less training data. To train models with limited data is itself a difficult problem in machine learning. So, to find more effective and efficient way to train the surrogate can be a good direction for future studies and research.

Bibliography

[1] Ram Bhushan Agrawal, K Deb, and RB Agrawal. Simulated binary crossover for continuous search space. *Complex systems*, 9(2):115–148, 1995.

[2] Taimoor Akhtar and Christine A. Shoemaker. Multi objective optimization of computationally expensive multi-modal functions with rbf surrogates and multi-rule selection. *Journal of Global Optimization*, 64(1):17–32, Jan 2016.

[3] Kalyanmoy Deb and Deb Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[4] Alan Díaz-Manríquez, Gregorio Toscano-Pulido, and Wilfrido Gómez-Flores. On the selection of surrogate models in evolutionary optimization algorithms. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2155–2162. IEEE, 2011.

[5] Marshall L Fisher. The lagrangian relaxation method for solving integer programming problems. *Management science*, 27(1):1–18, 1981.

[6] Okitsugu Fujiwara and Do Ba Khang. A two-phase decomposition method for optimal design of looped water distribution networks. *Water resources research*, 26(4):539–549, 1990.

[7] Zong Woo Geem. Optimal cost design of water distribution networks using harmony search. *Engineering Optimization*, 38(03):259–277, 2006.

[8] Fred Glover. Heuristics for integer programming using surrogate constraints. *Decision sciences*, 8(1):156–166, 1977.

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

[10] Joshua Knowles. Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

[11] Frank Kursawe. A variant of evolution strategies for vector optimization. In *Proceedings of*

the 1st Workshop on Parallel Problem Solving from Nature, PPSN I, pages 193–197, Berlin, Heidelberg, 1991. Springer-Verlag.

[12] Jonathan Law and John Smullen. *A Dictionary of Finance and Banking*. Oxford Quick Reference. OUP Oxford, 2018.

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

[14] Holger R Maier, Zoran Kapelan, Joseph Kasprzyk, Joshua Kollat, L Shawn Matott, MC Cunha, Graeme C Dandy, Matthew S Gibbs, Edward Keedwell, A Marchi, et al. Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions. *Environmental Modelling & Software*, 62:271–299, 2014.

[15] Saúl Zapotecas Martínez and Carlos A Coello Coello. A memetic algorithm with non gradient-based local search assisted by a meta-model. In *International Conference on Parallel Problem Solving from Nature*, pages 576–585. Springer, 2010.

[16] Barbara Spang Minsker and Christine A Shoemaker. Dynamic optimal control of in-situ bioremediation of ground water. *Journal of water resources planning and management*, 124(3):149–161, 1998.

[17] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, first edition, 1997.

[18] George L Nemhauser and Laurence A Wolsey. Integer programming and combinatorial optimization. *Wiley, Chichester. GL Nemhauser, MWP Savelsbergh, GS Sigismondi (1992). Constraint Classification for Mixed Integer Programming Formulations. COAL Bulletin*, 20:8–12, 1988.

[19] John Nicklow, Patrick Reed, Dragan Savic, Tibebe Dessalegne, Laura Harrell, Amy Chan-Hilton, Mohammad Karamouz, Barbara Minsker, Avi Ostfeld, Abhishek Singh, et al. State of the art for genetic algorithms and beyond in water resources planning and management. *J. Water Resour. Plan. Manag. ASCE*, 136(4):412–432, 2010.

[20] Saman Razavi, Bryan A Tolson, and Donald H Burn. Review of surrogate modeling in water resources. *Water Resources Research*, 48(7), 2012.

[21] Juan Reca, Juan Martínez, and Rafael López. A hybrid water distribution networks design optimization method based on a search space reduction approach and a genetic algorithm. *Water*, 9(11):845, 2017.

[22] Edgar Reehuis. Multiobjective robust optimization of water distribution networks. *A Masters thesis submitted to Leiden Institute of Advanced Computer Science (LIACS), Leiden University, Niels Bohrweg*, 1:2333, 2010.

