Multi-Objective Bi-Level Optimization for Parameter Adjustment in Machine Learning

Nuno Filipe Cortes Fernandes
nuno.c.fernandes@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

July 2021

Abstract

In Machine Learning (ML) problems, classical approaches such as grid search are not viable methods for the computation of hyperparameters in higher dimension problems due to combinatorial explosion. The hyperparameter adjustment can be formulated as a Bilevel Optimization Problem (BP). Furthermore, some problems might require multiple objectives to be optimized. This work tests the proof of concept of a Multi-Objective Bi-Level Optimization Problem (MOBP) algorithm, in particular evolutionary-based algorithms, to solve multi-objective Support Vector Machine (SVM) problems with an automatic selection of hyperparameters. The selected algorithm is the Hybrid Bi-Level Evolutionary Multi-Objective Optimization (H-BLEMO) and, in total, six formulations based on soft margin and total margin formulations were tested. The formulations with the best results had similar results to the traditional dual formulation SVM. The formulations with the objective based on the total margin formulation were found preferable since they achieved better performance in all datasets. However, the classification type problems were found to impact the observations and conclusions of the upper-level objective space of the MOBP. In conclusion, the concept was found to be a reliable alternative and a good competitor to the classical SVM algorithms.

Keywords: Hyperparameter Optimization; Multi-Objective Bi-Level Optimization; H-BLEMO; Multi-Objective Support Vector Machine.

1. Introduction

ML is the development of algorithms and techniques that create a model to predict information and making decisions. The learning is made by providing data and solving an optimization problem by finding the set of optimal parameters that minimize a predefined expected loss function [Claesen and Moor, 2015]. The construction of a model by the algorithm requires a selection of hyperparameters. These variables control the characteristics of the algorithm in training the model and have a significant influence on its performance.

Since it first appeared, several approaches have been developed to solve this optimization problem. The so-called classical approach consists of an exhaustive search or brute force strategies such as Cross-Validation (CV) strategy by employing grid search procedure. It suffers, however, of several adversities, the main one being the fact that the combinatorial nature of this strategy leads to a combinatorial explosion as the dimension (number of features) of the problem increases.

A recent alternative to the classical approach was proposed in the article by [Bennett et al., 2006]. The CV Hyperparameter Optimization (HO) problem was defined as a BP. The problem has two distinct levels, the outside one called upper-level or leader, and the other called lower-level or follower. The solution to the lower-level corresponds to the constraint functions of the leader. In HO, the lower-level corresponds to the optimization problem of the training stage and the upper-level to the optimization problem of the validation stage. Since, as mention above, the hyperparameters are chosen before training, they are upper-level variables. As for the model parameters, they are lower-level variables. The corresponding solution to the lower-level problem is the optimal model parameter set of training.

Although each level is composed of single objective function stage, the BP can be extended to include several objective functions in both or simple one level. This new formulation is referred to as MOBP. Multiple objectives are often considered and grouped together into the same optimization function. However, the inexistence of conflicts between two or more objects cannot be guaranteed. Using multi-objective bi-level Evolutionary Algorithms

based meta-heuristics and the selection of multiple objective functions in each layer is the motivation for this thesis. The main contribution of this thesis is the proof of concept of MOBP in the adjustment of hyperparameters and parameters, and the effect of SVM formulations with different objectives.

2. Revision of Literature

2.1. SVM

SVM is prediction model developed in the 1990s by [Cortes and Vapnik, 1995] for pattern recognition. Used in binary classification, it employs the determination of the optimal hyperplane that separates the two classes. In many real-world problems, the data is rarely perfectly separable and usually contains noise. To relax the original strict first SVM formulation it is allow slightly misclassified data points through the usage of a positive slack ξ. This variable measures the error of misclassified points and is defined as the distance points to their respective class hyperplane. The formulation is known as soft margin SVM and is given by

$$\begin{aligned} \min_{w,w_0} \quad & \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i(w^T x_i + w_0) \geqq 1 - \xi_i, \\ & \xi_i \geqq 0, i = 1, \ldots, l. \end{aligned} \tag{1}$$

where the parameter C is a non-negative regularisation variable that controls the importance of misclassified points. In other words, the significance given to the optimization of the margin decreases for higher values leading to a smaller margin.

The final result requires at least a pair of support vectors to define the hyperplane, one for each class, and only these points from all data are necessary to store. The optimization should be aiming to keep good performance and simultaneously contain a small set of support vectors. One disadvantage of a model with a large number of support vectors is the possibility of over-fitting. On this account, a validation stage in conjunction with the cross-validation technique is required after the training stage.

