Learning Human-Understandable Models for the Health Assessment of Li-Ion Batteries via Multi-Objective Genetic Programming

Yuviny Echevarría[a], Cecilio Blanco[b], Luciano Sánchez[a,*]

[a]*Departamento de Informática, Universidad de Oviedo, Spain*
[b]*Departamento de Electrónica, Universidad de Oviedo, Spain*

Abstract

The health of automotive Li-ion batteries depends on different side reactions on the electrodes that may degrade the cells, thereby reducing their usable capacity and sometimes producing catastrophic failures with serious economic and safety implications. In this paper, a method of detection and prognosis of battery deterioration is proposed in which an intelligent soft sensor is able to synthesize human-understandable health indicators from sequences of voltages, currents and temperatures streamed via on-vehicle sensors. This soft sensor is based on a dynamic model optimizing three different criteria obtained by means of multi-objective grammatical evolution. Different survival selection strategies suitable for this problem are discussed and compared.

Keywords:
Multiobjective Genetic Programming, Grammatical Evolution, Battery Model

1. Introduction

All automotive Li-ion batteries exhibit gradual degradation, given that their usable capacity decreases with time. The state of health of an automotive Li-ion battery at any point in time depends on the history of the different electrochemical reactions that have taken place on its electrodes. Some of these reactions are undesired and cause premature deterioration of the battery, but it is not easy to

[*] Corresponding author

Email addresses: yuviny@gmail.com (Yuviny Echevarría), cecilio@uniovi.es (Cecilio Blanco), luciano@uniovi.es (Luciano Sánchez)

perceive when these harmful changes are taking place, nor can these abnormal degradations be prevented immediately while the vehicle is being used (Weng et al., 2016).

Normally, a cause-agnostic approach to the problem is adopted in which the precise degradations are ignored and only the usable capacity of the battery is measured during routine charges of the vehicle. However, there are also “silent” deteriorations that do not influence the battery’s usable capacity and hence go undetected when using the agnostic approach. For instance, if the impedance of an aged battery is increased, then the maximum power of the vehicle is limited, though its range is not. In other cases, electrodepositions or mechanical fractures can degrade the electrodes and the battery may suddenly die. Apart from economic concerns, abrupt failures have safety implications and must be prevented (Weng et al., 2013; Anseán et al., 2016).

Diagnosing silent deterioration requires measuring different variables such as the capacities of the positive and negative electrodes, or the loss of lithium inventory (Torai et al., 2016). These are destructive measurements, but indirect measurements are also possible, as these values are correlated with the locations of the slope changes in the voltage vs. charge curve (known as Incremental Capacity Analysis (ICA); see (Dubarry et al., 2006)). However, this analysis must be performed off-vehicle under laboratory conditions (Dubarry et al., 2012). Moreover, the accuracy of the vehicle’s sensors is not sufficient to detect many incipient types of degradation. As removing the battery and testing it in the laboratory is not a viable option, soft sensors of its state of health are used instead to synthesize the irretrievable variables from sequences of voltages, currents and temperatures streamed via on-vehicle sensors. Most soft sensors are model based and require that a dynamic model of the battery is learned from data; see (Sánchez et al., 2014) for an empirical study in which the accuracies of different first-principle, semi-physical and intelligent models are compared when applied to this task.

Soft sensors can synthesize the unknown variables, but they do not constitute a diagnostic or prognostic method per se. Sensors that depend on black-box models of the battery are often combined with “virtual laboratory” experiments, in which the off-vehicle laboratory experiments mentioned above are applied to a computer simulation of the battery model. Alternatively, soft sensors based on grey-box or first-principle models are able to produce estimations of the variables of interest without additional procedures. Human-understandable models are sought because virtual labs are both computationally expensive and potentially inaccurate: well-fitted models may perform poorly when certain laboratory experiments are simulated on them.

Models with a human-understandable structure also have a further advantage: expert knowledge can be added to these models without difficulty. This "injected" knowledge can comprise linguistic rules (Fazzolari et al., 2013), mathematical expressions (Kozek and Sinanović, 2008; Greblicki, 1992), semi-physical analogies (Blanco et al., 2014) or many others. Fuzzy Rule-Based Systems (FRBS) arguably constitute the most popular mechanism for integrating data and expert knowledge, although in this paper we propose that Genetic Programming-Based Learning (GPBL) is more flexible (Tsakonas, 2014), as GPBL expressions include FRBS as a particular case. FRBS are valid chains in a context-free grammar and GPBL algorithms for FRBS exist (Sanchez et al., 2015). In contrast, there are many human-understandable models with a grammar-based definition that cannot be easily expressed in terms of IF-THEN rules; see, for instance, (Villar et al., 2015).

Nonetheless, the use of GPBL for learning the hidden health variables of a battery is not straightforward. This is because the learning algorithm must jointly minimize the residuals of the approximation to the output voltage, the temperature and the derivative of the output voltage, as previously mentioned. While Multi-Objective Genetic Programming (MOGP) is a well-known technique, it is generally thought of as an anti-bloat technique in single-objective problems (Vladislavleva et al., 2009; Giri et al., 2013; Ni et al., 2013; Ji Ni and Rockett, 2015; Nag and Pal, 2016), where complexity measures are added to the optimization problem as new goals. Recent advances in the survival selection mechanisms of Multi-Objective Evolutionary Algorithms (MOEA) have not yet been ported to MOGP. Examples of these as-yet unported mechanisms are the adoption of a set of well-distributed reference points in the fitness landscape (Deb and Jain, 2014), the use of new Pareto-preference relationships (Yuan et al., 2016) and decompositions based on the boundary intersection approach (Li and Zhang, 2009).

For all these reasons, the methodological aim of this paper is to extend some multi-criteria identification algorithms for dynamic models to GPBL with the ultimate goal of learning human-understandable health models of batteries. The algorithms SPEA2 (Zitzler et al., 2001), NSGA2 (Deb et al., 2002), NSGA3 (Deb and Jain, 2014), MOEA/D (Li and Zhang, 2009) and θ-DEA (Yuan et al., 2016) will be ported to GPBL, giving rise to the extensions named MOGAP-SPEA2, MOGAP-NSGA2, MOGAP-NSGA3, MOGAP-MOEA/D and MOGAP-θDEA, respectively. The suitability of these extensions will be assessed via a real-world battery diagnosis problem in which certain health-related variables of a LiFePO4 battery will be synthesized with the aid of these extensions. Apart from its practical interest, the problem is also relevant from a methodologic point of view because of the presence of *dominance resistant* individuals with respect to the third objective, i.e. the first derivative of the voltage with respect to the stored charge, which require the adoption of specialized survival selection strategies.

The structure of the present paper is as follows. Section 2 provides a review of the state-of-art in multi-objective genetic programming. The proposed extensions are introduced in Section 3, while Section 4 details the battery diagnosis problem motivating the present study. Section 5 contains the experimental design and discusses the results thus obtained and the conclusions of the paper are presented in Section 6.

2. Multi-objective Genetic Programming-based Learning Models

Although tree-based data structures associated with Genetic Programming-based Learning (GPBL) have been well-studied both in classification problems and in symbolic regression problems (La Cava et al., 2016)), there are only a few papers on the use of these structures in dynamic models (Villar et al., 2015). Most of these studies focus on optimizing scalar fitness functions, while the majority of the multi-objective implementations of Genetic Programming (GP) are intended to combine anti-bloat techniques with a numerical fitness function (Poli et al., 2008, 2010).

