A Pareto-Compliant Surrogate Approach for Multiobjective Optimization

Submitted to the *Evolutionary Multiobjective Optimization* track

ABSTRACT

Most surrogate approaches to multi-objective optimization build a surrogate model for each objective. The models for the objectives can then be used in different ways: 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 as 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 SVM (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, significantly reducing the number of evaluations of the actual objective functions on classical benchmarks problems.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artificial IntelligenceProblem 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 (see e.g. [10] for a survey of surrogate evolutionary optimization). Several types of meta-models have been used (quadratic models, neural networks, Regression Support Vector Machines, kriging or Gaussian Processes). 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 [12] 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 [18], or within a memetic algorithm for local seach improvement [**?**]. Meta-models can also be used to rank and filter out offspring (pre-screening mode), according to Pareto-related indicators like the hypervolume [7], or a weighted sum of the objectives, or a goal-oriented direction. Lastly, meta-models can be used within interactive multiobjective optimization like aspiration level methods.

Surrogate approaches generally consider the decision space, i.e. a meta-model associates some approximate objective information to any individual. A notable exception [19] considers the objective space and characterizes the region of the objective space which has already been visited. The rationale for this approach, based on One-Class SVM [15], 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 formulation and the resolution of the ASM model. 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 describes the Aggregated Surrogate Model (ASM), formalizes the constraints it should satisfy and details the ASM resolution. Due to space limitations, we assume the reader's familiarity with the Support Vector Machines principles [17].

2.1 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 a given time during the run of an EMOA, 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. 1.(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. 1.(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 MOO benchmarks, (e.g. IHR1, see Fig. 3 (c) and (d)).

Expectedly, the ASM model discriminates the Pareto set and the dominated region. However, a binary classification approach is ill-suited, in the sense that 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 [16] (mapping each point x onto some target value $f(x)$ up to some tolerance ϵ) and the One-class SVM [15], 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.

2.2 Lagrangian formulation

Let $X \subset \mathbb{R}^d$ denote the search (decision) space and let $x_1 \dots x_m$ denote points in X, with $x_1 \dots x_\ell$ being Pareto points and $x_{\ell+1}, \dots, x_m$ being dominated points. The sought ASM mapping, noted $\mathcal{F}$ ($\mathcal{F}: X \mapsto \mathbb{R}$), is finally subject to $m+\ell$ constraints: for each $x_i, 1 \leq i \leq \ell$, $\mathcal{F}(x_i)$ must belong to $[\rho-\epsilon, \rho+\epsilon]$ and for each $x_i, \ell < i \leq m$, $\mathcal{F}(x_j)$ must be less than $\rho-\epsilon$.

2.2.1 The primal problem

Using the kernel trick[2], mapping $\mathcal{F}$ will be defined as a linear function w w.r.t. some feature space $\Phi(X)$:

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

Then, introducing the usual slack variables $\xi^{(*)}$ (with notations borrowed from [16], $\xi^{(*)}$ represents the $(m+\ell)$-vector made of $(\xi_i^{up})_{i\in[1,\ell]}$, $(\xi_i^{low})_{i\in[1,\ell]}$, and $(\xi_i^{up})_{i\in[\ell+1,m]}$), and given positive constants C and ϵ, the primal problem is:

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

subject to

$$< w, \Phi(x_i) > \leq \rho + \epsilon + \xi_i^{up} \quad (i = 1 \dots \ell) \quad (2)$$

$$< w, \Phi(x_i) > \geq \rho - \epsilon - \xi_i^{low} \quad (i = 1 \dots \ell) \quad (3)$$

