IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 11, 2021, accepted November 1, 2021, date of publication November 4, 2021, date of current version November 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3125519

Multi-Objective Optimization of Electric Arc Furnace Using the Non-Dominated Sorting Genetic Algorithm II

MATHEUS F. TORQUATO[1], **GERMÁN MARTÍNEZ-AYUSO**[2], **ASHRAF A. FAHMY**[1,3], **AND JOHANN SIENZ**[1]

[1] Advanced Sustainable Manufacturing Technologies 2020, Swansea University, Swansea SA1 8EN, U.K.
[2] Medical School, Swansea University, Swansea SA2 8PP, U.K.
[3] Department of Electrical Power and Machines, Helwan University, Helwan 11795, Egypt

Corresponding author: Matheus F. Torquato (m.f.torquato@swansea.ac.uk)

This work was supported by the Advanced Sustainable Manufacturing Technologies (ASTUTE) 2020 operation supporting manufacturing companies across Wales, part-funded by the European Regional Development Fund through the Welsh Government and the participating Higher Education Institutions.

ABSTRACT Combining classical technologies with modern intelligent algorithms, this paper introduces a new approach for the optimisation and modelling of the EAF-based steel-making process based on a multi-objective optimisation using evolutionary computing and machine learning. Using a large amount of real-world historical data containing 6423 consecutive EAF heats collected from a melt shop in an established steel plant this work not only creates machine learning models for both EAF and ladle furnaces but also simultaneously minimises the total scrap cost and EAF energy consumption per ton of scrap. In the modelling process, several algorithms are tested, tuned, evaluated and compared before selecting Gradient Boosting as the best option to model the data analysed. A similar approach is followed for the selection of the multi-objective optimisation algorithm. For this task, six techniques are tested and compared based on the hypervolume performance indicator to just then select the Non-dominated Sorting Genetic Algorithm II (*NSGA-II*) as the best option. Given this applied research focus on a real manufacturing process, real-world constraints and variables such as individual scrap price, scrap availability, tap additives and ambient temperature are used in the models developed here. A comparison with an equivalent EAF model from the literature showed a 13% improvement using the mean absolute error in the EAF energy usage prediction as a comparative metric. The multi-objective optimisation resulted in reductions in the energy consumption costs that ranged from 1.87% up to 8.20% among different steel grades and scrap cost reductions ranging from 1.15% up to 5.2%. The machine learning models and the optimiser were ultimately deployed with a graphical user interface allowing the melt-shop staff members to make informed decisions while controlling the EAF operation.

INDEX TERMS Electric arc furnace, genetic algorithms, multi-objective optimisation, NSGA-II, optimisation, steel-making.

I. INTRODUCTION

Steel is the world's most popular construction material because of its unique combination of durability, workability, and cost. It is 100% recyclable and can be recycled indefinitely without losing its quality. As a consequence, steel is the most recycled material in the world [1]. It is produced via two main routes: the first and more common is the blast furnace-basic oxygen furnace (BF-BOF) route which is also known as primary steel-making and uses mostly new iron as feedstock. The second is the electric arc furnace (EAF) route which is also known as secondary steel-making and uses scrap steel as primary raw material.

Even amid joint efforts between private sector companies and governments to achieve a net-zero carbon goal, the steel industry is still responsible for about 5% of greenhouse gas emissions globally, making it one of the highest-emitting industrial sectors, alongside concrete. In the United Kingdom

The associate editor coordinating the review of this manuscript and approving it for publication was Ran Cheng.

only, emissions from steel reached approximately 17m tonnes of carbon dioxide (CO_2) in 2012.

Steel manufacturing is a highly energy-demanding sector [2]. Although the energy consumption per ton of produced steel has reduced by 50% in the past 30 years, the production and recycling of steel are processes that demand a great deal of energy and suffer from significant energy losses, something around 100 kWh/t [3]. This high energy consumption drives the search for alternative methods and new ways to optimise this process, not to mention the efforts on reducing the amount of CO_2 released to the environment. As a consequence, the need for primary steel production could diminish to near zero in the next 30 years in an effort to reduce even further the environmental impact.

In a global trend towards reducing the carbon footprint, the steel manufacturers have been gradually adopting the use of Electric Arc Furnaces (EAF) which melt the metal feedstock using electric arcs generated between carbon electrodes. This manufacturing technique is mainly used for recycling scrap steel and others scrap metals in general. Simply stated, steel recycling involves melting down the scrap metal and adjusting its chemistry to obtain the desired steel specifications.

Figure 1 illustrates the main stages in the steel-making process using an electric arc furnace. It is important to note that, in addition to EAF, there is also another furnace where refining operations are carried out to check and correct the chemistry of the steel. Each furnace is described in more detail in Section III-A4.

A. MAIN CONTRIBUTIONS

The presented research introduces a new approach for the optimisation of the EAF-based steel-making process based on multi-objective optimisation using evolutionary computing. Differently from other works from literature, this research models the two steps in the EAF steel production process which includes the electric arc furnace and the ladle furnace.

This is a data-driven approach using a large amount of real-world historical data collected from the melt shop in an established steel plant that not only creates machine learning models for both furnaces but also simultaneously minimises the total scrap cost and EAF energy consumption per ton of scrap.

Real-word constraints and variables such as individual scrap price, available quantity, tap additives and ambient temperature are used in the models developed here, but not found in any other model simultaneously.

In order to produce the optimisation software which is one of the greatest contributions from this work, several machine learning models using different algorithms are developed, trained, tuned, tested and evaluated. A similar approach was taken in the choice of the multi-objective optimisation (MOO) algorithm used. Such comprehensive analysis has not been found to date in researches involving this kind of manufacturing application.

II. RELATED WORKS

There have been several attempts to optimise, tune and control the different parameters of the EAF melting process. However, due to its complexity and the numerous sub-processes involved, the fundamental mechanisms involved are relatively poorly understood. Consequently, some crucial aspects are usually overlooked or simplified.

In [4] a detailed description of the controls used to perform an electric arc furnace electrical optimisation is presented. A mathematical model was developed to analyse the furnace's electrical characteristics and the dynamic behaviour of the automatic control system governing electrode motion. A prototype diagnostic system for identifying the phases of smelting based on the harmonics of the arc currents was developed. The 5% increase in the mean productivity of the DSP-180 furnace achieved by [4] has significant importance since it addresses the highest operating cost of an EAF, the power consumption. This diagnostic system can reach even superior results when combined with optimisation techniques involving the ladle furnace and the EAF scrap input, for example.

In [5], the melting process and chemical reactions are described and modelled as equilibrium zones with mass transport limitations. The main focus of this work involves the application of mathematical modelling techniques to develop a dynamic model of the EAF steel-making process. Although parameter estimation was conducted using industrial data, it seems like the amount of data available for the study was not enough since the only piece of information used to follow the progress of each batch was the off-gas composition data. A sensitivity analysis was carried out in order to identify the least sensitive parameters in the model, but still, it should be noticed that a degree of empiricism was introduced to model relationships where the real mechanisms are either too complex to be modelled or insufficient information was available.

Even more complex models were developed with the increase in computational power, which enabled the use

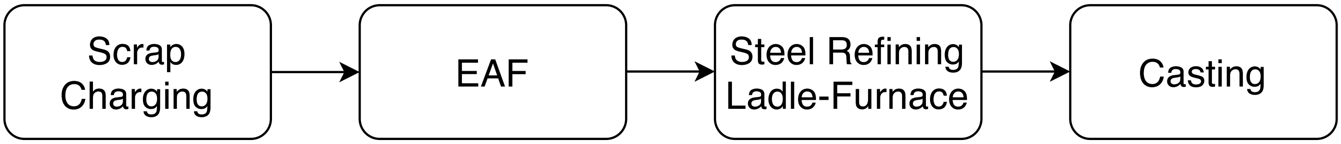


FIGURE 1. EAF steel production process.

of finite elements methods (FEM) or computational fluid dynamics (CFD) for such complex modelling problems.

The vibration of EAF electrodes was the focus of the research presented in [6]. A detailed model of the EAF system was built by introducing the flexible dynamics of structures and a suitable modelling of the arc, of the control and of the electromechanical forces. This model was developed by employing multibody dynamics (MBD) in cooperation with finite element methods. The effects of the arc dynamics and of the magnetic induction among the phases were analysed through a multiphysics approach. As a result, the proposed model was able to simulate the dynamic response of the EAF despite some approximations and a few structural optimisation recommendations were given.

A more complex model based on CFD simulations was presented in [7]. In this research, the authors studied the thermal effect of the coal particle combustion and the electrode base arc radiation. Then, the amount of CO and CO_2 was calculated as well as the temperature distribution inside the furnace. The CFD approach allowed also obtaining the flow field inside the furnace which severely affected the temperature field distribution and the furnace performance. The authors highlighted the contribution of radiation and emphasised how difficult it was to assess its impact on the model.

A critical disadvantage of numerical methods, such as FEM and CFD, is that the greater the complexity introduced in the model, the more computationally expensive it becomes. The time required to solve a system of equations depends on many factors, such as simulation time, number of elements, number of physics models, etc. Many of these parameters increase computation time exponentially. With these computational requirements, long working electric furnace cycles can take significant amounts of resources and time, making these methods unsuitable for a few optimisation problems. To overcome this problem, some assumptions and simplifications need to be made at the cost of reducing accuracy and resolution.

As an alternative to the previous computationally expensive FEM and CFD models, some authors have proposed to use different statistical approaches to represent, model, simulate and optimise the comportment of EAFs. These statistical approaches learn the behaviour of the furnace from available data such as temperature, electric energy consumption, *CO* release levels, etc. Opposite to the previously mentioned models, once the statistical models have learned the behaviour of the system (during the training phase), they can quickly provide a solution for a new set of input data, predicting the behaviour of the same system for a scenario never seen before during the training stage. There have been several studies that have proposed different statistical methods to approach the optimisation of electric arc furnaces.

The research presented in [8] proposed the optimisation of energy and scrap selection in Electric Arc Furnaces using multivariate data analysis (MVDA). For that, 12665 heats in total from six different Scandinavian melt-shops were used in the analysis. Three separate models were created to predict the three different end-conditions: chemical composition of the steel, specific electrical energy consumption and metallic yield. The Root Mean Square Error of Estimation (RMSEE) was calculated for each predicted variable and whereas the RMSEE for chemical prediction ranged from the order of magnitude of 10^{-3} to 10^{-1}, the equivalent value for energy prediction (kWh/t) was in the order of magnitude of 10^{0}. Naturally, some chemical elements could be predicted with much higher accuracy than others. For instance, carbon was found to be one of the most difficult elements to predict. Lastly, the optimisation of scrap recipes was mentioned, but no conclusions about this were presented. The author further develops on the topic at [9], but at this time focusing on estimating scrap properties based on the evaluation of historical process data.

In [10] a linear programming model was created with two main purposes: 1) optimising the scrap recipe with a focus on cost reduction and 2) optimising scrap transportation and loading operations to increase the production rate. Many steel companies and technology service firms are interested in developing tools able to find optimal recipes using different types of raw materials in order to reduce production cost and this is also one objective of the implementation here developed in this research. By using the model developed in [10] the author was able to reduce the scrap feedstock cost by between 2% and 6% and the charging time by between 2 and 10 minutes. As one of the conclusions of the research, the author suggests that the use of Artificial intelligence (AI) techniques might help to improve the model performance.

The research presented in [11] also used linear programming to approach the optimisation of seven key variables: the electricity consumption, electrode consumption, percentage of scrap melted, average arc current, oxygen input, gas input and carbon input. The optimal control design problem described in the paper is a linear programming problem with a linear objective function and linear equality and inequality constraints. The ultimate goal was to optimise the production cost while maintaining quality and efficiency. Reductions between 7% and 22% were achieved in terms of production cost. The author recognises that non-linear effects are not included in the model since they cannot be represented by linear programming.