Human-understandable structures must be concise and should be based on a limited catalogue of symbols. In this respect, Grammar-based Genetic Programming or Grammatical Evolution (GE) is one of the broadest-ranging sub-areas of GP. Its modularity has been widely used to represent restrictions on general domains describing constraints and interactions within systems (O'Neill and Ryan, 2001; Dempsey et al., 2007; Brabazon et al., 2008). There are also studies in which GE has been applied to learn linguistically understandable rules from data (Garcia et al., 1999), although some authors claim that GPBL may be preferred to rule-based learning systems in terms of the interpretability of the extracted knowledge in many applications (Tsakonas, 2014). GE is often criticized because of the disruptiveness of grammar-based crossover operators, while many authors have studied the definition of operators that make small changes in the genotype, thus avoiding large changes in the phenotype (McKay et al., 2010; Vanneschi et al., 2014; Fenton et al., 2014; Lourenço et al., 2016).

The key issue with respect to Grammatical Evolution is the handling of phenotypes to guide the search towards feasible genotypes (Lourenço et al., 2016). As already stated, some aspects of the combination of GE and multi-objective learning have not been studied in depth, the most relevant studies on Multi-Objective

Genetic Programming (MOGP) being related to the parsimony of the solutions and the joint optimization of different quality measurements during the learning, including information about the complexity of the problem, anti-bloat control and elitist-based dominance relationships.

2.1. MOGP and GE for reducing complexity and bloating

Most of the prevalent Multi-Objective Genetic Programming implementations have dealt with anti-bloat techniques in single-objective problems. Complexity measures are added as additional criteria and the evolution is guided towards the joint optimization of the quality of the solutions and their accuracy.

Complexity measures are important when a human-understandable form is needed (Vladislavleva et al., 2009; Giri et al., 2013). Given that ephemeral random floating-point constants are prone to produce large trees, hybrids between GA and GP have been devised that operate with compact representations (Howard and D'Angelo, 1995). Specific genetic programming mutation operators that reduce the complexity in the population have also been studied by Ni (Ni et al., 2013; Ji Ni and Rockett, 2015) for regression problems. Lastly, there are also applications of MOGP to system identification tasks, where the accuracy and the complexity of the model are treated as separate objectives (La Cava et al., 2016).

It should be noted that Grammar-Based Multi-Objective Genetic Programming does not require a separate discussion, as it shares the same pros and cons of MOGP, although there are fewer studies in this field (Pappa and Freitas, 2009; Bader-El-Den et al., 2009; Mariani et al., 2016)

2.2. Evolution control: dominance and elitism

As stated previously, dominance resistant solutions are present in the problem being addressed in this paper. These solutions require specific operators and survival selection strategies.

Custom operators for MOGP have been studied in different fields of application. For example, Bhowan (Bhowan et al., 2013) studied MOGP using either Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Zitzler et al., 2001) or Non-dominated Sorting Genetic Algorithm II (NSGA2) (Deb et al., 2002) for classification problems with unbalanced data. This paper highlights the importance of finding correct fitness evaluation strategies for MOGP and concludes that, for this particular problem, SPEA2 is preferable to NSGA2. Many other studies have focused on MOGP for classification problems that make use of custom operators; see for instance (Diosan and Andreica, 2015; Smid et al., 2015; Lazarus, 2015; Hiroyasu et al., 2015). One of the most comprehensive efforts to assess a range of MOGPs for classification problems through the receiver operating characteristic (ROC) was presented by Wang (Wang et al., 2014). The list of algorithms includes a survival selection mechanism via Multi-Objective Evolutionary Algorithms based on Decomposition (MOEA/D) (Qingfu Zhang and Hui Li, 2007). In Wang's study, it was observed that classical GP tree-based mutation operators are very disruptive for the problem under study; thus, tailor-made crossover and mutation operators are designed, with the NSGA2 survival selection mechanism being finally considered the preferable option. A subsequent paper proposed the Convex Hull-Based Multi-Objective Genetic Programming (CH-MOGP) algorithm for improving NSGA2.

The diversity in the population for grammar-driven MOGP and symbolic regression was studied by Tsakonas (Tsakonas, 2014). Rezaee (Rezaee et al., 2013) minimized two different objectives in the optimization of Quantum-Dot Cellular Automata electronic circuits. Finally, Olague performed an automatic synthesis of operators for image analysis (Olague and Trujillo, 2012), employing a three-objective approach. The environmental selection strategy in this study is Strength Pareto Evolutionary Algorithm 2 (SPEA2).

Lastly, a set of approaches for automating the selection of the heuristics that control the evolution has emerged in recent years. Hyper-heuristics are defined by Burke as “heuristics to choose heuristics" (Burke et al., 2013). Hyper-heuristics operate on a search space of heuristics as elements or components rather than directly on the search space of solutions to the underlying problem. Nguyen implemented a hyper-heuristic framework for a dynamic Multi-Objective Job Shop Scheduling problem with a cooperative GP (Nguyen et al., 2014). Three low level heuristics were compared: SPEA2, NSGA2, and Harmony distance-based MOEA. The above paper proposed two new algorithms: multi-objective GP together with hyper-heuristic (MO-GPHH) methods and diversified multi-objective cooperative evolution (DMOCC). The ability of GP to generate heuristics and represent complex programs or rules was highlighted in these studies.

3. Proposed methodology

This study aims to apply multi-objective GE to learn model-based soft sensors with a human-understandable form to measure the battery health of a vehicle without the help of “virtual lab” experimentation. As mentioned in the Introduction, three criteria need to be jointly optimized.

Different proposals related to MOGP could be adapted to this problem, although none of them can be used unless some changes are effected. This is because one of our criteria is related to the derivative of the prediction and this derivative has large deviations for small changes in the model. Hence, dominance resistant solutions are frequent. As of yet, this particular problem remains unsolved for GP/GE.

Given that many-objective GAs are designed to cope with dominance resistant solutions (Bechikh et al., 2017), we suggest that certain components of many objective GAs can be reused in MOGP even though the problem at hand has just three objectives. In particular, we propose to use a set of well-distributed reference points, as done in recent GAs such as NSGA3 (Deb and Jain, 2014) and θ-DEA (Yuan et al., 2016).

Furthermore, some other minor design decisions are described in this section. Soft sensors depend on models with many numerical constants and a compact, human-readable expression is sought. Thus, a codification similar to that of Genetic Algorithm-Programming (GAP) (Howard and D'Angelo, 1995) is used. GAP individuals have two parts: a tree-based expression and a chain of parameters. Ephemeral random constants are not used, but rather identifiers that point to a position in a chain of numbers. This chain is evolved alongside the expression tree. Mutations and crossovers take part in the chain and the tree part with different probabilities.

The pseudo-code of our grammar-based Multi-Objective Genetic Algorithm Programming (MOGAP) is shown in Algorithm 1. The list of input parameters is:

1. The problem, Pr, which is evaluated to a vector of numerical criteria.

2. A Multi-Objective Evolutionary Algorithm, MOEA, and its associated list of parameters, MOEA$_\theta = \{\theta_1, \theta_2, \ldots, \theta_p\}$.

3. The maximum number of generations, g.

4. The breeding operators. Crossover and mutation operators have different probabilities for the tree part and the chain of parameters. These are p_{hlx}; p_{llx} for the crossover, and p_{hlm}; p_{llm} for the mutation.