The formulation 1 is called the primal SVM formulation and is rarely used for solving the problem. A transformed formulation is instead used named the dual formulation, and it is particularly beneficial in nonlinear datasets and with kernel transformation. Since computing the mapping of the transformation in the primal formulation can be computationally expensive, using the kernel function $K(x, x) = \langle \Phi(x), \Phi(x') \rangle$ where Φ: $\chi \to \mathcal{F}$ and $\mathcal{F}$ is a Hilbert Space, the computation is reduced to dot-product between points of transformed dataset in a total of N by N evaluations. Also, the fact that only the support vectors have α non-zero values facilitates the optimization and reduces the complexity.

The soft margin formulation had only in mind the wrongly classified data points. The distances of correctly classified data can also be taken into consideration. The idea was proposed in [Min Yoon et al., 2003] where the opposite concept of the slack variable, called surplus variable, ξ^+ or η, was introduced in the problem 1 as a maximization objective. This extension is referred to as total margin SVM and is express in 2, where two hyper-parameter were introduced to control the trade-off of the slack vector and the surplus vector with respect to the margin minimization. The variable C_1 is selected to be higher than C_2, to ensure at that at least one ξ_i and η_i are zero.

$$\begin{aligned} \min_{w,w_0} \quad & \frac{1}{2}\|w\|_2^2 + C_1\sum_{i=1}^{l}\xi_i - C_2\sum_{i=1}^{l}\eta_i \\ \text{subject to} \quad & y_i(w^T \Phi(x_i) + w_0) \geqq 1 - \xi_i + \eta_i, \\ & \xi_i \geqq 0, \eta_i \geqq 0, i = 1, \ldots, l, \end{aligned} \tag{2}$$

2.2. MOP and MOBP

Considering the objective function $F : \mathbb{R}^n \to \mathbb{R}$, and the constraints $G_k : \mathbb{R}^n \to \mathbb{R}$, k = 1,. . . , K and H_p : $\mathbb{R}^n \to \mathbb{R}$, p = 1,. . . , P, the MOP is given by

$$\begin{aligned} \min_{x \in X} \quad & F(x) = (F_1(x), \ldots, F_t(x)) \\ \text{subject to} \quad & G_k(x) \leq 0, k = 1, \ldots, K \\ & H_p(x) = 0, p = 1, \ldots, P. \end{aligned} \tag{3}$$

Contrary to single-objective optimization, with more the two objective, final solution is a frontier in the objective space. Two concepts are necessary for defining it.

Definition 2.1 (Dominance)**.** Given two vector x, y $\in \mathbb{R}^k$, $x \leq y$ if $x_i \leq y_i$ for i = 1, . . . ,k, and that $x \prec y$ (x *dominates* y) if f $x \leq y$ and $x \neq y$.

Definition 2.2 (Non-dominated)**.** A variable vector $x \in \mathbb{X}$ is non-dominated with respect to $\mathbb{X}$ if there does no exist $x' \in \mathbb{X}$ such that $f(x') \prec f(x)$.

A point is then considered best or non-dominated if is best in one and not worst in all the other objectives. In the decision variable space, the vector containing non-dominated points is called efficient solution or Pareto optimal solution. As for the objective space, the vector for the same points is referred to as Pareto Front (PF).

The BP is a mathematical program composed of two levels of optimization. The upper-level is the main optimization problem, and the lower-level is the secondary optimization problem which is nested in the first one. The levels are characterized by their one objective function, constraints, and the class of decision vector variables. While the lower-level is optimized with respect to the lower-level decision vector, the upper-level decision vector act as a parameter. This implies a constraining nature of the lower-level concerning the upper-level. For the upper level objective function $F : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^p$ and the lower level objective function $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^q$, the MOBP is defined by

$$\begin{aligned} \underset{x_u \in X_U, x_l \in X_L}{\text{“min”}} \quad & F(x_u, x_l) = (F_1(x_u, x_l), \ldots, F_p(x_u, x_l)) \\ \text{subject to} \quad & x_l \in \underset{x_l \in X_L}{\text{argmin}} \{ f(x_u, x_l) = (f_1(x_u, x_l), \ldots \\ & \ldots, f_q(x_u, x_l)) : g_j(x_u, x_l) \leq 0, j = 1, \ldots, J \} \\ & G_k(x_u, x_l) \leq 0, k = 1, \ldots, K \end{aligned} \tag{4}$$

where $G_k : X_U \times X_L \rightarrow \mathbb{R}$, k = 1,. . . , K and $g_j : X_U \times X_L \rightarrow \mathbb{R}$ represent the upper level constraints and the lower level constraints, respectively. Both constraints can also have equality constraints.

Pioneer by the author of VEGA algorithm [Schaffer, 1985] in the 1980s, the application of Evolutionary Multi-Objective Optimization (EMO) has been widely used and improved since then. Contrary to classical techniques that require several separate runs to compute the PF, as stated in [Coello, 1999], the EMO are ideal for MOPs since a set of possible PF solutions in parallel is computed in a single run as well as being less susceptible to shape and continuity.