In [12], two models were presented: the first was based on linear regression, and the second was based on genetic programming. The purpose of these models was to deliver an EAF electric energy consumption model. For the genetic programming model, the fitness function took into consideration the average relative deviation between the predicted and the experimental energy consumptions. The in-house genetic programming system was run 100 times in order to develop 100 models for the prediction of electric energy consumption. Each run lasted approximately two and a half hours on a desktop computer. As a result, the authors claimed that the average electric energy consumption could be reduced by up to 1.04% and 1.16% by using linear regression and the genetic model, respectively.

Recently, AI, more specifically, artificial neural networks (ANN), were employed in the optimisation of certain specific parameters of EAFs, but with a different focus than what has been discussed so far.

Gajic *et al.* [13] used artificial neural networks for modelling EAFs with a focus on exploring the extent and effect that fluctuations in the chemical composition had on electrical energy consumption. A feed-forward neural network composed of three layers trained with the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm was employed. The content of carbon, chromium, nickel, silicon and iron were used as inputs and their impact on electrical consumption was investigated. As a result, the proposed model successfully predicted the values of the electrical energy consumption in function of the chemical composition of the charge material mix. The carbon content was highlighted as the most relevant parameter.

In [14], two supervised machine learning algorithms were developed, one using ANN and the other using support vector regression. These models were trained to predict the amount of alloying additives need to obtain the desired chemical composition of white cast iron. The 300 melting batches of data used in the research were collected from a foundry and the parameters used in the modelling involved chemical composition as input and the amount of alloying additives added during the alloying process as output.

ANNs were also used in [15] where a regression model was used to predict the decarburisation rate based on lance height and total oxygen flow. The authors managed to achieve a coefficient of determination of 0.98 in comparison to the value of 0.45 achieved previously by a Linear regression. The data used in this research was collected from a six-ton BOF (Basic Oxygen Furnace) pilot plant converter located at Swerea MEFOS, Sweden.

Similar to previous references, [16] also produced a prediction model. This time, a deep learning model was used to predict the energy usage for an electric arc furnace (EAF). Using 10,990 instances of data collected from the melt shop in an established steel plant in South Wales, UK, the group of researchers managed to produce a model that achieved a mean absolute error of $1.5mWh$. In order to train the single-output deep learning model, 16 attributes were used as input including scrap materials and additives.

III. METHODS

A. BACKGROUND INFORMATION AND VARIABLES DEFINITION

1) REGRESSION MODEL

Machine learning allows computer systems to tackle tasks that are too difficult to solve with fixed programs written and designed by human beings [17]. Therefore, data scientists use many different types of machine learning algorithms to discover hidden patterns in large data sets that often lead to actionable insights. At a high level, these different algorithms can be classified into two groups, based on how they learn about the data to make predictions: supervised and unsupervised learning [18].

Regression is an example of a machine learning algorithm based on supervised learning which models dependent variables as a non-linear combination of parameters and one or more independent variables. The model can be univariate (single dependent variable) or multivariate (multiple dependent variables) [19], [20].

A large number of industrial machine learning applications use supervised learning. In this case, an algorithm is used to learn a mapping function, f, from a set of input variables $\mathbf{X}$ to a set of output variables $\mathbf{Y}$.

Mathematically, this can be expressed as:

$$\mathbf{Y} = f(\mathbf{X}, \mathbf{B}) + \varepsilon, \tag{1}$$

where $\mathbf{B}$ represents an array of nonlinear coefficients to be computed during the training phase of the regression model. Once these coefficients are found they are used in the mapping function f. $\mathbf{B}$ can be expressed as

$$\mathbf{B} = [\beta_1, \beta_2, \ldots, \beta_k]. \tag{2}$$

$\mathbf{X}$ is an array of the independent variables which are observed in the training data (inputs) and can be expressed as

$$\mathbf{X} = [x_1, x_2, \ldots, x_m]. \tag{3}$$

$\mathbf{Y}$ is an array of the dependent variables which are also observed in the data used for training the model and can be expressed as

$$\mathbf{Y} = [y_1, y_2, \ldots, y_n]. \tag{4}$$

ε represents the error term.

A machine learning model can be a mathematical representation of a real-world process and its purpose is to train the function f so well that accurate predictions of the output variable $\mathbf{Y}$ can be made even when input data $\mathbf{X}$ not previously seen is presented to the model.

2) MULTI-OBJECTIVE OPTIMISATION

Multi-objective optimisation (also known as multi-objective programming, vector optimisation, multi-criteria optimisation or Pareto optimisation) is an area of multiple-criteria decision making concerning mathematical optimisation problems involving more than one target variable to be optimised simultaneously. The optimisation becomes even more challenging when the objectives are conflicting, that is, achieving the optimal value for one objective requires some compromise on one or more of other objectives, for example, maximising performance while minimising fuel consumption and emission of pollutants of a vehicle.

Multi-objective optimisation has been applied to many fields of science and engineering, where optimal decisions need to be taken in the presence of trade-offs. In such cases, a multi-objective optimisation study should be carried out, providing not one, but several solutions that represent the compromise between the objective functions [21]–[23].

A subset of MOO problems is the constrained multi-objective optimisation problems (CMOPs) which, as the name suggests, refer to multi-objective optimisation problems that have restrictions. In the literature, it is possible to find a wide range of applications of this kind of optimisation such as human resource allocation [24]; optimisation of reinforced concrete structures [25]; optimisation of carbon fibre drawing process [26]; urban planning [27] and many others [28].

In mathematical terms, a constrained multi-objective optimisation problem can be formulated as:

Given γ objective functions | $\gamma \in \mathbb{N}$:

$$\text{minimise } \mathbf{Y} = \mathbf{F}(\mathbf{X}) = (f_1(\mathbf{X}), \ldots, f_\gamma(\mathbf{X}))$$

$$\text{subject to } \begin{cases} \mathbf{X} \in \mathbf{D} \\ \mathbf{G}(\mathbf{F}(\mathbf{X})), \end{cases}$$

$$\text{where } \begin{cases} f_i : \mathbf{D} \longrightarrow \mathbf{D}_y, \; i = 1, 2, \ldots, \gamma \\ \mathbf{G}(\mathbf{F}(\mathbf{X})) \subset \mathbf{D}_y \\ \mathbf{D} \subset \mathbb{R}^m, \end{cases} \tag{5}$$

X is described in Equation 3 and x_{lower} and x_{upper} are, respectively, the lower and upper limits of all independent variables $x_i, \; i = 1, 2, \ldots, m$, which make **D**:

$$\mathbf{D} = \begin{Bmatrix} x_{1\_lower} \leqslant x_1 \leqslant x_{1\_upper} \\ x_{2\_lower} \leqslant x_2 \leqslant x_{2\_upper} \\ \vdots \\ x_{m\_lower} \leqslant x_m \leqslant x_{m\_upper} \end{Bmatrix}^T . \tag{6}$$

D is called the feasible domain or decision space. As a consequence, this feasible domain limits the co-domain from $\mathbb{R}^n$ to $\mathbf{D}_y$ [26], [29]–[31].

$\mathbf{G}(\mathbf{F}(\mathbf{X}))$, the restricted co-domain, is defined by w inequality constraints which can be expresses as

$$\mathbf{G}(\mathbf{F}(\mathbf{X})) = \begin{bmatrix} y_{1\_lower} - y_1 \leqslant 0 \\ y_1 - y_{1\_upper} \leqslant 0 \\ y_{2\_lower} - y_2 \leqslant 0 \\ y_2 - y_{2\_upper} \leqslant 0 \\ \vdots \end{bmatrix}^T \tag{7}$$

Both $\mathbf{D}_y$ and $\mathbf{G}(\mathbf{F}(\mathbf{X}))$ are illustrated in Figure 2.

In multi-objective optimisation, there is usually no single, viable solution that minimises all objective functions simultaneously. Therefore, attention is paid to the Pareto optimal set **P** i.e. solutions that cannot be improved in any of the objectives without degrading at least one of the other objectives in **Y**. In mathematical terms, a feasible solution $\mathbf{Y}_e \mid \mathbf{Y}_e \in \mathbf{G}(\mathbf{F}(\mathbf{X}))$ is said to Pareto dominate another solution $\mathbf{Y}_b \mid \mathbf{Y}_b \in \mathbf{G}(\mathbf{F}(\mathbf{X}))$, if:

$$\bullet\, f_{e_i}(\mathbf{X}) \leqslant f_{b_i}(\mathbf{X}), \quad \text{for } i = 1, 2, \ldots \gamma$$
$$\bullet\, f_{e_j}(\mathbf{X}) < f_{b_j}(\mathbf{X}) \text{ for at least one index } j, \quad \text{for } j = 1, 2, \ldots \gamma \tag{8}$$

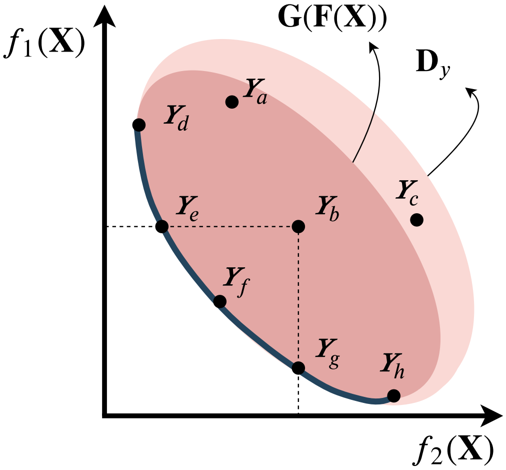


FIGURE 2. Co-domain $\mathbf{D}_y$ and restricted co-domain $\mathbf{G}(\mathbf{F}(\mathbf{X}))$.

In other words, a feasible solution $\mathbf{Y}_e \mid \mathbf{Y}_e \in \mathbf{G}(\mathbf{F}(\mathbf{X}))$ is said to Pareto dominate another solution $\mathbf{Y}_b \mid \mathbf{Y}_b \in \mathbf{G}(\mathbf{F}(\mathbf{X}))$, if

- A solution $\mathbf{Y}_e$ is no worse than $\mathbf{Y}_b$ in all γ objectives
- A solution $\mathbf{Y}_e$ is strictly better than $\mathbf{Y}_b$ in at least one objective.

A solution $\mathbf{Y}_e$ belongs to the Pareto optimal set **P** if there is no other solution that dominates it. The **P** set is also often called as Pareto front, Pareto frontier or Pareto boundary [31].

Figure 2 illustrates some of the concepts previously defined. In this particular case, a multi-objective optimisation problem with $\gamma = 2$ is shown, where the external ellipsoidal area represents the $\mathbf{D}_y$ co-domain and the internal ellipsoid area represents the $\mathbf{G}(\mathbf{F}(\mathbf{X}))$ restricted co-domain. Within $\mathbf{D}_y$, eight viable solutions can be identified ($\mathbf{Y}_{a-h}$), however, only five of them ($\mathbf{Y}_{d-h}$) belong to the Pareto boundary, as described in Equation 8. Finally, it is important to note that, although the solution $\mathbf{Y}_c$ belongs to $\mathbf{D}_y$, it does not belong to $\mathbf{G}(\mathbf{F}(\mathbf{X}))$. Therefore, $\mathbf{Y}_c$ cannot be considered a viable solution.