[23] Rommel G Regis and Christine A Shoemaker. Constrained global optimization of expensive black box functions using radial basis functions. *Journal of Global optimization*, 31(1):153–171, 2005.

[24] Alejandro Rosales-Pérez, Carlos A Coello Coello, Jesus A Gonzalez, Carlos A Reyes-Garcia, and Hugo Jair Escalante. A hybrid surrogate-based approach for evolutionary multi-objective optimization. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 2548–2555. IEEE, 2013.

[25] Dragan A Savic and Godfrey A Walters. Genetic algorithms for least-cost design of water distribution networks. *Journal of water resources planning and management*, 123(2):67–77, 1997.

[26] Pradnya A Vikhar. Evolutionary algorithms: A critical review and its future prospects. In *Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), 2016 International Conference on*, pages 261–265. IEEE, 2016.

[27] Stefan M Wild, Rommel G Regis, and Christine A Shoemaker. Orbit: Optimization by radial basis function interpolation in trust-regions. *SIAM Journal on Scientific Computing*, 30(6):3197–3219, 2008.

[28] Stefan M Wild and Christine Shoemaker. Global convergence of radial basis function trust-region algorithms for derivative-free optimization. *SIAM Review*, 55(2):349–371, 2013.

[29] Zheng. Xu, Carlos A. Velez, Francesca Pianosi, and Dimitri P. Solomatine. Multi-objective opti-

mization of urban wastewater systems design by progressive improvement of surrogate machine learning models. Unpublished Manuscript, 2010.

[30] Shengquan Yan and Barbara Minsker. Optimal groundwater remediation design using trust region based metamodels within a genetic algorithm, 2005. Manuscript.

[31] Shengquan Yan and Barbara Minsker. Optimal groundwater remediation design using an adaptive neural network genetic algorithm. *Water Resources Research*, 42(5), 2006.

[32] Shengquan Yan and Barbara Minsker. Applying dynamic surrogate models in noisy genetic algorithms to optimize groundwater remediation designs. *Journal of water resources planning and management*, 137(3):284–292, 2010.

[33] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, 2000.

Appendices

A Hanoi Data

Table I-1 Hanoi Node Data

ID	Elevation(m)	Demand(m^3/s)
1	100	(Reservoir)
2	30	247.22
3	30	236.11
4	30	36.11
5	30	201.39
6	30	279.17
7	30	375
8	30	152.78
9	30	145.83
10	30	145.83
11	30	138.89
12	30	155.56
13	30	261.11
14	30	170.83
15	30	77.78
16	30	86.11
17	30	240.28
18	30	373.61
19	30	16.67
20	30	354.17
21	30	258.33
22	30	134.72
23	30	290.28
24	30	227.78
25	30	47.22
26	30	250
27	30	102.78
28	30	80.56
29	30	100
30	30	100
31	30	29.17
32	30	223.61

Table I-2 Hanoi Pipe Data

ID	From Node	To Node	Length(m)
1	1	2	100
2	2	3	1350
3	3	4	900
4	4	5	1150
5	5	6	1450
6	6	7	450
7	7	8	850
8	8	9	850
9	9	10	800
10	10	11	950
11	11	12	1200
12	12	13	3500
13	10	14	800
14	14	15	500
15	15	16	550
16	17	16	2730
17	17	18	1750
18	18	19	800
19	19	3	400
20	3	20	2200
21	20	21	1500
22	21	22	500
23	20	23	2650
24	23	24	1230
25	24	25	1300
26	26	25	850
27	27	26	300
28	16	27	750
29	23	28	1500
30	28	29	2000
31	29	30	1600
32	30	31	150
33	32	31	860
34	25	32	950

The unit price of the pipes is calculated by: $UnitPrice(D) = 1.1 \times inch(D)^{1.5}$ [$]

