Multi objective particle swarm optimization using enhanced dominance and guide selection

Gérard Dupont, Sébastien Adam, Yves Lecourtier, Bruno Grilhère

International Journal of Computational Intelligence Research.
ISSN 0973-1873 Vol.2, No.X (2006), pp. XXX–XXX

Multi objective particle swarm optimization using enhanced dominance and guide selection

Gérard Dupont[1 2], **Sébastien Adam**[1], **Yves Lecourtier**[1] **and Bruno Grilheres**[1 2]

[1]Laboratoire d'Informatique de Traitement de l'Information et des Systmes (LITIS),
Universit de Rouen, Saint-Étienne-du-Rouvray, France

[2]EADS Defense and Systems, Information Processing and Competence Center,
Val de Reuil, France

Abstract:
Nowadays, the core of the Particle Swarm Optimization (PSO) algorithm has proved to be reliable. However, faced with multi-objective problems, adaptations are needed. Deeper researches must be conducted on its key steps, such as solution set management and guide selection, in order to improve its efficiency in this context. Indeed, numerous parameters and implementation strategies can impact on the optimization performance in a particle swarm optimizer. In this paper, our recent works on those topics are presented. We introduce an ε dominance variation which enables a finer neighborhood handling in criterion space. Then we propose some ideas concerning the guide selection and memorization for each particle. These methods are compared against a standard MOPSO implementation on benchmark problems and against an evolutionary approach (NSGAII) for a real world problem: SVM classifier optimization (or model selection) for a handwritten digits/outliers discrimination problem.

Keywords: Optimization, particle swarm, SVM model selection, multi objective optimizer, epsilon-dominance.

I. Introduction

In several technical fields, engineers are dealing with complex optimization problems which involve contradictory objectives. Such multi-objective optimization problems have been extensively studied during the last decades. Existing approaches can be classified with respect to the hypotheses which are required for the computation. A common hypothesis is the derivability or continuity of the functions to be optimized. Unfortunately, such hypotheses are not verified for problems with complex models. Thus other ways have been found through meta-heuristic algorithms. Genetic algorithms are famous techniques in that domain and they have shown to be efficient on many optimization problems (see [13]). Recently, some researchers also tackle those problems with multi-objective particle swarm optimizer (see [10]).
Based on the work of James Kennedy and Russel Eberhart presented in [15], the particle swarm optimizers try to find solutions of optimization problems by using techniques inspired by the nature, as the genetic algorithms mimic evolution in species. In the last few years, PSO has been extensively studied and some results have shown that it can compete with other evolutionary algorithms such as genetic algorithms (see [16, 21, 31]). Multi-Objective PSO algorithms (refered as MOPSO in the paper) have also been implemented and have opened a large new field of interest (see [28]).
The aim of this paper is to propose some improvements of particle swarm optimizer dealing with multi-objective problems. These improvements concern the introduction of a new dominance and an original strategy for guide selection.
The paper is organized as follows: section II gives a brief overview on basic definitions involved in multi-objective optimization problems and in particle swarm optimization. In section III, our contributions concerning the dominance and the guide selection strategy are described. In section IV, these contributions are discussed through experimental results on benchmark problems. Finally, the proposed variant of the MOPSO algorithm is applied on a real world problem which concerns SVM multi-model selection for handwritten digit identification.

II. Basic definitions

This section presents the basic formalization of multi-objective optimization problems. Then it describes the particle swarm core algorithm and its classical multi-objective implementation (see [10]).

A. Multi-objective optimization problems

Many definitions can be found for multi-objective optimization problems (see [9] for a precise definition of all the following equations). Such problems seek to minimize simultaneously N objective functions f_k depending on n parameters in the form:

$$\begin{aligned} f_k: \quad & \mathbb{R}^n && \longrightarrow \mathbb{R} \\ & \overrightarrow{x} && \longrightarrow f_k(\overrightarrow{x}) \\ & with && k \in [1; N] \end{aligned} \tag{1}$$

In order to express parameter limitations that can be met in real world problems (such as material characteristics in engineering applications), some constraints must be introduced.

They reduce the feasible region of $\mathbb{R}^n$ to a smaller one noted S. Usually, these constraints are modeled as M equations expressed as inequalities or equalities:

$$g_k(\overrightarrow{x}) \geq 0 \quad \text{with} \quad k \in [1; M] \tag{2}$$
$$h_k(\overrightarrow{x}) = 0 \quad \text{with} \quad k \in [1; M] \tag{3}$$

The global multi-objective problem can thus be defined as the minimization of:

$$\overrightarrow{f}\ (\overrightarrow{x}) = \{f_1(\overrightarrow{x}), \ldots, f_k(\overrightarrow{x}), \ldots, f_N(\overrightarrow{x})\} \tag{4}$$
$$\text{given } \overrightarrow{x} \in S \quad \leftrightarrow \left\{ \begin{array}{l} \overrightarrow{g}\ (\overrightarrow{x}) \geq 0 \\ \overrightarrow{h}\ (\overrightarrow{x}) = 0 \end{array} \right. \tag{5}$$

B. Multi-objective solutions

In most case, multi-objective problems do not have a single global optimal solution according to equation 4 and a new definition of minimizing $\overrightarrow{f}\ (\overrightarrow{x})$ has to be used. The concept of optimum changes, because in multi-objective optimization problems the purpose is to find trade-off solutions rather than a single solution. Thus to compare those solutions and determine which are useful, the well-known Pareto dominance is commonly used. Based on the work of Vilfredo Pareto (see [25]), it can be expressed as follows:

$$\overrightarrow{x}_i \succ \overrightarrow{x}_j \leftrightarrow \left\{ \begin{array}{l} \forall k \in [1, N], f_k(\overrightarrow{x_i}) \leq f_k(\overrightarrow{x_j}) \\ \exists k' \in [1, N] \mid f_k(\overrightarrow{x_i}) < f_k(\overrightarrow{x_j}) \end{array} \right. \tag{6}$$

In accordance with [9], this expression means that a given decision vector $\overrightarrow{x}_i$ dominates another one $\overrightarrow{x}_j$ if, and only if none of the corresponding objective function values $f_k(\overrightarrow{x_i})$ is worse than $f_k(\overrightarrow{x_j})$ and if there is a dimension in the criterion space where it is strictly better. Using such a definition, the Pareto optimal set P^* can be defined as the set of all non-dominated vectors (see [29]).

$$P^* \subset \mathbb{R}^n, \overrightarrow{x}_i \in P^* \leftrightarrow \nexists\ \overrightarrow{x}_j \in \mathbb{R}^n \mid \overrightarrow{x}_j \succ \overrightarrow{x}_i \tag{7}$$

The set of corresponding objective values in the criterion space constitutes the so-called Pareto front.
The aim of a multi-objective optimization algorithm is to find a good estimation of P^* noted $\hat{P}$ in accordance to some other concepts which can be linked to the problem. As stated in Deb's book [12], the quality of this estimation must be at least measured in terms of diversity of the distribution and spread along the front.

C. PSO core

The PSO is a population based algorithm which deals with swarm intelligence. Each particle in this swarm has a n dimensional vector used as a position in the parameter space. At each iteration, particles are moving using some core equations to compute their velocity and decide their movements. The main advantage of PSO is its simple implementation as it can be reduced to the two following equations (see [29]):

$$\begin{aligned} v_{i,t+1} = \quad & \omega.r_0.v_{i,t} + \\ & c_1.r_1.\left(p_{i,best} - x_{i,t}\right) + \\ & c_2.r_2.\left(p_{i,guide} - x_{i,t}\right) \end{aligned} \tag{8}$$
$$x_{i,t+1} = \quad x_{i,t} + \chi(v_{i,t+1}) \tag{9}$$

$x_{i,t}$ is the position of the i^{th} particle at time t, and $v_{i,t}$ its velocity. $p_{i,best}$ and $p_{i,guide}$ are respectively the best position (in term of optimization) that the current particle has found in its path and the position of a particle that has been chosen as a guide. The weights applied to those positions are called the individual and social factors because they respectively depend on the current particle memory of its own best position and on another particle position from the swarm. They are both weighted independently by a coefficient c_x and a random value r_x in $[0, 1]$. The particles will either tend to explore the parameter space or to further investigate around a previously found solution according to their variations. Thus they have a significant impact on the convergence. ω is the inertia weight which can be constant or time-dependant like in [36]. Large values of this parameter tend to make the particle following its last direction with a turbulence factor r_0 whose value is chosen in $[0, 1]$. A last part is modeled by the function $\chi()$. It is generally implemented as a simple factor known as the turbulence factor like in [20] and thus replacing the random part of the inertia weight. However some implementations use it as a velocity normalization function or a constriction factor, keeping the direction but avoiding speed divergence (see [23]).