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

$$\xi_i^{up} \geq 0 \quad (i = 1 \dots \ell) \quad (5)$$

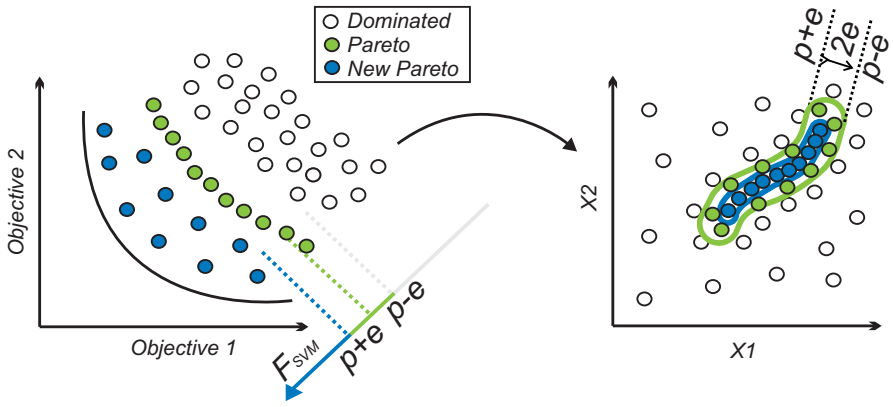
$$\xi_i^{low} \geq 0 \quad (i = 1 \dots \ell) \quad (6)$$

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

Introducing the non-negative Lagrangian multipliers $\alpha_i^{(*)}$ and $\beta_i^{(*)}$ for each above constraint respectively (where $^{(*)}$

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

[2]The SVM approach, initially aimed at finding linear functions, only computes scalar products of sample points. The so-called kernel trick supports the extension to non-linear functional spaces: the search space X is mapped onto a more expressive space referred to as feature space $\Phi(X)$, where the scalar product $< \Phi(x), \Phi(x') >= K(x, x')$ can be calculated without computing explicitly $\Phi(x)$ or $\Phi(x')$.



(a). Objective space (b) Decision space

Figure 1: A schematic view of the Pareto front, depicting dominated points (white), current Pareto (grey) and new Pareto (black) respectively in objective and decision space.

is either $^{(up)}$ or $^{(low)}$), the Lagrangian is:

$$\begin{aligned}L(w,\rho,\xi^{(*)},\alpha^{(*)},\beta^{(*)}) &= \frac{1}{2}||w||^2 \\ &+C\sum_{i=1}^{\ell}(\xi_i^{up}+\xi_i^{low})+C\sum_{i=\ell+1}^{m}\xi_i^{up}+\rho \\ &-\sum_{i=1}^{\ell}\alpha_i^{up}(\rho+\epsilon+\xi_i^{up}-<w,\Phi(x_i)>) \\ &-\sum_{i=1}^{\ell}\alpha_i^{low}(<w,\Phi(x_i)>-\rho+\epsilon+\xi_i^{low}) \\ &-\sum_{i=\ell+1}^{m}\alpha_i^{up}(\rho-\epsilon+\xi_i^{up}-<w,\Phi(x_i)>) \\ &-\sum_{i=1}^{\ell}\beta_i^{up}\xi_i^{up} \\ &-\sum_{i=1}^{\ell}\beta_i^{low}\xi_i^{low} \\ &-\sum_{i=\ell+1}^{m}\beta_i^{up}\xi_i^{up}\end{aligned}$$

Computing the KKT conditions leads to:

$$\frac{\partial L}{\partial w} = w+\sum_{i=1}^{\ell}(\alpha_i^{up}-\alpha_i^{low})\Phi(x_i)+\sum_{i=\ell+1}^{m}\alpha_i^{up}\Phi(x_i)=0 \quad (8)$$

$$\frac{\partial L}{\partial \rho} = 1-\sum_{i=1}^{\ell}(\alpha_i^{up}-\alpha_i^{low})-\sum_{i=\ell+1}^{m}\alpha_i^{up}=0 \quad (9)$$

$$\frac{\partial L}{\partial \xi_i^{up}} = C-\alpha_i^{up}-\beta_i^{up}=0 \quad (10)$$

$$\frac{\partial L}{\partial \xi_i^{low}} = C-\alpha_i^{low}-\beta_i^{low}=0 \quad (11)$$

$$\frac{\partial L}{\partial \xi_i^{up}} = C-\alpha_i^{up}-\beta_i^{up}=0 \quad (12)$$