5. A local search operator, LS, that can take up to ils iterations.

The details of this algorithm are as follows:

- **Initialization**: Tree parts are generated using the ramped half-and-half method. Floating-point values in the chain part of the chromosome are initialized at random with a uniform distribution with bounds lb and ub. The chain

Algorithm 1: MOGAP

```
Input: $Pr$; MOEA; MOEA$_\theta$; $g$; $p_{hlx}$; $p_{llx}$; $p_{hlm}$; $p_{llm}$; LS; $ils$
pop ← Initialize_Population()
Evaluate_Population()
for $i \leftarrow 0$ to $g$ do
    MOEA.Pre_Breeding_Procedure()
    MOEA.Parent_Selection()
    GaP.Crossover( $hlxp, llxp$)
    GaP.Mutation( $hlmp, hlmp$)
    Evaluate_Generated_Individuals()
    for $j \leftarrow 0$ to $ils$ do
        Low_Level_Local_Search()
    MOEA.Survival_Selection()
```

part not only codifies the values of the numerical constants in the tree part but also other additional values that shall be evolved. For instance, in the particular case of θ-DEA, the list of parameters MOEA$_\theta$ contains a tuple $\{H_1, H_2\}$ from which three sets are to be generated: reference points, nadir points and ideal points. H_1 and H_2 represent the divisions of boundary and inner layers, respectively. The generated linear hyperplanes are needed for locating well-distributed reference points in the feasible space; (see (Yuan et al., 2016) for details).

- **Pre-breeding**: This procedure ensures that the algorithm recomputes some values before crossover and mutation. Continuing with the preceding example (θ-DEA), the pre-breeding step requires updating the ideal points and applying the cluster operator.

- **Recombination**: Two individuals are selected from the mating pool via a parent selection strategy based on tournament selection with a randomly selected subpopulation (of size seven in this paper). Separate crossover operators are applied to the chain and tree parts. Chain parts are recombined by means of Simulated Binary Crossover (SBX) with a large distribution index and probability p_{llx} by means of an integer-based two-point crossover operator. Arbitrary-length integer vector coding is used for GE implementations. Furthermore, MOGAP also includes a Low-Level Local Search that optimizes the numerical constants in the chain of parameters for a given tree part. This local search is an essential part of this approach because it makes the crossover less disruptive, helping to keep offspring as near to parents as possible.

- **Mutation**: The mutation operator is also independently applied to the chain and tree parts with probabilities p_{hlm} and p_{llm}, respectively. A polynomial mutation with a large distribution index (Cordón et al., 2003) is applied to the chains. The tree-part mutation consists in selecting a node at random and replacing it by a valid alternative in the grammar-based context.

- **Fitness**: The multi-objective fitness function for MOGAP is a vector whose components are the different criteria: residuals of the approximation for the observable variables (voltage and temperature), and differences between the derivatives of predicted and measured voltages. Note that battery models are stateful, but initial conditions are unknown; thus these conditions are evolved alongside the tree-based expression of the model. The assessment of the fitness function depends on these conditions (initial charge, initial temperature, etc.)

- **Survival Selection**: The fitness landscape for the first two objectives (residuals of the approximations of voltage and temperature) is different to that of the third (mean of the squared differences between the derivatives of the true voltage and the model prediction). Both residuals of the model (in voltage and temperature) are smooth, but the derivative of the voltage is not; small changes in the parameters can elicit large deviations of the derivative of the voltage. In practical terms, this means that the offspring of two individuals with a good fitness in the three criteria often include an individual also with a good fitness in the two first objectives, but a bad fitness in the third, which neither dominates the parents nor is dominated by them. In other words, there are many dominance resistant solutions. As a consequence of this, if a pure dominance ranking were used to guide the selection (as is done, for instance, in NSGA-II), the population may evolve to a set of non-dominated individuals and hence the selection would be purely guided by the crowding distance, thereby hindering the evolutionary process.

  Interestingly enough, this same problem also occurs with many-criteria algorithms and therefore we propose to copy the survival selection strategies for MOGAP from those of many-objective algorithms. These strategies include indicator-based approaches, decomposition techniques with different aggregation functions (Weighted Sum and Tchebycheff), and boundary intersection approaches, as implemented in SPEA2 (Zitzler et al., 2001), NSGA2 (Deb et al., 2002), NSGA-3 (Deb and Jain, 2014), MOEA/D (Li and Zhang, 2009) and θDEA (Yuan et al., 2016), among others. Each of these survival strategies has been generalized to MOGAP. The identifiers MOGAP-SPEA2, MOGAP-NSGA2, MOGAP-NSGA3, MOGAP-MOEA/D and MOGAP-θDEA will be used in the remaining part of the paper to refer to these alternatives. In particular, the following parameters were chosen:

- A fixed-size elite archive in MOGAP-SPEA2 equal to 7.
- Tchebycheff's aggregation function with a neighborhood size of 10 and predefined weights in MOGAP-MOEA/D.
- Divisions of boundary and inner layers for MOGAP-NSGA-3 and MOGAP-θDEA, with $H_1 = 4$ and $H_2 = 3$ respectively.
- A predefined penalty parameter (θ) equal to 2 for MOGAP-θDEA.

The next section contains an empirical study of the respective performances of MOGAP-SPEA2, MOGAP-NSGA2, MOGAP-NSGA3, MOGAP-MOEA/D and MOGAP-θDEA when these methods are applied to learn a soft sensor of the health of a rechargeable Li-ion battery.

4. Practical application: Detection and prognosis of battery deterioration

State-of-the-art techniques for assessing the health of a battery are based on the measurement of certain electrochemical properties of the battery's electrodes which manifest as slope changes in the Open Circuit Potential (OCP) curves of the negative (graphite) and positive (phosphate) electrodes. OCP curves cannot be perceived separately, but rather via their difference, which is the Open Circuit Voltage (OCV) of the battery (see Figure 1). This is the voltage of the battery when it is being charged (or discharged) at an infinitesimal rate. In practice, a reasonable approximation of the OCV curve can be obtained if a slow charge/discharge cycle is applied for 25 hours (the so-called "C25" charge/discharge current).

The slope changes in the OCP curve of the negative electrode appear in the discharge OCV curve when there are phase changes between the different compounds of lithium and carbon co-existing in this electrode. These phase changes can be very precisely traced to different health problems (lithium loss, electrodeposition, mechanical cracks, etc.). In short, the position of the peaks of the derivative of the discharge curve vs. the voltage (Incremental Capacity Analysis or ICA

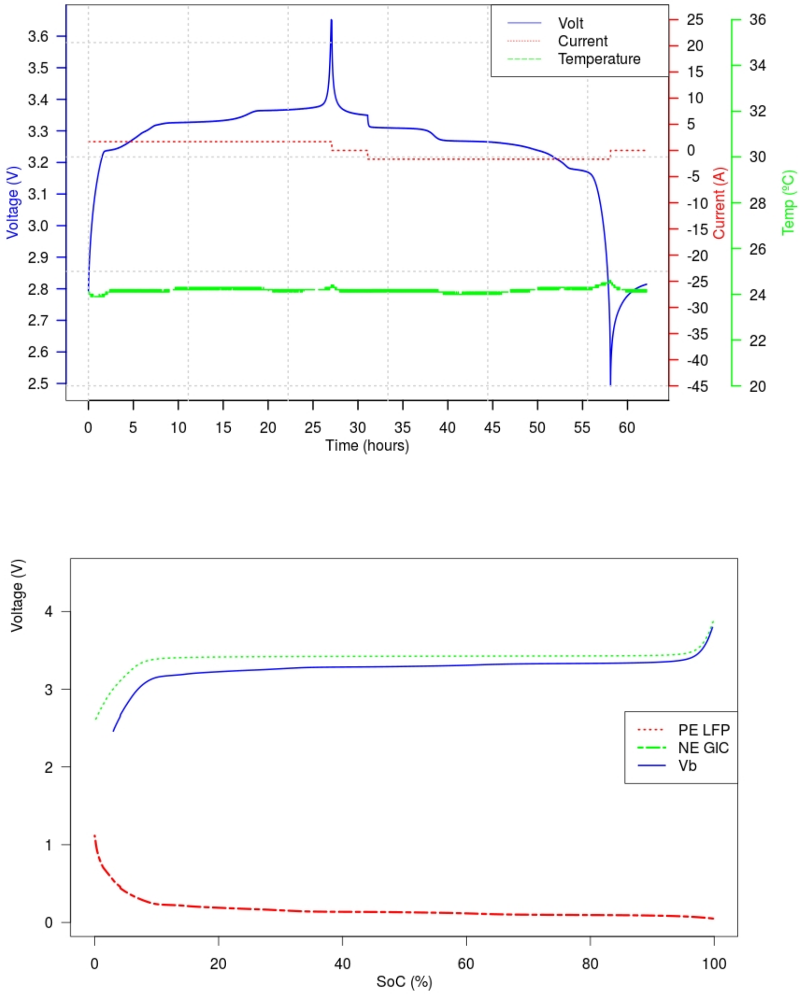

Figure 1: Top part: Current, voltage and battery temperature during a slow charge/discharge cycle; Bottom part: OCV curve, LFP and graphite OCPs. The OCV is the difference between the potentials of the negative and positive electrodes.