D. From PSO to MOPSO

Only few modifications need to be made on the core algorithm to adapt it to multi-objective problems. These modifications are presented in algorithm 1. The main changes are to consider a criterion space of dimensions N and to compare the solutions offered by each particle. It increases the algorithm computation cost, but does not change its core. An elitist strategy should be engaged in order to remember only the good parameter combinations and therefore an archive has to be built. It retains only the particle position that can be included in $\hat{P}$, the current Pareto set estimation. In accordance to the cooperative approach in PSO, this system is called the collaborative memory.

```
Input: S_O an initialized swarm of particles
Output: P̂_{t_end} the archive of non-dominated particles
t = 0;
while stopping criterion not reached do
    Compute objective functions on (S_t);
    foreach particle p_{i,t} in S_t do
        * if current position x_{i,t} is best then
            p_{i,best} = x_{i,t};
        end
        * Select new guide p_{i,guide} in archive (p_{i,t}, P̂_t);
        Compute velocity (p_{i,t});
        Test constraints (p_{i,t});
    end
    Update particle position in swarm (S_t);
    * Update Archive (S_t, P̂_t);
    t = t + 1;
end
```

Algorithm 1: MOPSO pseudo-code with $*$ on features enhanced by our contributions (based on [28]).

Reyes-Sierra proposed a review of state-of-the-art MOPSO variants in [28]. A categorization of the various approaches is presented. It allows to point out that despite the youth of this research field, the variants of MOPSO proposed are very diversified. The most discriminative aspect is the strategy used to manage the multidimensionality of the solution. The simplest technique is to refine the problem through a single objective using aggregation methods (such as a weighted summarization) or to apply an ordering strategy on the different objectives. Sub-population approaches use multiple swarms, optimizing separately each objective but sharing information to propose a global set of solutions. However, as presented in the bibliography, a consensus seems to be established on Pareto dominance based approaches (or combination of approaches) which appear to have better performance (see [28] for a complete description of the MOPSO variants and references).

The study of existing MOPSO variants also allows to point out that dominance and guide selection strategy have a significant impact on the algorithm performance. Thus, our contributions, described in the next sections, are mainly focused on them.

III. An enhanced epsilon dominance and guide selection

In accordance to [28], the major difficulties in the adaptation of PSO to multi objectives problems are : (i) the guide selection (called the leader in the paper), (ii) the maintenance of the non-dominated solutions and (iii) the diversity of the swarm. Our contributions, described in the next sections, are mainly focused on the two first of them. Our proposal can be described as a Pareto dominance based one, using an external archive of non-dominated solutions and a density estimator to select the guide. Indeed, we propose a new guide selection strategy and a variation of the domination concept to ease the archive maintenance. The steps of the MOPSO algorithm impacted by such contributions are highlighted with stars in the algorithm 1.

A. Building the archive

As mentioned before, an archive of solutions eligible for the Pareto set has to be maintained. In order to determine if a particle should be included in the archive, the most common method has been to retain all non-dominated solutions in accordance to the Pareto dominance. The drawback of such an approach is the control of the archive size, which can quickly become very large and hard to maintain, whereas only some key values are needed to obtain a good Pareto Set description. Thus other strategies have to be found to limit the archive size while preserving its diversity and spread along the front.

The ε dominance introduced in [17] and evaluated in [19] presents good capabilities to tackle this problem. Two definitions exist based on the deviation type: absolute (additive ε see equation 10 from [17]) or relative (multiplicative ε see equation 11 from [18]). According to previous studies, the relative definition is commonly chosen as it permits to easily define the ε value and provides more results for smaller objective values.

$$\vec{x}_i \succ \vec{x}_j \quad \leftrightarrow \quad \begin{cases} \forall k \in [1,N], f_k(\vec{x_i}) + \varepsilon \leq f_k(\vec{x_j}) \\ \exists k' \in [1,N] \mid f_k(\vec{x_i}) + \varepsilon < f_k(\vec{x_j}) \end{cases} \tag{10}$$

$$\vec{x}_i \succ \vec{x}_j \quad \leftrightarrow \quad \begin{cases} \forall k \in [1,N], \frac{f_k(\vec{x_i})}{1+\varepsilon} \leq f_k(\vec{x_j}) \\ \exists k' \in [1,N] \mid \frac{f_k(\vec{x_i})}{1+\varepsilon} < f_k(\vec{x_j}) \end{cases} \tag{11}$$

The difference with the classic Pareto dominance can clearly be focused on the figure 1. As noticed in [18], this definition allows to quickly achieve an estimation of the Pareto front by modifying the domination area of a particle proportionally to its criterion values. It is one way to manage simultaneously the dominance between particles and the neighborhood in the criterion space and will yield a better diversity along the Pareto front.

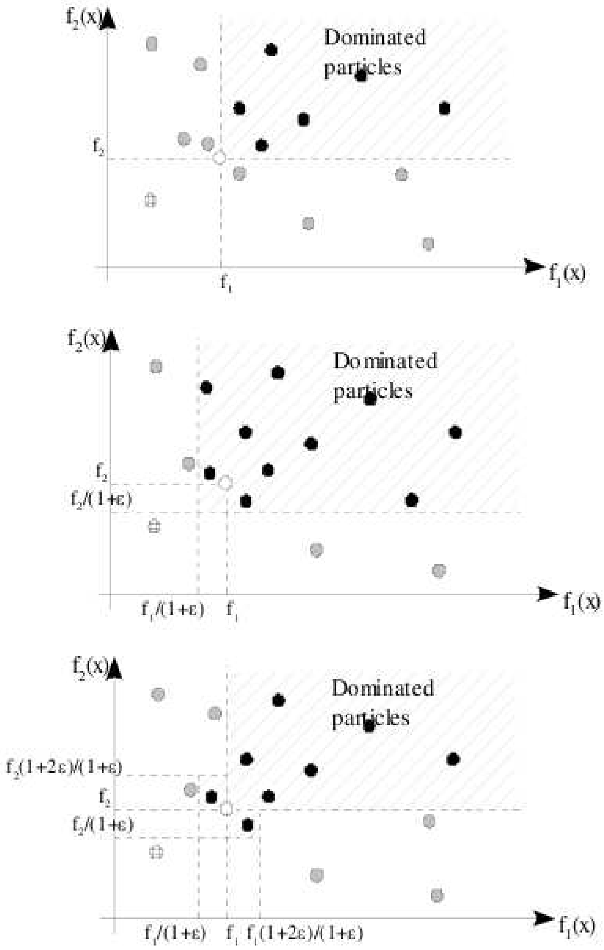

Figure. 1: Illustration of Pareto dominance, ε dominance and our ε dominance variant.

However, with such a definition, the difference with the Pareto domination area is larger for particle with bigger objectives values. This could induce a drawback as shown in figure 2 on a benchmark problem, where the domination area of the considered element ($p1$) limits the front description. Particular shapes of the Pareto front estimation (for instance areas with only minor variations on one objective and large variations on another) can thus be mistaken. This is a conse-

quence of the ε dominance definition, which limits the number of particles used to describe the extremes or the parts of the front where one of the criteria is almost constant.
Such a problem was noticed in [18], but surprisingly, no work exists in the literrature about the study of the effects on Pareto front results and no solution has been proposed to avoid this. In order to tackle this problem without involving a CPU greedy clustering method, we introduce an ε dominance variant. It limits the domination area introduced by the standard ε dominance to local neighborhood in order to avoid the limitations on large criteria value. The figure 1 presents a schematic illustration of this variant in comparison with Pareto and ε dominance. One can see that the classic ε dominance allows to handle the neighborhood of the considered particle (white one) in the objective space by extending the domination area. Thus closest solutions, which reduce the diversity of the solutions set, are removed. However, it also removes some other particles not present in the local neighborhood because of the global extension of the domination area. Using the ε dominance variant allows to limit such extension, keeping its benefits and avoiding the highlighted drawbacks.