Therefore, at the saddle point we have:

$$w=\sum_{i=1}^{\ell}\alpha_i^{low}\Phi(x_i)-\sum_{i=1}^{m}\alpha_i^{up}\Phi(x_j) \quad (13)$$

$$1=\sum_{i=1}^{m}\alpha_i^{up}-\sum_{i=1}^{\ell}\alpha_i^{low} \quad (14)$$

$$C=\alpha_i^{up}+\beta_i^{up}=\alpha_i^{low}+\beta_i^{low} \quad (15)$$

Reporting these equalities, the Lagrangian becomes:

$$L(w,\rho,\xi,\alpha,\beta) = -\frac{1}{2}||w||^2-\epsilon\left(2\sum_{i=1}^{\ell}\alpha_i^{up}-1\right)$$

Eliminating the $\beta^{(*)}$ thanks to Eq. (15), the dual problem to solve in $(\alpha^{(*)})$ is:
Maximize

$$\widetilde{\mathcal{L}}(\alpha^{(*)}) = -\frac{1}{2}||w||^2-\epsilon\left(2\sum_{i=1}^{\ell}\alpha_i^{up}-1\right) \quad (16)$$

subject to

$$\sum_{i=1}^{m}\alpha_i^{up}-\sum_{i=1}^{\ell}\alpha_i^{low}=1 \quad (17)$$

$$0\le\alpha_i^{(*)}\le C \quad (18)$$

2.2.2 *Solving the dual problem*

Following [16], the idea is to iterate exact resolutions of the maximization problem by varying only two of the α's multipliers. Thanks to the sum constraint (Eq. (14) or (17)), one of the α variables can be eliminated. As the resulting function, now depending on a single variable, is quadratic, its optimization can be solved analytically. It remains to choose the pair of α indices; this choice has a large impact on the overall computational cost for large regression problems, and several heuristics have been proposed [8]. It turns out that the best results in our problem were obtained for a uniform selection of the α indices.

For the sake of completeness, but due to space limitations, the detailed derivation of the solution, that has to distinguish all possible cases between Pareto and dominated points, is temporarily anonymously available at `http://sites.google.com/site/paretosvm/`.

3. PARETO-SVM FILTER ALGORITHM

This section describes the use of the ASM meta-model for Evolutionary Multi-Objective Optimization, defining the Pareto-SVM algorithm.

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 updating the model

from time to time using recently evaluated individuals as examples.

The situation here is different as i/ the optimization problem is a multi-objective one; ii/ the presented approach involves the single surrogate ASM model. 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. 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 [14, 7]. 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 (implicitly including any choice among multiple operators for instance).

The Pareto-SVM algorithm is described similarly 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. Note that, at this level, the maintenance of the archive is limited to adding to it all newborn offspring (line 7). Actual update, including the ASM update, takes place every K_{learn} generations (line 4).

3.3 Model Update

The procedure for updating the model is given in Algorithm 3. When entering the update procedure, the archive is made of the archive at the end of the previous update, augmented by all newborn offspring (line 7 of Algorithm 2). The possible duplicates are removed (line 1). In most cases (depending on K_{learn} and the number of offspring generated per generation, the size of the archive will increase 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. The idea is to keep in the archive points that will ensure a good coverage of the complete 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 [**?**], the objective space will be divided regularly into $N_{archive}$ boxes, and only one point will be kept in each box. The boxes are computed in lines 2 and 3, the 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. The training data for the Pareto-SVM learning is made of one point per box (line 8), plus the current population (that is likely to contain non-dominated points). It is then pruned from duplicates, and sorted using non-dominated sort to distinguish between Pareto and dominated points (line 11) before be passed on to the Pareto-SVM learning algorithm that returns the surrogate model to the main algorithm (line 12).