Over the years, several techniques and approaches were invented, and, by far, the most popular are the PF based approaches. One of the most used, tested, and widely established EMO is the improved version of the algorithm by the same authors in [Srinivas and Deb, 1994] and is referred to as NSGAII [Deb et al., 2002]. The main improvement is the introduction of elitism in the algorithm. With this concept, the previous parent population members can be contained in the child population allowing the prevention of loss of good solutions and helping an overall better convergence [Zitzler et al., 2000].

The main operators of changing or diversify the population are the crossover, and mutation with a previous selection. The selection operator is the method where members of a population are selected by ranking the population with a fitness value. The principal fitness measure is the Non-Dominated Rank (ND) based on 2.2. Another important aspect of solutions in the PF is the requirement of diversity to ensure the complete representation of the PF. For this reason, and to help differentiate solutions with equal ND, the Crowding Distance (CD) is used. Measuring the density outside of the point by computing the cuboid form with the nearest neighbours as vertices when two solutions have the same ND, a solution with a bigger cuboid or less crowded region is preferable. This operator is called tournament selection.

The crossover or recombination is, as the name suggests, a reconfiguration of the parents' solutions to obtain new child solutions similar to the process of chromosomal recombination in biology.

After the recombination, the mutation operator helps in diversifying the child solutions and preventing local minima by slightly changing the solutions. The NSGAII uses four parameters to control the progression of the operators: crossover probability, index for SBX operator, mutation probability, and the index of polynomial mutation and are crucial to the performance of the algorithm.

The overall procedure of NSGAII is shortly described below. Using a parent population of members P_t of size N, the NSGAII for each generation creates another population Q_t called child population with the above operators. When the total number of new creation is equal to the parent population, the two equal size populations are combined in a new population R_t. This population is used to create the new parent population by removing half the members. Ranking and sorting R_t with ND and in turn CD, the worst solutions of size N are rejected. The previous steps are repeated until the predefined maximum generation is achieved.

It is important to mention the above algorithms since most MOBP algorithm used in each level of a MOP algorithm. One of the original authors of NSGAII also developed an approach for MOBP called Bi-Level Evolutionary Multi-Objective Optimization algorithm [Deb and Sinha, 2009]. Although the described procedure uses the previous EMO to solve both levels of optimization, as indicated by the authors, any other developed algorithm can be used. This algorithm was later acknowledged to contain several drawbacks leading therefore to a new extended version named H-BLEMO algorithm [Deb and Sinha, 2010].

2.3. Hyperparameter Optimization

Traditionally, hyperparameters in ML are determined by a series of trial and choosing, in the end, the set of values that achieved the best performance. This is done by an exhaustive n-dimension grid search. Typically, a CV technique is combined with the grid to improve validation performance. Also called brute force, the approach has the downside of a combinatorial explosion causing it to be unreliable in problems of dimension higher than two [Bergstra and Bengio, 2012] which can reach up to hundreds [Bergstra et al., 2013].

To remove the problems of using the CV technique for a higher number of hyperparameters, [Bennett et al., 2006] proposed a new program of bilevel CV and tested on support vector regression model. In this way, for each fold of the CV an automatic selector was included with the training of the model with the respective set of hyperparameters in the lower-level and the validation in upper-level.

3. Methodology

As mentioned in Section 2, the H-BLEMO algorithm was the result of several improvements to the BLEMO algorithm mostly involving unnecessary computation for already found good solutions.

The main structure of the algorithm is composed of n_s subpopulations. Each one shares the same upper-level variable vector, and, in total, the subpopulations have N_u members. For each subpopulation, a lower-level NSGAII is performed followed by a local search optimization. In every generation, the archive is updated after every lower-level optimization.

At the beginning of a generation, every member in the population P_t of size N_u and archive have computed the corresponding values of ND and CD for both levels. Step 1 deals with creating a new upper-level vector and respective lower-level vectors for the current generation. For a single upper-level vector creation, a binary tournament selection is applied to population P_t and archive. Of the four outcome parents, two are selected stochastically are recombined using the SBX operator. Finally, one is mutated with the polynomial operator. This final vector is the new child upper-level vector. The aforementioned vector is then used to create N_l child solutions, number which is based on its location in the current archive members' space.

After all child solutions are created for the upper-level vector, step 2 performs the NSGAII to the lower-level. The algorithm differs solely from the original on the selection. Taking advantage of previous found archive solutions, if the subpopulation is present in the archive, only these are used in the binary tournament selection. Otherwise, the normal process is used. At the end of lower-level optimization, the solutions are sorted and ranked by ND and CD.