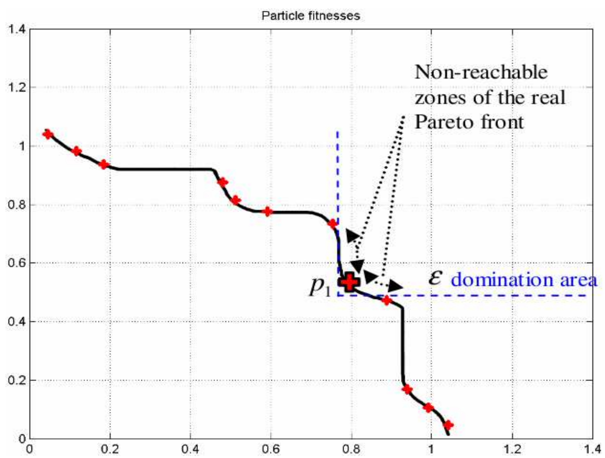

Figure. 2: Example of limitations introduced by ε dominance against an estimation of the Pareto front (black line). The highlighted zone will never be covered by new elements as they are under the ε domination area of already present elements (red crosses).

The principle of this variant is to use the implicit neighborhood management introduced by the ε dominance. The dominated neighborhood is proportional to ε (i.e. multiplicative ε) which is easy to implement and define. The mathematical formalization of such a variant is expressed in equation 12. The first part is simply the Pareto dominance whereas the second part defines the local domination areas in the neighborhood.

$$\vec{x}_i \succ \vec{x}_j \leftrightarrow \left\{ \begin{array}{l} \left\{ \begin{array}{l} \forall k \in [1, N], f_k(\vec{x_i}) \leq f_k(\vec{x_j}) \\ \exists k' \in [1, N] \mid f_k(\vec{x_i}) < f_k(\vec{x_j}) \end{array} \right. \\ OR \\ \left\{ \begin{array}{l} \exists k' \in [1, N] \mid \\ f_{k'}(\vec{x_j}) < f_{k'}(\vec{x_i}) < \frac{1+2\varepsilon}{1+\varepsilon} f_k(\vec{x_j}) \\ \forall k \in [1, N], \frac{f_k(\vec{x_i})}{1+\varepsilon} \leq f_k(\vec{x_j}) \end{array} \right. \end{array} \right. \quad (12)$$

This variant of ε dominance allows to overcome the problem mentionned above while maintaining the benefits of classical ε dominance. It keeps a good diversity while avoiding the maintenance of a complex data structure for the non-dominated particles induced by methods based on clustering. Such criterion space clustering approaches have been largely tested in [10] with the hypercube strategy, in [21] with the sigma method or in [14] with the dominated trees. The advantages of our variant will be highlighted in the experimentations presented in section IV.
As it is presented in the papers mentioned above, the maintenance of the archive of the non-dominated particles is strongly linked to the guide selection which is one of the core step of the MOPSO. Thus we also contribute on the guide selection behavior.

B. Guide selection behavior

Performance of PSO algorithm depends on the factors which will influence each particles movement through the core equation 8. The particle will be influenced by its previous position, which is regulated through the inertia factor, its personal memory $p_{i,best}$ and a guide $p_{i,guide}$. Between the numerous possible implementations of personal memory influence, we choose to select the last non-dominated position of the particle to be the individual memorization of its best position. [5] has shown that more complex strategies can provide small improvements, but this approach (called newest strategy in [5]) allows good performance with a very small computational cost.
Then the most important factor is the global guide who will try to help the particle to find to the Pareto front by modifying its trajectory. According to [28], the guide has to be selected in the archive of non-dominated solutions. Nevertheless the selection heuristic can drastically change the swarm convergence behavior.
Our approach is based on the use of a probabilistic framework since it has shown to have better performance in [1]. The idea is to select each particle guide through a roulette wheel selection where each non-dominated solution will have a different selection probability evaluated at each iteration. However, instead of using a computation based on the Pareto domination to determine the probability, we use a local density evaluation in order to tend the swarm to fill the holes of the current Pareto front estimation. Thus for each archive member, the probability is computed as an inverted density measure on its local neighborhood in the criterion space. Such an approach has also been tested in [5] for local best selection with quite good results. A similar approach can also be found in [2] but unfortunately without any further detail on the chosen estimator. However, the choice of the density measurement is not trivial because some particular shapes of the Pareto front or specific constraints can introduce discontinuities. A classic density measure, based on the counting of particles in a fixed area around the current archive element, will be biased by configurations similar to figure 2 : the area could be almost empty because of the front discontinuity. We propose a simple and intuitive solution which provides density estimation on an adaptive local neighborhood. It computes the sum of the inverted dis-

tances between the current particle and its K nearest neighbors. Then the selection probability is computed by inverting this estimation and normalizing it as a probability as shown hereafter (where ψ is the set of the K nearest neighbors of the current particle in the criterion space).

$$D(x_i) = \sum_{x_j \in \psi} (x_i - x_j) \qquad (13)$$

$$then \qquad P(x_i) = \frac{D(x_i)}{\sum_{x_k \in \psi} D(x_i)}$$

According to equation 13, a particle with closest neighbors will have an important local density evaluation and thus a small selection probability.

The last problem to solve is the choice of a decision rule for changing the guide of a particle. Indeed the guide selection strategy has a computational cost. Moreover if the particles change their guide too often (at each iteration) their movements cannot be really influenced by their guide and the social effect can be lost. In mono-objective optimization, this behavior is not a problem because the new guide should always be better than the previous one. However in MOPSO, guides are equivalent since they're all included in $\hat{P}$. This problem is partly solved by using complex swarm clustering (for example by sub-swarming on each criterion, see [28]), but we propose a more simple technique: enabling a particle guide memorization. Indeed, we did not find any studies on a guide memorization influence. Thereby the guide selection step, highlighted by a star in Algorithm 1, is modified. This is described in Algorithm 2.

Input: a particle p and
$\hat{P}_t$ the current archived of non-dominated particles
Output: a particle p_{guide} out of $\hat{P}_t$
if p *current's position is best position* **then**
  no guide selection;
else
  r = random([0,1]);
  if $r > p$*'s memory threshold*
  AND current guide in $\hat{P}_t$ **then**
    keep current guide;
  else
    select guide($\hat{P}_t$);
  end
end

Algorithm 2: Enhanced guide selection pseudo-code with memory threshold.

The idea is to allow a particle p to keep its previous guide in particular case. To avoid the swarm to only explore locally the front because of the stronger influence of guides, a particle which has been recently added in the archive (which means, when it reaches a non-dominated position) does not select any guide. This is the reason of the first test in the algorithm. In such case, p can be considered to be a pioneer and it is assumed that it does not need any guide. It is completely free to explore any part of the parameter space using only its personal best position and its inertia. In the other case, the particle uses a new characteristic added to the swarm: a guide memory threshold which will define a global behavior of guide memorization. A new guide will be selected for this particle only if its threshold is exceeded as shown hereafter (i.e. the particl remembers its guide) and if its previous guide has not been deleted from archive.

The main advantage of this implementation is that the memorization is under control with the threshold. Experimentations have been conducted on the standard problems in order to select a good trade-off for this new parameter. The obtained results are presented in the following section.

IV. Evaluation on standards problems

In this section, benchmark problems are used in order to validate our approach against a baseline MOPSO with basic implementation.

A. Evaluation strategies

1) Algorithm setting

As explained in [26] and theoretically studied in [35], the numerous parameters of a PSO algorithm can be adapted to maximize the convergence on each problem. However our experimental approach was to select values which present a good trade-off in order to have a problem-free implementation. As the aim was to study the performance of our contributions concerning dominance and guide selection, there was no need for fine tuning of these parameters. Thus they have been uniformly chosen in controled domains which best fit the state of the art advices (see [26] and [28]):

- Inertia weight ωr_0 in $[0.8; 1.0]$
- Individual cognitive factor $c_1 r_1$ in $[1.6; 1.8]$
- Social cognitive factor $c_2 r_2$ in $[1.4; 1.6]$
- The constriction function $\chi()$ implemented as a velocity threshold: when a dimension of the velocity vector exceeds the threshold, the whole vector is normalized such as the global direction is kept. Thus it constricts the velocity when it has a dimension greater than 0.1 (with criteria values normalized in $[0; 1]$).