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
```

3.4 Informed Operators

The basic idea used here is that of *informed operators* [14]. When a variation operator is called, it generates a given number of *pre-children*. The value of the surrogate model for all these pre-children is computed, and the operator returns the best one according to those surrogate values. However, the particular multi-objective context raises an additional issue here: a better surrogate value does not imply a smaller distance from the Pareto set. Indeed, because of the errors in the surrogate model, and because of the ϵ tolerance in the formulation (see Section 2), 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 lying on the current Pareto front – and preliminary experiments confirmed that. In order to minimize such risk, the choice of which offspring to keep is based on the ASM gain with respect to the closest point in the current Pareto set.

A more formal description is given in Algorithm 4, that 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 eventually chosen (line 3) if more than one are available, depending on the type of MOEA. It is then applied N_{inform} times (line 6). For each such pre-offspring, the nearest point from current non-dominated parents is sought (line 7), and depending on the improvement of the surrogate model F with respect to this parent, the pre-offspring is kept or not (9).

4. EXPERIMENTAL RESULTS

4.1 Experimental Settings

Two state-of-the-art EMOA were chosen from literature: $(\lambda+\mu)-\mathcal{S}$-NSGA-2 [3, 5] and $\mu\times(1+\lambda)$-MO-CMA-ES [9]. 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

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 ← ⋃_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
```

independent trials with at most 100000 fitness evaluations.

The results of the original and SVM-informed versions of algorithms are compared on the widely used ZDT1:3-6 [21] and their rotated variants IHR1:3-6 [9]. For ZDT1-3 problems, dimension is 30, while it is set to 10 for all other problems. Note that the optimal Pareto front of all ZDT problems lies on the boundary of the decision space. Hence, in order to prevent the exploitation of this specificity by MO-CMA-ES, its penalization term is set to $\alpha = 1$ instead of the original 10^{-4} [9].

The specific parameters of PARETO-SVM were calibrated by preliminary experiments. K_{learn} was set to 10. The widely used Radial Basis Functions (RBF) was chosen as a kernel for PARETO-SVM . The choice of σ was made by computing the average distance D_{avr} of all points in training set, and, for ZDT problems $\sigma = 2D_{avr}$, $C = 10$, while for IHR problems $\sigma = D_{avr}$, $C = 100$. For all problems $N_{archive} = 400$, and $\epsilon = 10^{-5}$. PARETO-SVM model optimization was stopped after 300000 choices of indices pairs. The CPU cost of one PARETO-SVM learning on ZDT1 is then approximately 0.5 – 1.0 sec. on a 2.26 GHz processor.

The SVM-informed versions of MOEAs was developed as described in Algorithm 2. To generate i-th pre-child in SVM-informed version of MO-CMA-ES the global mutation step size can be additively changed : $\sigma_i^{'} = \sigma_i \exp^{(-d+2dk)}$, 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. This study uses Pareto-compliant quality indicators as recommended in [13]. The widely used hypervolume indicator was chosen for comparison of MOEAs which in fact use hypervolume indicator as second sorting criterion.

Let P be an approximation of Pareto front with $|P| = \mu$. Let P^* be the approximating μ-optimal distribution of optimal Pareto points [2]. The error of the Pareto front approximation is defined by $\Delta H(P^*, P) = I_H(P^*) - I_H(P)$.

4.3 Results Analysis

Two sets of experiments have been conducted to validate the proposed approach. The goal of the first experiments is to empirically evaluate the accuracy of the PARETO-SVM model. 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 problem, the PARETO-SVM model was optimized using specific 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, leading to fronts P_0 (the closest from true Pareto), P_1, ...

Figure 3 illustrates the result distribution of F_{svm} values for training and test data in decision and objective space. The dominated and Pareto points of training data were P_{80} and P_{100} non-dominated fronts respectively. To evaluate the distribution of F_{svm} in new regions, several fronts with smaller indices were used. Figure 3 shows that indeed, for all k, $F_{svm}(P_k) > F_{svm}(P_{k+20})$ on average.