The multi-objective optimiser used in this paper is a non-dominated sorting-based multi-objective evolutionary algorithm (MOEA) called Non-dominated Sorting Genetic Algorithm II or *NSGA-II* [32], [33] which has the following three features:

1) It uses an elitist principle, i.e. the elites of a population are given the opportunity to be carried to the next generation.
2) Is uses an explicit diversity preserving mechanism (Crowding distance).
3) It emphasises the non-dominated solutions.

3) GENETIC ALGORITHMS

As its name suggests, the *NSGA-II* is a variation of a Genetic Algorithm (GA). The GAs, firstly introduced in [34], are used to solve search and optimisation problems where an optimal solution can be found through an iterative process in which the search starts from an initial population

$$\overline{\mathbf{X}}(0) = [\mathbf{X}_1(0), \mathbf{X}_2(0), \ldots, \mathbf{X}_\eta(0)] \tag{9}$$

and then, combining the best representatives of it, a new population is obtained

$$\overline{\mathbf{X}}(1) = [\mathbf{X}_1(1), \mathbf{X}_2(1), \ldots, \mathbf{X}_\eta(1)] \tag{10}$$

that replaces, partially or totally, the previous one until the last population

$$\overline{\mathbf{X}}(k) = [\mathbf{X}_1(k), \mathbf{X}_2(k), \ldots, \mathbf{X}_\eta(k)] \tag{11}$$

is reached.

GA is an iterative algorithm that is started from a population $\mathbf{X}$ formed by η chromosomes randomly created. In every iteration, also called generation or epoch, the η chromosomes from $\mathbf{X}$ are evaluated, selected, recombined and mutated to form a new population also of η chromosomes. Then, the new population is used as input to the algorithm's next iteration and this population update procedure is repeated k times, where k is the number of GA generations. For each i-th chromosome at the j-th generation, $\mathbf{X}_i(j)$, there is an associated fitness value, $\mathbf{Y}_i(j)$, calculated through the evaluation function.

The flowchart shown in Figure 3 illustrates a generic structure of a GA.

4) STEEL-MAKING PROCESS WITH ELECTRIC ARC FURNACE

As mentioned in Section I, one of the main reasons for adopting the EAF's steel-making method is the need for steelmakers to find possible ways to contribute to the global challenge of reducing CO_2 emissions. As the use of EAFs allows the steel to be made from a scrap feedstock, it can be seen as a much more environmental-friendly option compared to the BF-BOF traditional steel manufacturing process. The environmental factor combined with much lower capital costs has attracted the attention of, not only the steel-making industry but also of smaller emerging enterprises.

The melting of steel scrap in an electric arc furnace is divided into four stages. The first stage is called loading, where the furnace is filled with scrap metal, limestone and the additional additives needed to make a particular grade of steel. The scrap types and quantities selected at the beginning of the process, the recipe, are chosen according to their approximate chemical composition and the targeted grade of steel to be produced. However, given the aggregate nature of scrap where different types of metal might be mixed, predicting the composition of the scrap and hence the final composition of the steel is a real challenge.

After all the elements have been loaded into the EAF, its lid is closed and the electrodes are introduced. In the second stage, the melting, a high electric current flows through the electrodes, forming electrical arcs that heat the scrap up to 1500 °C. During heating, some impurities are burned and escape into the atmosphere as gases. The unburned impurities float on the molten steel in the form of slag. Given the lower density of scrap, impurities and other metal additions, molten steel occupies the bottom part inside the furnace. Then, these first two stages are repeated between 2 or 3 times, until the oven is full, reaching its maximum weight or maximum volume load since the density of the scrap is variable. The chemical composition of the mixture is generally tested during each charge.

Once the furnace is fully loaded and a flat bath is reached, the third stage, the ladle furnace step starts. In this stage, the steel is poured (tapping process) from the EAF into another smaller furnace, the ladle furnace, where refining operations take place to check and correct the steel chemistry and superheat the melt above its freezing temperature. Additions such as alloys and lime might be incorporated into the mix in order to achieve the desired steel grade. Finally, the last step is casting the molten steel to the desired geometry, usually as long bars.

B. IMPLEMENTATION DESCRIPTION

The implementation here developed was based on an actual production dataset containing real industrial data collected from 6423 consecutive batches (or heats) produced within 12 months in a melt-shop located in the UK. Due to the presence of business-sensitive information, the availability of the used dataset is not possible in its entirety, however, information regarding the most important variables and parameters is discussed in the following sections.

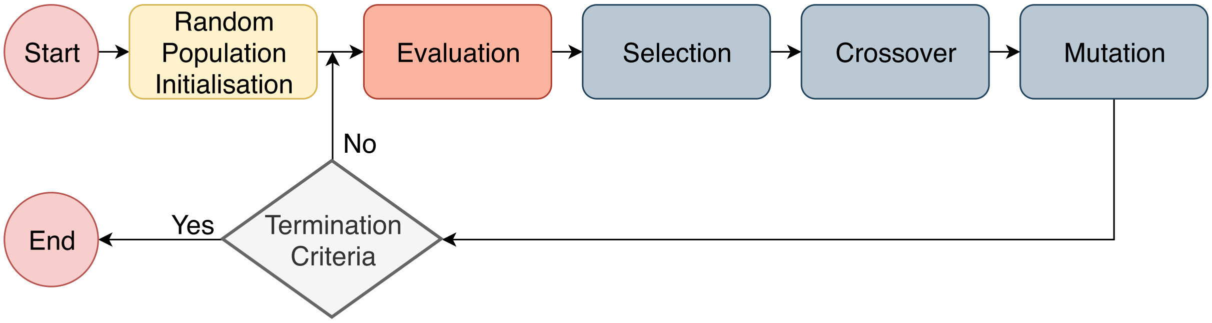


FIGURE 3. Genetic algorithm main structure.

TABLE 1. Parameters of the EAF dataset.

Heat ID	1	2	$\cdots$	6423
Ambient Temperature ($°C$)	15	13	$\cdots$	17
Quantity Scrap 1 (Ton)	23.44	84.21	$\cdots$	110.88
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
Quantity Scrap δ **(Ton)**	27.02	0	$\cdots$	6.11
EAF Additives 1	146	486	$\cdots$	124
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
EAF Additives θ **(kg)**	477	364	$\cdots$	519
EAF Output Composition 1	0.12	0.45	$\cdots$	0.56
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
EAF Output Composition p	0.16	0.08	$\cdots$	0.19
Energy (kWh)	51644	53242	$\cdots$	53017
Tap Additives 1 **(kg)**	101	103	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
Tap Additives κ **(kg)**	39	29	$\cdots$	40
Final Composition 1	0.02	0.05	$\cdots$	0.06
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
Final Composition p	0.02	0.001	$\cdots$	0.66
Steel Grade	A	C	$\cdots$	H

The dataset size is 6423 × 88, which means that for all of the 6423 heats, there are 88 columns containing information such as: ambient temperature; steel grade; heat ID; EAF timing measurements, additives, scrap input quantities, average power and energy used; EAF chemical output composition; ladle additives and chemical output composition. There is a predefined conversion between the final chemical composition measured after secondary refining and each different steel grade. For this, a standard composed of intervals with the minimum and maximum desired values for each chemical element is used. It is important to mention that the chemical composition attributed to each steel grade varies according to the steelmaker.

Table 1 illustrates a transposed snippet of the dataset used to train and test the regression models used in the optimisation. Here, the main variables used in this implementation are represented and, as it can be seen, the majority of the dataset is composed of numeric values, however, the steel grade (bottom row of Table 1) is a categorical variable.

Each individual component necessary for the implementation here presented is detailed in the following sections.

1) EAF REGRESSOR

The EAF regressor has the role of modelling the EAF behaviour and its output is used in the calculation of the fitness of the *NSGA-II* optimiser (Section III-B3). This module has three inputs: 1) an array of δ elements representing the quantity in tonnes of each different category of scrap to be charged into the EAF; 2) an array of θ elements representing the quantity of each additive to be charged into the EAF and 3) the ambient temperature. The output of this regressor is the chemical composition described by p elements measured at the tapping and the energy consumption (kWh) in the EAF.

All models developed in this research were coded in Python 3.8 and used *scikit-learn* [35] as the machine learning library. For the EAF Regressor model, seven different algorithms were used to train the predictive models. K-nearest Neighbours (KNN), Neural Networks (MLP), Random Forest (RF), Linear Regression (LR), Support Vector Regression (SVR), Decision Tree (DT) and Gradient Boosting (GB) were the selected algorithms.

Before the training process, during the data cleaning stage, a filtering technique was applied in order to remove the outlier values in each column of the used dataset. The standard score (*zscore*) threshold used was $\sigma = 4$, i.e. all rows from the 6423 batches that had any element with a *zscore* greater than four were removed from the training set. This operation resulted in a reduction of 22.55% in the dataset size. Figure 4 illustrates this reduction in the distribution of one variable from the dataset, the energy variable.

After that, the numeric columns from the dataset were standardised using a scaling transformation to the range of 0 to 1. As some ML algorithms calculate the distance between two points by the Euclidean distance during the training process, models using data with features highly varying in magnitude, unit, and range would be governed by features with a higher range of values. Hence, the standardisation so that each feature contributes approximately proportionately to the final distance.

In the training step, 85% of the heats available in the filtered dataset were selected randomly as training data, while the remaining 15% was used afterwards for testing the created model.

Using the same training set, the seven previously mentioned ML regression algorithms were trained using their correspondent *scikit-learn* implementations: *KNeighborsRegressor, MLPRegressor, RandomForestRegressor, LinearRegression, SVR, DecisionTreeRegressor* and *GradientBoostingRegressor*. The hyperparameters used for each model were selected programmatically through a hyperparameter optimisation using a randomised search over 25 iterations and a 3-fold cross-validation splitting strategy based on the *RandomizedSearchCV scikit-learn* implementation.

The hyperparameters used for each machine learning model following the hyperparameter optimisation are based on the parameters available in the library used [35]. A detailed list of the tuned hyperparameters is presented below.

- K Neighbors Regressor.
  - Number of Neighbours = 31
  - Leaf Size = 36
  - Weights = *uniform*
  - p = 2
- MLP Regressor.
  - Hidden layer sizes = (45,)
  - Activation = 'relu'
  - Solver = 'adam'
  - Alpha = 0.001

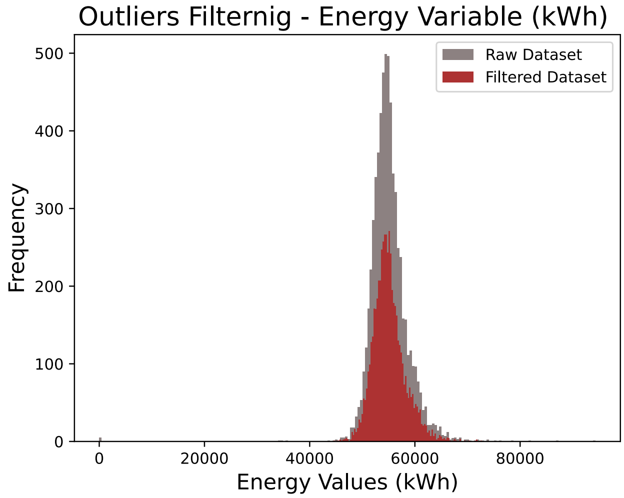


FIGURE 4. Histogram illustrating data reduction in the energy variable after removal of outliers values.