This approach can be linked to [27]. However we limit the scales for the social and individual cognitive factors to different values since it has shown a statistically significant improvement in mono-objective PSO (see [35]) and in our multi-objective studies. We chose to introduce the uniform randomization through the specified domain instead of using secondary random factor r_x in order to control their variability. The swarm size was limited to 40 elements in order to offer a good trade-off between the number of potential solutions at each iteration and the update rate of the swarm. The number of iterations is not fixed and depends on the problems. For performance comparisons on the experiments, our stopping criteria was a limitation on the number of objective function evaluations, empirically fixed in order to obtain an acceptable estimation of the Pareto front.

2) Benchmark problems

Four problems from the literature have been chosen for the experiments. The first one is BNH, or also called MOPC1

Table 1: Benchmark functions ($f()$) and constraints ($g()$).

Name	**Criteria/constraints**
BNH	$f_1(\overrightarrow{x}) = 4x_1^2 + 4x_2^2$
	$f_2(\overrightarrow{x}) = (x_1-5)^2 + (x_2-5)^2$
	$g_1(\overrightarrow{x}) \equiv (x_1-5)^2 + x_2^2 \leq 25$
	$g_2(\overrightarrow{x}) \equiv (x_1-8)^2 + (x_2+3)^2 \geq 7$
	$x_1 \in [0,5] \quad x_2 \in [0,3]$
MOP5	$f_1(\overrightarrow{x}) = \frac{x_1^2+x_2^2}{2} + \sin\left(x_1^2+x_2^2\right)$
	$f_2(\overrightarrow{x}) = \frac{(3x_1-2x_2+4)^2}{8} + \frac{(x_1-x_2+1)^2}{27} + 15$
	$f_3(\overrightarrow{x}) = \frac{1}{x_1^2+x_2^2+1} - 1.1e^{-x_1^2-x_2^2}$
	$x_1 \in [-30,30] \quad x_2 \in [-30,30]$
MOP6	$f_1(\overrightarrow{x}) = x_1$
	$f_2(\overrightarrow{x}) = (1+10x_2)\left[1 - \left(\frac{x_1}{1+10x_2}\right)^2 - \frac{x_1\sin(8\pi x_1)}{1+10x_2}\right]$
	$x_1 \in [0,1] \quad x_2 \in [0,1]$
TNK	$f_1(\overrightarrow{x}) = x_1$
	$f_2(\overrightarrow{x}) = x_2$
	$g_1(\overrightarrow{x}) \equiv x_1^2 + x_2^2 - 1 - 0.1\cos\left(16\arctan\left(\frac{x_1}{x_2}\right)\right) \geq 0$
	$g_2(\overrightarrow{x}) \equiv (x_1-0.5)^2 + (x_2-0.5)^2 \geq 0.5$
	$x_1 \in [0,\pi] \quad x_2 \in [0,\pi]$

(see [3]). It is considered to be simple because constraints do not introduce serious difficulties in finding the Pareto set and the front does not have any discontinuity or complex convexity. The MOP5, proposed by Viennete, and MOP6 (see [6] for complete references) are two unconstrained problems used to test optimization algorithms against two major difficulties: an increase of the criterion number and a discontinued Pareto front. Then the last problem, called TNK by Tanaka [33], is considered to be quite difficult because of the restriction of the solution space introduced by the constraints. The descriptions of the mathematical functions, as they have been implemented, are shown in table 1.

3) Metrics

Comparing different executions of two multi-objective algorithms is a very complicated task. However, in our case, we only need to compare different variants of the same algorithm. Thus we use only simple metrics to compare the spread and diversity of the front obtained by each implementation.

The spacing metric S (see [30]) measures the homogeneity of the front description by computing the mean distance between each element of the Pareto set estimation. Thus small values are better than large ones. A null value means that the elements are equidistant. This limit cannot be reached with the relative implementation of the ε dominance because of its intrinsic definition which introduces a neighborhood limitation relative to the criterion value. The maximal extension D simply measures the diagonal between the extremes elements on each criterion and must be maximized in order to cover the entire front. Then the set coverage SC proposed in [37] tries to evaluate the domination of a Pareto front estimation $\hat{P}_A$ against another one, $\hat{P}_B$, by counting the number of elements of $\hat{P}_B$ which are dominated by a least one element of $\hat{P}_A$. By definition if $SC(\hat{P}_A, \hat{P}_B) = 1$ and $SC(\hat{P}_B, \hat{P}_A) = 0$ we can say that the estimation $\hat{P}_A$ is better than $\hat{P}_B$. They were respectively computed as presented in equations 14, 15 and 16 with normalized objective values.

Table 2: Metrics for dominance comparison (left columns results for MOPSO baseline with ε dominance and right with enhanced ε dominance).

	BNH		MOP5		MOP6		TNK	
Objectives evaluations	4000		2000		4000		4000	
Archive size	17.6	**58.4**	66.1	47.3	9.68	**23.9**	10.4	**26.8**
Spacing metric	3.55	**3.2**	0.04	43.4	0.19	**0.12**	0.09	**0.03**
Maximal extension	99.7	**105**	1.55	**65.8**	0.97	**1.05**	1.30	1.30
Set coverage	0.98	0.96	0.12	1	0.92	0.9	0.97	0.95

$$S = \sqrt{\frac{1}{|\hat{P}|}\sum_{i=A}^{|\hat{P}|}\left(d_i - \overline{d}\right)^2} \tag{14}$$

$$with \quad d_i = \min_{\overrightarrow{x_i}\in\hat{P}\wedge k\neq i}\sum_{n=1}^{N} \mid f_n(\overrightarrow{x_i}) - f_n(\overrightarrow{x_k}) \mid$$

$$D = \sqrt{\sum_{n=1}^{N}\left(\max_{\overrightarrow{x_i}\in\hat{P}} f_n(\overrightarrow{x_i}) - \min_{\overrightarrow{x_k}\in\hat{P}} f_n(\overrightarrow{x_k})\right)^2} \tag{15}$$

$$SC(\hat{P}_A, \hat{P}_B) = \frac{\mid \left\{x \in \hat{P}_B \mid \exists y \in \hat{P}_A : y \prec x\right\} \mid}{\mid \hat{P}_B \mid} \tag{16}$$

As the algorithm involves random values in its execution, many differences can appear in two different runs. Thus in our experimental protocol, the different configurations of MOPSO used the same initial swarm with random position vectors assigned in the parameter space. Then we repeat 100 times the execution (with different initial swarms) of each implementation of the algorithm. Our aims were to obtain a good estimation of the general algorithm behavior and to enable statistical estimators computation for each metric at each iteration.

The computional cost involved by the enhancement of neighborhood and guide selection was evaluated both on benchmark and real life problems. It appears that the most critical point was the objective computations and that the computational overload in comparison to the baseline was not significative. Thus it has not been studied in the following results.

B. Results and discussion

1) Dominance

We compare the ε dominance variant to the ε dominance classically used in MOPSO on the benchmark problems. Table 2 presents the metric mean values over all executions of our approach (in the right columns and bolded when there is some improvement) against standard ε dominance approach (in the left columns). As the set coverage is a non-symmetric binary measure, we present both the results of our approach against the standard and the standard against our variant.

The results summarized in table 2 must be carefully interpreted. First of all we can see that MOP5 is a problem that highlights the standard ε dominance drawbacks. Since one of the objectives has small variability, the front is extended on very high values. The limitation introduced by the standard ε dominance does not allow to describe those parts and thus the final estimation is very different (and worst) than the one obtained with our variant. Closely considering the set

coverage allows a better understanding of the situation: the dissymmetry on the metric implies that all the elements from the Pareto front estimated with our dominance variant dominates the ones from the other approach estimation.
The consequence of this is the large differences on the other metrics: the maximal extension is clearly improved and the spacing metric values are not comparable since the objective values are too different. So on this particular problem, our variant allows to perform a better (or faster) estimation of the Pareto front.