B Benchmarking Results with Embedded NSGA-II

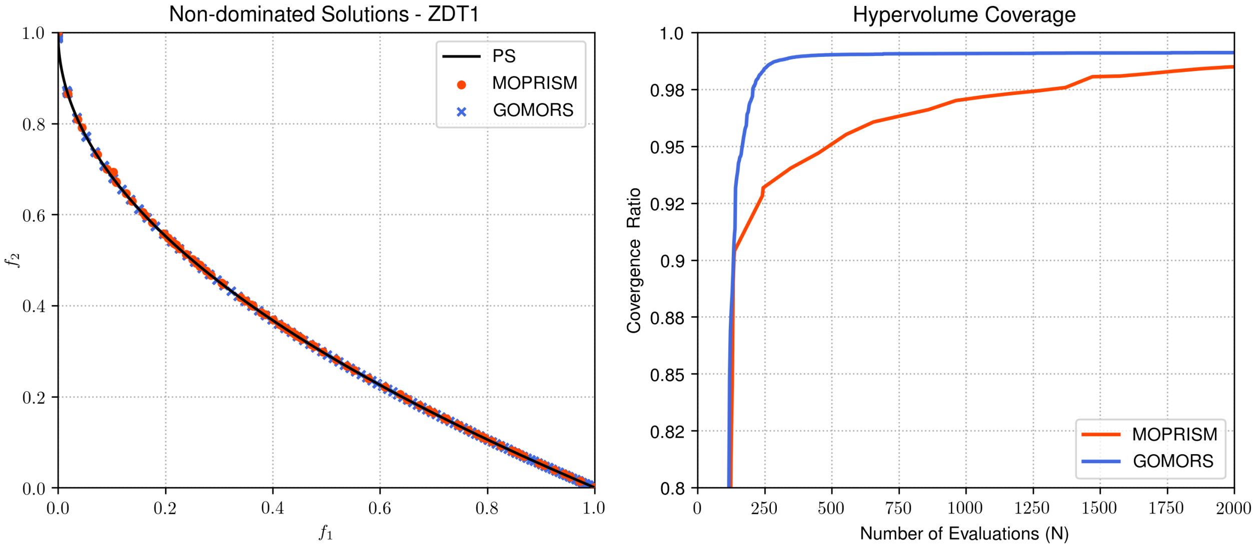

Figure II-1 *Compare MOPRISM and GOMORS results on ZDT-1 with 2000 expensive evaluations*