  - Batch size = 'auto',
  - Learning rate = 'constant'
  - Maximum number of iterations = 279
  - Tolerance = 0.0001
  - Momentum = 0.9
  - Epsilon = $1e - 08$
- Random Forest Regressor.
  - Max Depth = 6
  - Number of estimators = 185
  - Min weight fraction leaf = 0
  - Min samples leaf = 18
  - Min samples split = 7
  - Max leaf nodes = 137
  - Criterion = mse
- Linear Regression.
  - Fit intercept = True
- Support Vector Regression.
  - Kernel function = 'rbf'
  - Kernel coefficient = 'scale'
  - Tolerance = $1e - 05$
  - C = 214
  - Epsilon = 0.04
  - Shrinking = True
- Decision Tree Regressor.
  - Criterion = 'mse'
  - Splitter = 'best'
  - Min samples split = 20
  - Min samples leaf = 18
  - Min weight fraction leaf = 0.0
  - Max Depth = 7
  - Max Leaf Nodes = 5
  - Min Impurity Decrease = 0.45
- Gradient Boosting Regressor.
  - Loss = 'lad'
  - Learning rate = 0.21
  - Number of estimators = 36
  - Min Impurity Decrease = 0.6
  - Criterion = friedman_mse
  - Min samples split = 6
  - Min samples leaf = 19
  - Max depth = 9

After fitting all the seven models with tuned hyperparameters the regressors were tested over a prediction using the 15% unseen portion of the filtered dataset dedicated for testing. The metric used to evaluate the performance was the mean square error (MSE). This value measures the average of the squares of the errors i.e. the average squared difference between the estimated values and the actual values. This accuracy metric was used in order to select the model associated with the lowest obtained MSE value.

2) LADLE REGRESSOR

Equivalently to the EAF Regressor described in Section III-B1, the ladle regressor has the role of modelling the ladle furnace behaviour. Also in the same way as the EAF regressor (as illustrated in Figure 5), the ladle regressor has three inputs: 1) an array of p elements containing the chemical composition measured at the EAF tapping (the output of the EAF regressor); 2) an array of κ elements representing the quantity of each additive to be charged into the ladle furnace and 3) the ambient temperature, the same value used in the EAF model. The output of this regressor is the final chemical composition of the steel produced after the secondary refining that happens in the ladle. The final chemical composition is also described by an array of p different elements measured before the casting process.

The ladle model plays a fundamental role in the search for scrap compositions that minimise scrap and energy costs. In order to be considered a feasible solution, every output from this model must meet all the chemical composition constraints based on the target steel grade to be produced.

This model followed the same steps used in the EAF model training. The filtered portion of the training dataset (85%) containing the tap additives values and ambient temperature were used as input to this model alongside the correspondent chemical composition coming from the EAF.

3) OPTIMISER IMPLEMENTATION

The multi-objective optimisation presented in this section was implemented using Platypus [36]. It is a framework for evolutionary computing in Python with a focus on multi-objective evolutionary algorithms (MOEAs).

The diagram from Figure 5 is the main component of the multi-objective optimiser here presented. It uses the two regression models previously described (Sections III-B1 and III-B2) in order to find a selection of scrap quantities that minimises energy consumption and scrap costs while meeting all the chemical composition constraints. This diagram illustrates a high-level abstraction of the fitness function used for solving the inverse problem previously described.

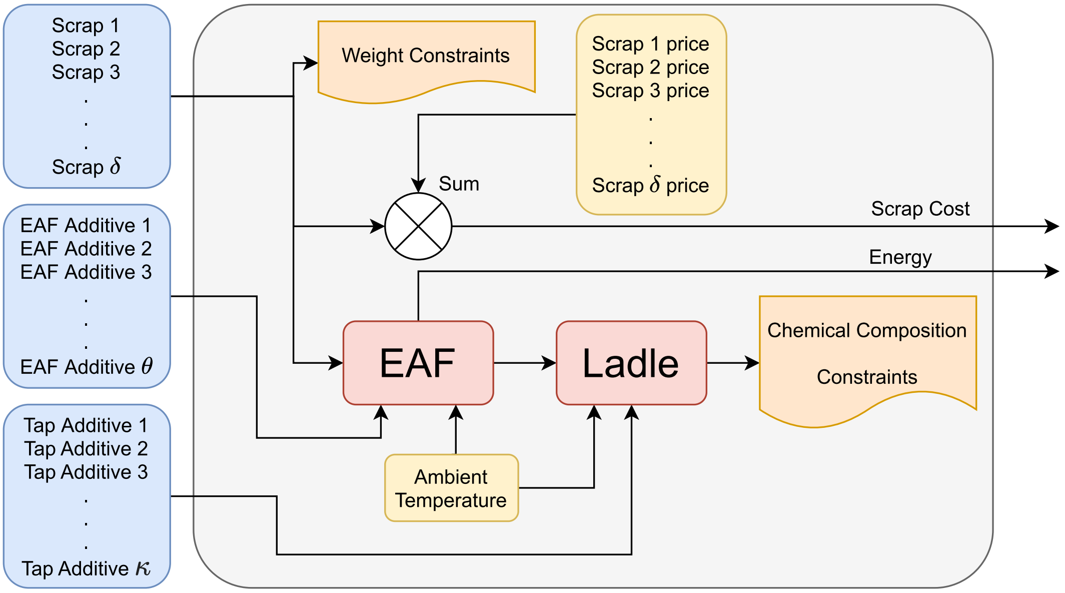


FIGURE 5. High level representation of the *NSGA-II* optimiser input/output.

In summary, given a target grade of steel to be produced (user input), the optimiser provides, given a defined search space **D** (as in Equation 6), the best set of input **X** that optimises (minimises) both ($\gamma = 2$) objective variables **Y** respecting the imposed restrictions **G**(**F**(**X**)) (as in Equation 7).

Here, the set of input variables **X** consist of:

1) The quantity of each different category of scrap to be charged into the EAF;
2) The quantity of each additive to be charged into the EAF, called EAF Additives;
3) The quantity of each additive to be added to the ladle furnace during the tapping, called Tap or Ladle Additives.

Algorithm 1 presents an overview of the *NSGA-II* optimiser used here. The optimisations carried out in this research used a fixed population size of $\eta = 100$ and $k = 10000$ generations.

In the evaluation function or fitness calculation (lines 2 and 10 of Algorithm 1), each chromosome $\overline{\mathbf{X}}(j)$ from the j-th population receives a score according to its correspondent value in the evaluation function (Algorithm 2). For this particular constrained multi-objective optimisation problem where the target is to minimise both the energy consumption and the total scrap cost, the lower both values are, the better is the fitness value for each chromosome.

Algorithm 2, which uses the bold font style to represent arrays, illustrates in detail how the two minimisation targets *energy_per_weight* and *total_scrap_cost* are calculated. It is possible to see that the input values used in the regression models *eaf_regressor* and *ladle_regressor* (lines 6 and 7 of Algorithm 2) are the variables x_i contained in the chromosomes **X**, where i ranges from 1 to $\delta + \theta + \kappa$.

Algorithm 1 *NSGA-II* Pseudocode

```
1:  Randomly Initialise the first Population $\overline{\mathbf{X}}(0)$ of size $\eta$
2:  Evaluate $\overline{\mathbf{X}}(0)$ for all the $\gamma$ objective values
3:  Non-dominated sorting Based on Pareto Dominance
4:  Generate sets of non-dominated fronts
5:  Binary Tournament Selection
6:  Recombination and Mutation
7:  Generate Child Population $\overline{\mathbf{Q}}(0)$ of size $\eta$
8:  for $i = 0$ to $k - 1$ do
9:      Merge $\overline{\mathbf{Q}}(i) \cup \overline{\mathbf{X}}(i)$
10:     Evaluation and Non-dominated sorting based on Pareto dominance
11:     Generate sets of non-dominated fronts
12:     Select chromosomes to compose the new generation
13:     Create Next Generation of $\overline{\mathbf{Q}}(i+1)$ and $\overline{\mathbf{X}}(i+1)$
14: end for
```

Once the objective values are calculated the *NSGA-II* sorts the population based on these newly calculated values and the Pareto Dominance criteria described in Equation 8. The sorting is performed by the Fast Non-dominated Sorting Approach [33] (lines 3 and 10 of Algorithm 1) which assigns non-domination fronts based on fitness values to each chromosome of the population.

After sorting and assigning fitness values for the entire population, the selection process starts. The selection method used here is the binary tournament selection which picks two random chromosomes from the current population and copies the fittest one to the next population. The fittest chromosomes are chosen based on the crowding distance, constraint violation and evaluation function values.

As the genetic algorithm seeks to maintain a good spread of solutions, the crowding distance (a measure of density of solutions in the neighbourhood) helps to identify solutions in the least crowded regions of the Pareto-optimal set. Diversity among non-dominated solutions is implicitly introduced by using the crowding comparison procedure.

Meanwhile, the constraint violation value, as its name suggests, is the score assigned to each chromosome based on how well they fit problem constraints. For this optimisation, the variables *scrap_weight* and *final_composition* from Algorithm 2 are the ones tested against the constraints. Here, the scrap weight inequality constraint is met when the total weight of the scrap selection (*scrap_weight*) is equal to or less than the weight constraint and the final composition is met when all chemical elements of the *final_composition* are within the ranges established by the target grade (as in Equation 7). In this case, one pair of inequalities is used for each chemical element.

If either solution violates constraints, then the solution with a smaller constraint violation is preferred. If both solutions are feasible, then Pareto dominance is used to select the preferred solution.

Once the selection process is finished, the selected chromosomes will then be submitted to the crossover and mutation steps. In the crossover, also called recombination, the previously selected chromosomes are mixed in order to stochastically generate new solutions from existing individuals. The crossover technique used in the *NSGA-II* is the Simulated Binary Crossover (SBX) [37], a real-parameter recombination operator which simulates the working principle of the single-point crossover operator on binary strings [38], [39], firstly introduced in [40]. Here the crossover used probability = 1.0 and distribution index = 15.0.

For the mutation step, Polynomial Mutation (PM) is used. The mutation is a genetic operator used to maintain genetic diversity among generations of the genetic algorithm. The PM employs a polynomial probability [37] distribution to perturb a solution in a parent's vicinity [41]. Probability = 1.0 and distribution index = 20.0 were used.

Given that all the input and output variables used in the *NSGA-II* are real values and both the SBX and the PM operators are able to handle numeric inputs, no encoding/decoding was needed for any of these operators.

After repeating the evaluation, selection, crossover and mutation for k generations, the last population $\overline{\mathbf{X}}(k)$ is likely to have at least one chromosome $\mathbf{X}(k)$ that meets all the $\mathbf{G}(\mathbf{F}(\mathbf{X}))$ constraints and provides, in this case, the lowest value as possible to the target minimisation variables in $\mathbf{Y}(k)$. It's important to notice that for the developed implementation the only termination criteria for the GA was the number of generations, k.

IV. RESULTS

The EAF and ladle regression models were validated using the test portion of the data from the train/test split. Figure 6 presents the mean squared errors obtained by the algorithms presented in Section III-B1 for both EAF and ladle models using the unseen test set.

The results obtained for the EAF regression modelling can be compared to the results from [16] since both implementations seem to refer to the same research object and use a similar dataset. Figure 8 shows a comparison between the mean absolute errors (MAEs) obtained for the prediction of the energy variable based on the EAF model of both implementations, each one using multiple ML algorithms. Here, this research trained models using the LR, SVR and DT algorithms in order to match the results obtained from [16]. The most successful model from [16] was a model using deep learning (DL) which achieved an MAE of 1.5 mWh. Even though the DL model was not replicated here, a GB model (not present in [16]) achieved an MAE value ≈ 14% below the best regressor from this reference.

Figure 7 shows the regression line for one of the outputs of the EAF regression model, the energy consumption. It can be seen that there many values with significant errors, but the majority of the predictions lay close to the main regression diagonal.