For the other problems, one can observe that the set coverage metrics of both approaches are quite similar and thus we can conclude that the Pareto front estimations are both near the real Pareto front (or near the limit of the algorithm capacities for the number of iterations). As the archive size is always significantly improved by our approach, we can argue that it generally permits to obtain a finer description of the front. This is confirmed by the spacing metric which is also improved and proves that the results are well distributed along the front. Finally, we provide the maximal extension in a specific way in order to allow a better interpretation. The evaluation has been made not on the final front estimation on each runs but on the filtered front. It means that the archive obtained with one approach is reduced by removing all the elements that are dominated by at least one element from the other approach archive. We choose this method because some front estimations contain incorrect elements which corrupt the maximal extension value. The results show that if our approach appears to yield less satisfactory results at first, it is only due to the presence of dominated solutions in the other estimation. Thus its maximal extension artificially grows because of such false Pareto front estimation. This particular difficulty on the metric interpretation highlights the difficulty of quantitative comparison.
A more thorough comparison requires a qualitative observation of the estimated Pareto front. As seen in Figure 3, the quality of the front is clearly enhanced with our variant: the extremes are better described and the description of parts where a criterion is almost invariant is also enhanced. This is highlighted on MOP5, where the classic ε dominance does not allow describing the right part of the front because of the particular shape of the Pareto front.
It is obvious that the classic ε dominance can also tackle those problems by reducing the epsilon value and allow more elements to be included in the archive. But other parts of the front which are well described will also suffer from this by more and more elements inclusion and thus the archive size bounds can be quickly broken. Moreover, it will not resolve the problem involved in ‘flat‘ parts of the front as our approach can do.

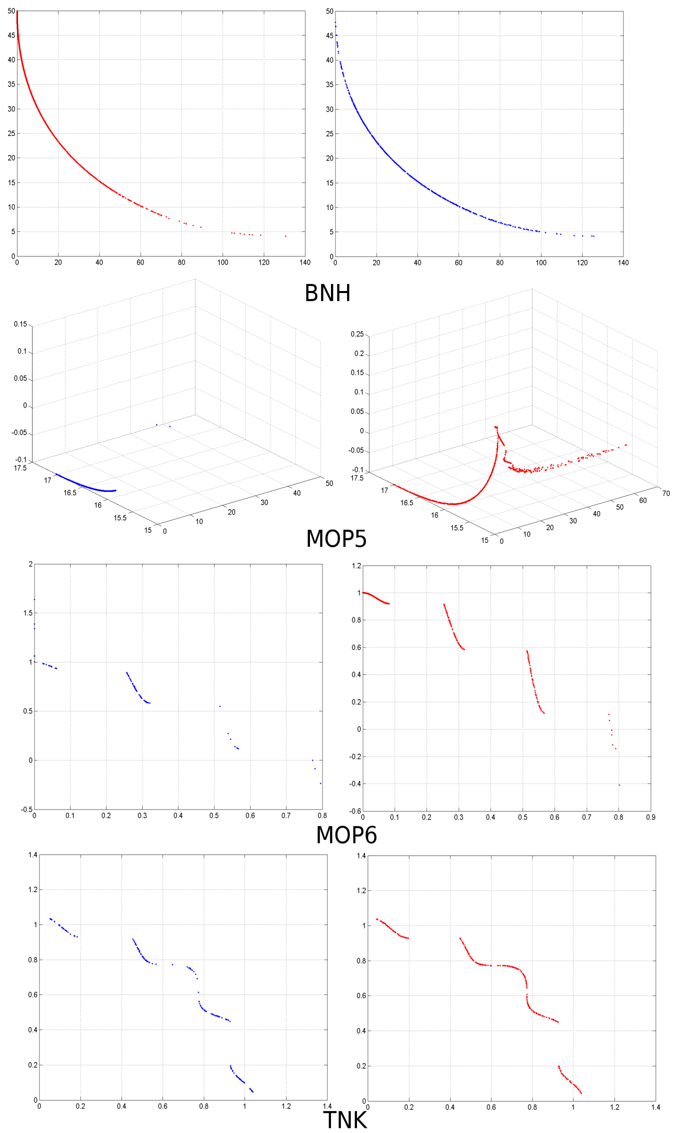

Figure. 3: Dominance comparison on the benchmark problems (the left blue front is for the standard ε dominance and the right red one for our variant).

Table 3: Metrics for guide selection behavior comparison (left columns results for MOPSO baseline with random guide selection and right with enhanced guide selection).

	BNH		MOP5		MOP6		TNK	
Objectives evaluations	4000		2000		4000		4000	
Archive size	55.6	44.8	111	104	18.7	**21.5**	27.5	**27.8**
Spacing metric	2.95	**2.55**	29.9	58.9	0.24	**0.13**	0.03	**0.02**
Maximal extension	108	**121**	1357	**1681**	2.59	2.08	1.29	**1.30**
Set coverage	0.97	**0.99**	0.91	**0.99**	0.95	**0.96**	0.97	0.97

2) *Guide selection strategy*

Both configurations in this study use the proposed enhanced ε dominance. Their differences are only on the guide management: the first uses a full random selection and no guide memorization whereas the other involves the density based probability to select the guide that can be kept through the next iteration. The number of neighbors was experimentally limited to 4 and the memory factor to 0.6 as it appears to be the most effective values in our experiments (not presented here). Figure 4 shows the evolution of the different metrics through the iterations on each problem. Table 3 presents the mean improvements over all executions of our approach (right columns) against random selection (left columns).

BNH: The improvement is not obvious on BNH tests. Such a result is quite logical since the objective functions are quite simple and do not need a strong strategy to allow a good estimation of the Pareto set. Improvements of the front diversity can be seen but through a reduction of archive size.

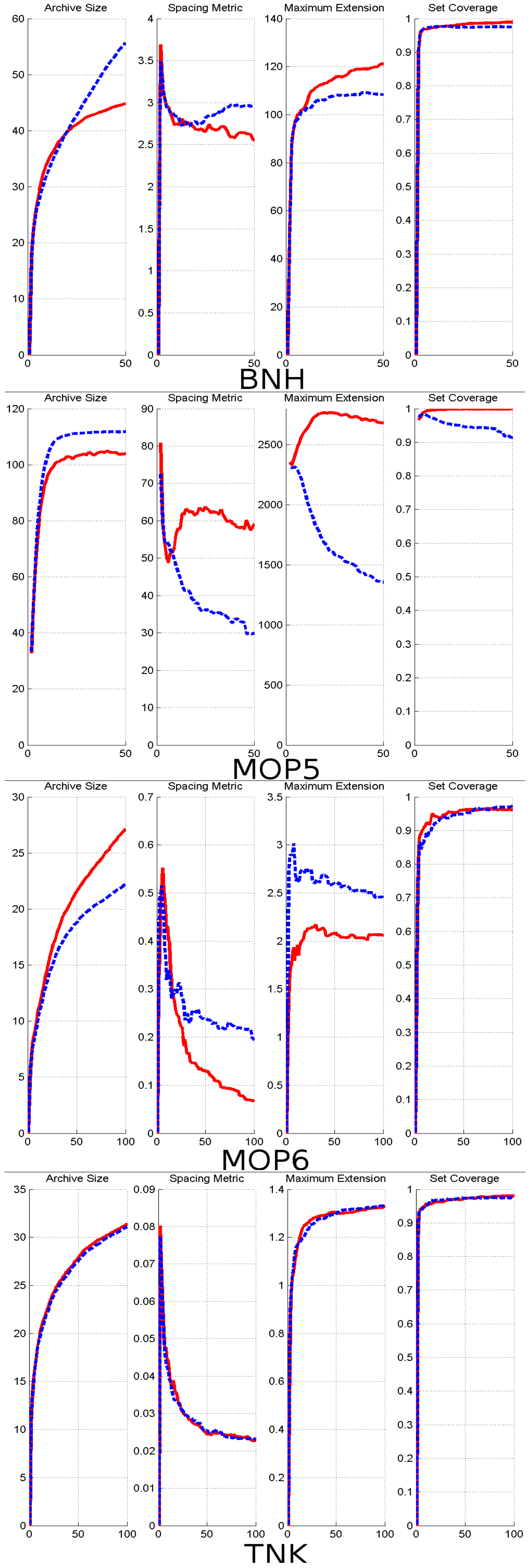

Figure. 4: Evolution of metrics through iteration on different problems (means values for standard guide selection in blue dashed lines and our variant in red lines).