Step 3 involves the new optimization addition of the Local Search operator to achieve the locally PF. Since the operator can be expensive, as later verified to represent 50% of all computation effort, the operator was only applied to solutions that follow certain properties to exclude inadequate solutions. The operator is defined by the optimization of achievement scalarizing function problem [Wierzbicki, 1980].

Step 4 is for updating the archive after the Local Search. For only the deemed optimal solutions, these are compared with all archive members. If the solutions are non-dominated, these enter the archive, and the dominated members are excluded. In case of exceeding the maximum size of the archive, until the size is reached, the members are removed according to CD.

In Step 5 the creation of all new solutions finalizes, meaning the above steps are repeated until the population of new solutions has the exact size of the parent population P_t. What follows is the combination of both populations after a ranking by ND and CD for future selection of N_u members.

This selection of half of the combined population is step 6 of the algorithm. The members first considered are those that have upper-level ND equal to 1, and then lower-level ND equal to 1 in order of reducing by lower-level CD. If the entire lower-level subpopulation is already present in the side population and the future solutions are from the same subpopulations and have both ND equal to 1, no further copy to the side population is done. The process is repeated for all upper-level ND equals 1 and future values until members reach N_u size.

In the last step, for each subpopulation in the side population not created on the above steps, a lower-level NSGAII is utilized for helping the individual approximation of PF. The termination criteria metric is computed, and if the value reaches lower than the threshold on generations multiple of τ, the algorithm comes to an end. Otherwise, the steps above are repeated.

4. Proposed SVM Problems for MOBP

4.1. SVM Formulations

The two objectives in the selected primal formulations are the minimization of slack variable and maximization of surplus variable, and in total, six different formulations were created. The constraints remain the same as in the original formulation and for that reason are not shown below. The first two formulas were based on the formulation 1 and were defined as

- Formulation 1:

$$\min_{w,w_0} \left\{ F_1 = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{l}\xi_i \right. \tag{5}$$

- Formulation 2:

$$\min_{w,w_0} \begin{cases} F_1 = \|w\|_2^2 \\ F_2 = \sum_{i=1}^{l}\xi_i \end{cases} \tag{6}$$

while the remaining were originated from the formulation 2 and were defined as

- Formulation 3:

$$\min_{w,w_0} \left\{ F_1 = \frac{1}{2}\|w\|_2^2 + C_1\sum_{i=1}^{l}\xi_i - C_2\sum_{i=1}^{l}\eta_i \right. \tag{7}$$

- Formulation 4:

$$\min_{w,w_0} \begin{cases} F_1 = \|w\|_2^2 \\ F_2 = C_1 \sum_{i=1}^{l} \xi_i - C_2 \sum_{i=1}^{l} \eta_i \end{cases} \quad (8)$$

- Formulation 5:

$$\min_{w,w_0} \begin{cases} F_1 = \|w\|_2^2 \\ F_2 = \sum_{i=1}^{l} \xi_i \\ F_3 = -\sum_{i=1}^{l} \eta_i \end{cases} \quad (9)$$

- Formulation 6:

$$\min_{w,w_0} \begin{cases} F_1 = \frac{1}{2}\|w\|_2^2 - C \sum_{i=1}^{l} \eta_i \\ F_2 = \sum_{i=1}^{l} \xi_i \end{cases} \quad (10)$$

4.2. Objectives for SVM Evaluation

For evaluating the lower-level solutions in the training and validation stage, two types of objectives were selected. The first corresponds to the error of a hyperplane of the dataset and the other to the classification task.

The Hinge loss is the most commonly loss function use in training SVM. It's defined for a particular point as

$$L_{Hinge}(y, w^T x + w_0) = max\{0, y(w^T x + w_0)\}, \quad (11)$$

where y is the output ± 1. This is an equivalent definition to the slack variable.

For the classification of classes, the metric used is a specific case of the F-score. To transform this metric into a minimization problem the same approach in [Musicant et al., 2003]. Using the minimization approach of F1-score in [Musicant et al., 2003], the new F1-score metric is given by

$$\text{F1-score} = \frac{1}{1 + \frac{1-C}{2z}}, \quad (12)$$

with C representing the global performance using the accuracy and z the ratio of true positive classifications. The maximization of the F1-score is achieved by the minimization of fraction in the denominator, assuming z $\neq$ 0, and that results in the following minimization problem

$$(1 - C) - 2z. \quad (13)$$

Since the metric takes only into account only the true positive cases, a similar approach to 13 can be formulated called negative F1-score with a new z variable, z_{neg}, representing the ratio of true negative classifications. In conclusion, in both levels and in all formulations, the hinge loss, the F1-score, and the negative F1-score were employed.