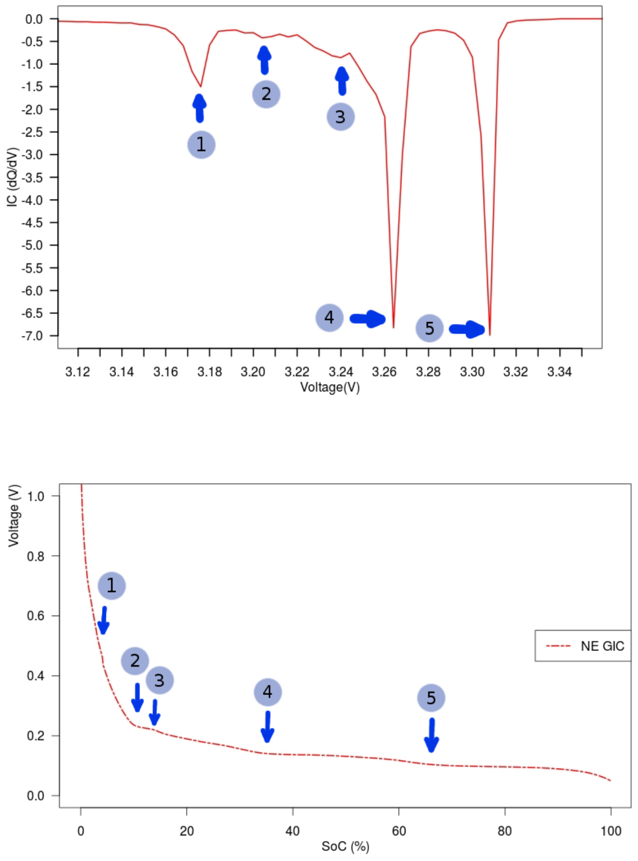

Figure 2: Incremental Capacity (IC) analysis, $\frac{\Delta Q}{\Delta V}$, of the Voltage discharge segment, and characteristic points of the NE GIC and ICA curves.

curve) constitutes an effective diagnostic tool (see Figure 2, top part). An important open problem is how to retrieve the position of these peaks while the battery is being used in an electric vehicle (thus enabling the driver to become aware of any health problems) without the need to take the battery apart and test it in the laboratory for a day or more. To the best of our knowledge, there are no methods that can retrieve the position of the ICA curve peaks using online data.

The proposal made in this paper is to use Grammatical Evolution to learn a model-based soft sensor of the battery health that can be used on-vehicle. The intended output of this sensor is a list of locations where the slope of the OCP curve of the negative electrode changes (see Figure 2, bottom) which are also the locations of the peaks of the ICA curve. The underlying model in the sensor hence describes the OCP curves of the battery. As a set of input-output data pairs cannot be obtained, this model will be learned indirectly. By “indirectly”, we mean that a linguistically-understandable input-output model is obtained first. The structure of this model is chosen so that it is a valid chain in a grammar defining the algebraic expression of OCP curves. Once the model is learned and validated with input-output data, the output of the soft sensor is obtained by parsing the tree part of the GE model, as explained in the following paragraphs.

4.1. Battery model

The outputs of the battery model depend on the state of charge, which in turn depends on the entire history of the battery. Hence, the modelling algorithm has to evolve both the model expression and the initial state of charge (Sanchez et al., 2014). This is troublesome, because the voltage of certain Li-ion batteries (and, in particular, automotive LFP batteries constituting the subject of this study) is almost flat for a wide range of states of charge. Thus, there is very little information for estimating the charge from the voltage and temperature, except when the battery is almost completely charged or discharged (Sanchez et al., 2015).

The method of least squares is not sufficient for fitting a model that can be used for locating the peaks in the ICA curve; a term that depends on the derivative of the voltage curve must be added. The reason is that a model can converge to the data in the least squares sense while, at the same time, the derivatives of the data and model may be very different. For instance, if the output of the model is discrete, then the squared error of the approximation can be made arbitrarily low if the discretizing grid is refined, but the derivative of the model is zero almost everywhere (see Figure 3).

The inputs to the proposed model are the current, i_B, applied to the battery, and the ambient temperature, T_A. The outputs are the battery voltage, v_B, and the

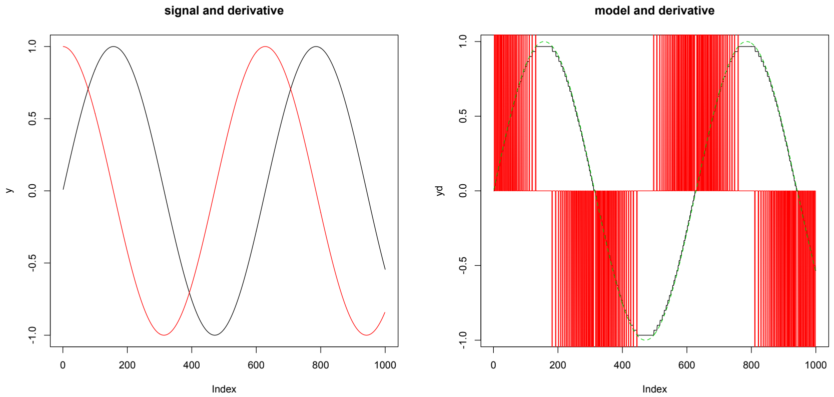

Figure 3: Least squares convergence does not mean that the derivatives of the models are similar. Left: Training data (black) and the derivative (red) of the actual system. Right: Least squares fitting of a linear combination of the step functions (black), the derivative of the model (red) and the training data (green, dashed). The model error can be made as small as required if the number of steps is increased, but its derivative is zero almost everywhere. There are many other cases in which a model converges in least squares, but the derivative of the model does not converge to the derivative of the signal being learned.

battery temperature, T_B. The latent variables are the charge, Q, and the amount of lithium in the positive and negative electrodes, respectively Q_1 and Q_2. Open-Circuit Voltage (OCV) and Open-Circuit Potential (OCP) curves for the positive (PE) and negative electrodes (NE) will be derived from the structure of the model. Both the OCV and OCP are functions of Q. In turn, the output, v_B, depends on the electrode OCPs. For near-zero currents, the following approximation holds:

$$OCV = PE(Q_1) - NE(Q_2). \tag{1}$$

If the current is between low and moderate, then

$$v_B = PE(Q_1) - NE(Q_2) + R * i_B, \tag{2}$$

where the term $R * i_B$ models the transport of lithium from the PE (where it is in the form of lithium-iron-phosphate) to the NE Graphite Intercalation Compound (GIC) during charge ($i_B > 0$). The opposite transport phenomena occur during discharge ($i_B < 0$). It is well known that the aging process mainly alters the negative electrode, NE; thus the PE can be regarded as time-invariant (Anseán et al., 2016). That is to say, the PE can be determined in the laboratory and left unchanged, but a learning model for the NE is needed to assess the State of Health (SoH) of the battery.

