A Mono Surrogate for Multiobjective Optimization

Ilya Loshchilov, Marc Schoenauer, Michèle Sebag

A Mono Surrogate for Multiobjective Optimization

Ilya Loshchilov
TAO, INRIA Saclay
U. Paris Sud, F-91405 Orsay

Marc Schoenauer
TAO, INRIA Saclay
U. Paris Sud, F-91405 Orsay

Michèle Sebag
CNRS, LRI UMR 8623
U. Paris Sud, F-91405 Orsay

firstname.lastname@inria.fr

ABSTRACT

Most surrogate approaches to multi-objective optimization build a surrogate model for each objective. These surrogates can be used inside a classical Evolutionary Multiobjective Optimization Algorithm (EMOA) *in lieu* of the actual objectives, without modifying the underlying EMOA; or to filter out points that the models predict to be uninteresting. In contrast, the proposed approach aims at building a global surrogate model defined on the decision space and tightly characterizing the current Pareto set and the dominated region, in order to speed up the evolution progress toward the true Pareto set. This surrogate model is specified by combining a One-class Support Vector Machine (SVMs) to characterize the dominated points, and a Regression SVM to clamp the Pareto front on a single value. The resulting surrogate model is then used within state-of-the-art EMOAs to pre-screen the individuals generated by application of standard variation operators. Empirical validation on classical MOO benchmark problems shows a significant reduction of the number of evaluations of the actual objective functions.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artificial Intelligence Problem Solving, Control Methods, and Search

General Terms

Algorithms

Keywords

Multiobjective Optimization, Surrogate Models, Support Vector Machine

1. INTRODUCTION

In the classical optimization framework, surrogate approaches (aka Surface Response Methods) have been proposed decades ago to deal with computationally expensive objective functions, and decrease the overall optimization cost. Surrogate optimization proceeds by building an approximation of the objective function, referred to as *surrogate model* or meta-model; the optimization algorithm then uses the meta-model *in lieu* of the actual objective function. Of course, the meta-model must be regularly updated as the search proceeds and new information about the search space is gathered; considering an inaccurate meta-model for long would mislead the search and miss the optima of the actual objective function.

Surrogate methods have received a particular attention in the realm of Evolutionary Algorithms (EAs), all the more so as EAs are known to require a high number of objective function computations. Several types of meta-models have been considered, ranging from quadratic models, neural networks, Regression Support Vector Machines to kriging or Gaussian Processes; the interested reader is referred to [7] for a survey of surrogate evolutionary optimization. Meta-models can aim at either a global approximation of the objective function, or a local one, focusing on the neighborhood of the best current individuals. The meta-model can be used to replace the objective function for a given number of generations; it can be used to generate new individuals (the optima of the meta-model) from scratch; and it can also be used to filter out unpromising offspring. A key issue in surrogate evolutionary optimization is how and when the meta-model is updated. The exact objective function can be computed for the top-ranked individuals in each generation, or the individuals with best Expected Improvement after the kriging meta-model. The update can proceed by revising the model (e.g., a Neural Net), or relearning it from scratch (e.g., a Support Vector Machine (SVM)).

Unsurprisingly, Evolutionary Multi-Objective (EMO) algorithms facing even more severe computational issues than single-objective optimization, the use of meta-models has been intensively investigated in the EMO literature (see [8] for a comprehensive survey). Most approaches carry over the single-objective surrogate approach, learning one meta-model for each objective and embedding the meta-models within a standard EMO with little modification [14], or within a memetic algorithm for local seach improvement [15]. Meta-models can also be used to rank and filter out offspring (pre-screening mode), according to Pareto-related indicators like the hypervolume [5], or a weighted sum of the objectives, or a goal-oriented direction.

Surrogate approaches generally consider the decision space. A notable exception, Yu et al. aim at characterizing the re-

GECCO'10, July 7–11, 2010, Portland, Oregon, USA.

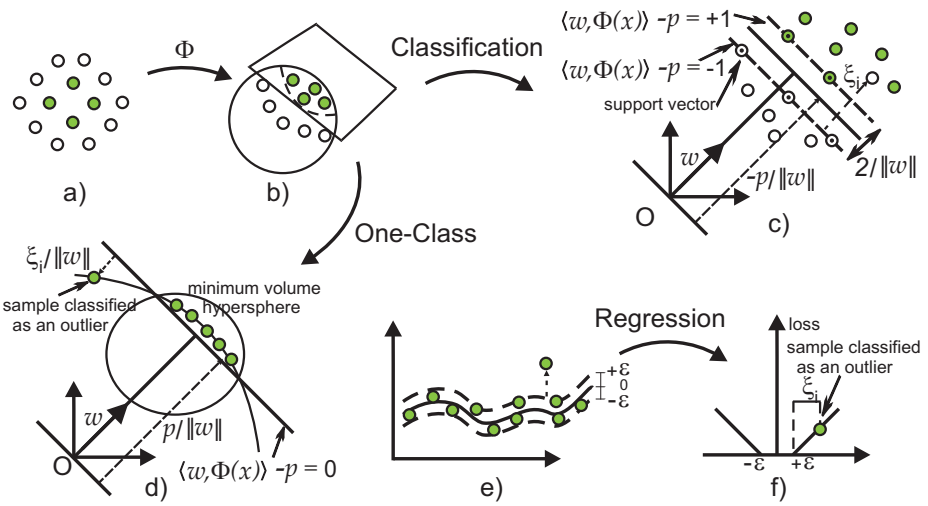


Figure 1: SVMs rely on the kernel trick: map the original representation (a) onto a high-dimensional space (b), hopefully making the learning problem linearly separable. SVMs achieve classification (c) or regression (e,f). They can also be applied to characterize a dataset (One-Class SVM: d).

gion of the objective space which has already been visited [16]. The rationale for this approach, based on One-Class SVM [11], is that the envelope of the visited region excludes the Pareto front. Unfortunately, the Pareto front in the objective space does not tell much about the Pareto set in decision space[1], and can hardly be used to guide the EMO search.