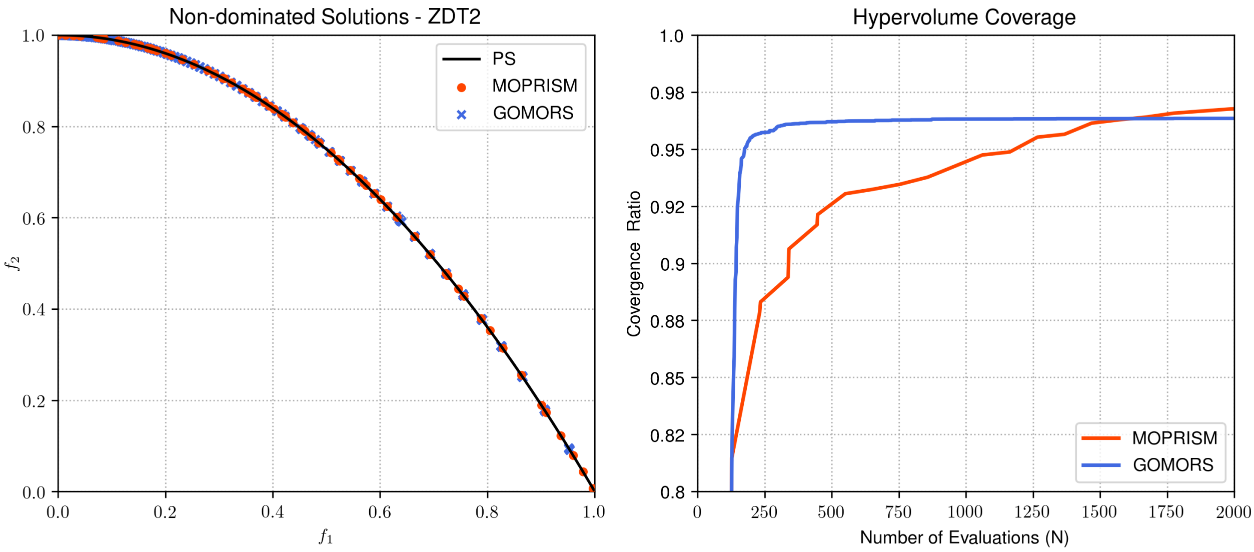

Figure II-2 *Compare MOPRISM and GOMORS results on ZDT-2 with 2000 expensive evaluations*

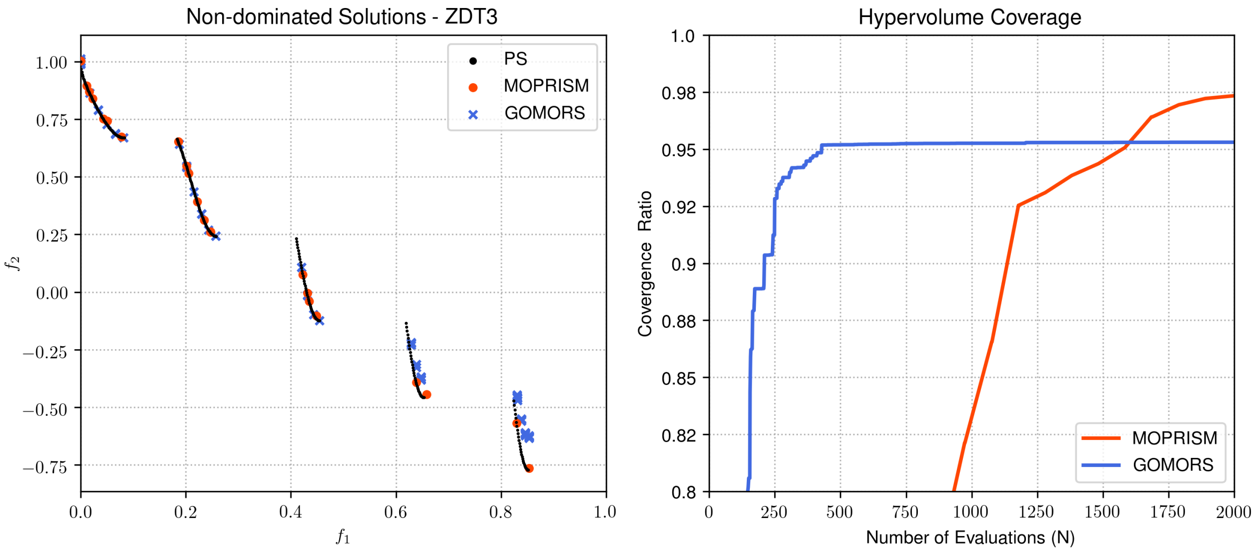

Figure II-3 *Compare MOPRISM and GOMORS results on ZDT-3 with 2000 expensive evaluations*