We also propose approximating the concentration of lithium in the electrodes via a linear model with coefficients r_p, s_p, r_n, and s_n that take into account the capacity of each electrode and the initial amount of lithium in the phosphate and the graphite:

$$Q_1 = r_p * Q + s_p \tag{3}$$

$$Q_2 = r_n * Q + s_n \tag{4}$$

Lastly, the output T_B is modelled in line with the heat balance equation introduced in (Sanchez et al., 2015),

$$\dot{T}_B = (\alpha(Q_{T_B}, i_B) + \beta T_B) \cdot {i_B}^2 + \gamma \cdot (T_B - T_A) \tag{5}$$

which depends on an additional function, $\alpha(Q_{T_B}, i_B)$, and two parameters, β and γ, that respectively describe the thermal inertia of the battery, the amount of heat that is generated when the battery is charged or discharged, and the thermal resistance between battery and ambient.

4.2. Implementation

Overall, the most challenging part of this health model is that of evolving an expression for the OCP of the negative electrode. On the one hand, there are no input-output pairs and therefore this problem cannot be solved via symbolic regression techniques. We propose to learn a model of the two outputs that depends on the sought OCP, the different parameters (R, β, γ) and the initial values of the latent variables (Q, Q_1, Q_2). The function α is represented by a 2D lookup table, whose components are also evolved by MOGAP.

The three objectives of the problem are the:

1. The mean squared error (MSE) between the measured voltage output and the model v_B (Eq. 2.)

2. The MSE between the temperature output (Eq. 5) and the model output T_B.

3. The MSE between the derivative of the output v_B and the measured battery voltage, in the discharge segments.

The voltage profile of the negative electrode is also assumed to be a valid chain in the context-free grammar included in Listing 1. The terminal symbols $k0$ to $k24$ are pointers to positive numbers in the chain of parameters. The voltage changes

Listing 1: Grammar file for MOGAP in this study

```
    START ::= POS - POS * logit(QSCALED) - SUMSTEPS
 SUMSTEPS ::= POS |
              POS * q |
              POS * SUMSTEPS |
              SUMSTEPS + SUMSTEPS |
              sigmoid(QSCALED)
  QSCALED ::= POS * q - POS
      POS ::= k0 | k2 | ... | k24
```

that arise when the battery is almost empty or full are modelled by means of the following logit function:

$$\text{logit}(p) = \log(\frac{p}{1-p}) \tag{6}$$

while the slope changes associated with characteristic points on the OCP curves are modelled via sigmoids:

$$\text{sigmoid}(x) = \frac{1}{1+\exp(-x)} \tag{7}$$

Thus, the location of the peaks in the ICA curve is expected to be associated with the arguments of the "sigmoid" function in the chain, without the need to obtain the ICA curve numerically, which is a computationally expensive procedure. For instance, given the model

```
k0 - k1 * logit(k2*q-k3)
  - (k4*sigmoid(k5*q-k6)+k7*sigmoid(k8*q-k9)+k10*q+k11)
```

the output of the soft sensor is the pair $(k6/k5, k9/k8)$; i.e. the values of q where the arguments of the sigmoid function is zero. Note that this grammar enforces the monotonicity of the OCP curves because all valid models are additions of monotonic functions or products of a positive constant and a monotonic function.

5. Experimental design, validation of results and discussion

The benchmark used for validating the proposed algorithms was set out in (Echevarria et al., 2017) and comprises data from two $LiFePO_4$ cells manufactured by European Batteries (see Figure 4). The average operating voltage of these

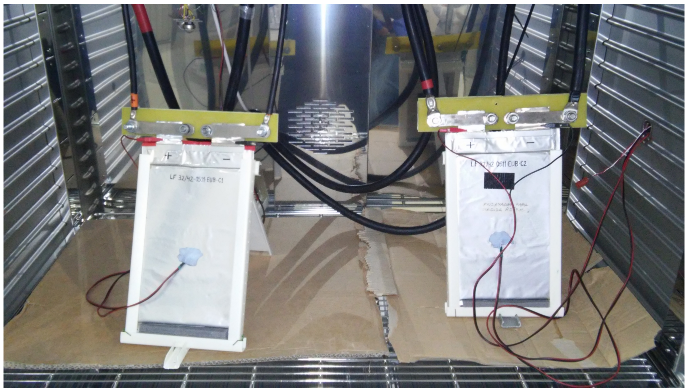

Figure 4: ICP 750 climate chamber with the $\mathrm{LiFePO_4}$ pouch-type cells from European Batteries used in this study.

batteries is 3.2 V. The discharge and charge cut-off voltages are 2.5 V and 3.65 V, respectively. The dimensions are: height, 275 mm; length, 166.5 mm; and width, 13.3 mm. The cells weigh 1010 g and 1008 g, respectively. Full charge/discharge cycles, at a constant current rate, were applied to the batteries. The current, i_B, voltage, v_B, ambient temperature, T_A, and battery temperature, T_B, were sampled every 2 seconds.

5.1. Experimental design

The experimental design for MOGAP reported in this paper comprises the assessment of the following five algorithms: MOGAP-SPEA2, MOGAP-NSGA2, MOGAP-NSGA3, MOGAP-MOEA/D and MOGAP-θDEA. Each algorithm was run twenty times. Two batteries were used for testing purposes. Half of the times, the training dataset comprised data from Bat #1 and the test dataset comprised the data from Bat #2, while the roles of training and test were reversed the other half of the times.

The same population sizes and mutation and crossover probabilities were used for all algorithms. The values of the learning parameters are shown in Table 1. The local search algorithm for the present experimentation is a variant of Nelder Mead on a sequence of subspaces proposed by Tom Rowan, called the “Subplex” algorithm (Rowan, 1990).

Each Pareto Front (PF) was built by combining all the non-dominated models that were found by each algorithm in these twenty runs. The decision to generate a combined Pareto Front (or “Super Pareto Front”) is not the only possible alternative. However, other approaches such as the attainment surfaces or the

Parameter	**Value**
Generations	50 000
Population size	≈ 60
Local search iterations $ipls$	25
High level crossover probability P_{hlx}	0.9
Low level crossover probability P_{llx}	0.8
High level mutation probability P_{hlm}	0.01
Low level mutation probability P_{llx}	0.01

Table 1: Parameters for the experimentation

identification of a set with representative points are also implicit in the numerical study if a binary indicator measure is selected. Moreover, statistical studies with unary quality indicators, such as Inverted Generational Distance, require the true Pareto of the problem. The resulting Super Pareto Front was assessed via Zitzler's binary additive ϵ-indicator ($I_{\epsilon+}$) (Zitzler et al., 2002): Given two PF A and B, $I_{\varepsilon+}(A, B) = \inf\{\varepsilon \in \mathbb{R} \mid \forall b \in B \, \exists a \in A : a \succeq_{\varepsilon+} b\}$, where ε expresses the minimum factor that can be added to each objective value in $b \in B$, and the resulting objective remains weakly dominated by $a \in A$. If $I_{\epsilon+}(A, B) = 0$ and $I_{\epsilon+}(B, A) = 0$, then A and B represent the same PF ($A = B$). If $I_{\epsilon+}(A, B) \leq 0$ and $I_{\epsilon+}(B, A) > 0$ then A $I_{\epsilon+}$-dominates over B ($A \lhd B$). Otherwise, $I_{\epsilon+}(B, A) \leq 0$ and $I_{\epsilon+}(A, B) > 0$ and hence B $I_{\epsilon+}$-dominates over A ($A \lhd B$). If both epsilons are greater than zero, then A and B are not comparable ($A \parallel B$).