The presented work aims at building a global surrogate model in decision space, characterizing whether an individual belongs to i/ the current Pareto set; or ii/ the dominated region; or iii/ the rest of the decision space (not yet visited, and containing the *true* Pareto set). This surrogate model, providing an aggregated perspective on all objective functions simultaneously, is used to guide the search in the vicinity of the current Pareto set, and speed up the population move toward the true Pareto set. This Aggregated Surrogate Model (ASM) is constructed by combining ideas from Regression and One-class SVMs.

Section 2 describes the ASM problem and its resolution. Section 3 gives an overview of the EMO algorithm using ASM, referred to as Pareto-SVM. Section 4 analyzes the experimental validation of Pareto-SVM on different classical benchmark functions. Finally, Section 5 discusses our contributions and concludes the paper.

2. PARETO SUPPORT VECTOR MACHINE

This section formalizes the ASM as a constrained quadratic optimization problem and describes its resolution. Let us briefly remind the basics of Support Vector Machines (SVMs), referring the interested reader to [13] for a comprehensive presentation.

2.1 Support Vector Machines

SVMs, originally developed for pattern classification (Fig. 1.(c)), have been extended to regression (Fig. 1.(e,f)) and later on to the characterization of a sample (one-class) dataset (Fig. 1.(d)).

Considering a two-class training set $\mathcal{E} = \{(x_i, y_i), i = 1 \dots \ell, x_i \in X = \mathbb{R}^n, y_i \in \{-1, 1\}\}$, classification SVM solves the following primal problem:

$$\underset{\{w,\ \rho,\ \xi\}}{\text{Minimize}}\ \frac{1}{2}||w||^2 + C\sum_{i=1}^{\ell}\xi_i$$

subject to $y_i(< w, x_i > -\rho) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1 \dots \ell$ where w is the normal to the separating hyperplane, its Euclidean norm $||w||$ is the minimal margin to be maximized (Fig. 1.(c)), ξ are the slack variables introduced to account for the non-separable case, and constant $C > 0$ determines the trade-off between margin maximization and training error minimization.

Associating to each above constraint a Lagrange multiplier α_i, the primal problem defines a dual problem which is a quadratic optimization problem:

$$\underset{\alpha}{\text{Maximize}} \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j y_i y_j \ < x_i, x_j >$$

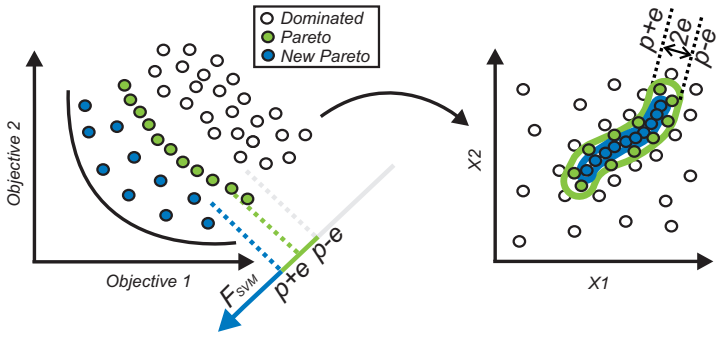
subject to $\quad 0 \leq \alpha_i < C$ for $i = 1 \dots \ell$ and $\sum_{i=1}^{\ell}\alpha_i y_i = 0$.

Along the above formulation, the separating hyperplane can be characterized from scalar products of the training points. The SVM extension to non-linear functional spaces relies on the so-called kernel trick, mapping instance space X (Fig. 1 (a)) onto a more expressive space (Fig. 1 (b)) referred to as feature space $\Phi(X)$. Defining the kernel as the scalar product $(K(x, x') =_{def} < \Phi(x), \Phi(x') >)$ in the feature space, a non-linear separating function can be found without mapping explicitly X onto $\Phi(X)$, by resolving:

$$\underset{\alpha}{\text{Maximize}} \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j y_i y_j \ K(x_i, x_j)$$

subject to $\quad 0 \leq \alpha_i < C$ for $i = 1 \dots \ell$ and $\sum_{i=1}^{\ell}\alpha_i y_i = 0$.

[1]Except for specific problems where the Pareto front in the objective space corresponds to a set of rectangles in the decision space.



(a) Objective space (b) Decision space

Figure 2: An *idealistic* schematic view of the Pareto front, depicting dominated points (white), current Pareto (grey) and new Pareto (black) respectively in objective and decision space.

The decision function is $f(x) = sgn(\sum_{i=1}^{\ell} y_i \alpha_i K(x_i, x) - \rho)$. One infinite-dimensional kernel which will be used in the following is the Gaussian Radial Basis kernel function (RBF), parameterized from bandwidth parameter σ:

$$K(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\sigma^2} \quad (1)$$

2.2 Rationale and Assumption

The goal of the present approach is to build a single surrogate model in the decision space, usable to drive the population toward the *true* Pareto set. This surrogate model will be learned from i/ points belonging to the current Pareto set, and ii/ dominated points.

