Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Interactive data-driven multiobjective optimization of metallurgical properties of microalloyed steels using the DESDEO framework

Bhupinder Singh Saini [a,*], Debalay Chakrabarti [b], Nirupam Chakraborti [c,b], Babooshka Shavazipour [a], Kaisa Miettinen [a]

[a] *University of Jyvaskyla, Faculty of Information Technology, University of Jyvaskyla, P.O. Box 35 (Agora), FI-40014, Finland*
[b] *Department of Metallurgical and Materials Engineering, Indian Institute of Technology Kharagpur, Kharagpur, West Bengal, 721302, India*
[c] *Faculty of Mechanical Engineering, Czech Technical University, Prague, Czech Republic*

ARTICLE INFO

Keywords:
Data-driven evolutionary computation
Multiple criteria optimization
Surrogate-assisted optimization
Multiple decision makers
Interactive optimization
Open-source software

ABSTRACT

Solving real-life data-driven multiobjective optimization problems involves many complicated challenges. These challenges include preprocessing the data, modelling the objective functions, getting a meaningful formulation of the problem, and supporting decision makers to find preferred solutions in the existence of conflicting objective functions. In this paper, we tackle the problem of optimizing the composition of microalloyed steels to get good mechanical properties such as yield strength, percentage elongation, and Charpy energy. We formulate a problem with six objective functions based on data available and support two decision makers in finding a solution that satisfies them both. To enable two decision makers to make meaningful decisions for a problem with many objectives, we create the so-called MultiDM/IOPIS algorithm, which combines multiobjective evolutionary algorithms and scalarization functions from interactive multiobjective optimization methods in novel ways. We use the software framework called DESDEO, an open-source Python framework for interactively solving multiobjective optimization problems, to create the MultiDM/IOPIS algorithm. We provide a detailed account of all the challenges faced while formulating and solving the problem. We discuss and use many strategies to overcome those challenges. Overall, we propose a methodology to solve real-life data-driven problems with multiple objective functions and decision makers. With this methodology, we successfully obtained microalloyed steel compositions with mechanical properties that satisfied both decision makers.

1. Introduction

Digitalization sheds light on new data collection and information sharing methods, leading the world towards a data-centered era and opens up many opportunities for novel data-based methodology developments in data analytics and decision-making. However, various elements and challenges are involved in any decision-making process, starting from data.

Decision makers (DMs), in many real-life problems, often need to consider multiple objectives functions (or objectives, in brief), simultaneously when making decisions. In a decision-making process involving data, they first need to identify the objectives to be optimized with the help of the data available and the independent variables that control them. This process may require the data to be preprocessed. A multiobjective optimization problem (MOP) that considers the objectives important to the DMs can then be formulated. A DM is expected to be a domain expert. An analyst, who is an expert in multiobjective optimization,[1] typically coordinates the formulation of the MOP with the DMs and supports in solving it.

One of the ways to solve data-driven MOPs is to use surrogate-assisted optimization algorithms (Chugh et al., 2019; Jin et al., 2021). They use regression models, called surrogate models, to mimic the behaviour of the objectives as recorded in the data. In this, we assume that the data has been obtained from phenomena that can be treated as objectives to be optimized. The choice of surrogate modelling techniques to model the objectives and the methods of training and validating the models significantly impact the solutions found by the optimization method.

MOPs generally do not have a single optimal solution. Instead, due to potentially conflicting objectives, there exist many so-called Pareto optimal solutions that reflect the trade-offs between the various objectives (Miettinen, 1999). Many optimization algorithms aim to find

[*] Corresponding author.
E-mail address: bhupinder.s.saini@jyu.fi (B.S. Saini).