Although, the values of F_{svm} of training Pareto points lie in a small tube constrained by ϵ, the new Pareto front may be non-linear as we can see for IHR1 problem. This behavior is quite normal when we deal with difficult problems and may lead to premature convergence if we use very selective F_{svm} -based filter. Indeed, high F_{svm} -based selection pressure may accelerate the exploration of the perspective regions of Pareto front with loss of diversity. PARETO-SVM Filter deals with the acceleration of the convergence of the EMOA and not with the diversity, therefore it may be inefficient in approximation of the μ-optimal distribution of nearly-optimal Pareto points.

The first experiments with SVM-informed MOEAs show that PARETO-SVM indeed allows to accelerate both S-NSGA-2 and MO-CMA-ES on most problems. Figure 2 shows the on-line behavior of the algorithms for ZDT1 and IHR1.

The optimal Pareto front of ZDT1 problem is linear and lies on the boundary of the decision space. Therefore, the dominated points often lie in the center of decision space, while Pareto points goes toward the boundary. In this case

the solution of Pareto-SVM learning is fairly simple: the One-Class SVM for dominated points covers the center of decision space, and small subspace of Pareto points are covered by Regression with a given ϵ value.

SVM-informed S-NSGA-2 works nearly 1.5 times faster with $p = 2$ and more than 2 times faster with $p = 10$ than original version in the sense of minimization of ΔH value and number of function evaluations. The value $\Delta H = 0.001$ for ZDT problems corresponds to the situation when all points are non-dominated and only the diversity of points makes small difference of ΔH value.

The IHR problems are rotated variants of ZDT problems, therefore they are non-separable and 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-a). The MO-CMA-ES inherits invariance properties from the CMA-ES, therefore it is also efficient on these rotated problems, while S-NSGA-2 can approximate only 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-2 approximate only small part of Pareto front in first generations, but in contrast to S-NSGA-2, MO-CMA-ES can gradually approximate the whole front. This can be seen clearly on Figure 2-b, witnessed by the flat line between 10000 and 40000 evaluations. In this case, Pareto-SVM model helps MO-CMA-ES to converge faster to the Pareto front, but 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 Pareto-SVM learning. The hypervolume contribution as an additional information may be useful, especially because the extreme points will have the highest importance. 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 line "Best" on top). Hence a 1 indicates the best result, and 2 for instance indicates that this algorithm needed twice the number of evaluations of the best algorithm to reach this value of ΔH.

In general, the results are similar in the sense that increasing the selection pressure leads to the 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}tough 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) decreases this impact of this drawback, but more efficient strategy should be used. One option could be to compare all pre-children of all parents together, and select μ of them according to the diversity and the closeness to the parents in decision space. This topic will be addressed in future work.

5. DISCUSSION AND CONCLUSION

This paper has introduced the first surrogate model that is truly multi-objective: a single model defined on the decision space gives an aggregated perspective on the position of any point with respect to the current Pareto set and the dominated region. This aggregated surrogate model, ASM, enables to guide the offspring generation and speeds up the population move toward the true Pareto set. Built by combining One-class and regression SVMs and 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 error on the rest of the Pareto archive). This problem was fixed using an additional k factor, replacing ρ by $k\rho$ in Equation[3] (1). The best k value in the sense of the ASM generalization error was determined for each problem using one 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 underlying structure of the multi-objective problem.

6. REFERENCES

[1] K. Abboud and M.Schoenauer. Surrogate deterministic mutation: Preliminary results. In P. Collet et al., editor, *Artificial Evolution'01*, pages 103–115. LNCS 2310, Springer Verlag, 2002.

[2] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point.

[3]This modification entails replacing 1 by k in Eq. (9,14) and (17); the rest is unchanged.