At any given time during the EMOA run, the relative position of the Pareto set and the dominated points can be schematically depicted as follows. The situation might be simple in the objective space (Fig. 2.(a)), with the true Pareto front and the dominated region located on the two opposite sides of the current Pareto front. It can be much more intricate in the decision space; Fig. 2.(b) illustrates the case where the true Pareto set (respectively the dominated region) lies within (resp. outside) the convex hull of the current Pareto set. Further, the Pareto set can include many disjoint regions in the decision space. The assumption made in this paper is that the Pareto region includes a small number of connected components; note that this assumption holds for most classical multi-objective optimization benchmarks, (e.g. IHR1, see Fig. 3 (c) and (d)).

While ASM expectedly discriminates the Pareto set and the dominated region, a binary classification approach is ill-suited as it would not give any precise indication about where the *true* Pareto set is located. More generally, the Pareto set (true or current) and the dominated points cannot be handled in a symmetrical way: dominated points span over a subspace whereas the Pareto set should better be viewed as a manifold.

It thus comes to map all Pareto points onto a single value ρ (up to some tolerance ϵ); meanwhile, the dominated points would be mapped onto the half space $]-\infty, \rho - \epsilon[$. Such a mapping might actually provide useful indications: expectedly, points mapped onto the half space $[\rho + \epsilon, +\infty[$ would belong to the yet unexplored region, which is bound to contain the *true* Pareto set, and these points could thus be considered promising.

The above constraints on the ASM mapping can be expressed by combining the SVM-regression formulation [12] (mapping each point x onto some target value $f(x)$ up to some tolerance ϵ) and the One-class SVM [11], mapping a set of points onto a connected interval and thus characterizing the support of the underlying sample distribution. The main difference is that the target value ρ associated to the Pareto points is free in the ASM problem.

2.3 Lagrangian formulation

Let $X \subset \mathbb{R}^d$ denote the instance (decision) space. The available set of points $\{x_1 \dots x_m, x_i \in X\}$ includes the current Pareto points $\{x_1 \dots x_\ell\}$ and the dominated ones $\{x_{\ell+1}, \dots, x_m\}$. By construction, the ASM noted $\mathcal{F}$ ($\mathcal{F} : X \mapsto \mathbb{R}$) is subject to $m + \ell$ constraints:

$$\begin{array}{ll} \text{for } 1 \leq i \leq \ell & \mathcal{F}(x_i) \text{ must belong to } [\rho - \epsilon, \rho + \epsilon] \\ \text{for } \ell < i \leq m & \mathcal{F}(x_i) \text{ must be less than } \rho - \epsilon \end{array}$$

2.3.1 The primal problem

Using the kernel trick, mapping $\mathcal{F}$ will be defined as a linear function w w.r.t. some feature space $\Phi(X)$:

$$\mathcal{F}(x) = \ < w, \Phi(x) >$$

Slack variables are introduced to account for insatisfiable constraints. With notations borrowed from [12], $\xi^{(*)}$ represents the $(m+\ell)$-vector made of $(\xi_i^{up})_{i \in [1,\ell]}$, $(\xi_i^{low})_{i \in [1,\ell]}$, and $(\xi_i^{up})_{i \in [\ell+1,m]}$). Letting C and ϵ be two positive constants, the primal ASM problem is:

$$\underset{\{w,\ \xi^{(*)},\ \rho\}}{\text{Minimize}} \ \frac{1}{2}||w||^2 + C \sum_{i=1}^{\ell} (\xi_i^{up} + \xi_i^{low}) + C \sum_{i=\ell+1}^{m} \xi_i^{up} + \rho \quad (2)$$

subject to

$$< w, \Phi(x_i) > \leq \rho + \epsilon + \xi_i^{up} \quad (i = 1 \dots \ell) \quad (3)$$

$$< w, \Phi(x_i) > \geq \rho - \epsilon - \xi_i^{low} \quad (i = 1 \dots \ell) \quad (4)$$

$$< w, \Phi(x_i) > \leq \rho - \epsilon + \xi_i^{up} \quad (i = \ell + 1 \dots m) \quad (5)$$

$$\xi_i^{up} \geq 0 \quad (i = 1 \dots \ell) \quad (6)$$

$$\xi_i^{low} \geq 0 \quad (i = 1 \dots \ell) \quad (7)$$

$$\xi_i^{up} \geq 0 \quad (i = \ell + 1 \dots m) \quad (8)$$

Due to space limitations, the detailed derivation of the solution is available in appendix at `http://sites.google.com/site/paretosvm/`.

3. PARETO-SVM FILTER ALGORITHM

This section describes the PARETO-SVM algorithm, exploiting the single ASM surrogate to speed up Evolutionary Multi-Objective Optimization.

3.1 Discussion

As mentioned earlier, surrogate (multi-objective) optimization most commonly proceeds by replacing the objective function with the surrogate model, computing the true objective on carefully selected points, and retraining the model from time to time using recently evaluated individuals.

The situation here is different as the optimization problem is a multi-objective one, and the single ASM surrogate model is being built. The most natural idea, optimizing directly the ASM model, raises the following two issues. Firstly, the true Pareto set expectedly lies away from the dominated points and beyond the current Pareto set; the ASM would thus be used to explore yet unexplored regions, i.e. for extrapolation. In contrast, single-objective surrogate models are mostly used for interpolation, except perhaps during the very first generations. Secondly and more importantly, identifying the Pareto set critically relies on the population diversity. While all individuals in the current Pareto set are equally mapped on the same ρ value, some will be 'more equal than others', in the sense that they will get a higher ASM value by chance. Optimizing *ex abrupto* the ASM model would thus favor some regions of the Pareto set and hinder the population diversity.