[1] Note that optimization is required in problems where solution alternatives are not known. Instead, we optimize objective functions which depend on the independent, so-called decision variables to find the solutions.

https://doi.org/10.1016/j.engappai.2023.105918
Received 4 July 2022; Received in revised form 15 November 2022; Accepted 24 January 2023
Available online xxxx

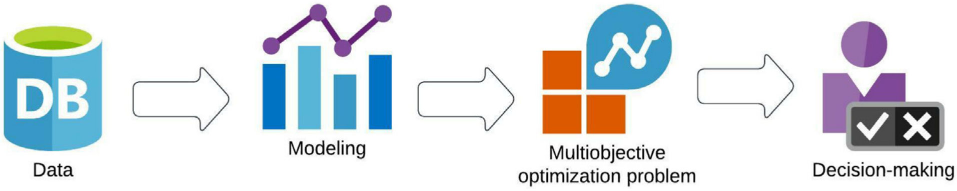


Fig. 1. A schematic of the seamless chain from data to decision-making.
Source: Adapted from http://jyu.fi/demo.

a representative approximation of the Pareto optimal solutions, see, e.g. Branke et al. (2008), Miettinen (1999). However, such algorithms can return upwards of a thousand solutions to the DMs. Comparing many solutions, especially in problems with many objectives, can be a cognitively challenging task. Thus, without assistance, DMs may find it difficult to make decisions in such problems.

One way to tackle this challenge is to use interactive methods for multiobjective optimization (Miettinen, 1999). Such methods incorporate the preferences of a DM during the optimization process to focus on Pareto optimal solutions that the DM prefers. Interactive methods search for new solutions iteratively, i.e., once the DM sees the solutions discovered using their preferences, they can learn about the trade-offs and interdependencies among the objective functions as well as about the feasibility of preferences. If satisfied, the DM can select a solution they like as the final solution. Alternatively, if they do not like the solutions, they can *interact* with the method by providing new preferences to the method, signifying interest in a different set of trade-offs. The interactive method then provides them with new solutions that reflect the new preferences. Focusing the search in a smaller area helps interactive methods find Pareto optimal solutions quicker and reduces the number of alternative solutions that a DM must consider at a time. This process enables the DM to learn about the possible trade-offs in the MOP in manageable and iterative steps, making the task of finding preferable solutions easier.

The process of problem formulation, optimization and decision-making is called a seamless chain (Heikkinen et al.). We show a simple schematic of the steps involved in data-based decision-making in Fig. 1. Similar schematics for data-driven MOPs solved using non-interactive methods have also been proposed (Jin et al., 2021). The figure presents a simple, linear pathway from data and modelling to multiobjective optimization and decision-making.

However, solving real-life MOPs can be a more complicated process. It must involve lengthy deliberations between the DM and the analyst to formulate a meaningful MOP. The data may introduce constraints limiting which objectives the DMs can consider in their MOP and how the analyst can model such objectives. The optimization and decision-making steps may reveal significant issues in the problem formulation of modelling phases. This would require going back and fixing the issues in the earlier steps and conducting optimization and decision-making again. The presence of more than one DM can also complicate the consideration, as most interactive methods are designed for a single DM. Although various data-driven optimization applications have been introduced in the literature, most of them are limited to explaining the results and miss reporting practical challenges and how they overcome those issues. Indeed a suitable framework or guidelines are lacking. To fill this gap, in this paper, we propose a more realistic representation of the seamless chain that will benefit both analysts and DMs by providing them with a structured guideline to tackle challenges associated with real-life data-driven MOPs.

The major contributions of this paper are as follows:

1. Microalloyed steel design problem: We formulate and solve a data-driven MOP with six objectives to obtain alloy compositions that optimize multiple mechanical properties of microalloyed steel. To achieve this, we follow the seamless chain structure to make the most efficient use of the data available. We discuss in detail the challenges we faced during the modelling and the solution process, and the steps we took to overcome them.
2. Novel interactive method: We developed the interactive MultiDM/IOPIS (multiple decision makers supported using IOPIS) algorithm, an extension of the IOPIS algorithm (Saini et al., 2020), to support multiple DMs to find a solution that meets their different preferences. The new method can be applied for group decision making. In this paper, we use the MultiDM/IOPIS algorithm to support two DMs simultaneously to solve the microalloyed steel design problem.
3. An updated seamless chain: We provide a detailed explanation of all the steps we took to solve the microalloyed steel design problem. We used many established techniques, and created new and novel techniques. We provide an updated schematic of the seamless chain to reflect the challenges faced while solving real data-driven MOPs. By providing our detailed observations of the tools and techniques used, **we present a guideline to solve data-driven MOPs interactively**.