5. Results & Discussion

5.1. Pre-Testing

The H-BLEMO constructed to this work is based on the Matlab tool Evolutionary multi-objective optimization platform PlatEMO [Tian et al., 2017] made available by BIMK Group, specifically the multi-objective algorithm NSGAII. As for the termination criteria, the hypervolume indicator algorithm used was proposed by [Fonseca et al., 2006]. For both levels, the standard parameters of NSGAII (crossover probability of 0.9, index for SBX operator of 15, mutation probability of 0.1, index of polynomial mutation of 20) were selected. The number of population members was defined by 20 times the total number of variables in the problem following the indication of the authors. This number achieves best performance with smallest number of function evaluations. A constrained non-linear multivariable function from Matlab library was utilized for the Local Search quadratic optimization.

For the empirical analysis of SVM formulations selected, several datasets were retrieved from the UCI Machine Learning Repository [Dua and Graff, 2017]: Iris flower dataset or Fisher's Iris dataset(*Iris Setosa* and *Iris Versicolour* and *Iris Versicolour* and *Iris Virginia*); Haberman's Survival; and Wisconsin Breast Cancer Database (January 8, 1991) [Bennett and Mangasarian, 1992]. Three self-made dataset were also created: linearly separable, non-linearly separable and non-linearly separable with noise (Noisy) datasets. For the generalization of the classification task, the CV technique was implemented in the algorithm. For every generation, the dataset is randomized and 70% used in the lower-level or training stage. The remainder is applied to validate the training result.

5.2. Behavior of Classical Multi-Objective Bi-Level

Before testing ML problems in the H-BLEMO, the algorithm was simulated in classical MOBPs, the two first test problems in [Deb and Sinha, 2010] called TP1 and TP2, to evaluate the performance of the adapted algorithm.

The results of the constructed algorithm approximately achieves the true PF but the solutions are considerably fragmented and incomplete. This indicates a lack of performance since several variable combinations are not present in the final results. The test required a modification on the mutation probability parameter of the algorithm. The substitution was intended to aim for more diverse results because of the poor results. However, the parameters used in the Sections below are the standard values. It is important to state that the H-BLEMO was constructed by the interpretation of the step procedure and the utilization of different algorithms from the original.

5.3. Upper-Level Objective Space

In the upper-level objective space of classical problems, the PF represents a direct outcome of the variation of the upper-level variables in the optimization while in classifications problems two steps exist: the computation of hyperplane, and classification task evaluated with a metric function. This means that in MOBP the hyperplane is defined by the variables and the objectives function are defined not by the variables but by the hyperplane. This difference changes the dynamic of the algorithm, for example, in the termination criteria, and the evaluation of the performance of the final results.

First, the values of objective functions evaluating the classification belong to a set of finite numbers and are dependent on the number of points in the dataset and the ratio of positive cases of the F1-score metrics. The consequence of this is a non-smooth and depleted PF. However, these vacant spaces in the PF are not a sign of the ineffectiveness of the algorithm but simply an intrinsic characteristic of the classification problem. Second, the points in the objective space can have the same values even though they represent different hyperplanes, but the opposite can also occur. These two cases take place due to not just the minimization of both the F1-scores but also the addition of the CV technique.

The PF can also cease to be a frontier. Using the hinge function and only the true F1 or the negative F1 instead of both, the PF becomes a single point. This occurs for two reasons: the hinge function being lower bound by 0, and the ND. Given the ND, when one point with hinge equal to 0 and an arbitrary F1 score value, the archive will only accept points with smaller F1 while removing the remaining.

The objective space becomes unusable when concluding or comparing the performance of the algorithm and the different formulations. However, this does not imply an useless utility to the validation and training stage of the hyperplanes.

5.4. Termination Criteria

The effect in upper-level objective space is especially concerning because of its dependency on the termination of levels of the algorithm. A comparison of the algorithm with the criteria and without was done to evaluate its impact. In replacement, maximum number of upper-level generations is defined.

For the linearly separable dataset, the differences are very slim, chiefly due to being an easily separable dataset. Nonetheless, some hyperplanes were found to not perfectly separate the data. In the non-linearly separable dataset, some of the tests with the termination criteria accomplish results similar to previous test, but not using it seems favourable to better performance. Its impact is even more expressive in more complex datasets such as the Noisy dataset. The reason for these discrepancies in the results lies in the fact that the termination criteria depends on the maximum and minimum values of the HV. In most runs of the algorithm, the upper-level PF did not change in ten consecutive generations, making both values equal and criteria zero, immediately stopping the algorithm.

Despite the identical utilization in the lower-level problem, the lower-level objective space is less prone to consecutively remaining the same due to the presence of a higher number of objective functions and in particular non-classification objectives such as, for example, the norm and the sum of the distances to hyperplane of misclassified points.

The termination criteria used was then removed for not being suitable for these problems and was substituted with a maximum number of generations.