The final result of the optimiser presented in Section III-B3 is depicted in Figure 9. The EAF and ladle predictors, together with the optimiser were encapsulated in a computer

Algorithm 2 *NSGA-II* Evaluation Function

```
$evaluation\_function(\mathbf{X}, ambient\_temperature, \mathbf{scrap\_prices})$:
    $\mathbf{scrap\_quantities} = \mathbf{X}[x_1, x_2, \ldots, x_\delta]$
    $\mathbf{eaf\_additives} = \mathbf{X}[x_{\delta+1}, x_{\delta+2}, \ldots, x_{\delta+\theta}]$
    $\mathbf{tap\_additives} = \mathbf{X}[x_{\delta+\theta+1}, x_{\delta+\theta+2}, \ldots, x_{\delta+\theta+\kappa}]$
    $total\_scrap\_cost = \mathbf{scrap\_quantities} \cdot \mathbf{scrap\_prices}$
    $total\_energy, \mathbf{eaf\_composition} = eaf\_regressor(\mathbf{scrap\_quantities}, \mathbf{eaf\_additives}, ambient\_temperature)$
    $\mathbf{final\_composition} = ladle\_regressor(\mathbf{eaf\_composition}, \mathbf{tap\_additives}, ambient\_temperature)$
    $scrap\_weight = sum(\mathbf{scrap\_quantities})$
    $energy\_per\_weight = total\_energy \backslash scrap\_weight$
    $\mathbf{Y} = [energy\_per\_weight, total\_scrap\_cost]$
    return $\mathbf{Y}$
```

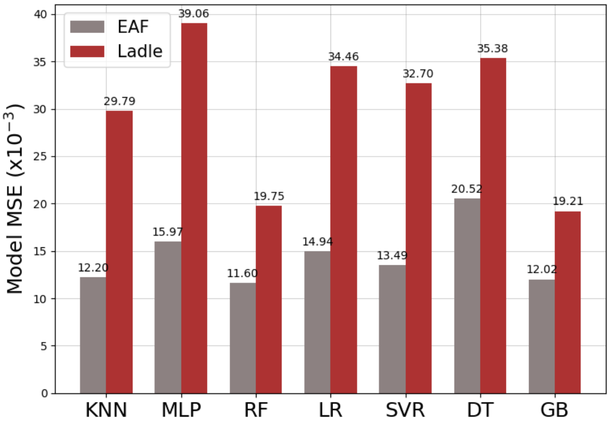


FIGURE 6. MSEs obtained using different ML algorithms.

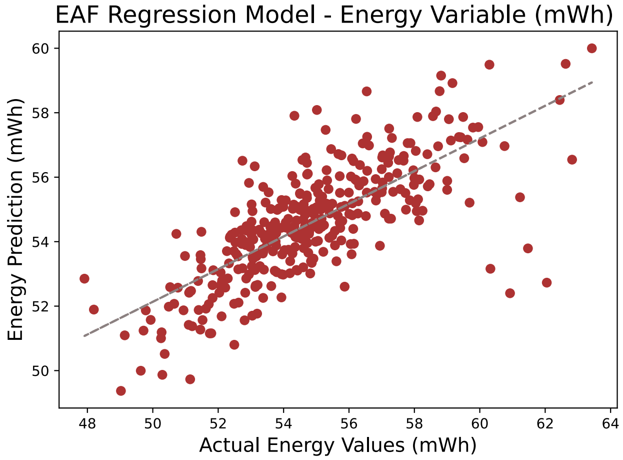


FIGURE 7. EAF regression line for energy (mWh).

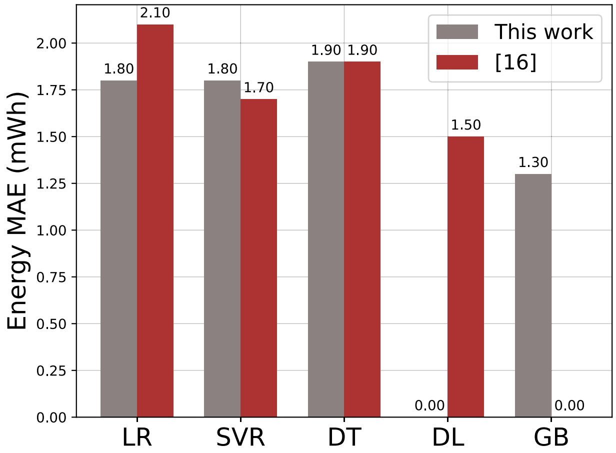


FIGURE 8. Energy MAE comparison with [16].

software which gives access to each of the previously mentioned models separately. The produced software allows the user to insert the input variables such as ambient temperature, target steel quality, the price per ton for each different scrap category in addition to upper and lower limits for each individual scrap category as well. The predictors, as exemplified in Figure 10, allow the user to interact with the models designed to replicate the behaviours of both EAF and ladle furnaces.

Figure 11 shows the output of all function evaluations carried out during the optimisation process, together with the feasible solutions found along the way and optimal solutions from the last generation $\overline{\mathbf{Y}}(k)$. This Figure shows each minimisation variable from $\mathbf{Y}$ in a different axis, where the energy per ton of scrap is plotted in the x-axis and the total scrap price is plotted in the y-axis. For simplicity, all results presented here refer to the production of steel grade A.

This plot shows a clear trade-off between both minimisation variables. Here, even though all points from Figure 11 belong to the co-domain $\mathbf{D}_y$, only the ones in the P Pareto optimal set are of most importance.

Figure 12 shows the hypervolume performance indicator [42], [43] obtained in the minimisation of a Grade A steel. It measures the size of the dominated space, bound from above by a reference point, also called Nadir Point.

As multi-objective evolutionary algorithms using the hypervolume indicator transform multi-objective problems into single objective ones by searching for a finite set of solutions maximising the corresponding hypervolume indicator, this plot is indispensable for future comparison of similar optimisers.

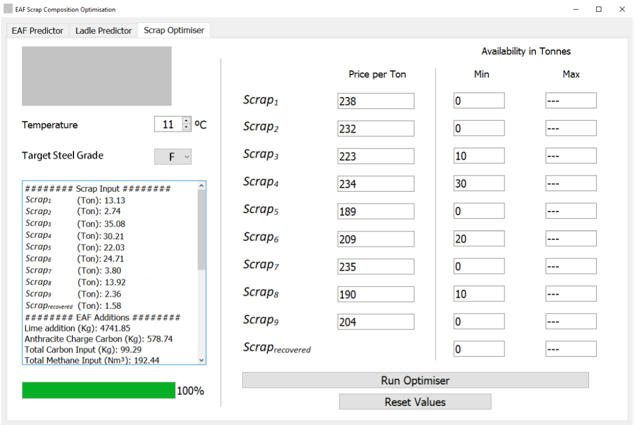
Figure 13 shows also some output values $\mathbf{Y}$ from the last population $\overline{\mathbf{Y}}(k)$ of the *NSGA-II* optimiser, however, only the P values that belong to the restricted co-domain $\mathbf{G}(\mathbf{F}(\mathbf{X})) \mid \mathbf{G}(\mathbf{F}(\mathbf{X})) \subset \mathbf{D}_y$ are being shown this time. The constraints responsible for limiting $\mathbf{D}_y$ to $\mathbf{G}(\mathbf{F}(\mathbf{X}))$ are the total scrap weight and, most importantly, the final composition of the target steel which determines ranges to be met for each individual chemical element according to the steel grade to be produced.

Figure 13 also shows the crowding distance value for each solution $\mathbf{Y}$ from P. These values are represented by the colour map and, as it can be seen, the extreme points get assigned a crowding distance significantly higher than the others.

In order to test and validate the developed optimiser, a final experiment was carried out in order to identify, measure and compare the values achieved for the minimisation variables by the *NSGA-II*. In this experiment, different grades of steel (A-H) were optimised and the final scrap cost and energy consumption variables were compared to the average values for each equivalent grade in the dataset.

As the prices for each individual category of scrap varies on a weekly basis and such values are not recorded in the dataset used for this implementation, a fixed average price was defined for the purpose of comparing the optimisation values obtained for scrap usage with the recorded values from the dataset. The used prices are found in Table 2.

All of these prices are in $\$/1000 \times kg$ and the $Scrap_{recovered}$ which costs $\$0$ per ton refers to the scrap recovered from unavoidable spillages during the basket loading or scrap



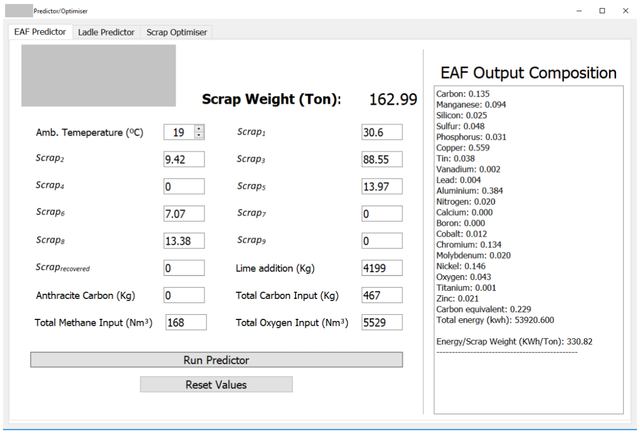
Temperature: 11 ºC

Target Steel Grade: F

```
######## Scrap Input ########
Scrap1          (Ton): 13.13
Scrap2          (Ton): 2.74
Scrap3          (Ton): 35.08
Scrap4          (Ton): 30.21
Scrap5          (Ton): 22.03
Scrap6          (Ton): 24.71
Scrap7          (Ton): 3.80
Scrap8          (Ton): 13.92
Scrap9          (Ton): 2.36
Scraprecovered  (Ton): 1.58
######## EAF Additions ########
Lime addition (Kg): 4741.85
Anthracite Charge Carbon (Kg): 578.74
Total Carbon Input (Kg): 99.29
Total Methane Input (Nm³): 192.44
```

		Availability in Tonnes	
	Price per Ton	Min	Max
$Scrap_1$	238	0	---
$Scrap_2$	232	0	---
$Scrap_3$	223	10	---
$Scrap_4$	234	30	---
$Scrap_5$	189	0	---
$Scrap_6$	209	20	---
$Scrap_7$	235	0	---
$Scrap_8$	190	10	---
$Scrap_9$	204	0	---
$Scrap_{recovered}$		0	---

FIGURE 9. Developed software displaying the optimiser tab.



Scrap Weight (Ton): 162.99

Amb. Temeperature (ºC)	19	$Scrap_1$	30.6
$Scrap_2$	9.42	$Scrap_3$	88.55
$Scrap_4$	0	$Scrap_5$	13.97
$Scrap_6$	7.07	$Scrap_7$	0
$Scrap_8$	13.38	$Scrap_9$	0
$Scrap_{recovered}$	0	Lime addition (Kg)	4199
Anthracite Carbon (Kg)	0	Total Carbon Input (Kg)	467
Total Methane Input (Nm³)	168	Total Oxygen Input (Nm³)	5529

EAF Output Composition

```
Carbon: 0.135
Manganese: 0.094
Silicon: 0.025
Sulfur: 0.048
Phosphorus: 0.031
Copper: 0.559
Tin: 0.038
Vanadium: 0.002
Lead: 0.004
Aluminium: 0.384
Nitrogen: 0.020
Calcium: 0.000
Boron: 0.000
Cobalt: 0.012
Chromium: 0.134
Molybdenum: 0.020
Nickel: 0.146
Oxygen: 0.043
Titanium: 0.001
Zinc: 0.021
Carbon equivalent: 0.229
Total energy (kwh): 53920.600

Energy/Scrap Weight (KWh/Ton): 330.82
-----------------------------------------------
```

FIGURE 10. Developed software displaying the EAF predictor tab.