For these reasons, the ASM model will be used to implement a filter-based approach [10, 5]. Next subsections respectively outline the full algorithm, and describe the two specific modules of the PARETO-SVM algorithm, the surrogate model update and its use within informed operators.

3.2 The algorithm

The general description of an MOEA (Algorithm 1) is based on the usual parent-selection → variation → survival selection loop, with optionally some archive maintenance (line 5), as many popular MOEAs need to maintain some archive of the non-dominated individuals encountered during the search [3]. Note that line 4 describes both the parental selection and the application of the variation operators; it implicitly accounts for any choice procedure among multiple operators.

The PARETO-SVM algorithm is described likewise in Algorithm 2. The main differences are the model update (line 5) and the call to the informed operators (line 6) that replaces the standard call to variation operators, with the surrogate model F_{SVM} as additional argument. The archive maintenance is limited to storing all newborn offspring (line 7). Actual update, including the ASM update, takes place every K_{learn} generations (line 4).

3.3 Model Update

The model update (Algorithm 3) starts from the current archive (as produced from the previous update) augmented by all newborn offspring (line 7 of Algorithm 2) and first removes the possible duplicates (line 1). In most cases (depending on K_{learn} and the number of offspring generated per generation), the size of the archive increases far too much to make it possible to efficiently apply the Pareto-SVM learning. Furthermore, pruning the archive should not be done solely based on Pareto dominance, as in most standard MOEAs where only the best Pareto points are of interest. We need instead to ensure a good coverage of the dominated region that has been visited in the past, to make sure that the ASM will label these regions as 'dominated'. Borrowing ideas from PESA [2], the objective space is equally partitioned into $N_{archive}$ boxes, and the archive keeps one point per box. Boxes are computed in lines 2 and 3, points are put in their respective boxes in line 5, and all boxes are pruned (line 7), keeping either a uniformly chosen point among the non-dominated points of the box if any, or a uniformly chosen point in the box.

Algorithm 1 Standard MOEA

```
1: Archive ← ∅
2: Pop ← MOEA.Init()
3: while NOT Stopping Criterion do
4:     Offspring ← VarOp(ParentSelect(Pop))
5:     UpdateArchive(Pop,Offspring)
6:     Pop ← SurvivalSelect(Pop,Offspring)
7: return Pop.BestIndividual
```

ASM is learned from a training set made of one point per box (line 8), plus the current population (that is likely to contain non-dominated points). The training set is pruned to remove the duplicates and thereafter sorted using non-dominated sort to distinguish between current Pareto and dominated points (line 11). Finally it is passed to the ASM learning algorithm that returns the new ASM surrogate model to the main algorithm (line 12).

3.4 Informed Operators

The PARETO-SVM algorithm uses the ASM to yield *informed operators* [10]. Upon calling a variation operator, a given number of *pre-children* is generated, the ASM value is computed for all pre-children, and the operator returns the best ones according to those surrogate values.

An additional difficulty is raised in the multi-objective context, as a better surrogate value does not imply a smaller distance from the Pareto set. Quite the contrary, a child that is far from its parent can have a better ASM value than its parent while being nevertheless farthest from the Pareto front than some other points, because of the errors in the surrogate model, and/or the ϵ tolerance in the ASM formulation (section 2). In order to handle this issue, confirmed from preliminary experiments, the pre-children filtering is based on their ASM gain with respect to the closest point in the current Pareto set.

Formally, Algorithm 4 describes how all offspring are generated from the current parent population. For each offspring to be generated (outer loop, lines 2 to 12), a variation operator is chosen (line 3) if more than one are available (depending on the type of MOEA) and applied N_{inform} times (line 6). To each pre-child thus obtained, is associated its nearest neighbor among current non-dominated parents (line 7), and the ASM improvement of the pre-child compared to its nearest neighbor determines whether the pre-child is kept (9).

4. EXPERIMENTAL RESULTS

This section describes the experimental setting used to empirically validate the presented PARETO-SVM approach on classical MOO benchmark problems.

Algorithm 2 PARETO-SVM

Archive ← ∅
Pop ← MOEA.Init()
while NOT Stopping Criterion **do**
  if #generation ≡ 0 [K_{learn}] **then**
    F_{SVM} = UpdateModel(Archive, Pop)
      // every K_{learn} generation
  Offspring ← InfOp(ParentSelect(Pop), F_{SVM})
  Archive ← Archive ∪ Offspring
  Pop ← SurvivalSelect(Pop, Offspring)
return Pop.BestIndividual

Algorithm 3 UpdateModel(Archive, Pop)

EliminateDuplicates(Archive)
ComputeObjectiveBounds(Archive)
PartitionObjectiveSpace($N_{Archive}$)
for all P ∈ Archive **do**
  FindBox(P) // Assign P to the box it belongs to
for all Boxes B **do**
  Ind[B] ← Random(NonDominated(B))
    // Select one point per box
Archive ← $\bigcup_B$ Ind[B] // at most $N_{Archive}$ points
TrainingData ← Archive ∪ Pop
EliminateDuplicates(TrainingData)
NonDominatedSort(TrainingData)
return Pareto-SVM(Training Data) // returns F_{SVM}

Algorithm 4 InfOp(Parents, F)

Require: OP(s) // variation operator(s)
Offspring ← ∅
for iOff = 1 to RequiredSize **do**
  Choose variation operator Op // Eventually
  GainBest ← 0
  for i = 1 to N_{inform} **do**
    Ind ← Op(Parents)
    IndPop ← NearestNeighbor(Ind,ND-Parents)
    Gain ← F(IndPop) - F(Ind)
    if Gain > GainBest **then**
      GainBest ← Gain
      Best ← Ind
  Offspring ← Offspring ∪ {Best}
return Offspring

4.1 Experimental Setting

Two state-of-the-art EMOA algorithms are considered: $(\mu+\lambda)-S-NSGA-II$ [3, 4] and $\mu \times (1+\lambda)$-MO-CMA-ES [6]. Both algorithms use the hypervolume indicator as second-level sorting criterion to rank individuals on the same level of non-dominance. Population size is μ=100 for both algorithms, and offspring population sizes are λ=100 and $\mu \times (\lambda = 1)$ respectively. All reported results are based on 50 independent trials with at most 100000 fitness evaluations.