5.5. Local Search

The Local Search optimization is the attempt to guarantee the lower-level solutions to be locally PF. ML problems change the relationship between variables, the objective function and objective space, and the problem itself. A study of the effect of the Local Search optimization was carried out.

The first aspect observed is how the solutions are more sparse and, seemingly, more diverse in terms of the number of different hyperplanes. Several hyperplanes of in the archive of linearly separable dataset do not correctly divide space. In this case, archive contains only perfectly classification solutions, but not good solutions were highlighted after measuring the accuracy in the test stage. In terms of the overall performance of the H-BLEMO, this non-perfect solutions can detriment the creation step of new child solutions due to the dependency on the archive.

In the non-linearly separable dataset with a more restricted space for the best separation, the algorithm, in the long run, tends to have all solutions in the archive alike or coinciding with each other. The utilization of the Local Search did not aid the results.

The removal of the this step does not negatively affect the H-BLEMO. On the contrary, it improved or corrected the performance of solutions. Another advantage of this disposal is the significant reduction of the computation time. The reason for these problems has two possible origins: the algorithm constructed has worse performance when compared to the original one, and/or the choice of the Local Search algorithm and respective parameters. This operator was removed from the H-BLEMO.

5.6. Classification Results

A testing stage was created to evaluate the performance by using 20% of the original dataset, computed by the accuracy. At the end of each run, the algorithm contains multiple solutions. The accuracy percentages represent the average of 5 runs of the average of all solution in a single run. In Gaussian tests, the comparison done between hyperplanes and also with different σ values.

The algorithm for the *Iris Setosa* and *Iris Versicolour* dataset, independently of the formulation, has a perfect separation rate. This result is expected, since the dataset has ample space between the two classes.

In table 1, there is no perfect flat hyperplane that separates the space. However, a slight improvement can be observed along with the different formulations in the original dataset. For this dataset, the slack objective improves the optimization when separated from the norm objective. Although in the formulation 7 (3) that is not verified, the presence of the surplus objective is what improves the performance in 1.5 %, or 2.56 % in 200 gen test, when compared with formulation 5 (1). The introduction of the surplus objective likewise helps the performance, notably, of formulation 8 (4), for which the best result is achieved of roughly 95%. As for the Gaussian transformation, the values are seemingly beneficial in formulation 5 (1) and 7 (3) but have a high enough value of standard deviation to indicate overfitting. However, both formulations 8 (4) and 10 (6) attain similar and slightly better results than without transformation, respectively.

In table 2, the effects of the slack and surplus objectives are similar to the *Iris Versicolour* and *Iris Virginia* dataset, albeit the best performances in the original dataset are the formulations 7 (3) and 9 (5), around 93.50% and 93.25%, respectively. With the Gaussian transformation, the improvement is barely significant. In the case of the formulation 8 (4), the best mean value is achieved in the 100 generation test whereas, in the 200 generation test, the performance is worse than formulation 7 (3), 8 (4) and 10 (6) meaning this formulation is not as good as the formulation 7 (3) with no transformation.

The Haberman's Survival dataset was by far the dataset with worst performance as well as the most standard deviation. The reason for this disparity is the extreme complexity of the dataset. Despite this, the some advantages of the formations are visible. Again the separation of slack objective and the introduction of the slack objective increased the overall performance. Especially, formulation 8 (4) in the original transformation achieves around 10% and 5% better results than formulation 5 (1) and 6 (2), respectively. The best results are the 72% of the formulation indicated above and 73% of formulation 10 (6). In the Gaussian transformation tests, the performance is similar or worse than that of the original transformation. Comparing the tests of different generations, no advantage is found in using more than 100 generations, since no considerable reduction in the standard deviation is attained. Again, this is a consequence of the complexity of the dataset.

In table 4, the results for the most challenging dataset in terms of the total number of features is presented. The dataset is the most notable for the positive influence of the introduction of the slack objective, achieving around 15% better accuracy and significantly better precision between formulation 5 (1) and 7 (3). The better results are attained when the two variable objectives are solely in the same function, in particular the best in formulation 8 (4) reached approximately 97%. With the Gaussian transformation, a drastic improvement is reached in formulation 5 (1) and 6 (2) though insufficient to surpass the remaining. Also, there is no distinction in the performance when the surplus objective is introduced, yet they still achieved similar results to the best formulation in the original dataset.

Comparing the overall datasets, the test of the *Iris Setosa* and *Iris Versicolour* had similar computational times when compared with the Wisconsin Breast Cancer dataset. Since the former is not as complex as the latter, the explanation for this is in the total number of generations. For 100, the value is simply excessive for the optimization reaching a point of repetitive comparisons of newfound solutions and the solution in the archive. The total number of possible archive members can be reduced since there is no necessity for a complete filled PF. However, the value can maybe be chosen according to the dataset. The values used here were the same for all test simply because no prior sensibility was known.