5.2. Numerical results and discussion

In Figure 5 the sigmoids in the model expression are plotted together with the discharge curve. Note that not all the peaks were successfully retrieved, only the most relevant ones (in this case, peaks 2, 3, 4 and 5 of the ICA curve in Figure 2). Both the accuracy and the number of peaks that are detected decrease when the current is higher: up to four (out of five peaks) can be retrieved for low currents, but only two (peaks 4 and 5) are found for moderate currents. If the current is high, the changes in the slope are smoothed out by the kinetic effects of the battery and the placement of the sigmoids will not be reliable.

In order to validate the methodological decisions taken in the design of the algorithm, a 3D view of the Pareto Fronts thus obtained is shown at the top of Figure 6. This front is decomposed in three drawings at the bottom of the same figure.

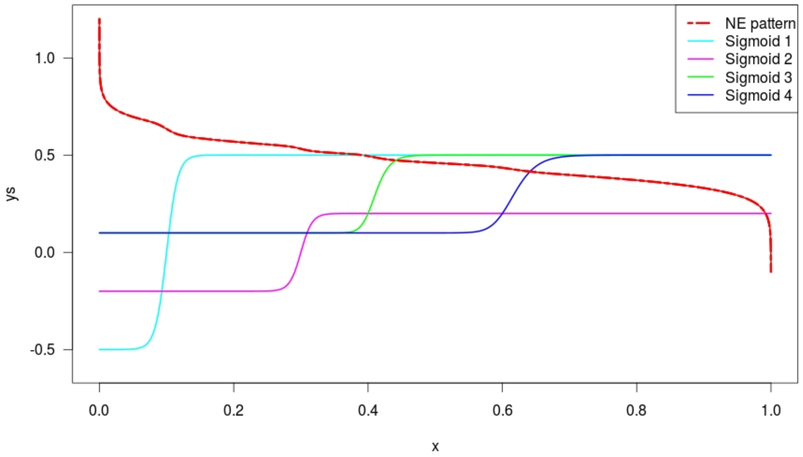

Figure 5: The location of the peaks in the ICA curve is associated with the positions of the sigmoids in the model expression.

Note that selection strategies with either MOGAP-SPEA2 or MOGAP-NSGA2 produce solutions with high dispersion, while MOGAP-NSGA3, MOGAP-MOEA/D and MOGAP-θDEA are concentrated.

Numerically, Table 2 shows the Additive Epsilon Indicator for each pair of studied algorithms. The lower triangular matrix shows the relationships based on the indicators $I_{\epsilon+}(A, B)$ and $I_{\epsilon+}(B, A)$. Note that the SPEA2 and NSGA2 strategies do not dominate one another. Notice also that MOGAP-NSGA3 outperforms the rest of the assessed setups for the proposed MOGAP. In addition to these data, the binary extension of the Inverted Generational Distance Indicator, $I_H(A, B)$, provides complementary information on the diversity in PFs and is displayed at the bottom of the same table.

Finally, note that the proposed soft sensor has no comparable counterparts in the specialized battery literature. Furthermore, OCP curves cannot be observed and it is difficult to state how accurate the battery model in this sensor is. Nonetheless, we propose that the “voltage error” criterion for small currents (the OCV curve) serves as an informal measurement of the attainable accuracy. In order to obtain a numerical reference of the quality of the proposed method, a recurrent neural network and a Genetic Fuzzy System (GFS) were used to learn a dynamic model of the battery. The neural network is a Long Short-Term Memory (LSTM) network with 50 hidden nodes and an output linear layer. The inputs to the net-

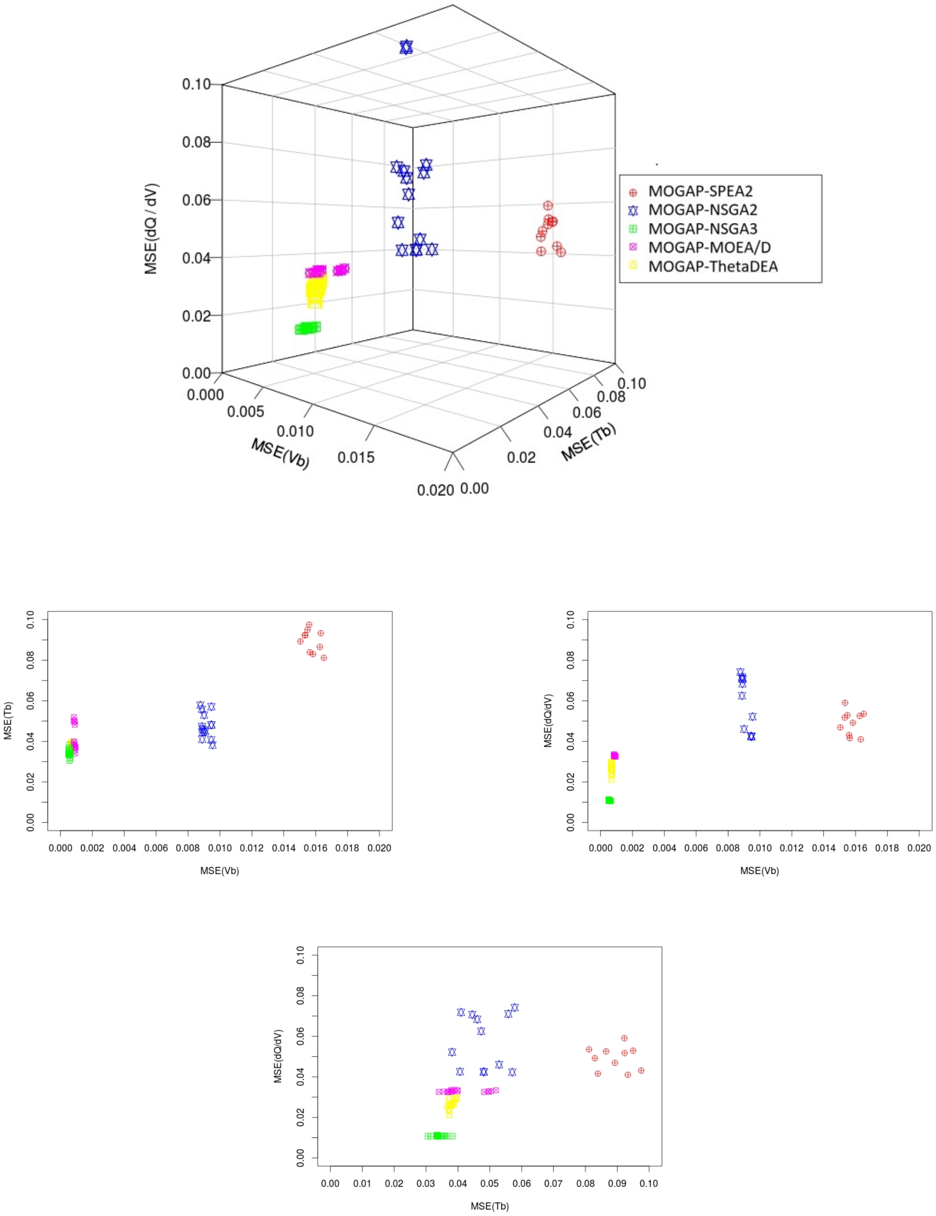

Figure 6: Pareto Fronts obtained from the different survival strategies assessed in MOGAP. Top: 3D view. Bottom: Projections over pairs of criteria.

work were the current, i_B, and the charge of the battery, Q. The neural estimation of the OCV curve was obtained by simulating a C25 cycle with the learned battery model. The GFS is a soft sensor described in the paper (Sánchez et al., 2017). Table 3 provides a summary of the results.

6. Concluding remarks and future work

A model-based soft sensor of battery health has been developed that outputs the location of the characteristic points of the negative electrode of an automotive battery when the battery is being charged or discharged at a low current. Grammatical Evolution has been used to enforce the monotonicity of the models and also to produce a human-readable structure that allows the position of the characteristic points to be obtained symbolically, parsing the model expression.