MOP5: The performance of our approach must be well interpreted for this problem. As shown by the dynamic evolution of the metric in figure 4, the results are biased. Indeed after about 20 iterations the values of metrics fall drastically for the random selection. The reason is that the front of this problem is particularly difficult to find as it has a lot of local optimal solutions as explained previously. This is confirmed by the evolution of the set coverage and maximal extension which allow concluding that the front estimated by the probabilistic approach is quite better.
MOP6: The solution is significantly improved by our approach on MOP6 tests. It is quite obvious that this particular problem, which contains much discontinuities on its Pareto front, is better solved by our enhanced guide selection behavior. The only exception is the maximal extension. The reasons are the same as in the precedent study on dominance.
TNK: The problem involves a lot of hard constraints which strongly limit the parameter space. Thus our approach based on a density estimator evaluated in the criterion space does not improve the global results since it does not permit to tackle the specific difficulties introduced in this problem.
Such results can be difficult to analyze since some behavioral particularities are kept undetected even when using several metrics. Thus, we interpret the values as relative improvement in order to facilitate the analysis on each problem. The classical qualitative evaluation of the Pareto front has also led us these interpretations. With respect to all the measures, we can conclude that our approach obtained a significant improvement in most cases. As we saw, the higher improvement is reached with difficult problems (i.e. with discontinued front) without strong constraints. However such results are limited to the context of our experiments, which is the comparison between different MOPSO approaches on standard problems. Thus we have also tested our MOPSO in a real world environment against an evolutionary algorithm.

V. SVM model selection using the proposed MOPSO

This section proposes an original application of the proposed MOPSO for tuning the hyperparameters of a classifier. Such a problem is a critical step for building an efficient classification system as this crucial aspect of model selection strongly impacts the performance of a classification system. For a long time, this problem has been tackled using a mono-objective optimization process, with the predictive accuracy or error rate as objective. Now, it is well-known that a single criterion is not always a good performance indicator. Indeed, in many real-world problems (medical domain, road safety, biometry, etc...), the missclassification costs are (i) asymetric as error consequences are class-dependant ; (ii) difficult to estimate, for instance when the classification process is embedded in a more complex system. In such cases, a single criterion might be a poor indicator. Since the works of Bradley [4] concerning the Receiver Operating Characteristics (ROC) curve, classifier model selection has been implicitly considered to be a multi-objective optimization problem, particularly in the context of a two-class classification problem. Indeed, a classifier ROC curve represents the set of trade-offs between False Rejection (FR) and False Acceptance (FA) rates (also known as sensitivity vs. specificity trade-off). As a consequence, some approaches have been proposed in order to choose the classifier hyperparameters using the ROC curve as a performance indicator. Unfortunately, these approaches are always based on a reduction of the FR and FA rates into a single criterion such as the Area Under Curve (AUC) or the FMeasure (FM).

In this section, classifier hyperparameters tuning is explicitly considered to be a multi-objective optimization problem aiming at optimizing simultaneously FA and FR. It is tackled using the proposed MOPSO optimizer. Consequently, the aim is to use the proposed MOPSO to find a set of classifiers in order to select the best set of FA/FR trade-offs. Such a strategy is evaluated on data extracted from a real-world application which takes place in the context of a handwritten digit/outlier discrimination problem.

On can note that some other combinations of SVM classifier and particle swarm optimization (limited to mono-objective optimization) can be found in the litterature with different approaches. Two examples can be found in [32] and [24]. In the first one, the PSO is used to select the characteristics (genes in a tumor classification problem) exploited by the SVM classifier and thus appears as a very efficient preprocessing module in the overall classification system. And in the second one, a Modified PSO called the Converging Linear Particle Swarm Optimizer is proposed to replace the traditionnal learning algorithm. Tested against baseline algorithms on the handwritten characters database from MNIST, it has shown to have similar capabilities. In both studies, an original combination is proposed and promising results are presented. The following sections will describe our own proposal.

The application is quickly described in subsection V-A, in order to justify our choices. The SVM classifier used and its optimization strategy are described in subsection V-B. Finally, obtained results are presented and discussed in V-C.

A. Digits/outliers discrimination

The work described in this section is part of the design of a more complex system which aims at extracting numerical fields (phone number, zip code, customer code, etc.) from incoming handwritten mail document images. The proposed approach is applied to a particular stage of this numerical field extraction system [7]. More precisely, the classifier to be optimized is used as a fast two-class classifier which has to identify the digits among a huge number of irrelevant shapes (words, letters, fragments of words, etc). Consequently, the classifier objective is to reject as many outliers as possible, while accepting as many digits as possible. However, rejecting a digit has a much more serious consequence than accepting an outlier. The rejected data will never be processed and thus a numerical field can be lost. If a non-digit is accepted, it will increase the computation cost on non-relevant data. This problem is a good example of a classification task with asymmetric and unknown misclassification costs since the influence of a FA or a FR rate on the whole system results is unknown a priori.

Concerning the classifier to be optimized, the Support Vector Machines classifier has been chosen for its well-known efficiency in a two-class context.

B. SVM classifier and optimization strategy

Support Vector Machines are a well-founded and largely used learning machine algorithm which have been proved to be very effective on several real-world problems. In order to take into account asymmetric misclassification costs, we adopt the strategy proposed in [22] that consists in the introduction of two distinct penalty parameters C^- and C^+ (also called positive and negative margins).

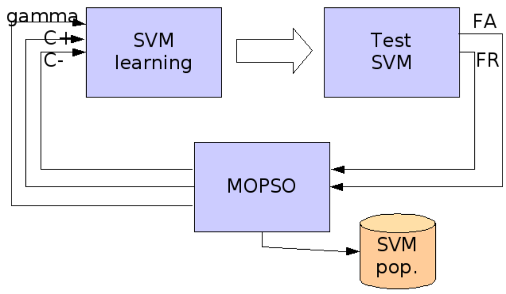

Figure. 5: Schematic view of the SVM optimization strategy through MOPSO.

In such a case, given a set of m training examples x_i belonging to the class y_i, the classical maximization of the dual Lagrangian with respect to the α_i becomes:

$$\max_{\alpha}\left(\sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j \kappa(x_i, x_j)\right) \quad (17)$$

subject to the constraints :

$$\begin{cases} 0 \leq \alpha_i \leq C^+, for\, y_i = 1 \\ 0 \leq \alpha_i \leq C^+, for\, y_i = 1 \\ \sum_{i=1}^{m}\alpha_i y_i = 0 \end{cases}$$

Where α_i denotes the Lagrange multipliers, C^- and C^+ are respectively the cost factors for the two classes (-1) and $(+1)$, and $\kappa(x_i, x_j)$ denotes the kernel transformation. In the classical case of a Gaussian (RBF) kernel, $\kappa(x_i, x_j)$ is defined as:

$$\kappa(x_i, x_j) = e^{-\gamma \times ||x_i - x_j||^2} \quad (18)$$

In accordance with [8], we choose to keep the intrinsic optimization of support vector in SVM using the Lagrangian maximization and we apply the optimization process to the classifier hyper-parameters. Hence, our optimization parameters are:

- the kernel parameter of the SVM-rbf : γ
- the penalty parameters introduced above: C^- and C^+.

As explained before, the criteria to be optimized are both the FA rate and the FR rate which are obtained by testing the hyperparameters set on a test database. The proposed strategy is illustrated on figure 5.

C. MOPSO on SVM experimentation and comparison

In this section, the experimental results obtained using the approach shown on figure 5 are presented and discussed. Two kinds of tests are presented. The first one aims at showing the interests of our MOPSO improvements. The second one consists in a comparison of the proposed MOPSO with respectively a state of the art multi-objective algorithm (NSGA-II [11]) and a classic SVM model selection approach.
Our first comparison has been made against a baseline MOPSO (standard ε dominance and random guide selection) in order to ensure that our contributions concerning MOPSO are efficient on a real world problem. The comparative results are presented on figure 6. As one can see, the problem does not appear to be difficult. The Pareto front estimation does not contain any discontinuity. However the gain of our contributions can be clearly observed. The standard MOPSO mainly focusses its search on the middle part of the front and has a poor description of the extremes. The results obtained using our approach are quite better. One can be observed a better homogeneity of the description and well defined extremes parts.