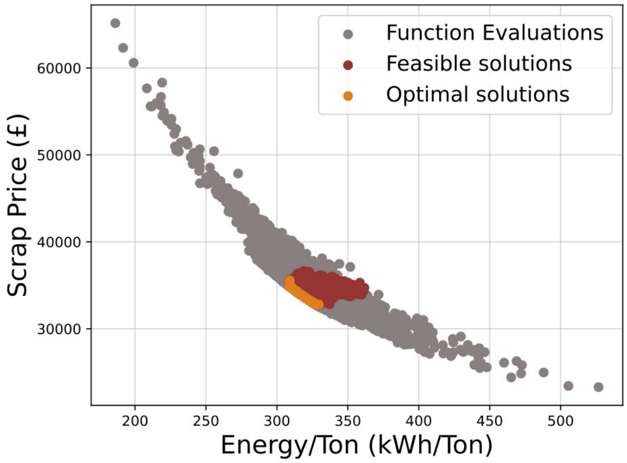


FIGURE 11. Pareto optimal set - grade A.

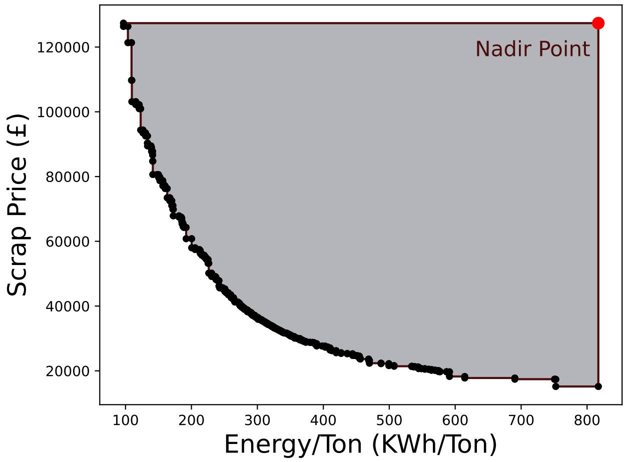


FIGURE 12. Hypervolume with both objectives being minimised.

transportation in the scrapyard, for example, and it is composed of a mix from all previous scrap categories.

Figure 14 shows a comparison between the average total scrap cost required to produce each steel grade (identified as A-H) taken from the dataset provided. The scrap recipe optimiser resulted in a price reduction for all the analysed steel grades. The scrap cost reductions for each individual steel grade ranged from 1.15% up to 5.2%. It is valid to mention that these figures are compatible with values obtained from state-of-the-art researches as seen in [10].

Equally notable results were achieved in the energy consumption minimisation. Figure 15 shows a comparison between the average energy consumption per ton of scrap (kWh/Ton) of each steel grade taken from the dataset and the respective optimised values obtained by the implementation here developed. The optimiser generated solutions with energy consumption reductions that ranged from 1.87% up to 8.20% among the different steel grades.

V. DISCUSSION

As can be seen from Figure 8, the EAF model developed using the gradient boosting algorithm managed to achieve a mean absolute error of 1.30 *mWh*. This is 13% lower than the best EAF model developed by [16] which uses a more complex algorithm. Even though a significantly larger dataset was used in [16], it is believed that this difference is mainly due to two reasons: the filtering by removing outlier values and the model hyperparameters tuning.

The raw version of this industrial dataset had many flaws such as missing, repeated, noisy and wrongly recorded data. In addition to that, many outlier values were detected, not to mention that the final version of the raw dataset was composed by a concatenation of a few separate datasets extracted from different systems within the partner industry. In this scenario, the removal of the outliers (Figure 4) proved to have a significant effect on the improvement of model performance.

It is important to mention that even after a thorough process of cleaning and pre-processing the dataset used in this data-driven optimisation, the machine learning models are still susceptible to inaccuracies and bias due to potential undetected issues with the dataset. This can be seen as the most significant limitation of the approach here presented as it could lead to skewed models. Considering that, the

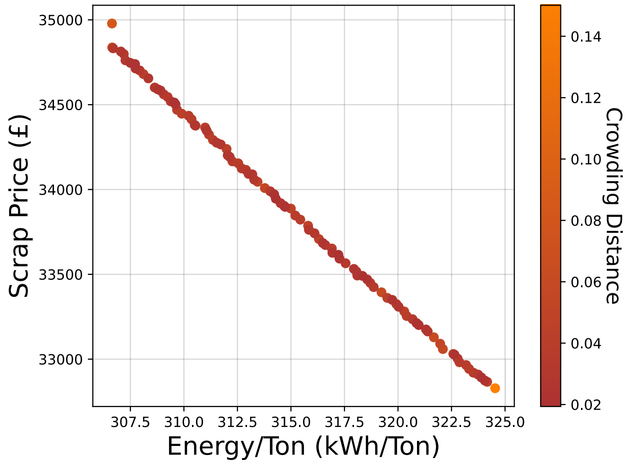


FIGURE 13. Pareto optimal set with restrictions - grade A.

TABLE 2. Prices used for each scrap category.

Scrap Category	Price Per Ton
$Scrap_1$	£238
$Scrap_2$	£232
$Scrap_3$	£223
$Scrap_4$	£234
$Scrap_5$	£189
$Scrap_6$	£209
$Scrap_7$	£235
$Scrap_8$	£190
$Scrap_9$	£204
$Scrap_{recovered}$	£0

development of the furnace models assumed that the data quality improvements made during the pre-processing were enough to mitigate any major data issue.

The hyperparameters tuning produced also a great improvement in both regression models performances. In the preliminary test, a comparison between the MSE achieved by a fitted model and the same model after hyperparameters tuning showed a reduction of up to 5%.

As can be seen in Figure 6, the MSEs achieved, for both EAF and ladle, are all in the order of magnitude of 10^{-3}. These values represent a great fit to the modelled furnaces with no overfitting as the models were tuned using a randomised search and a 3-fold cross-validation splitting strategy with test data only. In general, the MSEs for the ladle regressor are greater than the MSEs for the EAF regressor and this can be explained by the fact that the second model inherits the error from the first one when it uses the EAF model's output composition as part of its input. When the error from the EAF regressor is added to the error of the ladle regressor, the MSE of this second model naturally becomes higher.

For generating the final version of the optimisation software, the models trained by the GB algorithm were the chosen ones since they achieved significantly lower MSEs (alongside RF) compared to the others algorithms.

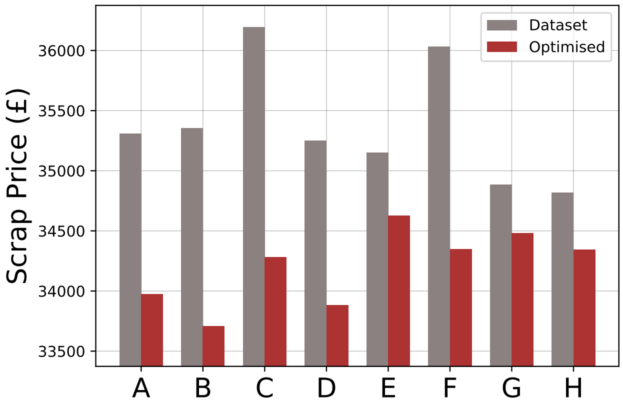


FIGURE 14. Scrap price optimisation results.

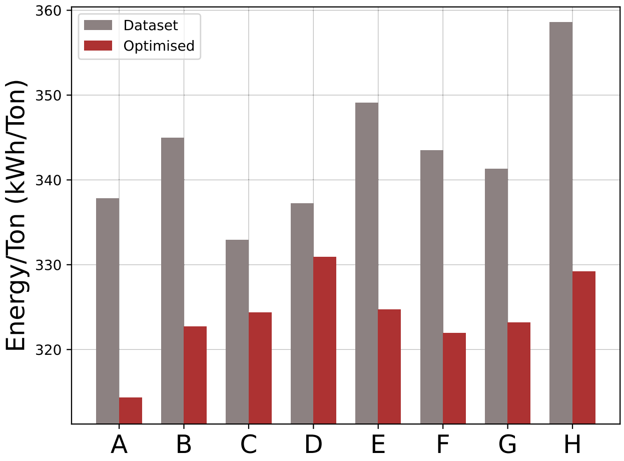


FIGURE 15. Energy consumption optimisation results.

In order to evaluate the goodness of the Pareto front achieved by the *NSGA-II* optimiser, other five multi-objective optimisation algorithms were tested and compared using the hypervolume metric. The MOO algorithms were *NSGA-II*; non-dominated sorting genetic algorithm III (*NSGA-III* - [44]); a developed version of generalised differential evolution (*GDE3* - [45]); multi-objective evolutionary algorithm based on decomposition (*MOEA/D* - [46]); strength Pareto evolutionary algorithm 2 (*SPEA2* - [47]); and a fast multi-objective evolutionary algorithm (ε-*MOEA* - [48]).

The parameters used for each MOO algorithm were empirically chosen. This choice was based on the parameters available in the optimisation library used [36] and focused on providing equivalent and compatible capabilities for each algorithm given specific limitations. A detailed list of parameters is presented below.

- NSGAII
  - Population Size = 100
  - Tournament Size = 2
  - Max Function Evaluations = 10000
  - Crossover = SBX (Probability = 1.0, Distribution Index = 15.0)

  - Mutation = Polynomial Mutation (Probability = 1, Distribution Index = 20.0)
- NSGAIII
  - Population Size = 100
  - Tournament Size = 2
  - Divisions Outer = 12
  - Max Function Evaluations = 10000
  - Crossover = SBX (Probability = 1.0, Distribution Index = 15.0)
  - Mutation = Polynomial Mutation (Probability = 1, Distribution Index = 20.0)
- GDE3
  - Population Size = 100
  - Crossover Rate = 0.1
  - Step Size = 0.5
  - Max Function Evaluations = 10000
- MOEA/D
  - Neighbourhood Size = 10
  - $\delta = 0.8$
  - $\eta = 0.1$
  - Max Function Evaluations = 10000
  - Crossover = SBX (Probability = 1.0, Distribution Index = 15.0)
  - Mutation = Polynomial Mutation (Probability = 1, Distribution Index = 20.0)
- SPEA2
  - Population Size = 100
  - Tournament Size = 2
  - k = 1
  - Max Function Evaluations = 10000
  - Crossover = SBX (Probability = 1.0, Distribution Index = 15.0)
  - Mutation = Polynomial Mutation (Probability = 1, Distribution Index = 20.0)
- *ε-MOEA*.
  - Population Size = 100
  - Tournament Size = 2
  - Max Function Evaluations = 10000
  - Crossover = SBX (Probability = 1.0, Distribution Index = 15.0)
  - Mutation = Polynomial Mutation (Probability = 1, Distribution Index = 20.0)

The hypervolume indicator, also known in the literature as Lebesgue measure or S-metric, was the chosen metric since it is one of the most popular indicators in MOO problems. Opposite to other indicators, for calculating the hypervolume a target Pareto set does not need to be known and only a reference point needs to be provided. Here, the hypervolume implementation used is from [49] and implemented in [50]. It calculates the area/volume, which is dominated by the provided set of solutions with respect to the reference point, also called Nadir point. Here, the reference point is set at $energy\_per\_weight = 371$ and $total\_scrap\_cost = 37338$.

Figure 16 shows the achieved Pareto front for each one of the six tested MOO algorithms. It can be seen that the *NSGA-II* achieved the lowest values for both minimisation variables, followed by *GDE*3. The correspondent hypervolume of each MOO is presented in Table 3.

TABLE 3. MOO algorithms and its hypervolume values.