In Section 2, we establish the core background concepts. In Sections 3 and 4, we describe the various steps involved in formulating and solving the MOP, respectively. Section 3 covers the first two steps of the seamless chain, whereas Section 4 covers the last two steps. We start in Section 3 by describing the dataset which contains information about alloys of microalloyed steels, training surrogate models, and formulating the MOP. More specifically, the dataset contains the compositions and the corresponding values of various metallurgical properties of the alloys. We then discuss the tools and strategies used to preprocess the data to make it suitable for use in MOPs. Following this, we test a large number of surrogate modelling techniques to find the ones that work best with the data. Finally, using the results of the previous steps, and with the help of the DMs, we formulate meaningful MOPs to be solved. In Section 4 we use the open-source software framework called DESDEO[2] (Misitano et al., 2021) to implement the MultiDM/IOPIS algorithm and solve the MOPs interactively with two DMs. We also use two non-interactive optimization methods and compare the results.

It should be noted that the process of problem formulation and solution is usually an iterative one: we update the methodologies used in earlier steps based on the results obtained in the later steps. However, in Sections 3 and 4 we provide a linear narrative of the steps involved for the benefit of the reader. In Section 5, we discuss the effectiveness of the various steps we took to solve the MOP, including steps that did not succeed. We also discuss the insights gained via the seamless chain process. In doing so, we provide a general framework (i.e., not limited to the application considered) and a guideline to solve data-driven MOPs. Finally, in Section 6, we conclude and mention some future research directions.

The aim of the article is not just to solve a challenging MOP. Instead, we use the MOP to showcase how to tackle various challenges encountered in data-driven MOPs and decision making with multiple DMs. We describe the tools we used during various steps of the seamless chain, and also discuss why we use them. We also implemented the MultiDM/IOPIS algorithm to enable multiple DMs to control the interactive optimization process and find a mutually satisfactory solution.

[2] https://desdeo.it.jyu.fi

We make all algorithms proposed in the article openly available to facilitate future research in data-driven optimization and group decision making.

2. Background

2.1. Multiobjective optimization

This paper considers the following form of a multiobjective optimization problem:

$$\begin{aligned} \text{minimize} \quad & \{f_1(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ \text{subject to} \quad & \mathbf{x} \in S \subset \mathbb{R}^n, \end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1, \dots, x_n)^T$ represents *decision vectors* (vectors of decision variables) in the feasible region S of the decision space $\mathbb{R}^n$. The number of objective functions is k and the vectors of objective function values, denoted by $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$, are defined as *objective vectors* belong to the *objective space* $\mathbb{R}^k$.

Because the analytical form of functions is not available in data-driven optimization problems, the decision variables and their corresponding objective function values are often collected from simulators, real-life processes, or experiments and are only available in a dataset format. In such problems, approximation models, also called surrogates (or metamodels), are created, using the available data to approximate objective function values. Then, these surrogates are utilized to perform the optimization process. Surrogates may also be utilized as replacements for computationally expensive functions or simulators to reduce the evaluation time and save computations resources (Chugh et al., 2016; Tabatabaei et al., 2015). When collecting new data is not possible during the optimization process, as in the case of this paper, the problem is called an offline data-driven optimization (Wang et al., 2018) problem. It means that the constructed surrogates cannot be updated with some new data.

Typically, in MOPs, identifying a single optimum is not possible because of the existing conflict between the objective functions. Instead, multiple (can be infinitely many) so-called Pareto optimal solutions exist, where improving any objective function value is impossible without impairment in at least one of the other objective function values. The set of Pareto optimal objective vectors in the objective space is called a Pareto front. DM needs to compare Pareto optimal solutions, study the existing trade-offs between objectives, and choose the most preferred one based on their preferences. Besides the DM, an analyst (or a group of analysts) is responsible for performing the analyses, computations, and supporting the interactive decision-making process. Generally, an analyst can be a human or a computer program (Miettinen et al., 2008).

Based on when preference information is incorporated, multiobjective optimization methods can be classified as a priori, a posteriori, and interactive methods (Miettinen, 1999). DM provides their preferences before and after the solution process, respectively, in a priori and a posteriori methods. Providing unrealistic preferences is the major shortcoming of the a priori method, particularly if the DMs do not have prior deep insight into the problem. On the other hand, computational cost of generating a representative set of Pareto optimal solutions and heavy cognitive loads of many comparisons are the main difficulties in using the a posteriori methods, especially when there are many objectives (Deb and Saxena, 2006; Ishibuchi et al., 2008).