The main difficulty arose from the need to minimize the prediction error in both the voltage and its derivative. Given that small changes in the model lead to a large spread of these derivatives, dominance resistant individuals appear that demand a custom multi-objective GP implementation. The proposed MOGAP algorithm incorporates recent advances developed for many-objective GAs, such as a set of well-distributed reference points in the fitness landscape and decompositions based on boundary intersections. Different alternatives were compared and it was concluded that the MOGAP-NSGA3 survival strategy is a reasonable choice for solving this problem.

The method presented in this paper is competitive with other data-driven intelligent algorithms, though there is still room for improvement. In future work, both the positive and the negative electrodes shall be included in the model. Lastly, let us recall that not all the characteristic points, but only the most relevant ones can be retrieved (recall Figure 5). This means that the sensor performs a partial battery diagnostic and there are certain health events that will not be detected. However, this problem is inherent to the kinetic behaviour of batteries and it is unlikely that an algorithmic solution can be found, as phase changes in the electrodes cannot be perceived in the output voltage for high currents.

Acknowledgements

This work was partially funded by the Eureka SD project (Agreement no. 2013-2591), which is supported by the EU's Erasmus Mundus programme. It also received funding from the Spanish Ministry of Science and Innovation (MICINN) under Grants TIN2017-84804-R, TEC2016-80700-R and DPI2013-46541-R and

	A = MOGAP-SPEA2		A = MOGAP-NSGA2		**A = MOGAP-NSGA3**		A = MOGAP-MOEA/D		A = MOGAP-θDEA	
	$I_{\epsilon+}(A,B)$	relation	$I_{\epsilon+}(A,B)$	relation	$I_{\epsilon+}(A,B)$	relation	$I_{\epsilon+}(A,B)$	relation	$I_{\epsilon+}(A,B)$	relation
B = MOGAP-SPEA2	0	=	0.042963		0.050633		0.047121		0.044606	
B = MOGAP-NSGA2	0.001319	$\|$	0	=	0.031650		0.010031		0.021061	
B = MOGAP-NSGA3	**-0.014538**	$\lhd$	**-0.007669**	$\lhd$	0	=	-3.401E-4		-1.555E-4	
B = MOGAP-MOEA/D	-0.008463	$\lhd$	-0.004158	$\lhd$	**0.021867**	$\rhd$	0	=	0.011278	
B = MOGAP-θDEA	-0.014382	$\lhd$	-0.001642	$\lhd$	**0.010588**	$\rhd$	0.002515	$\|$	0	=

	A = MOGAP-SPEA2		A = MOGAP-NSGA2		A = MOGAP-NSGA3		A = MOGAP-MOEA/D		A = MOGAP-θDEA	
	$IGD(A,B)$	relation	$IGD(A,B)$	relation	$IGD(A,B)$	relation	$IGD(A,B)$	relation	$IGD(A,B)$	relation
B = MOGAP-SPEA2	0	=	0.011668		0.013865		0.010969		0.011326	
B = MOGAP-NSGA2	0.010879		0	=	0.007705		0.003129		0.004139	
B = MOGAP-NSGA3	0.020926		0.016336		0	=	0.005385		0.003577	
B = MOGAP-MOEA/D	0.013899		0.010898		0.004990		0	=	0.001423	
B = MOGAP-θDEA	0.017720		0.011958		0.002517		0.001497		0	=

Table 2: Top part: Binary Additive Epsilon Indicator-based assessment. Bottom part: Inverted Generation Distance providing complementary information on diversity.

Tested Battery	MOGAP	LSTM	GFS
bat #1	0.00458	0.00781	0.00946
bat #2	0.00470	0.00786	0.00951

Table 3: Compared error values (average of the absolute error in volts) of the proposed method MOGAP method, an LSTM neural network and a Genetic Fuzzy System.

by the Regional Ministry of the Principality of Asturias under grant FC-GRUPIN-IDI/2018/000226.

References

Anseán, D., Dubarry, M., Devie, A., Liaw, B. Y., García, V. M., Viera, J. C., González, M., 2016. Fast charging technique for high power LiFePO 4 batteries: A mechanistic analysis of aging. Journal of Power Sources 321, 201–209.

Bader-El-Den, M., Poli, R., Fatima, S., 2009. Evolving timetabling heuristics using a grammar-based genetic programming hyper-heuristic framework. Memetic Computing 1 (3), 205–219.

Bechikh, S., Elarbi, M., Ben Said, L., 2017. Many-objective Optimization Using Evolutionary Algorithms: A Survey. In: Recent Advances in Evolutionary Multi-objective Optimization. Springer, pp. 105–137.

Bhowan, U., Johnston, M., Zhang, M., Yao, X., 2013. Evolving Diverse Ensembles Using Genetic Programming for Classification With Unbalanced Data. IEEE Transactions on Evolutionary Computation 17 (3), 368–386.

Blanco, C., Sanchez, L., Gonzalez, M., Anton, J. C., Garcia, V., Viera, J. C., 2014. An Equivalent Circuit Model With Variable Effective Capacity for LiFePO4 Batteries. IEEE Transactions on Vehicular Technology 63 (8), 3592–3599.

Brabazon, A., O'neill, M., Dempsey, I., 2008. An Introduction to Evolutionary Computation in Finance. IEEE Computational Intelligence Magazine 3 (4), 42–55.

Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Qu, R., 2013. Hyper-heuristics: a survey of the state of the art. Tech. Rep. 12, University of Nottingham, School of Computer Science, Nottingham, UK.

Cordón, O., Herrera-Viedma, E., Luque, M., De Moya, F., Zarco, C., 2003. Analyzing the performance of a multiobjective GA-P algorithm for learning fuzzy queries in a machine learning environment 2715, 611–619.

Deb, K., 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, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6 (2), 182–197.

Dempsey, I., O'Neill, M., Brabazon, A., 2007. Constant creation in grammatical evolution. International Journal of Innovative Computing and Applications 1 (1), 23.

Diosan, L., Andreica, A., 2015. Multi-objective breast cancer classification by using multi-expression programming. Applied Intelligence 43 (3), 499–511.

Dubarry, M., Svoboda, V., Hwu, R., Yann Liaw, B., 2006. Incremental Capacity Analysis and Close-to-Equilibrium OCV Measurements to Quantify Capacity Fade in Commercial Rechargeable Lithium Batteries. Electrochemical and Solid-State Letters 9 (10), A454.

Dubarry, M., Truchot, C., Liaw, B. Y., 2012. Synthesize battery degradation modes via a diagnostic and prognostic model. Journal of Power Sources 219, 204–216.

Echevarria, Y., Sanchez, L., Blanco, C., 2017. Li-Ion Battery charge/discharge benchmark.
URL `https://data.mendeley.com/datasets/r4n22f4jfk/1`

Fazzolari, M., Alcala, R., Nojima, Y., Ishibuchi, H., Herrera, F., 2013. A Review of the Application of Multiobjective Evolutionary Fuzzy Systems: Current Status and Further Directions. IEEE Transactions on Fuzzy Systems 21 (1), 45–65.

Fenton, M., McNally, C., Byrne, J., Hemberg, E., McDermott, J., O'Neill, M., 2014. Automatic innovative truss design using grammatical evolution. Automation in Construction 39, 59–69.

Garcia, S., Gonzalez, F., Sanchez, L., García, S., González, F., Sánchez, L., 1999. Evolving Fuzzy Rule Based Classifiers with GA-P: A Grammatical Approach. In: European Conference on Genetic Programming. Springer, pp. 203–210.

Giri, B. K., Hakanen, J., Miettinen, K., Chakraborti, N., 2013. Genetic programming through bi-objective genetic algorithms with a study of a simulated moving bed process involving multiple objectives. Applied Soft Computing 13 (5), 2613–2623.