Algorithm	Hypervolume
$NSGA\text{-}II$	28.29×10^4
$NSGA\text{-}III$	20.37×10^4
$GDE3$	28.04×10^4
$MOEA/D$	10.55×10^4
$SPEA2$	27.29×10^4
$\varepsilon\text{-}MOEA$	26.73×10^4

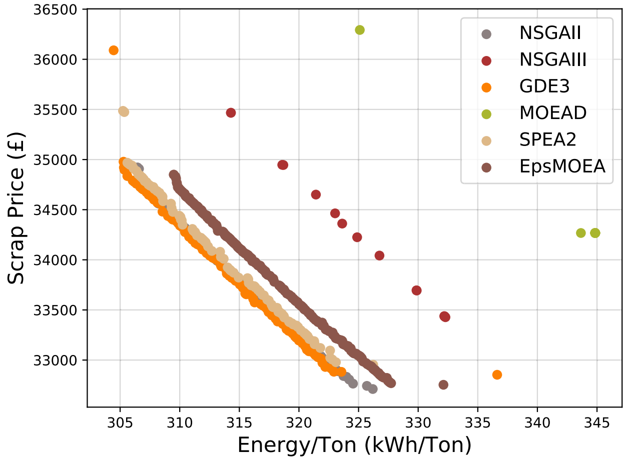


FIGURE 16. Comparison between MOO algorithms.

Figure 17 illustrates the convergence of each MOO algorithm throughout their respective generations. It is possible to see that both minimisation variables are shown for each algorithm. The Scrap Price (£) target variable uses the left side of the Y-axis while the Energy/Ton (kWh/Ton) uses the opposite side. As a result of both variables being minimised simultaneously, opposite trends can be observed between both optimisation variables throughout the optimisation process i.e. when one fitness value decreases, the other variable increases in proportional magnitude.

While *NSGA-II* and *GDE3* performed notoriously better, *NSGA-III* and *MOEAD* showed poor optimisation performance. From Figure 17, it can be seen that both variables in the *MOEAD* optimisation did not show any significant improvement during the generations. On the other hand, *NSGA-III* demonstrated an unstable pattern and did not manage to converge to a minimum value comparable to the other best performing algorithms.

NSGA-II and *GDE3* showed the best overall performances, as per hypervolume values from Table 3, and these can be observed in the convergence process as well. Both algorithms showed a smooth, steady and relatively constant performance during the minimisation process which allowed them to achieve the lowest values for Scrap Price (£)

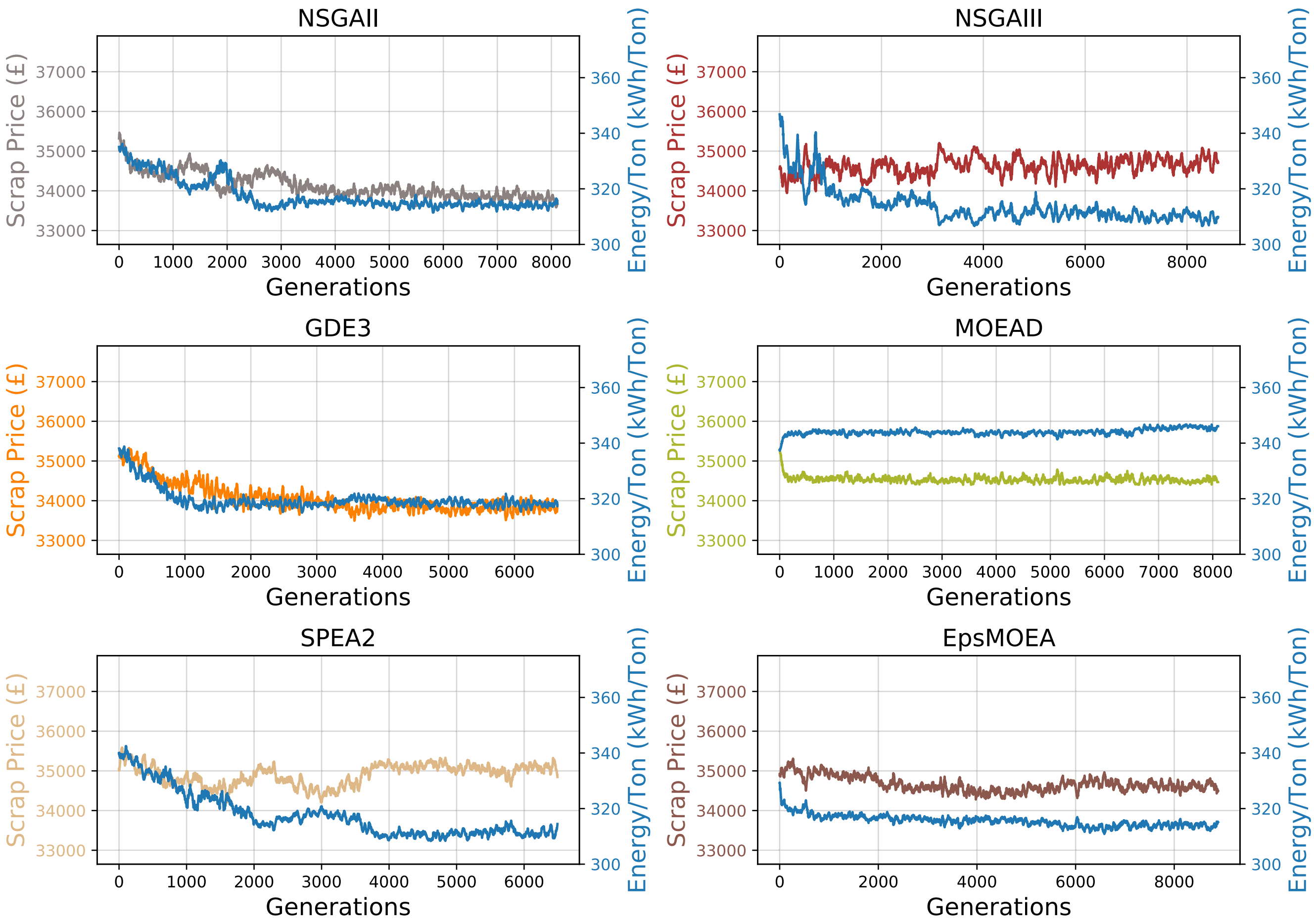


FIGURE 17. Convergence comparison between MOO algorithms.

and Energy/Ton (kWh/Ton), while covering a wide area in the minimisation space which results in high hypervolume values.

Apart from *MOEA/D*, all other algorithms achieved hypervolume values relatively similar, but *NSGA-II* was chosen as it was able to cover a wider area in solutions set.

The comparative optimisation results displayed in Figures 14 and 15 not only showed to be compatible with actual EAF operation values, but also presented a significant reduction both in Energy/Ton cost and total scrap cost.

VI. CONCLUSION

The obtained mean squared errors presented in Figure 6 highlight the accuracy of the developed models and enable real tests in the analysed melt-shop. Even though the EAF regressor presented lower mean squared error values compared to the Ladle regressor, the former is believed to have a more volatile behaviour during practical use since the general scrap chemical composition is known to vary according to different factors such as supplier, season, geographical region, etc. On the other hand, the behaviour of the ladle furnace is known to be more consistent and predictable since both the input and output chemical composition are controlled and monitored.

As presented in Figure 8, a direct comparison to an equivalent model from the literature showed that the model presented in this paper achieved a MAE of 200 *kWh* lower than the deep learning model from [16]. Even though this represents a reduction of only 13% it has significant importance to the accuracy of the EAF model, and consequently, to the *NSGA-II* optimiser.

The obtained results presented in Section IV suggest significant cost savings for both energy consumption and scrap usage. Even by using the most conservative estimates and considering the lowest minimisation values obtained, 1.15% of reduction for scrap cost and 1.87% of reduction for energy consumption, considerable potential savings are indicated by the optimiser described in Section III-B. Taking into consideration that approximately 16 to 20 heats are produced per day and that the minimisation values previously presented refer to every single batch, savings in the hundreds of thousands of pounds per month are expected.

The comparative analysis between six MOOs involving solutions set coverage and hypervolume values indicated *NSGA-II* as the best algorithm for this application of multi-objective optimisation. This algorithm achieved a hypervolume value of 28.29 × 10^4, slightly higher than the second best MOO algorithm, *GDE3*.

The resulting Pareto optimal set from the *NSGA-II* optimiser (Figure 13) provides not only one, but a set of scrap recipes and additives quantities that optimise the minimisation variables. This feature of the proposed implementation allows the melt-shop operator to choose between prioritising any of the target variables from **Y** according to its empirical knowledge.

A. FUTURE WORK

In order to make the software even more precise, the team intend to add the cost of each individual additive to the calculation of the total scrap cost. Even though the majority of the cost related to the EAF input feedstock comes from the scrap cost, adding additional minor costs would potentially produce a more accurate optimisation.

REFERENCES

[1] World Steel Association. (2018). *Steel Industry Key Facts*. [Online]. Available: https://www.worldsteel.org/en/dam/jcr: ab8be93e-1d2f-4215-9143-4eba6808bf03/20190207_steelFacts.pdf

[2] G. Bisio, G. Rubatto, and R. Martini, ''Heat transfer, energy saving and pollution control in UHP electric-arc furnaces,'' *Energy*, vol. 25, no. 11, pp. 1047–1066, Nov. 2000.

[3] D. Ameling, F. Strunck, H.-G. Pottken, and H. Strohschein, ''Energy recovery from UHP electric arc furnaces using hot cooling,'' *Energy*, vol. 25, pp. 1047–1066, 2000, doi: 10.1016/s0360-5442(00)00037-2.

[4] A. A. Nikolaev, G. P. Kornilov, A. V. Anufriev, S. V. Pekhterev, and E. V. Povelitsa, ''Electrical optimization of superpowerful arc furnaces,'' *Steel Transl.*, vol. 44, no. 4, pp. 289–297, Apr. 2014, doi: 10.3103/s0967091214040135.

[5] R. D. M. MacRosty and C. L. E. Swartz, ''Dynamic modeling of an industrial electric arc furnace,'' *Ind. Eng. Chem. Res.*, vol. 44, no. 21, pp. 8067–8083, Oct. 2005, doi: 10.1021/ie050101b.

[6] E. G. M. Brusa and S. Morsut, ''Design and structural optimization of the electric arc furnace through a mechatronic-integrated modeling activity,'' *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 3, pp. 1099–1107, Jun. 2015, doi: 10.1109/TMECH.2014.2364392.

[7] C. Yigit, G. Coskun, E. Buyukkaya, U. Durmaz, and H. R. Güven, ''CFD modeling of carbon combustion and electrode radiation in an electric arc furnace,'' *Appl. Thermal Eng.*, vol. 90, pp. 831–837, Nov. 2015, doi: 10.1016/j.applthermaleng.2015.07.066.

[8] E. Sandberg, ''Energy and scrap optimisation of electric arc furnaces by statistical analysis of process data,'' Ph.D. dissertation, Luleå Univ. Technol., Luleå, Sweden, 2005. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.399.4857&rep=rep1&type=pdf

[9] E. Sandberg, B. Lennox, and P. Undvall, ''Scrap management by statistical evaluation of EAF process data,'' *Control Eng. Pract.*, vol. 15, no. 9, pp. 1063–1075, Sep. 2007, doi: 10.1016/j.conengprac.2007.01.001.

[10] W. Wang, ''Cost optimization of scrap when making steel with an electric arc furnace,'' Ph.D. dissertation, Dept. Mech. Eng., McGill Univ., Montreal, QC, Canada, Feb. 2012. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:990816/FULLTEXT01.pdfhttps://escholarship.mcgill.ca/concern/theses/db78tg80w

[11] E.-W. Bai, ''Minimizing energy cost in electric arc furnace steel making by optimal control designs,'' *J. Energy*, vol. 2014, pp. 1–9, Jan. 2014, doi: 10.1155/2014/620695.