When the DM actively directs the solution process by providing preferences iteratively, the multiobjective optimization method is called interactive (Miettinen, 1999; Miettinen et al., 2008). In this way, the DM can learn about the interdependencies of the objectives in the problem and the feasibility of their preferences. Furthermore, these methods limit both cognitive and computational load since only a limited amount of information needs to be analyzed at a time, and only solutions reflecting the DM's preferences need to be generated, respectively. The DM can pursue the interactions by adjusting the preferences until they get satisfied and converge to the most preferred solution (see Miettinen et al., 2008 for more details).

There are different ways to solve MOPs (see, e.g., Branke et al., 2008; Miettinen, 1999; Miettinen et al., 2008); among them, one widely used approach is to transform the MOP into an equivalent single-objective problem utilizing a so-called scalarization function while incorporating DM's preferences (Miettinen, 1999; Miettinen and Mäkelä, 2002; Ruiz et al., 2009). The scalarized MOP can then be solved using an appropriate single objective optimizer, resulting in one or more Pareto optimal solutions that reflect the preferences to satisfy the DM.

Another way to solve MOPs is to use multiobjective evolutionary algorithms (MOEAs) (Branke et al., 2008; Ishibuchi et al., 2008). They are metaheuristic approaches which use a "population" of solutions simultaneously to mimic the process of evolution. Popular MOEAs such as RVEA (Cheng et al., 2016) and NSGA-III (Deb and Jain, 2014) have proven to be successful at generating a representative set of Pareto optimal solutions for MOPs with many objectives. However, MOPs become exponentially more difficult to solve with an increasing number of objectives (Deb and Saxena, 2006). The DESDEO framework provides open-source and modular Python implemenetations of RVEA and NSGA-III (and many other MOEAs), as well as many scalarization functions and interactive methods (Misitano et al., 2021). The DESDEO framework is the only open-source framework designed for optimization using interactive methods (Misitano et al., 2021). Its modularity enables easy creation of new interactive methods using components of other interactive methods implemented in the framework.

The IOPIS algorithm (Saini et al., 2020) provides a middle ground between using scalarization functions (which optimize in a single dimension) and MOEAs (which generally optimize in the objective space with many dimensions). The IOPIS algorithm incorporates a DM's preferences using multiple scalarization functions (typically fewer than the number of objectives in the original problem). Together, these functions form a new space called a preference incorporated space (PIS). An appropriate MOEA is then used to optimize in this new space, making the MOEA interactive (since preferences are incorporated). An analyst can therefore control the number of dimensions in which the MOEA optimizes by changing the number of scalarization functions, which form the PIS.

The IOPIS algorithm is of note as it enables easy and modular creation of interactive MOEAs. Moreover, as shown in Saini et al. (2020), it guarantees that the interactive MOEAs will have the beneficial properties of optimality (the solutions found by the MOEA are Pareto optimal), preferability (the solutions found by the MOEA follow the preferences of the DM), and searchability (the MOEA enables the DM to find any Pareto optimal solution by changing the preferences). However, the IOPIS algorithm was originally designed for solving MOPs with a single DM.

2.2. Microalloyed steels

As mentioned, we consider a data-driven problem of microalloyed steel. Steels used for structural, linepipe and naval applications must meet strict performance standards and withstand mechanical stresses imposed in such applications without failure. Microalloyed steels[3] exhibit the required high strength, toughness, ductility, and weldability capacity (Kim, 1983). Entities such as the British and European standards (BS EN standards), the American petroleum institute (API standards), and the US naval sea systems command (military or MIL standards) have published standards for usage of microalloyed steels in various domains. Each published standard includes one or more grades of microalloyed steel which set the minimum requirements and maximum bounds that should be met by the steel for specific applications.

Yield strength (YS) and ultimate tensile strength (UTS) are two important measures of the strength of materials. The YS measures the

[3] A subcategory of high-strength low-alloy steels.

maximum stress (force per unit area) a material can sustain before undergoing permanent (plastic) deformation. Below the YS, the material deforms elastically, i.e., it reverts to its original shape and size after removing the external force. The UTS measures the maximum stress a material can sustain before undergoing a fracture. Percentage elongation (ELON) measures the ductility as the fractional increase in the length of a specimen that undergoes fracture upon reaching the UTS.