Greblicki, W., 1992. Nonparametric identification of Wiener systems. IEEE Transactions on Information Theory 38 (5), 1487–1493.

Hiroyasu, T., Shiraishi, T., Yoshida, T., 2015. A feature transformation method using multiobjective genetic programming for two-class classification. In: 2015 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 2989–2995.

Howard, L., D'Angelo, D., 1995. The GA-P: a genetic algorithm and genetic programming hybrid. IEEE Expert 10 (3), 11–15.

Ji Ni, Rockett, P., 2015. Tikhonov Regularization as a Complexity Measure in Multiobjective Genetic Programming. IEEE Transactions on Evolutionary Computation 19 (2), 157–166.

Kozek, M., Sinanović, S., 2008. Identification of Wiener models using optimal local linear models. Simulation Modelling Practice and Theory 16 (8), 1055–1066.

La Cava, W., Danai, K., Spector, L., Fleming, P., Wright, A., Lackner, M., 2016. Automatic identification of wind turbine models using evolutionary multiobjective optimization. Renewable Energy 87, 892–902.

Lazarus, C., 2015. Pareto-Dominance Based MOGP for Evolving Soccer Agents. In: 2015 IEEE Symposium Series on Computational Intelligence. IEEE, pp. 280–287.

Li, H., Zhang, Q., 2009. Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation 13 (2), 229–242.

Lourenço, N., Pereira, F. B., Costa, E., 2016. Unveiling the properties of structured grammatical evolution. Genetic Programming and Evolvable Machines 17 (3), 251–289.

Mariani, T., Guizzo, G., Vergilio, S. R., Pozo, A. T. R., 2016. Grammatical Evolution for the Multi-Objective Integration and Test Order Problem. In: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference - GECCO '16. ACM Press, New York, New York, USA, pp. 1069–1076.

McKay, R. I., Hoai, N. X., Whigham, P. A., Shan, Y., O'Neill, M., 2010. Grammar-based Genetic Programming: a survey. Genetic Programming and Evolvable Machines 11 (3-4), 365–396.

Nag, K., Pal, N. R., 2016. A Multiobjective Genetic Programming-Based Ensemble for Simultaneous Feature Selection and Classification. IEEE Transactions on Cybernetics 46 (2), 499–510.

Nguyen, S., Zhang, M., Johnston, M., Tan, K. C., 2014. Automatic Design of Scheduling Policies for Dynamic Multi-objective Job Shop Scheduling via Cooperative Coevolution Genetic Programming. IEEE Transactions on Evolutionary Computation 18 (2), 193–208.

Ni, J., Drieberg, R. H., Rockett, P. I., 2013. The Use of an Analytic Quotient Operator in Genetic Programming. IEEE Transactions on Evolutionary Computation 17 (1), 146–152.

Olague, G., Trujillo, L., 2012. Interest point detection through multiobjective genetic programming. Applied Soft Computing 12 (8), 2566–2582.

O'Neill, M., Ryan, C., 2001. Grammatical evolution. IEEE Transactions on Evolutionary Computation 5 (4), 349–358.

Pappa, G. L., Freitas, A. A., 2009. Evolving rule induction algorithms with multi-objective grammar-based genetic programming. Knowledge and Information Systems 19 (3), 283–309.

Poli, R., Langdon, W. B., McPhee, N. F., Koza, J. R., 2008. A Field Guide to Genetic Programming. http://lulu.com.

Poli, R., Vanneschi, L., Langdon, W. B., McPhee, N. F., 2010. Theoretical results in genetic programming: the next ten years? Genetic Programming and Evolvable Machines 11 (3-4), 285–320.

Qingfu Zhang, Hui Li, 2007. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation 11 (6), 712–731.

Rezaee, R., Houshmand, M. M., Houshmand, M. M., 2013. Multi-objective optimization of QCA circuits with multiple outputs using genetic programming. Genetic Programming and Evolvable Machines 14 (1), 95–118.

Rowan, T., 1990. Functional Stability Analysis of Numerical Algorithms. Ph.D. thesis, University of Texas at Austin.

Sanchez, L., Blanco, C., Anton, J. C., Garcia, V., Gonzalez, M., Viera, J. C., 2015. A Variable Effective Capacity Model for LiFePO4 Traction Batteries Using Computational Intelligence Techniques. IEEE Transactions on Industrial Electronics 62 (1), 555–563.

Sánchez, L., Couso, I., González, M., 2014. A design methodology for semi-physical fuzzy models applied to the dynamic characterization of LiFePO4 batteries. Applied Soft Computing 14, 269–288.

Sánchez, L., Couso, I., Otero, J., Echevarría, Y., Anseán, D., 2017. A model-based virtual sensor for condition monitoring of li-ion batteries in cyber-physical vehicle systems. Journal of Sensors 2017.

Sanchez, L., Couso, I., Viera, J. C., 2014. Online SOC estimation of Li-FePO4 batteries through a new fuzzy rule-based recursive filter with feedback of the heat flow rate. In: Vehicle Power and Propulsion Conference (VPPC), 2014 IEEE. IEEE, pp. 1–6.

Smid, J., Pilat, M., Peskova, K., Neruda, R., 2015. Multi-Objective Genetic Programming for Dataset Similarity Induction. In: 2015 IEEE Symposium Series on Computational Intelligence. IEEE, pp. 1576–1582.

Torai, S., Nakagomi, M., Yoshitake, S., Yamaguchi, S., Oyama, N., 2016. State-of-health estimation of LiFePO 4 /graphite batteries based on a model using differential capacity. Journal of Power Sources 306, 62–69.

Tsakonas, A., 2014. An analysis of accuracy-diversity trade-off for hybrid combined system with multiobjective predictor selection. Applied Intelligence 40 (4), 710–723.

Vanneschi, L., Castelli, M., Silva, S., 2014. A survey of semantic methods in genetic programming. Genetic Programming and Evolvable Machines 15 (2), 195–214.

Villar, J. R., de la Cal, E. A., Sedano, J., Garcia Tamargo, M. A., 2015. Simple heuristics for enhancing GP learning. Logic Journal of IGPL 23 (3), 472–484.

Vladislavleva, E., Smits, G., den Hertog, D., 2009. Order of Nonlinearity as a Complexity Measure for Models Generated by Symbolic Regression via Pareto Genetic Programming. IEEE Transactions on Evolutionary Computation 13 (2), 333–349.

Wang, P., Tang, K., Weise, T., Tsang, E., Yao, X., 2014. Multiobjective genetic programming for maximizing ROC performance. Neurocomputing 125, 102–118.

Weng, C., Cui, Y., Sun, J., Peng, H., 2013. On-board state of health monitoring of lithium-ion batteries using incremental capacity analysis with support vector regression. Journal of Power Sources 235, 36–44.

Weng, C., Feng, X., Sun, J., Peng, H., 2016. State-of-health monitoring of lithium-ion battery modules and packs via incremental capacity peak tracking. Applied Energy 180, 360–368.

Yuan, Y., Xu, H., Wang, B., Yao, X., 2016. A New Dominance Relation-Based Evolutionary Algorithm for Many-Objective Optimization. IEEE Transactions on Evolutionary Computation 20 (1), 16–37.

Zitzler, E., Laumanns, M., Thiele, L., 2001. SPEA2: Improving the Strength Pareto Evolutionary Algorithm, 95–100.

Zitzler, E., Laumanns, M., Thiele, L., Fonseca, C. M., da Fonseca, V. G., Eckart Zitzler, Marco Laumanns, L. T. C. M. F. V. G. d. F., da Fonseca, V. G., 2002. Why Quality Assessment of Multiobjective Optimizers Is Difficult. In: GECCO '02 Proceedings of the Genetic and Evolutionary Computation Conference. Morgan Kaufmann Publishers Inc., pp. 666–674.