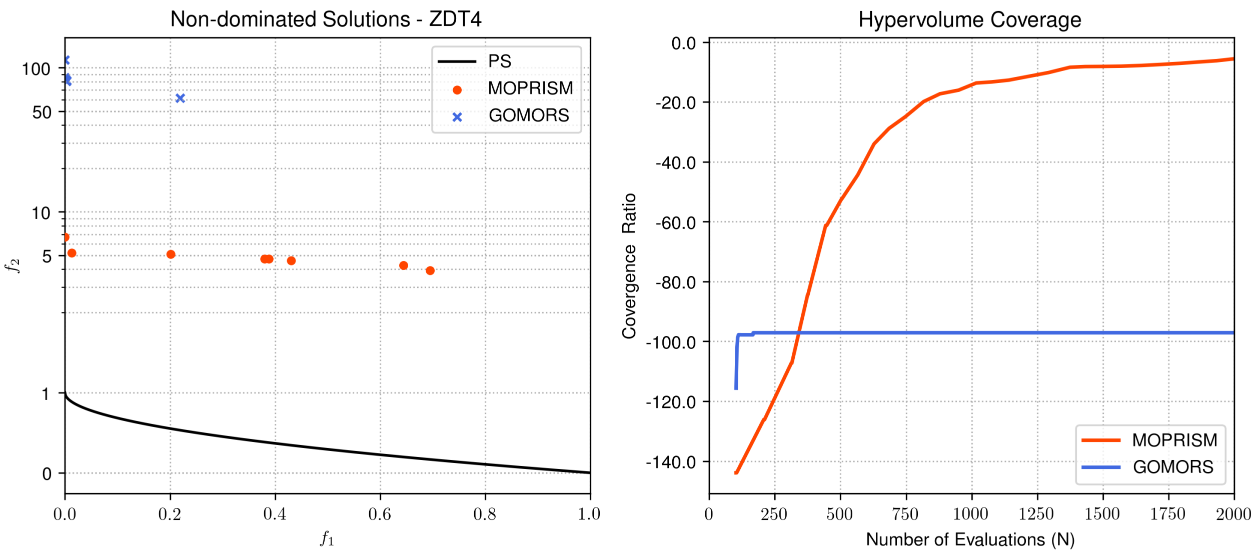

Figure II-4 *Compare MOPRISM and GOMORS results on ZDT-4 with 2000 expensive evaluations*

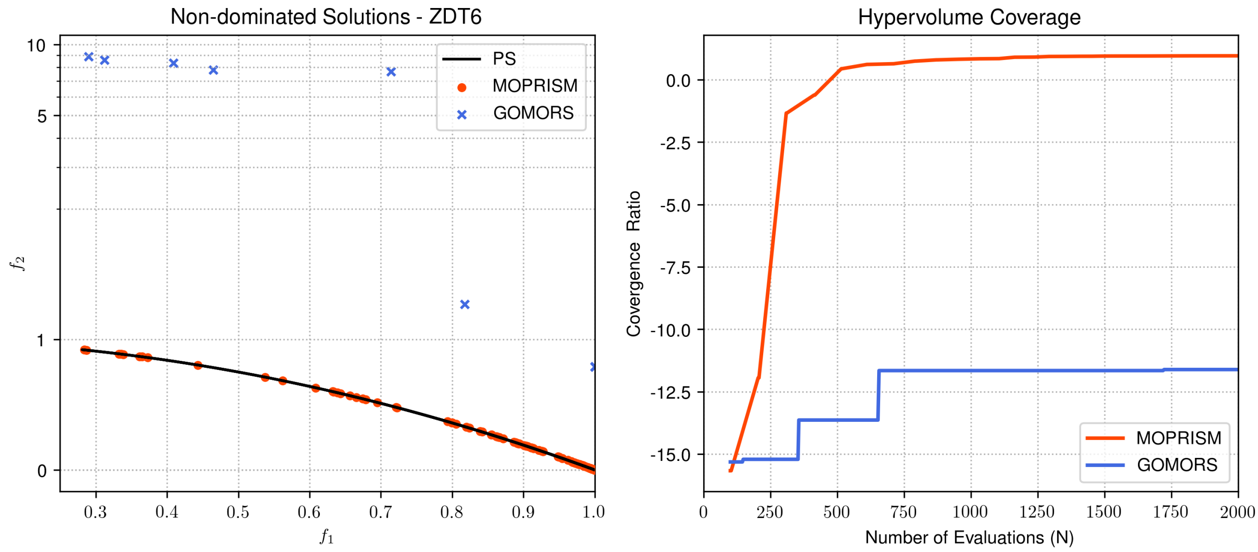

Figure II-5 *Compare MOPRISM and GOMORS results on ZDT-6 with 2000 expensive evaluations*