We can use Charpy impact tests to measure the toughness of materials at various temperatures. The test measures the energy required to fracture a standard specimen made from the material using an impact. The Charpy impact energy generally decreases at lower temperatures as ductile materials (such as steels) start showing brittle behaviour at such temperatures. Good toughness at low temperatures is required for steels used at such temperatures. The result of these tests can be reported as Charpy energy values at specific temperatures or impact transition temperature (ITT) values at specific Charpy energy values.

The aforementioned mechanical properties of microalloyed steels depend strongly on the composition of the steel alloy. For example, in minute quantities, alloying elements such as titanium, molybdenum, and vanadium generally positively affect the strength, ductility, and toughness of microalloyed steels. However, they can hinder the ability of structures made from such steels to be welded. The weldability of steels is correlated with the carbon equivalent (C_{eq}), which can be calculated as (Lancaster, 1999):

$$C_{eq} = \%C + \frac{\%Mn + \%Si}{6} + \frac{\%Cr + \%Mo + \%V}{5} + \frac{\%Cu + \%Ni}{15}. \tag{2}$$

The terms on the right-hand side of (2), $\%C, \%Mn, \%Si, \%Cr, \%Mo, \%V$, $\%Cu$, and $\%Ni$ measure the concentration of the alloying elements carbon, manganese, silicon, chromium, molybdenum, vanadium, copper, and nickel as a mass percentage respectively. Various steel grades set different upper limits to the acceptable levels of C_{eq} to account for weldability needs.

For different applications, different compositions of steels are needed which satisfy the various standards mentioned previously. Therefore, there is a need to formulate and solve MOPs to design microalloyed steel compositions. Past studies (Roy et al., 2020; Chakraborti, 2022) have considered MOPs with up to three objectives to design microalloyed steels. These studies solved the MOPs using non-interactive MOEAs.

3. Preprocessing, surrogate modelling and problem formulation

In this section, we follow the first two steps of the seamless chain to formulate an MOP. We first describe the characteristics of the raw data we used for our study in the text below. Then in Section 3.1, we preprocess the raw data to use it in later steps. In Sections 3.2 and 3.3, we train and validate surrogate models using the processed data. Finally, in Section 3.4, we formulate two MOPs using the surrogate models for the microalloyed steel composition design problem.

The raw dataset, compiled from a database and available in the DESDEO framework, contains details about the metallurgical properties of microalloyed steels. There are 736 rows and 51 columns in the dataset. Each row denotes information related to microalloyed steel of a specific composition, which we later refer to as a sample. The first twenty columns contain information about the concentration of various alloying elements. The rest of the columns contain information about various metallurgical properties of the samples. The first three among these are YS, UTS, and ELON. The next ten columns denote the ITT at ten different Charpy impact energy levels (ranging from 13 J to 80 J). The next six columns measure the Charpy impact energy value at different temperatures (ranging from −80 °C to 19 °C). The remaining columns contain properties such as fracture toughness, pearlite content and hardness.

As in may cases with real data, there are many issues with the raw dataset. As the raw dataset is a compilation of data from various sources, the data quality is not consistent across the rows. For example, some cells have exact numbers, whereas others have a range. Moreover, measurements of properties such as the Charpy energy are very noisy. While the raw dataset contains 736 rows of samples, many cells are empty in various columns. Consequently, many rows do not completely contain information about the samples' composition, grain size, and metallurgical properties. For example, there are only 599 samples that measure YS, 537 UTS measurements, and 296 ELON measurements. The number of rows that contain information about other metallurgical properties is much lower. Moreover, these measurements are spread across the data points such that the overlap in rows that measure two different metallurgical properties is very small. This means that the different properties are measured for entirely different alloy compositions, with minimal overlap.

3.1. Preprocessing the data

The first step of the seamless chain is preprocessing of the data. We cleaned and divided the raw dataset into multiple sets to be used in later steps. Firstly, all empty cells in the alloy composition columns were assigned a value of zero. As the dataset was collected from various studies, empty cells signify that those alloying elements were not a part of the study, and hence did not exist in the alloy. Secondly, we merged the columns named "Nb" (niobium) and "Cb" (columbium) as they refer to the same alloying element. Finally, we replaced the cells in the alloy composition columns that were represented as a range of values by the average value of the range. At this step, all cells in the alloy composition columns had numerical values.