The PARETO-SVM approach is assessed by comparing the original algorithm with its ASM-enhanced version, considering the widely used ZDT1:3-6 [17] and their rotated variants IHR1:3-6 [6] benchmark problems. The dimension is set to 30 (resp. to 10) for ZDT1-3 problems (resp. for all other problems). As the true Pareto front of all ZDT problems lies on the boundary of the decision space, and for the sake of an unbiased assessment (to prevent MO-CMA-ES from exploiting this specificity), the penalization term is set to $\alpha = 1$ instead of the original 10^{-4} [6].

The PARETO-SVM parameters have been calibrated using a few preliminary experiments; their automatic tuning will be considered for further study. The ASM surrogate model is based on the Radial Basis Functions kernel (Eq. 1), where the bandwidth σ is set after the average distance D_{avr} of all training points; for ZDT problems $\sigma = 2D_{avr}$, $C = 10$ while for IHR problems $\sigma = D_{avr}$, $C = 100$. For all problems $N_{archive} = 400$ and $\epsilon = 10^{-5}$. The ASM learning was stopped after 300,000 iterations, corresponding to circa 0.5 – 1.0 sec. on a 2.26 GHz processor for ZDT1. The ASM update frequency K_{learn} is set to 10.

The ASM-enhanced operators were computed as described in Algorithm 2. In the MO-CMA-ES case, the global mutation step size was additionally modified to $\sigma' = \sigma \exp^{(-d+2dk)}$ where $d = 0.7$ and k is uniformly distributed in $[0, 1]$.

4.2 Performance Measures

Many ways of measuring the performance of MOO algorithms have been proposed. As recommended in [9], this study uses Pareto-compliant quality indicators. The widely used hypervolume indicator I_H was chosen for comparison of MOEAs (which in fact use hypervolume indicator as second sorting criterion).

Let P be a μ-size approximation of Pareto front and let P^* be the approximate μ-optimal distribution of optimal Pareto points [1]. The approximation error of the Pareto front is defined by $\Delta H(P^*, P) = I_H(P^*) - I_H(P)$.

4.3 Result Analysis

Two sets of experiments have been conducted to validate the proposed approach. The goal of the first experiments is to empirically evaluate the ASM accuracy. The second set of experiments investigates the effect of using PARETO-SVM within existing MOEAs on different benchmark functions.

In order to evaluate its accuracy on ZDT1 and IHR1 problems, the ASM model was trained using calibrated training data: 20000 points were generated at a given distance from the (known) nearly-optimal Pareto points, and non-dominated sorting was applied to rank those points. Front P_0 denotes the closest front from the true Pareto front, P_1 denotes the second one and so forth.

Figure 3 illustrates the distribution of F_{svm} values for training and test data in decision and objective spaces, where the training set respectively includes P_{80} and P_{100} as non-dominated and dominated points. As shown in Fig. 3, F_{svm} approximates the Pareto-dominance in the sense that for all k, $F_{svm}(P_k) > F_{svm}(P_{k+20})$ on average.

As seen for the IHR1 problem, although the F_{svm} value lies in an ϵ-width tube for training Pareto points, the new Pareto front may be non-linear. This behavior is quite normal when we deal with difficult problems. It may lead to premature convergence if we use very selective F_{svm} -based filter, as high F_{svm}-based selection pressure may accelerate the exploration of the prospective regions of Pareto front and entail some loss of diversity. Note that, as PARETO-SVM was devised to speed up the EMOA convergence and does not specifically take into account the Pareto diversity, it may be inefficient with regard to the approximation of the μ-optimal distribution of nearly-optimal Pareto points.

The first experiments with ASM-based MOEAs show that Pareto-SVM indeed speeds up both S-NSGA-II and MO-CMA-ES on most problems. Figure 4 shows the on-line behavior of the algorithms for ZDT1 and IHR1.

For the ZDT1 problem, the optimal Pareto front is linear and lies on the boundary of the decision space. Therefore, dominated points often lie at the decision space center, while Pareto points go toward the boundary, making the ASM model fairly simple: the One-Class SVM for dominated points covers the internal region of the decision space, while a small subspace of the Pareto points is covered by SVM-Regression with a given ϵ value.

ASM-based S-NSGA-II works nearly 1.5 times faster with $p = 2$ and more than 2 times faster with $p = 10$ than the original version with regard to the ΔH value and the number of function evaluations. The value $\Delta H = 0.001$ for ZDT problems corresponds to the situation when all points are non-dominated; ΔH is weakly sensitive to the diversity of points.