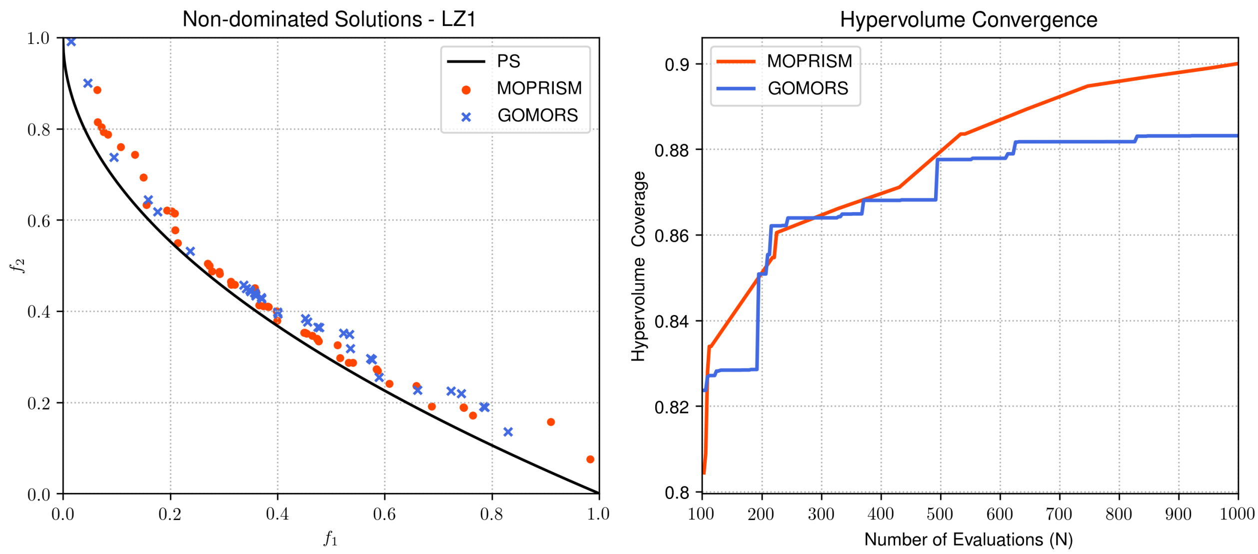

Figure II-6 *Compare MOPRISM and GOMORS results on LZ-1 with 1000 expensive evaluations*