We then divided the dataset into multiple sets such that each set contained all the alloy composition columns but only one metallurgical property. For this, we only considered rows which documented the respective metallurgical property. This led to, for example, a YS dataset with 599 samples. We repeated the process for UTS and ELON. None of the columns documenting ITT or the Charpy energy had more than 300 samples. Hence, instead of breaking these columns into multiple small sets, we combined the 10 ITT columns and the 6 Charpy energy columns into just two columns documenting the temperature and the corresponding Charpy energy.[4] This process led to a Charpy dataset with 781 samples.[5] We used the Pandas Python package to carry out the aforementioned tasks (Reback et al., 2020).

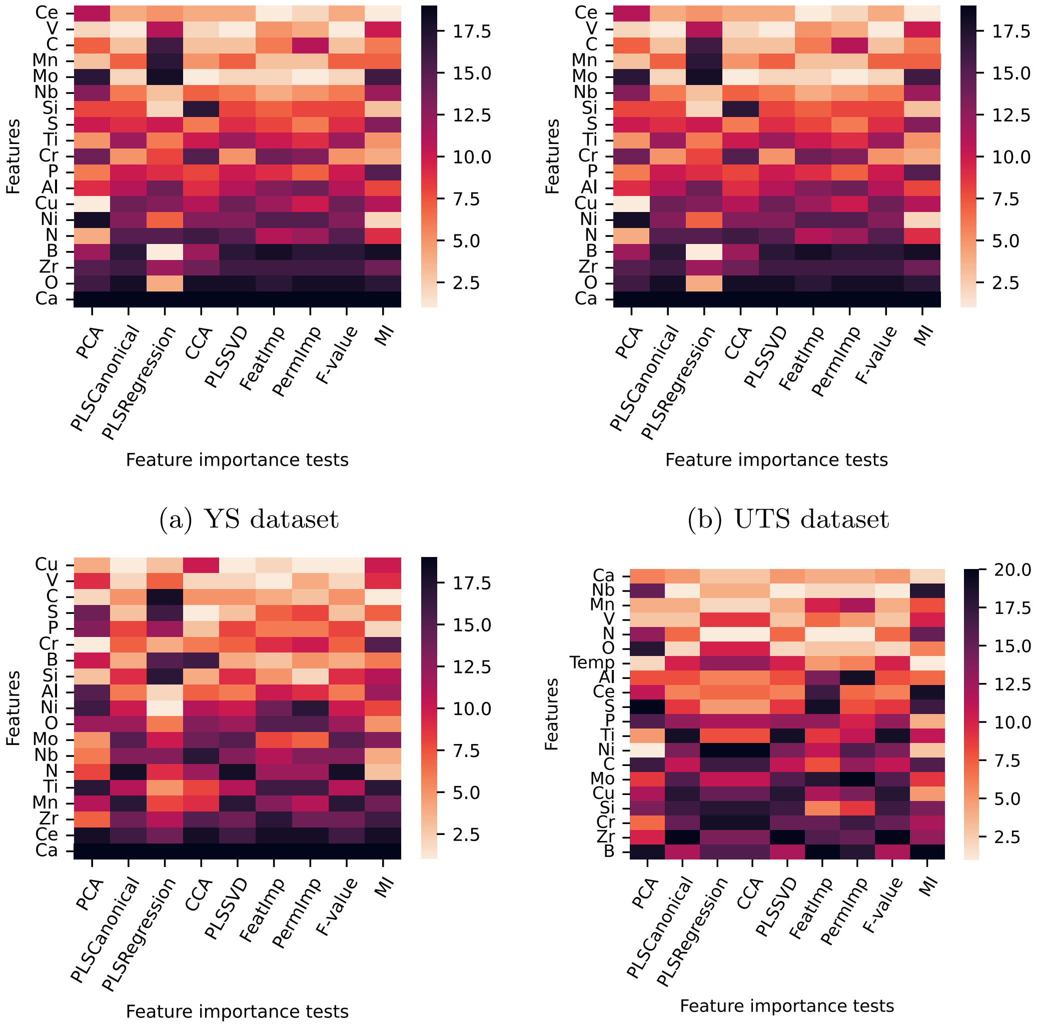
3.2. Model selection and training

The second step of the seamless chain is training surrrogate models. To begin the modelling process, we first analysed the dataset to identify which alloying elements significantly impacted the various metallurgical properties. We compared the results of the feature importance analysis against metallurgical literature to confirm the reliability of the data. In brief, we conducted the following tests to identify significant alloying elements for the YS, UTS, Elongation, and Charpy datasets using the `scikit-learn` Python package (Pedregosa et al., 2011):