The IHR problems, rotated variants of ZDT problems, are non-separable and thus significantly more difficult for the MOEAs with operators which use separability. The Pareto set of IHR1 for a given rotation matrix is shown on Figure 3-c). The MO-CMA-ES inherits invariance properties from the CMA-ES, therefore it is also efficient on these rotated problems, while S-NSGA-II can approximate only a small part of optimal Pareto front which corresponds to the center of decision space.

The variance of results on ZDT1 problem is small because this problem is very simple for surrogate modeling and even if some premature convergence initially leads to sample only a small part of the Pareto set, the algorithm quickly explores the rest of the set thanks to separability. On rotated IHR1 problem, such quick moving is difficult, hence the higher variance of results which corresponds to slowly moving along the Pareto front. A high selection pressure also accelerates this effect.

Both MO-CMA-ES and S-NSGA-II only approximate a small part of the Pareto front in first generations, but in contrast to S-NSGA-II, MO-CMA-ES can gradually approximate the whole front. This can be seen clearly on Figure 4-b, witnessed by the flat line between 10000 and 40000 evaluations. In this case, while the ASM model helps MO-CMA-ES to converge faster to the Pareto front, it can not give any preference to the extreme points which in fact help to move along the Pareto front.

This observation sustains the idea that quality indicators should probably be taken in account during the ASM learning. The hypervolume indicator may provide useful additional information, especially because extreme points have the highest importance whatever the reference point. Also, hypervolume or Epsilon indicators are very attractive for many-objective optimization, when most points are non-dominated.

Finally, Table 1 shows the comparative results of all original and SVM-informed MOEAs. Different target values for ΔH have been set, and the number of evaluation needed to reach those values are reported - normalized by the smallest value of the table (recalled on the top row, legend "Best"). Hence 1 indicates the best result, while e.g., 2 indicates that this algorithm needed twice the number of evaluations of the best algorithm to reach the target ΔH value.

A general trend is observed, that increasing the selection pressure leads to a faster convergence. However, increasing the number of pre-children can also lead to premature convergence, like for MO-CMA-ES on IHR problems with $p = 10$. This happens because the filter prefers the points which are possibly better than their parents according to F_{svm} though they might be farther from the true Pareto front than other parents. The comparison of the pre-children with the closest parent in decision space (Algorithm 4, line 7) addresses this drawback to some extent. Further study will be devoted to designing more efficient strategies. One option could be to globally compare all pre-children of all parents, and select μ of them according to the diversity and the closeness to the parents in decision space.

5. DISCUSSION AND CONCLUSION

The main contribution of this paper is to present a single surrogate, multi-objective optimization approach. This approach is the first one, to the best of our knowledge, where a single surrogate gives an aggregated perspective on the position of any point with respect to the current Pareto set and the dominated region, guiding the offspring generation and speeding up the population move toward the true Pareto set. The aggregated ASM model is built by combining One-class and regression SVMs; thanks to the kernel trick, ASM can be learned efficiently in non-linear functional spaces. It is conjectured that this model should be able to track non-linear Pareto sets, and the presented results on a few benchmark functions validate this idea.

Further work should of course push further such validation, and make a thorough comparison of the proposed Pareto-SVM approach with standard surrogate multi-objective approaches, building one surrogate model per objective. A main limitation of such approaches is to require precise surrogate models (in order to preserve the dominance relationship), which raises some difficulties for instance in noisy environments. On the opposite, Pareto-SVM does not need a high precision as long as dominated points are separated from the current Pareto set. Moreover, parameter ϵ could be tuned to account for the amount of noise in the objectives, in case such information is available.

Further work will investigate how to extend Pareto-SVM by taking into account the hypervolume indicator, already mentioned in Section 4.3. Optimizing the hypervolume does lead to the Pareto set; building a surrogate model estimating the hypervolume contribution therefore appears to be very relevant. On the other hand, the hypervolume contribution depends on all other points in the population, possibly leading to an unstable and ill-conditioned regression problem.

Another perspective for further study concerns the ASM learning problem. This constrained optimization problem happens to be over-constrained; in such cases, it results in a poor generalization error of the ASM (visible e.g. from its errors on the rest of the Pareto archive). This problem was fixed using an additional k factor, replacing ρ by $k\rho$ in Equation (2). The best k value w.r.t. the ASM generalization error was determined from a preliminary trial, leading to $k = 1$ for ZDT problems and $k = -1$ for IHR problems. On-going work aims at understanding this phenomenon, and relating it to the structure of the multi-objective landscape.