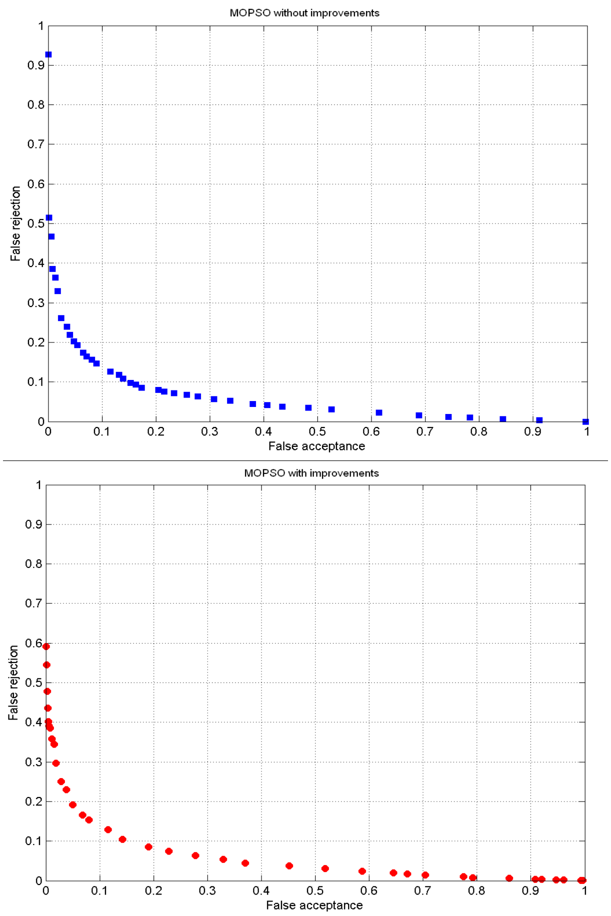

Figure. 6: Final Pareto Front estimation for both baseline MOPSO (up) and enhanced (down) MOPSO.

The second test concerns a comparison between the proposed MOPSO and a state-of-the-art MOEA: the NSGA-II (report to [11] for a complete description). As the approach differs from ours, some adaptations have been needed to

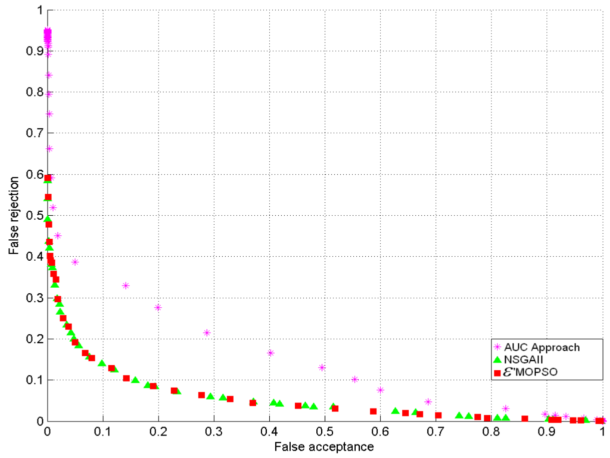

Figure. 7: Final Pareto Front estimation for both approaches (NSAGII in green and enhanced MOPSO in red).

offer a fair comparison. The most important parameter is the archive size which is limited to the initial population size in NSGA-II. Thus our MOPSO implementation was modified in order to limit its archive size. Using such a limitation, ε value was dynamically computed with a specific heuristic in order to rebuild the archive. Both algorithms were ran using the same population size (40) for a limited number of objective evaluations (1000). Such values appear as good trade-offs between the running time and the quality of the final Pareto set estimation. The results obtained are shown on figure 7 for the Pareto front estimation and on figure 8 for the metrics previously introduced.
One can note that we also introduce on figure 7 the results obtained using a classical SVM model selection called SVM-perf [34]. This approach has been configured to use the Area Under the ROC curve (AUC) as a single criterion during the classifier learning.

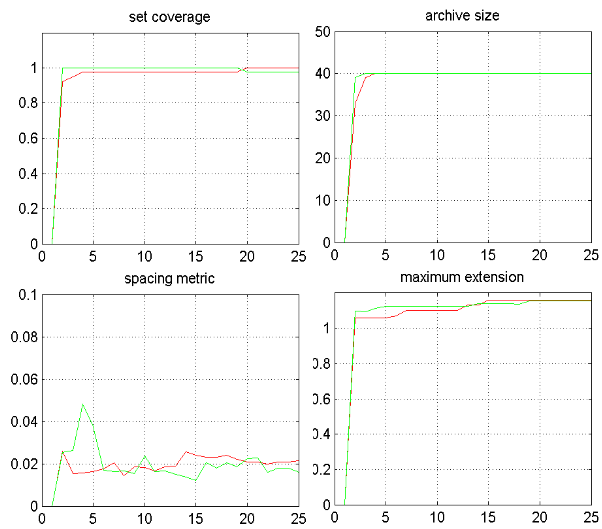

Figure. 8: Comparative values of metrics (NSAGII in green and enhanced MOPSO in red).

One can observe on figure 7 that both MO approaches allow a major improvement of the classic optimization w.r.t. SVM-

perf approach. Of course, such a comparison is not fair from a theoretical point of view since we compare a ROC curve obtained using a single parameterized classifier (using AUC as building criterion) with an approach that considers a set of classifiers. Nevertheless, from a practitioner point of view, these results aim at justifying the use of a multi-objective optimization framework in the context of SVM model selection. Indeed, for a chosen FA/FR trade-off, our framework provides a solution to the practitioner which is better than the solution obtained using a single classifier with a given output threshold.
Concerning the comparison of our approach with NSGA-II, the qualitative analysis proposed on figure 7 does not conclude to any dominance between the two multi objective optimizers. The quantitative comparison of metric values confirms this idea. The Figure 8 presents their variations per iteration and shows that both approaches obtain similar values very quickly. Thus the two approaches are quite competitive and perform both well on this problem. Such a result is quite interesting as it shows that our MOPSO implementation can compete with the state-of-the-art MOEA.

VI. Conclusion and further works

This paper introduces two contributions on two intrinsic difficulties faced when adapting the PSO to multi objective optimization: the archive and social guide management. Our variant on ε dominance enables a fast neighborhood management in criterion space and has proved to well maintain the diversity in the archive. Then our guide selection strategy and guide memorization have shown to allow the Pareto front estimation to be enhanced in its difficult parts. The validation of such methods has been made both on standard and real world problems and against a state-of-the-art multi objective optimizer. Our approach appears to be competitive and reliable.
Managing neighborhood, in order to avoid premature convergence and to promote a good spreading of solutions on the Pareto front estimation, is an open problem and several authors have proposed ideas to tackle this problem. This paper proposes an approach which has proven its low computational cost and its performance on a set of problems. A comparison with other proposal remains to be made in a near future.
However, what we tried to prove here was that our implementation allows obtaining a better Pareto set estimation than others using the classic ε dominance. Our proposition on the guide selection allows studying the guide memorization, a topic rarely discussed in other studies. It has shown to allow a significant improvement while keeping the MOPSO performance at the state-of-the-art level on a real world problem. Thus our approach appears as a good improvement to easily handle neighborhood in criterion space.
Much more experiments can then be conducted in order to compare to more MOPSO implementations. But before this, other improvements can be studied to go beyond the ones proposed in this paper. In particular, after proposing a new guide selection strategy, we are looking on the personal best management and selection which is the most natural continuation of our researches. The problem of the extremes handling, which has been partly solved by the neighborhood management, is always present because of the bias introduced by the relative ε dominance. This will also be one of the next big steps of our future work. The management of algorithms parameters also needs to be finer studied and our aim is to reduce the number of algorithm parameters (some successful tests have been conducted on an auto adaptive ε). Then, the neighborhood has to be enlarged to the parameter space. It will avoid a guide to be selected when it will add to much turbulence to its movements because its parameters combinaison is too different from the guided particle.
We also want to adapt our experimental approach to a more realistic environment in order to ensure the usability of our particle swarm optimizer. Some experiments will be conducted by considering the kernel choice as a new parameters in the optimization process for SVM model selection. This induces heterogeneity in the parameters but it can be tackled by MOPSO without too many difficulties. This research path is particularly valuable since it really helps the engineers to design their systems which have several heterogeneous parameters. Finally, we plan to enlarge our set of applications in terms of system complexity and domains. Information retrieval systems will be our most promising research paths especially for information extraction tasks through linguistic patterns which involve many parameters.

References

[1] J.E. Alvarez-Benitez, R.M. Everson, and J.E. Fieldsend. Mopso algorithm based exclusively on pareto dominance concepts. *Third International Conference on Evolutionary Mutli-Criterion Optimization*, pages 726–732, 2005.

[2] Alexandre M. Baltar and Darrell G. Fontane. A generalized multiobjective particle swarm optimization solver for spreadsheet models: application to water quality. In *AGU Hydrology Days 2006*, March 2006.

[3] To Thanh Binh and Urlich Korn. MOBES: A multiobjective evolution strategy for constrained optimization problems. In *The Third International Conference on Genetic Algorithms (Mendel 97)*, pages 176–182, Brno, Czech Republic, 1997.