[12] M. Kovačič, K. Stopar, R. Vertnik, and B. Šarler, ''Comprehensive electric arc furnace electric energy consumption modeling: A pilot study,'' *Energies*, vol. 12, no. 11, p. 2142, Jun. 2019, doi: 10.3390/en12112142.

[13] D. Gajic, I. Savic-Gajic, I. Savic, O. Georgieva, and S. Di Gennaro, ''Modelling of electrical energy consumption in an electric arc furnace using artificial neural networks,'' *Energy*, vol. 108, pp. 132–139, Aug. 2016, doi: 10.1016/j.energy.2015.07.068.

[14] N. Dučić, A. Jovičić, S. Manasijević, R. Radiša, V. Z. Ćojbašić, and B. Savković, ''Application of machine learning in the control of metal melting production process,'' *Appl. Sci.*, vol. 10, no. 17, p. 6048, Sep. 2020, doi: 10.3390/app10176048.

[15] A. Rahnama, Z. Li, and S. Sridhar, ''Machine learning-based prediction of a BOS reactor performance from operating parameters,'' *Processes*, vol. 8, no. 3, p. 371, Mar. 2020, doi: 10.3390/pr8030371.

[16] C. Chen, Y. Liu, M. Kumar, and J. Qin, ''Energy consumption modelling using deep learning technique—A case study of EAF,'' *Proc. CIRP*, vol. 72, pp. 1063–1068, Jan. 2018, doi: 10.1016/j.procir.2018.03.095.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[18] T. D. Bui, D. K. Nguyen, and T. D. Ngo, ''Supervising an unsupervised neural network,'' in *Proc. 1st Asian Conf. Intell. Inf. Database Syst.*, Apr. 2009, pp. 307–312.

[19] S. Ma, C. Yang, W. Liu, Z. Zhou, and J. Song, ''Comparison of univariate and multivariate predicted method based on support vector regression for silicon content in hot metal,'' in *Proc. 27th Chin. Control Decis. Conf. (CCDC)*, May 2015, pp. 4541–4545.

[20] M. T. M. Emmerich and A. H. Deutz, ''A tutorial on multiobjective optimization: Fundamentals and evolutionary methods,'' *Natural Comput.*, vol. 17, no. 3, pp. 585–609, 2018, doi: 10.1007/s11047-018-9685-y.

[21] N. Gunantara, ''A review of multi-objective optimization: Methods and its applications,'' *Cogent Eng.*, vol. 5, no. 1, Jan. 2018, Art. no. 1502242. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/23311916.2018.1502242

[22] K.-H. Chang, ''Multiobjective optimization and advanced topics,'' in *e-Design*, K.-H. Chang, Ed. Boston, MA, USA: Academic, 2015, ch. 19, pp. 1105–1173.

[23] K. Deb and D. Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.

[24] Z. Jia and L. Gong, ''Multi-criteria human resource allocation for optimization problems using multi-objective particle swarm optimization algorithm,'' in *Proc. Int. Conf. Comput. Sci. Softw. Eng.*, vol. 1, Dec. 2008, pp. 1187–1190.

[25] H. Afshari, W. Hare, and S. Tesfamariam, ''Constrained multi-objective optimization algorithms: Review and comparison with application in reinforced concrete structures,'' *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105631. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494619304119

[26] C. Ye and B. Shen, ''An improved constrained multi-objective optimization evolutionary algorithm for carbon fibre drawing process,'' *Syst. Sci. Control Eng.*, vol. 7, no. 1, pp. 133–145, Jan. 2019, doi: 10.1080/21642583.2019.1584774.

[27] W. Jiang-Ping and T. Qun, ''Urban planning decision using multi-objective optimization algorithm,'' in *Proc. Int. Colloq. Comput., Commun., Control, Manage. (ISECS)*, vol. 4, Aug. 2009, pp. 392–394.

[28] T. Stewart, O. Bandte, H. Braun, N. Chakraborti, M. Ehrgott, M. Göbelt, Y. Jin, H. Nakayama, S. Poles, and D. D. Stefano, ''Real-world applications of multiobjective optimization,'' in *Multiobjective Optimization*. Berlin, Germany: Springer, 2008, pp. 285–327, doi: 10.1007/978-3-540-88908-3_11.

[29] K. Amine, ''Multiobjective simulated annealing: Principles and algorithm variants,'' *Adv. Oper. Res.*, vol. 2019, pp. 1–13, May 2019, doi: 10.1155/2019/8134674.

[30] W. Huang, T. Xu, K. Li, and J. He, ''Multiobjective differential evolution enhanced with principle component analysis for constrained optimization,'' *Swarm Evol. Comput.*, vol. 50, Nov. 2019, Art. no. 100571, doi: 10.1016/j.swevo.2019.100571.

[31] C. M. Fonseca and P. J. Fleming, ''Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation,'' *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 28, no. 1, pp. 26–37, Jan. 1998, doi: 10.1109/3468.650319.

[32] N. Srinivas and K. Deb, ''Muiltiobjective optimization using nondominated sorting in genetic algorithms,'' *J. Evol. Comput.*, vol. 2, no. 3, pp. 221–248, Sep. 1994, doi: 10.1162/evco.1994.2.3.221.

[33] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, ''A fast and elitist multiobjective genetic algorithm: NSGA-II,'' *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[34] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Ann Arbor, MI, USA: Univ. of Michigan Press, 1975.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, ''Scikit-learn: Machine learning in Python,'' *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[36] *Multiobjective Optimization in Python*. Accessed: Nov. 4, 2021. [Online]. Available: https://platypus.readthedocs.io/en/latest/index.html

[37] K. Deb and A. Deb, ''Analysing mutation schemes for real-parameter genetic algorithms,'' *Int. J. Artif. Intell. Soft Comput.*, vol. 4, no. 1, p. 1, 2014, doi: 10.1504/ijaisc.2014.059280.

[38] M. F. Torquato and M. A. C. Fernandes, ''High-performance parallel implementation of genetic algorithm on fpga,'' *Circuits, Syst., Signal Process.*, vol. 38, no. 9, pp. 4014–4039, Sep. 2019, doi: 10.1007/s00034-019-01037-w.

[39] K. Deb, K. Sindhya, and T. Okabe, ''Self-adaptive simulated binary crossover for real-parameter optimization,'' in *Proc. 9th Annu. Conf. Genet. Evol. Comput. (GECCO)*, 2007, pp. 1187–1194, doi: 10.1145/1276958.1277190.

[40] K. Deb and R. B. Agrawal, ''Simulated binary crossover for continuous search space,'' *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.

[41] K. Deb and S. Agrawal, ''A niched-penalty approach for constraint handling in genetic algorithms,'' in *Artificial Neural Nets and Genetic Algorithms*. Vienna, Austria: Springer, 1999, pp. 235–243.

[42] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, ''Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications,'' *Theor. Comput. Sci.*, vol. 425, pp. 75–103, Mar. 2012, doi: 10.1016/j.tcs.2011.03.012.

[43] E. Zitzler and L. Thiele, ''Multiobjective optimization using evolutionary algorithms—A comparative case study,'' in *Proc. Int. Conf. Parallel Problem Solving Nature*, in Lecture Notes in Computer Science. Berlin, Germany: Springer, 1998, pp. 292–301, doi: 10.1007/bfb0056872.

[44] K. Deb and H. Jain, ''An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach. Part I: Solving problems with box constraints,'' *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014, doi: 10.1109/tevc.2013.2281535.

[45] S. Kukkonen and J. Lampinen, ''GDE3: The third evolution step of generalized differential evolution,'' in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2005, pp. 443–450, doi: 10.1109/cec.2005.1554717.

[46] A. Zhou, Q. Zhang, and G. Zhang, ''A multiobjective evolutionary algorithm based on decomposition and probability model,'' in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–8, doi: 10.1109/cec.2012.6252954.

[47] E. Zitzler, M. Laumanns, and L. Thiele, *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. ETH Zurich, 2001, doi: 10.3929/ETHZ-A-004284029.

[48] K. Deb, M. Mohan, and B. Mishra, ''A fast multiobjective evolutionary algorithm for finding wellspread Pareto-optimal solutions,'' Lndian Inst. Technol. Kanpur, KanGAL Rep. 2003002, Jan. 2003.

[49] C. M. Fonseca, L. Paquete, and M. Lopez-Ibanez, ''An improved dimension-sweep algorithm for the hypervolume indicator,'' in *Proc. IEEE Int. Conf. Evol. Comput.*, Jul. 2006, pp. 1157–1163, doi: 10.1109/cec.2006.1688440.

[50] J. Blank and K. Deb, ''Pymoo: Multi-objective optimization in Python,'' *IEEE Access*, vol. 8, pp. 89497–89509, 2020.

MATHEUS F. TORQUATO received the B.Sc. degree in science and technology, the B.Eng. degree in computer engineering, and the M.Sc. degree in computer engineering from the Federal University of Rio Grande do Norte, in 2013, 2015, and 2017, respectively. He is currently the Project Officer at Advanced Sustainable Manufacturing Technologies 2020. He is also an external member of the Research Group on Embedded Systems and Reconfigurable Hardware, where his main research topic is the acceleration of artificial intelligence (AI) algorithms through reconfigurable computing (RC) on FPGA. His research interests include AI, machine learning, RC, embedded systems, reconfigurable hardware, human–computer interaction, and tactile internet.

GERMÁN MARTÍNEZ-AYUSO received the B.Sc. degree in civil engineering from the University of Córdoba, Spain, the M.Sc. degree in structures from the University of Granada, Spain, and the Ph.D. degree in civil engineering from Swansea University, U.K. He currently works at Swansea University as a Research Officer in machine learning applied to image recognition and image processing for cardiovascular diseases. His research interests include many different topics, from machine learning application and development to material homogenization or structures simulation using finite element techniques.

ASHRAF A. FAHMY received the B.Eng. degree (Hons.) in electrical engineering and the M.Sc. degree with specialization in flux vector control of electric machines from Helwan University, Cairo, Egypt, in 1992 and 1999, respectively, and the Ph.D. degree with specialization in neuro-fuzzy control of robotic manipulators from Cardiff University, U.K., in 2005. He is currently the Senior Technical Manager at ASTUTE, Engineering College of Swansea University, an Associate Professor at Helwan University, and the HV Manager at Shaker Consultancy Group. He is also an electrical power and machines drives control engineer by education and worldwide experience, a robotics control engineer by research, and an industrial manufacturing consultant by U.K., and has worldwide experience. He has expertise in soft computing decision making, manufacturing systems, robotic manipulation, instrumentation, control systems, and electrical power generation.

JOHANN SIENZ received the B.Eng. degree (Hons.) from the University of Central Lancashire, U.K., in 1989, the Diplom-Ingenieur (FH) degree from the Augsburg University of Applied Sciences, Germany, in 1989, and the M.Sc. and Ph.D. degrees from Swansea University, U.K., in 1990 and 1994, respectively. He currently holds a position of the Personal Chair with the Faculty of Science and Engineering, Swansea University, where he is also the Deputy PVC and the Deputy Executive Dean. He is a fellow of the Institution of Mechanical Engineers (FIMechE) and the Royal Aeronautical Society (FRAeS). He is a Chartered Engineer (C.Eng.), a Chartered Mathematician (C.Math.), and a member of the Institute of Mathematics and its Applications (MIMA). He is the Editor-in-Chief of the international journal *Applied Mathematical Modelling*.