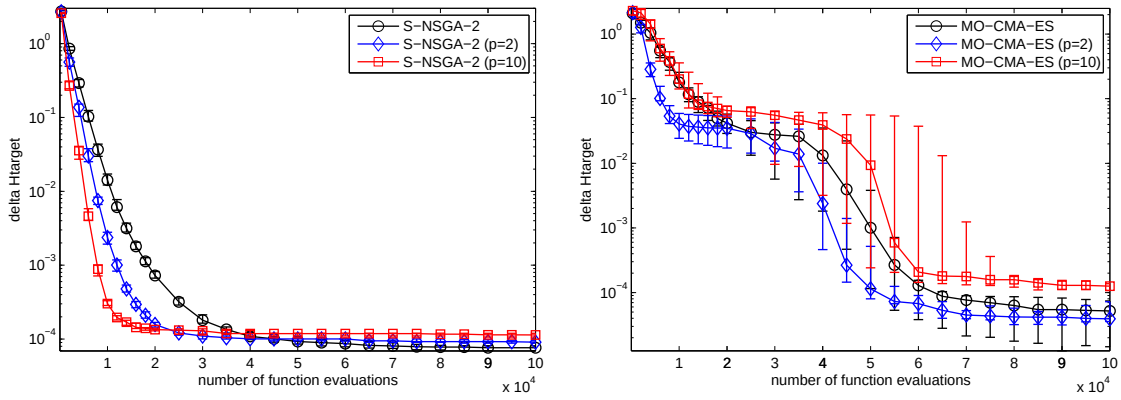


Figure 2: On-line performances of original and SVM-informed MOEAs on ZDT1 (left) and IHR1 (right) problems with different values of number of pre-children p. Error bars indicate the 20% and 80% percentiles (almost indistinguishable for ZDT1).