- Principal component analysis (PCA): This method can help us find "features" (alloying elements in our case) that have the most variance in the dataset.
- Cross decomposition: We use the PLSCanonical, PLSSVD, PLSRegression, and Canonical Correlation Analysis (CCA) algorithms to find out which features lead to the most variance in the various metallurgical properties.

[4] This process is known is data "melting" and converts "wide" data (more columns, fewer rows) to "tall" data (fewer columns, more rows)

[5] Note that, at this step, we have datasets for four mechanical properties of microalloyed steels already. Therefore, we can create an MOP with four objectives, which is more than the three objectives considered in earlier studies. However, in later subsections, we will add even more objectives to our MOP formulation.



(a) YS dataset

(b) UTS dataset

(c) Elongation dataset

(d) Charpy dataset

Fig. 2. Relative ranking of the features calculated for all datasets. A lower rank value (lighter colour in the heatmap) represents higher importance.

- Random forest regression: We calculate feature importance (FeatImp) and permutation importance (PermImp) using random forest models trained on the dataset.
- ANOVA tests: We rank various features according to their F-values.
- Mutual importance (MI): We rank the various features according to their MI values.

The results of the tests are presented in Fig. 2 as heatmaps. The x-axis in each of the sub-figures denotes the aforementioned tests, whereas the y-axis represents the alloying elements (and temperature values for the Charpy dataset). The colour of the heatmap represents the relative importance (calculated as a rank) of the alloying element as measured by each test. Elements with lighter hues achieved a better rank and are considered more important by the tests.

There are various reasons why a test may consider a certain alloying element unimportant for a given mechanical property. The alloying element may have no effect, or a mixed effect on the mechanical property. For example, nitrogen can form nitrides with titanium, which can lead to grain refinement, which leads to better YS. However, in the presence of aluminium, it can form AlN, which can lead to embrittlement of the steel, which lowers the YS. Hence, nitrogen can have a mixed effect on YS, and is ranked low by the tests. The dataset may also have a very skewed distribution of the values of alloying element concentrations. In the UTS dataset, only 25 (out of 537) samples contained Zirconium. The effect of such elements may not be adequately represented in the dataset, which can lead to a low rank. Finally, certain alloying elements may show a mixed effect on the mechanical properties because of noise in the dataset.

Based on these tests, we removed unimportant or noisy columns from the datasets. This can increase the overlap in the ranges of alloying element compositions for which the various properties are measured and lead to better modelling and optimization results. We discuss this further in Section 3.4.

We trained surrogate models for YS, UTS, ELON, and Charpy energy based on their respective datasets using many surrogate modelling algorithms and compared their training accuracy using the R^2 value. We considered the following surrogate modelling algorithms: neural networks (Gardner and Dorling, 1998), support vector machines (Steinwart and Christmann, 2008), Gaussian process regression (Matheron, 1963; Emmerich, 2005), and various ensemble modelling techniques. These surrogate modelling algorithms have been used extensively and successfully in data-driven multiobjective optimization (Jin et al., 2021). We conducted K-fold cross-validation to choose the best performing surrogate modelling technique for each metallurgical property. Details of the test, along with a Python implementation, can be found in the DESDEO framework.

The results of the test are shown in Table 1. The full names and details of the surrogate modelling techniques are presented in Table 6 in Appendix B. In general, ensemble techniques worked better

Table 1
The K-fold cross-validation performance (R^2 score) of various surrogate modelling techniques on YS, UTS, Elongation, and Charpy datasets.