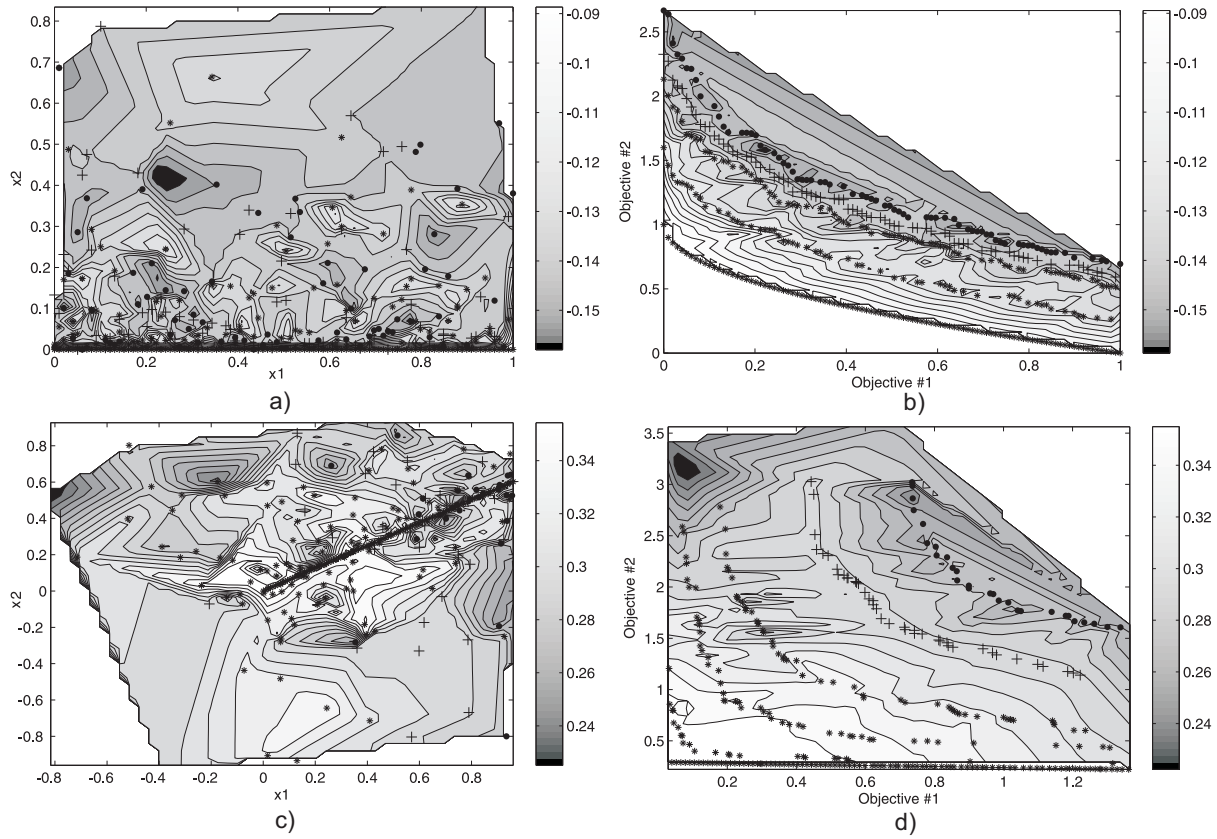


Figure 3: Surrogate Pareto-SVM model for ZDT1 (dim=30) in decision (a) and objective (b) space, for IHR1 (dim=10) in decision (c) and objective (d) space. Non-dominated fronts $P_{80}(+) \prec P_{100}(.)$ form the training data, while $P_k(*)$ for $k < 80$ represent the test data. See text for details.

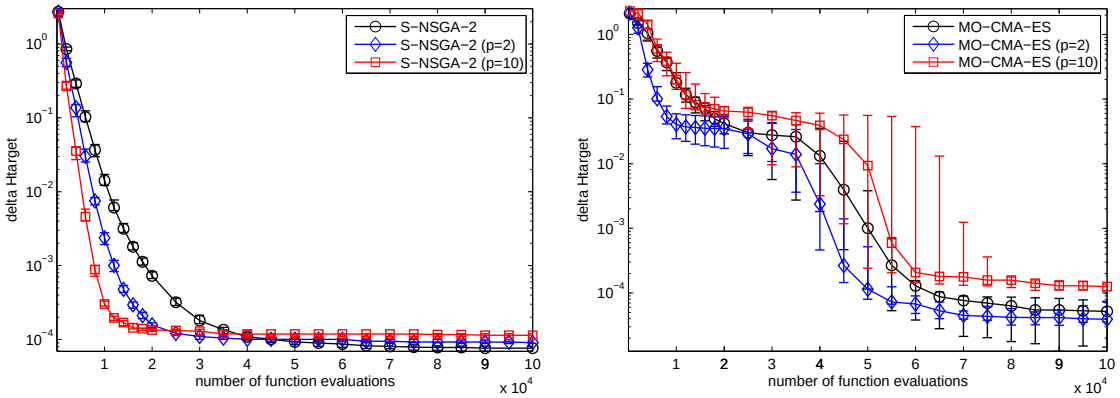


Figure 4: On-line performances of original and SVM-informed *S*-NSGA-II on ZDT1 (left) and MO-CMA-ES on IHR1 (right) problems with different values of number of pre-children p. Error bars indicate the 20% and 80% percentiles (almost indistinguishable for ZDT1).

6. REFERENCES

[1] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point. In *FOGA*, pages 87–102. ACM, 2009.

[2] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates. PESA-II: Region-based selection in evolutionary multiobjective optimization. In Lee Spector et al., editor, *GECCO-2001*, pages 283–290. Morgan Kaufmann, 2001.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast Elitist Multi-Objective Genetic Algorithm: NSGA-II. *IEEE TEC*, 6:182–197, 2000.

[4] M. Emmerich, N. Beume, and B. Naujoks. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In *Evolutionary Multi-Criterion Opt.*, pages 62–76. LNCS 3410, Springer Verlag, 2005.

[5] M. T. Emmerich, K. C. Giannakoglou, and B. Naujoks. Single- and Multiobjective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels. *IEEE TEC*, 10(4):421–439, 2006.

[6] C. Igel, N. Hansen, and S. Roth. Covariance Matrix Adaptation for Multi-objective Optimization. *Evolutionary Computation*, 15(1):1–28, 2007.

[7] Y. Jin. A Comprehensive Survey of Fitness Approximation in Evolutionary Computation. *Soft Computing*, 9(1):3–12, 2005.

[8] J. Knowles and H. Nakayama. Meta-modeling in multiobjective optimization. In J. Branke et al., editor, *Multiobjective Optimization*, number 5252 in LNCS, pages 245–284. Springer Verlag, 2008.