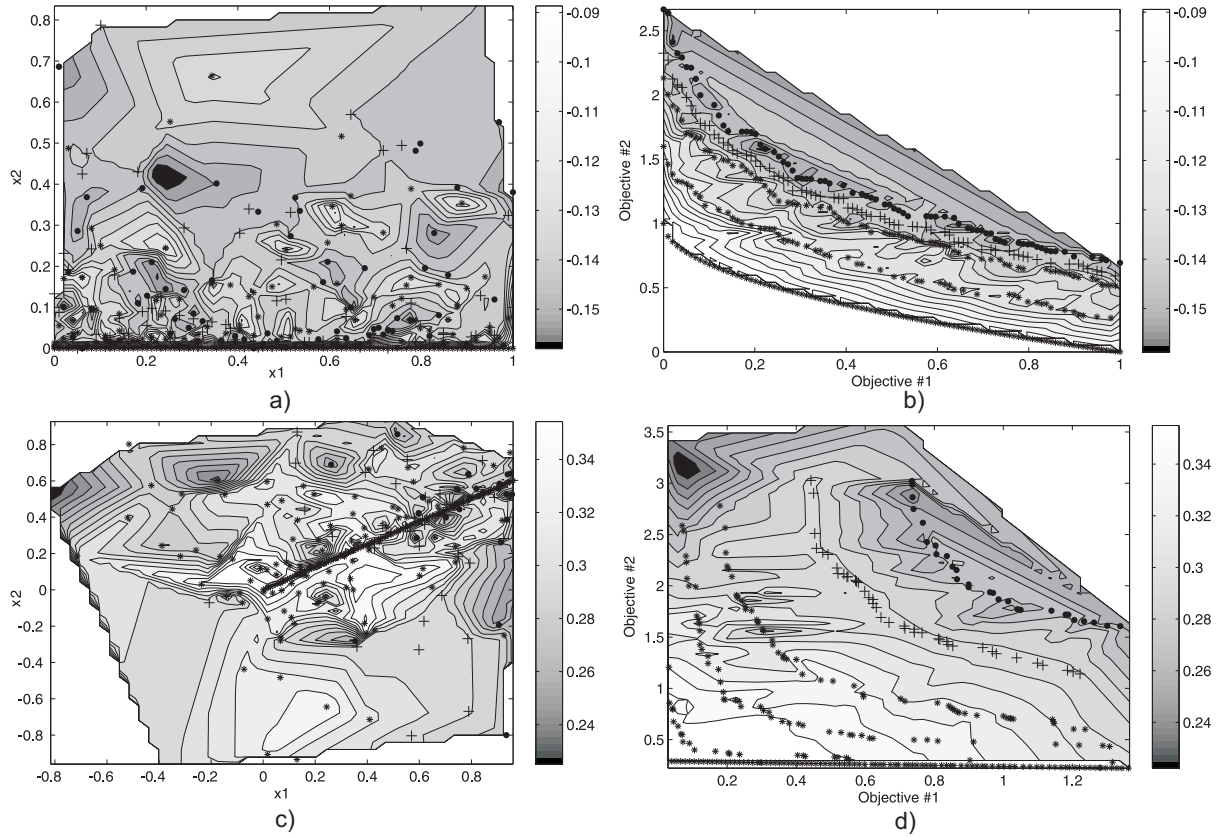


Figure 3: Surrogate Pareto-SVM model for ZDT1 (dim=30) in decision (a) and objective (b) space, for IHR1 (dim=10) in decision (c) and objective (d) space. See text for details.

In *Foundations of Genetic Algorithms (FOGA 2009)*, pages 87–102. ACM, 2009.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast elitist multi-objective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2000.

[4] M. El-Beltagy, P. Nair, and A. Keane. Metamodeling techniques for evolutionary optimization of computationally expensive problems: Promises and limitations. In D.E. Goldberg et al., editor, *GECCO'99*, pages 196–203. Morgan Kaufmann, 1999.

[5] M. Emmerich, N. Beume, and B. Naujoks. An emo algorithm using the hypervolume measure as selection criterion. In *2005 Intl Conference, March 2005*, pages 62–76. Springer, 2005.

[6] M. Emmerich, A. Giotis, M. Özdemir, T. Bäck, and K. Giannakoglou. Metamodel-assisted evolution strategies. In J. J. Merelo et al., editor, *PPSN VII*, number 2439 in LNCS, pages 361–370. Springer Verlag, 2002.

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

[8] T. Glasmachers and C. Igel. Second-Order SMO Improves SVM Online and Active Learning. *Neural Computation*, 20(2):374–382, 2008.

[9] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28, 2007.

[10] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1):3–12, 2005.

[11] S. Kern, N. Hansen, and P. Koumoutsakos. Local meta-models for optimization using evolution strategies. In Th. Runarsson et al., editor, *PPSN IX*, number 4193 in LNCS, pages 939–948. Springer Verlag, 2006.

[12] 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.

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

[14] 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.

[15] 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.

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

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

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

[19] Y.Yun, H. Nakayama, and M. Arakava. Generation of pareto frontiers using support vector machine. In *17th International Conference on Multi-Criteria Decision Making - MCDM'04*, 2004.

[20] Z. Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, , and K. Y. Lum. Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Trans. Systems, Man and Cybernetics - Part C*, 37(1):66–76, 2007.

[21] 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-2	1.6	2	2	2.3	**1**
S-NSGA-2 p=2	1.2	1.5	1.4	1.5	1.3
S-NSGA-2 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-2	1.8	1.5	1.8	2.3	1.2
S-NSGA-2 p=2	1.2	**1**	1.2	1.4	**1**
S-NSGA-2 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-2	1.4	1.9	1.6	1.9	2.1
S-NSGA-2 p=2	1.1	1.3	1.1	1.2	1.3
S-NSGA-2 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-2	1.8	1.8	1.6	1.3	.
S-NSGA-2 p=2	1.2	1.3	1.1	**1**	.
S-NSGA-2 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-2	1.6	**1**	.	.	.
S-NSGA-2 p=2	1.2	**1**	.	.	.
S-NSGA-2 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-2	1.1	2.3	4	.	.
S-NSGA-2 p=2	**1**	3.2	3.4	.	.
S-NSGA-2 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-2	1.3	.	.	.	.
S-NSGA-2 p=2	**1**	.	.	.	.
S-NSGA-2 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-2	15.8	.	.	.	.
S-NSGA-2 p=2	11.3	.	.	.	.
S-NSGA-2 p=10	.	.	.	.	.
MO-CMA-ES	1.6	1.4	.	.	.
MO-CMA-ES p=2	**1**	**1**	.	.	.
MO-CMA-ES p=10	1.9	.	.	.	.