[4] Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern-Recognition*, 30:11451159, 1997.

[5] Jürgen Branke and Sanaz Mostaghim. About selecting the personal best in multi-objective particle swarm optimization. In *Parallel Problem Solving from Nature*, volume 4193 of *Lecture Notes in Computer Science*, pages 523–532. Springer, September 2006. ISBN=3-540-38990-3.

[6] Leticia Cagnina, Susana Esquivel, and Carlos A. Coello Coello. A particle swarm optimizer for multi-objective optimization. *Journal of Computer Science & Technology*, 5(4), 2005.

[7] Chatelain Clément. *Extraction de squences numriques dans des documents manuscrits quelconques*. Phd thesis, University of Rouen, December 2006.

[8] Chatelain Clément, Adam Sébastien, Lecourtier Yves, Heutte Laurent, and Paquet Thierry. Multi-objective optimization for svm model selection. In *ICDAR07 - to be published*, 2007.

[9] Carlos A. Coello Coello. *Evolutionary Multi-Criterion Optimization: First International Conference*, volume 1993/2001 of *Lecture Notes in Computer Science*, chapter A Short Tutorial on Evolutionary Multiobjective Optimization, page 21. Springer Berlin / Heidelberg, emo edition, 2001.

[10] Carlos A. Coello Coello and Maximino Salazar Lechuga. A proposal for multiple objective particle swarm optimization. *Computational Intelligence*, pages 12–17, May 2002.

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm : Nsgaii. *IEEETransactions on Evolutionary Computation*, 6:182197, 2002.

[12] Kalyanmony Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons, 2001. ISBN 047187339X.

[13] David E. (edward) Goldberg. *Genetic algorithms in search, optimization & machine learning*. Addison-Wesley Publishing Co. - Reading, Mass, 1989.

[14] J. Fieldsend and S. Singh. A multi-objective algorithm based upon particle swarm optimisation. *In The 00 U.K. Workshop on Computational Intelligence*, pages 34–44, 2002.

[15] J. Kennedy and R. Eberhart. Particle swarm optimization. *Neural Networks, 1995. Proceedings., IEEE International Conference on*, 4:1942–1948, 1995.

[16] N. M. Kwok, D. K. Liu, and G. Dissanayake. Evolutionary computing based mobile robot localization. *Engineering Applications of Artificial Intelligence*, 19(8):857–868, December 2006.

[17] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *MIT Press in Evolutionary Computation*, 10, n3:263–282, 2002.

[18] Sanaz Mostaghim and Jürgen Teich. The role of ε-dominance in multi-objective particle swarm optimization. In *Proc. CEĆ03, the Congress on Evolutionary Computation*, volume 3, pages 1764–1771, Canberra, Australia, December 2003.

[19] Sanaz Mostaghim and Jürgen Teich. Strategies for finding good local guides in multi-objective particle swarm optimization. In *Swarm Intelligence Symposium*, Indianapolis, USA, April 2003. IEEE service center.

[20] Sanaz Mostaghim and Jürgen Teich. Covering pareto-optimal fronts by subswarms in multi-objective particle swarm optimization. In *IEEE Proceedings, World Congress on Computational Intelligence (CEC'04)*, volume 2, pages 1404–1411, Portland, USA, June 2004.

[21] C. R. Mouser and S. A. Dunn. Comparing genetic algorithms and particle swarm optimisation for an inverse problem exercise. In Rob May and A. J. Roberts, editors, *Proc. of 12th Computational Techniques and Applications Conference CTAC-2004*, volume 46, pages C89–C101, March 2005.

[22] Osuna, Freund R., and Girosi F. *Support vector machines: Training and applications*. 1997.

[23] Elpiniki Papageorgiou, Konstantinos Parsopoulos, Chrysostomos Stylios, Petros Groumpos, and Michael Vrahatis. Fuzzy cognitive maps learning using particle swarm optimization. *Journal of Intelligent Information Systems*, 25(1):95–121, July 2005.

[24] A.P. Paquet, U.; Engelbrecht. Training support vector machines with particle swarms. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 2, pages 1593 – 1598, 2003.

[25] Vilfredo Pareto. *Cours d'Economie Politique*. 1897.

[26] K. E. Parsopoulos and M. N. Vrahatis. Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, 1(2):235–306, June 2002.

[27] Margarita Reyes-Sierra and Carlos A. Coello Coello. Improving pso-based multi-objective optimization using crowding, mutation and epsilon-dominance. In *Evolutionary Multi-Criterion Optimization. Third International Conference*, volume 3410 of *Lecture Notes in Computer Science*, pages 505–519. Springer, 2005.

[28] Margarita Reyes-Sierra and Carlos A. Coello Coello. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International Journal of Computational Intelligence Research (IJCIR)*, 2:287–308, 2006.

[29] Mara Margarita Reyes-Sierra. *Use of Coevolution and Fitness Inheritance for Multi-Objective Particle Swarm Optimization*. PhD thesis, CenterofResearchandAdvancedStudies of the NationalPolytechnicInstitute, MexicoCity,Mexico, August 25th 2006.

[30] J. R. Schott. Fault tolerant design using single and multi-criteria genetic algorithms. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1995.

[31] Matthew Settles, Brandon Rodebaugh, and Terence Soule. Comparison of genetic algorithm and particle swarm optimizer when evolving a recurrent neural network. In Springer Berlin / Heidelberg, editor, *Genetic and Evolutionary Computation GECCO 2003*, volume 2723/2003 of *Lecture Notes in Computer Science*, pages 148–149, 2003.

[32] Qi Shen, Wei-Min Shi, Wei Kong, and Bao-Xian Ye. A combination of modified particle swarm optimization algorithm and support vector machine for gene selection and tumor classification. *Talanta*, In Press, Corrected Proof, 2006.

[33] M. Tanaka, H. Watanabe, Y. Furukawa, and T. Tanino. GA-based decision support system for multicriteria optimization. In *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century (Cat. No. 95CH3576-7)*, volume 2, pages 1556–61, New York, NY, USA, 1995. IEEE.

[34] Joachims Thorsten. A support vector method for multivariate performance measures. In *Conference on Machine Learning (ICML)*, 2005.

[35] F. van den Bergh and A. P. Engelbrecht. A study of particle swarm optimization particle trajectories. *Information Sciences*, 176(8):937–971, April 2006.

[36] Hong Zhang, C. M. Tam, and Heng Li. Multimode project scheduling based on particle swarm optimization. *Computer Aided Civil and Infrastructure Engineering*, 21(2):93–103, February 2006.

[37] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

Author Biographies

Gérard DUPONT was born in 1982 in Poitiers, France. He received two M.S. degrees in computer engineering and computer science at Rouen University in 2006. Since then, he began a Ph.D. degree in computer science at EADS-DS in Val de Reuil (France) and with the LITIS Laboratory of computer science in Rouen University on implicit feedback learning for semantic information retrieval. His research interests include evolutionary multi objective optimization, swarm intelligence, learning algorithm, information retrieval and semantic.

Sébastien ADAM was born in 1975 in Dieppe, France. He received a PhD in graphical document analysis from the University of Rouen in 2001. This PhD has been led for France Telecom, the historical French telecommunication operator and tackles the problem of multi-oriented and multi-scaled pattern recognition. Then he joined the LITIS labs in Rouen, France. His domains of interest are at the merging of document analysis and multi-objective optimization.

Yves LECOURTIER was born in Marseilles in 1950. After a thesis in signal processing in 1978, and a second thesis in physics (Automatic Control) in 1985 from the University of Paris-Sud, Orsay, France, he joined the University of Rouen as a Professor in 1987. His research domain is in pattern recognition and optimisation, especially for document analysis and text recognition. Pr. Lecourtier is a member of AFRIF, ASTI, IAPR. From 1994 to 2000, he was the chairman of the GRCE, a french society which gather most of the french researchers working in document analysis and text recognition fields.

Bruno GRILHERES joined EADS Information Processing Competence Center in 2002. He has been working on E-democracy and Text Mining. He led the technical architecture activity on IST CyberVote (IST Prize 2006) and Trade Chamber Elections. He has acted as information technology consultant for EADS Defense and Security Global Security and Mission Systems, Airbus. He is currently completing a PhD (to be presented in 2007) on statistical learning for information extraction.