[9] J. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. Technical report, 2006.

[10] K. Rasheed and H. Hirsh. Informed operators: Speeding up genetic-algorithm-based design optimization using reduced models. In D. Whitley et al., editor, *GECCO'2000*, pages 628–635. Morgan Kaufmann, 2000.

[11] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the Support of a High-Dimensional Distribution. *Neural Computation*, 13:1443–1471, 2001.

[12] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[13] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[14] I. Voutchkov and A. Keane. Multiobjective Optimization using Surrogates. In I. Parmee, editor, *ACDM'06*, pages 167–175. Institute for People-centred Computation, 2006.

[15] E. F. Wanner, F. G. G. aes, R. H. C. Takahashi, and P. J. Fleming. Local Search with Quadratic Approximations into Memetic Algorithms for Optimization with Multiple Criteria. *Evolutionary Computation*, 16(2):185–224, 2008.

[16] Y.Yun, H. Nakayama, and M. Arakava. Generation of pareto frontiers using support vector machine. In *MCDM'04*, 2004.

[17] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8:173–195, 2000.

Table 1: Median number of function evaluations to reach ΔHtarget values, normalized by Best

ZDT1					
ΔHtarget	1	0.1	0.01	1e-3	1e-4
Best	1100	3000	5300	7900	45700
S-NSGA-II	1.6	2	2	2.3	**1**
S-NSGA-II p=2	1.2	1.5	1.4	1.5	1.3
S-NSGA-II p=10	**1**	**1**	**1**	**1**	.
MO-CMA-ES	16.5	14.5	12.3	11.2	.
MO-CMA-ES p=2	6.9	8.5	8.4	7.9	.
MO-CMA-ES p=10	6.9	9.4	9.5	10.3	.
ZDT2					
ΔHtarget	1	0.1	0.01	1e-3	1e-4
Best	1400	4900	6800	8600	34300
S-NSGA-II	1.8	1.5	1.8	2.3	1.2
S-NSGA-II p=2	1.2	**1**	1.2	1.4	**1**
S-NSGA-II p=10	**1**	**1**	**1**	**1**	.
MO-CMA-ES	14.7	9.2	9.7	10.3	.
MO-CMA-ES p=2	5.5	6	6.9	7.4	.
MO-CMA-ES p=10	5	.	.	.	.
ZDT3					
ΔHtarget	1	0.1	0.01	1e-3	1e-4
Best	1300	3500	7100	10200	15400
S-NSGA-II	1.4	1.9	1.6	1.9	2.1
S-NSGA-II p=2	1.1	1.3	1.1	1.2	1.3
S-NSGA-II p=10	**1**	**1**	**1**	**1**	**1**
MO-CMA-ES	15.7	13.3	9.5	8.8	.
MO-CMA-ES p=2	6.2	9.8	9.1	7.9	.
MO-CMA-ES p=10	12.3	19.8	.	.	.
ZDT6					
ΔHtarget	1	0.1	0.01	1e-3	1e-4
Best	2900	6700	12400	25500	.
S-NSGA-II	1.8	1.8	1.6	1.3	.
S-NSGA-II p=2	1.2	1.3	1.1	**1**	.
S-NSGA-II p=10	**1**	**1**	**1**	1.1	.
MO-CMA-ES	6.6	6.7	5.3	3.4	.
MO-CMA-ES p=2	2.6	4.4	3.8	2.5	.
MO-CMA-ES p=10	3.7	6.4	5.2	3.4	.
IHR1					
ΔHtarget	1	0.1	0.01	1e-3	1e-4
Best	500	2800	36300	41800	50900
S-NSGA-II	1.6	**1**	.	.	.
S-NSGA-II p=2	1.2	**1**	.	.	.
S-NSGA-II p=10	**1**	1.1	.	.	.
MO-CMA-ES	8.4	4.7	1.1	1.1	1.2
MO-CMA-ES p=2	4.8	2.1	**1**	**1**	**1**
MO-CMA-ES p=10	9.4	4.3	1.3	1.2	.
IHR2					
ΔHtarget	1	0.1	0.01	1e-3	1e-4
Best	1800	10100	19900	45400	.
S-NSGA-II	1.1	2.3	4	.	.
S-NSGA-II p=2	**1**	3.2	3.4	.	.
S-NSGA-II p=10	1.3	4.8	3.1	.	.
MO-CMA-ES	5.2	1.8	1.4	1.1	.
MO-CMA-ES p=2	2.4	**1**	**1**	**1**	.
MO-CMA-ES p=10	5.7	1.8	1.5	.	.
IHR3					
ΔHtarget	1	0.1	0.01	1e-3	1e-4
Best	900	11500	36300	54200	.
S-NSGA-II	1.3	.	.	.	.
S-NSGA-II p=2	**1**	.	.	.	.
S-NSGA-II p=10	**1**	.	.	.	.
MO-CMA-ES	8.5	1.6	1.1	**1**	.
MO-CMA-ES p=2	5.8	**1**	**1**	1.1	.
MO-CMA-ES p=10	11	.	.	.	.
IHR6					
ΔHtarget	1	0.1	0.01	1e-3	1e-4
Best	5700	14500	.	.	.
S-NSGA-II	15.8	.	.	.	.
S-NSGA-II p=2	11.3	.	.	.	.
S-NSGA-II p=10	.	.	.	.	.
MO-CMA-ES	1.6	1.4	.	.	.
MO-CMA-ES p=2	**1**	**1**	.	.	.
MO-CMA-ES p=10	1.9	.	.	.	.