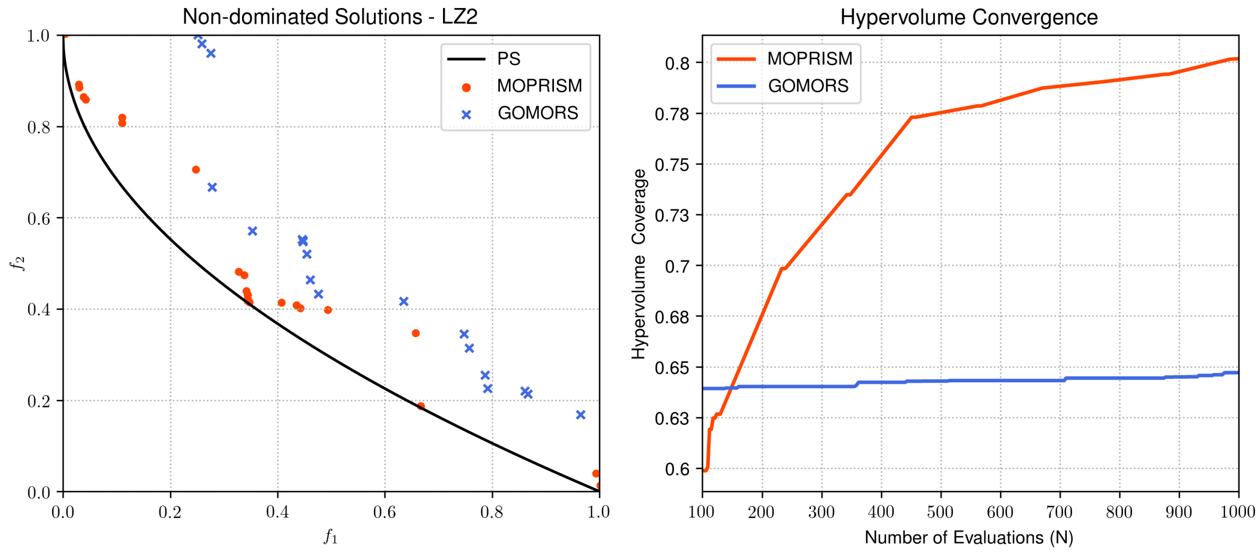

Figure II-7 *Compare MOPRISM and GOMORS results on LZ-2 with 1000 expensive evaluations*

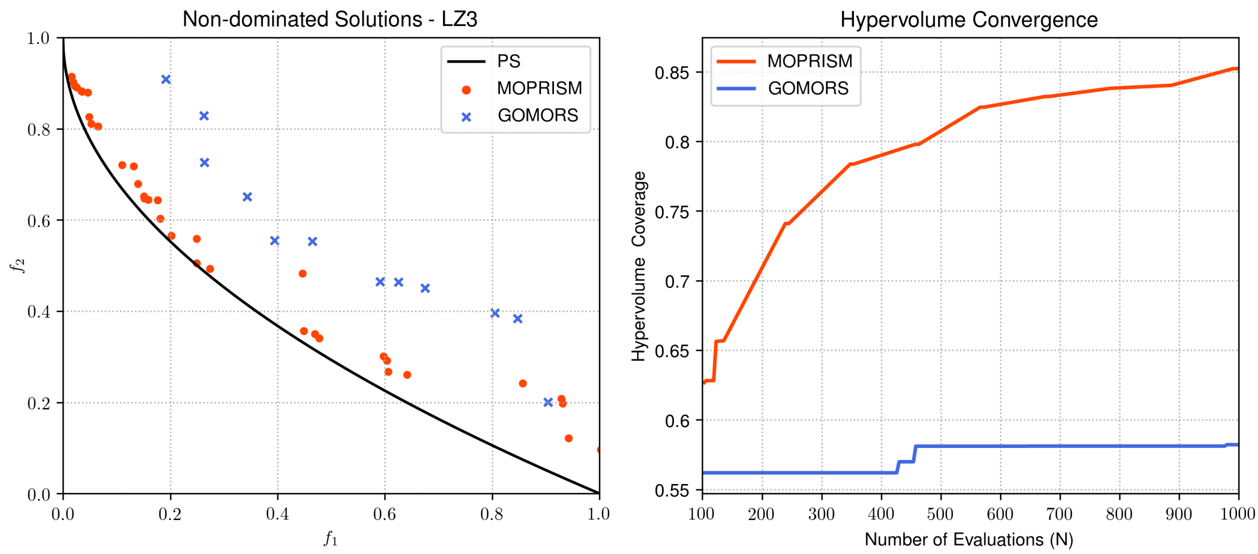

Figure II-8 *Compare MOPRISM and GOMORS results on LZ-3 with 1000 expensive evaluations*