Comparing the results of table 5 and the best results of previous tests, the H-BLEMO achieved similar results except for the most complex dataset, the Haberman's Survival. This proves how versatile MOBP algorithms can be in the optimization of different types of problems, as well being a competitor with traditional SVM algorithms allowing a shift in focus from also the selection of the best set

Table 1: Accuracy (%) and standard deviation of runs with different generations for each formulation for the *Iris Versicolour* and *Iris Virginia* dataset.

	Generations							
	100				200			
Transformation	**Original**		**Gaussian**		**Original**		**Gaussian**	
Formulation	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
1	91.67	6.1301	94.99	5.4773	91.43	5.0766	91.43	5.6057
2	92.00	4.0000	92.93	2.5386	92.00	2.4495	90.58	7.1666
3	93.16	3.8931	96.82	6.0383	93.99	3.6782	95.81	3.5094
4	94.93	1.5142	93.89	1.4285	95.44	4.1858	95.48	2.7633
5	94.00	4.8990	92.19	3.1306	95.02	3.1625	93.20	4.0513
6	94.01	3.4765	96.64	1.8939	94.68	3.0684	94.67	2.9887

Table 2: Accuracy (%) and standard deviation of runs with different generations for each formulation for the Noisy dataset.

	Generations							
	100				200			
Transformation	**Original**		**Gaussian**		**Original**		**Gaussian**	
Formulation	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
1	87.79	3.3481	86.84	5.9969	85.22	2.1326	88.17	6.1430
2	88.22	4.3825	89.91	4.5375	85.35	4.9404	89.35	3.7455
3	92.47	2.7534	93.02	2.7588	94.57	1.4614	92.63	3.6807
4	92.40	2.7183	95.13	2.3548	92.76	2.7014	92.36	0.9536
5	93.00	1.6956	92.24	2.9985	93.50	2.8940	92.88	3.7838
6	93.31	2.3762	89.63	3.0316	92.50	1.4309	92.80	3.2020

of the hyperparameters to just the choice of feature transformation, the type of hyperplane and different formulations and objectives.

6. Conclusions

Traditional SVM algorithms, despite efficient for simple or small dataset, for large datasets and the presence of high number of hyperparameters, suffer from combinatorial explosion, making them unusable. A new alternative consisted in transforming the problem into a mathematical problem of two levels, where one is the minimization problem of the training and evaluation and the other the minimization of the validation.

For this the MOBP algorithms based on EA and their concepts were used. In particular, the algorithm used is called H-BLEMO. Since the algorithms allow optimization of multiple objectives at the same time in each level, several formulations of the SVM problem were created, focusing on two objectives: the slack objective and the surplus objective to evaluate the existence of possible conflicts between objectives and the advantages of separating these objectives.

The several tests indicated that ML problems change how the MOBP is optimized and the respective conclusions of the final solution. There is no direct connection with the solution of the hyperplanes and of the PF. For this reason, the PF becomes a just a tool to achieve the best results and not to evaluate the final archive solutions and thus a different termination criteria is required.The overall tests indicate the introduction of the surplus objective in the formulation is preferable since the models achieved better results with it than without. The best formulation with this objective vary with the dataset. However, the results were slightly worse when objective was by itself. As for the soft margin based formulation, the results were similar with the best results also varying with the dataset.

In conclusion, the utilization of MOBP are a reliable alternative and a good competitor to the classical SVM algorithms, since it allow an automatize selection of hyperparameters and the testing for advantages and disadvantages of the relation between objectives and their separation.

7. Further Work

This work proved the potentiality of the concept. However, other studies are required for understanding the real impact of classification problems in MOBP and for fine-tuning.

First, although the performances were still good while not removing it, an analysis of the impact of

Table 3: Accuracy (%) and standard deviation of runs with different generations for each formulation of Haberman's Survival dataset.

	Generations							
	100				200			
Transformation	**Original**		**Gaussian**		**Original**		**Gaussian**	
Formulation	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
1	63.37	3.6062	69.51	6.6618	65.37	8.2017	63.15	5.9226
2	67.59	8.3339	62.26	4.7574	65.90	6.8304	62.19	5.6920
3	69.70	5.4575	67.37	4.2309	69.18	1.7316	71.81	6.4129
4	72.10	5.5328	69.79	5.6901	72.89	4.2815	69.93	3.2024
5	68.83	8.1508	71.55	1.9678	71.17	7.4315	71.59	0.7218
6	73.41	3.4715	72.00	4.1078	71.82	4.2295	72.83	5.1015

Table 4: Accuracy (%) and standard deviation of runs with different generations for each formulation for the Wisconsin Breast Cancer dataset.