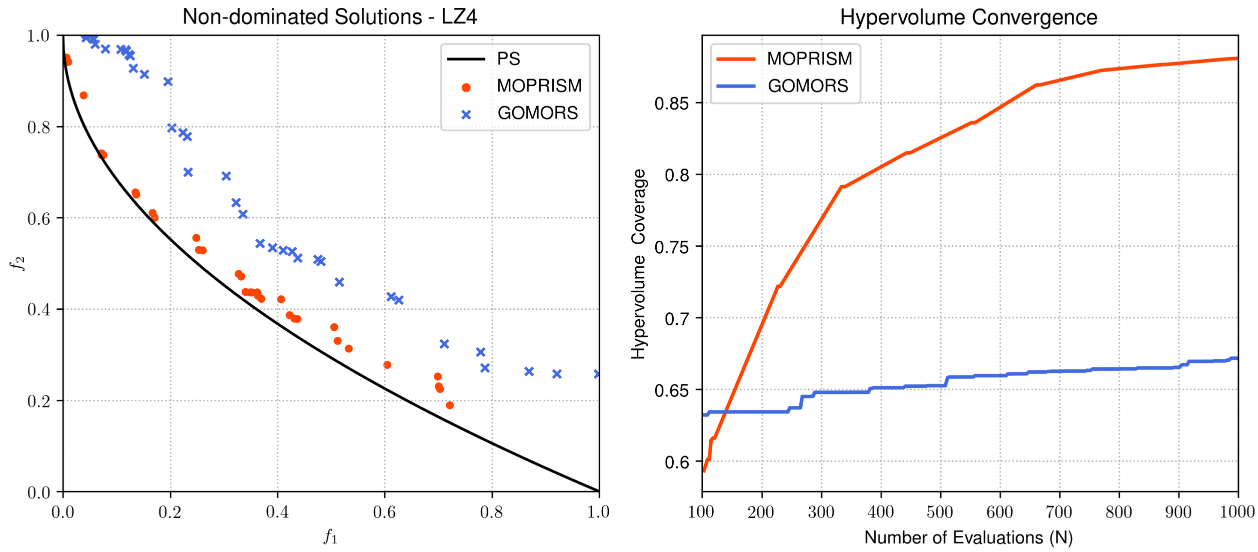

Figure II-9 *Compare MOPRISM and GOMORS results on LZ-4 with 1000 expensive evaluations*

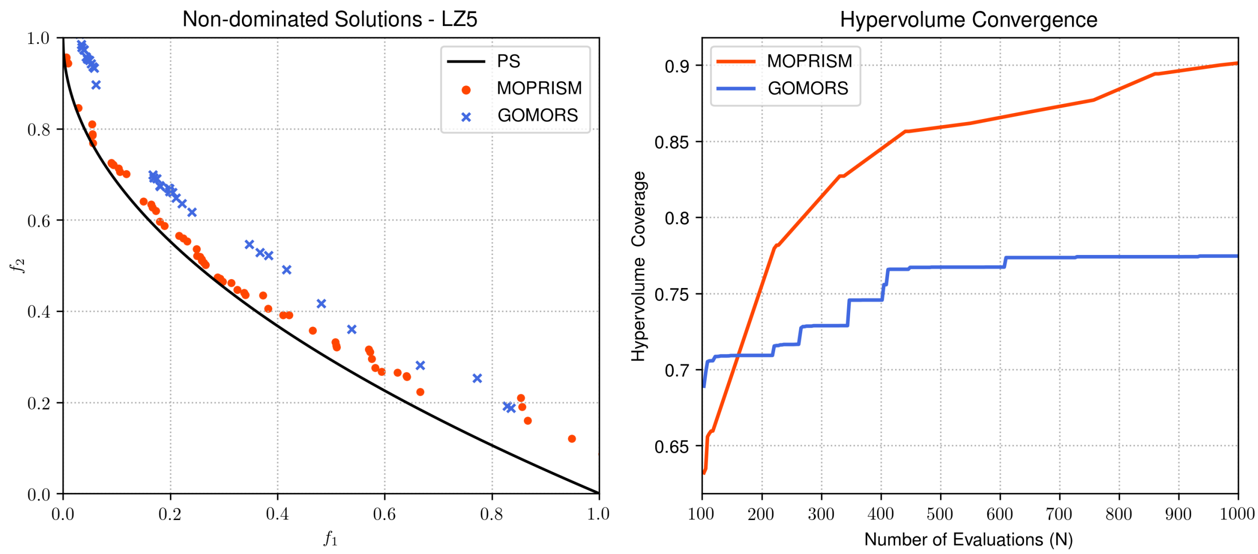

Figure II-10 *Compare MOPRISM and GOMORS results on LZ-5 with 1000 expensive evaluations*