	Generations							
	100				200			
Transformation	**Original**		**Gaussian**		**Original**		**Gaussian**	
Formulation	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
1	76.46	9.3958	91.66	2.0571	78.02	4.6195	92.31	1.5933
2	67.44	5.5326	86.61	6.7933	72.26	6.0014	90.57	5.6901
3	91.96	0.8946	96.58	1.4128	93.30	1.8187	95.12	0.1920
4	97.03	0.5305	96.51	1.5377	95.72	1.5826	96.53	1.5827
5	92.71	3.3031	95.80	1.3211	95.34	1.0050	94.95	1.4038
6	95.49	1.5539	96.16	1.6965	95.87	1.5312	96.62	1.3957

Table 5: Accuracy (%) and standard deviation of test with soft margin SVM in dual problem formulation for each dataset.

	Transformation			
	Original		**Gaussian**	
Dataset	**Mean**	**SD**	**Mean**	**SD**
Setosa & *Versicolour*	100	0	100	0
Noisy	93.34	2.0263	92.00	2.0484
Versicolour & *Virginia*	95.33	3.3993	96.40	3.5901
Haberman	73.07	3.9875	76.46	3.8640
Wisconsin Breast Cancer	96.93	0.9491	97.25	0.8552

the termination criteria in the lower-level optimization should be made and the reason for the negative effect of the Local Search optimization.

Second, due to the changes in the objective space, an evaluation should be effectuated to the ranking of solutions to verify if the CD criteria are relevant and necessary since there is no utility for scarcity in the PF.

Finally, more test should be done with a higher number of feature (> 10) and testing other formulations and taking into account different hyperparameters for each class in imbalanced datasets. Also, the test can include non-flat hyperplanes for separation and using the algorithm Reference-Point based Many-Objective NSGA-II [Deb and Jain, 2014] created for many-objective problems in the lower-level or both levels.

References

[Bennett et al., 2006] Bennett, K. P., Jing Hu, Xiaoyun Ji, Kunapuli, G., and Jong-Shi Pang (2006). Model selection via bilevel optimization. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 1922–1929.

[Bennett and Mangasarian, 1992] Bennett, K. P. and Mangasarian, O. L. (1992). Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods & Software*, 1(1):23–34.

[Bergstra and Bengio, 2012] Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(null):281–305.

[Bergstra et al., 2013] Bergstra, J., Yamins, D., and Cox, D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *ICML*.

[Claesen and Moor, 2015] Claesen, M. and Moor, B. D. (2015). Hyperparameter search in machine learning. *CoRR*, abs/1502.02127.

[Coello, 1999] Coello, C. A. C. (1999). Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems*, 1:269–308.

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

[Deb and Jain, 2014] Deb, K. and Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601.

[Deb et al., 2002] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

[Deb and Sinha, 2009] Deb, K. and Sinha, A. (2009). Solving bilevel multi-objective optimization problems using evolutionary algorithms. In Ehrgott, M., Fonseca, C. M., Gandibleux, X., Hao, J.-K., and Sevaux, M., editors, *Evolutionary Multi-Criterion Optimization*, pages 110–124, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Deb and Sinha, 2010] Deb, K. and Sinha, A. (2010). An efficient and accurate solution methodology for bilevel multi-objective programming problems using a hybrid evolutionary-local-search algorithm. *Evolutionary Computation*, 18(3):403–449. PMID: 20560758.

[Dua and Graff, 2017] Dua, D. and Graff, C. (2017). UCI machine learning repository.

[Fonseca et al., 2006] Fonseca, C. M., Paquete, L., and Lopez-Ibanez, M. (2006). An improved dimension-sweep algorithm for the hypervolume indicator. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1157–1163.

[Min Yoon et al., 2003] Min Yoon, Yeboon Yun, and Nakayama, H. (2003). A role of total margin in support vector machines. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 3, pages 2049–2053 vol.3.

[Musicant et al., 2003] Musicant, D. R., Kumar, V., and Ozgur, A. (2003). Optimizing f-measure with support vector machines. In *FLAIRS Conference*, pages 356–360.

[Schaffer, 1985] Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, page 93–100, USA. L. Erlbaum Associates Inc.

[Srinivas and Deb, 1994] Srinivas, N. and Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248.

[Tian et al., 2017] Tian, Y., Cheng, R., Zhang, X., and Jin, Y. (2017). PlatEMO: A MATLAB platform for evolutionary multi-objective optimization. *IEEE Computational Intelligence Magazine*, 12(4):73–87.

[Wierzbicki, 1980] Wierzbicki, A. P. (1980). The use of reference objectives in multiobjective optimization. In Fandel, G. and Gal, T., editors, *Multiple Criteria Decision Making Theory and Application*, pages 468–486, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Zitzler et al., 2000] Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195.