Surrogate modelling techniques	YS		UTS		Elongation		Charpy energy	
	Median R^2	Standard deviation	Median R^2	Standard deviation	Median R^2	Standard deviation	Median R^2	Standard deviation
ExTR	**0.746**	**0.059**	0.8380	0.0709	**0.673**	**0.124**	0.166	0.129
Ada	0.604	0.0818	0.7403	0.154	0.575	0.126	0.293	0.0724
Bagging	0.716	0.0492	0.8358	0.0839	0.612	0.108	0.275	0.109
GradBoost	0.712	0.0619	0.8437	0.0629	0.666	0.177	**0.436**	**0.0843**
XGBoost	0.742	0.0694	**0.8440**	**0.0781**	0.58	0.128	0.178	0.117
XGBRF	0.642	0.0571	0.7678	0.0932	0.651	0.136	0.423	0.0765
LightGBM	0.732	0.0511	0.7833	0.0587	0.642	0.141	0.305	0.103
RandomForest	0.726	0.037	0.8307	0.0679	0.64	0.12	0.317	0.0994
Kriging	−204	1920	−85.18	2030	−509	835	−6.5	3.6
SVM1	−0.0253	0.0435	−0.03087	0.0352	0.172	0.0707	−0.0803	0.11
SVM2	−0.232	0.14	−0.3843	0.143	0.0344	0.134	−0.0244	0.184
NN	−6.79	1.12	−11.35	7.29	−0.452	0.259	−0.0519	0.113

Table 2
Effect of alloying element composition on mechanical properties calculated using ICE plots. The alloying elements with no reported effects were not considered for the model.

Alloying element	Effect on YS	Effect on UTS	Effect on ELON	Effect on Charpy energy
Carbon	positive	positive	negative	mixed
Silicon	mixed	positive	negative	–
Manganese	positive	positive	mixed	positive
Phosphorus	negative	negative	negative	negative
Sulphur	mixed	negative	negative	negative
Molybdenum	positive	positive	negative	positive
Nickel	positive	positive	negative	mixed
Aluminium	negative	negative	positive	positive
Nitrogen	–	–	mixed	negative
Niobium	positive	positive	positive	positive
Vanadium	positive	positive	negative	negative
Boron	–	–	mixed	–
Titanium	positive	positive	negative	mixed
Chromium	positive	positive	negative	negative
Cerium	positive	positive	–	–
Copper	negative	positive	negative	–
Zirconium	negative	negative	mixed	–

than other surrogate modelling techniques tested. The extra trees regression (Geurts et al., 2006), gradient boost regression (Friedman) and random forest regression (Breiman, 2001) (from Python package `scikit-learn`), XGBoost (Chen and Guestrin, 2016) (from Python package `xgboost`, and LightGBM (Ke et al., 2017) (from Python package `lightgbm`) performed the best. It should be noted that we use the default hyperparameter settings, as provided in their respective Python packages, for training the surrogate models. As the Charpy dataset was much noisier than the other datasets, the surrogate models for Charpy energy performed much worse. The model with the highest median R^2 value was chosen for each mechanical property, as highlighted in bold in Table 1.

3.3. Model validation

In the absence of the ability to create and test new alloys, we can validate the models by comparing the effect of changing the alloying element compositions on the models' responses. We can do so using individual conditional expectation (ICE) plots (Goldstein et al., 2015). They plot the changes in the output of a surrogate model based on changes in one of the input variables, which are the alloying element concentrations in our problem. The effect of alloying elements on the mechanical properties is presented in Table 2.

In the table, we divide the effect of changing alloying element concentrations into four categories. A positive effect means that the alloying element benefits the property modelled by the surrogate model, whereas a negative effect implies a negative correlation. A mixed effect implies that the alloying element has a complex relationship with the modelled property. It can have a beneficial or a detrimental effect based on other criteria, for example, the concentration of other alloying elements. In contrast, a nil effect (signified with "-") means that the alloying element has no effect on the surrogate model, or its effect could not be detected. We present a complete discussion of the effect of all alloying elements in Appendix A. The discussion is based on an extensive review (Abbaschian and Reed-Hill, 2009; Bhadeshia and Hansraj, 2019; Bhadeshia and Honeycombe, 2017; Gladman, 1997; Bae et al., 2002; Chen et al., 2004; Li et al., 2001; Rozhkova et al., 1981; Mesquita and Kestenbach, 2011; Erhart and Grabke, 1981; Vervynckt et al., 2012; Kim et al., 2001; Norström and Vingsbo, 1979; DeArdo, 2003; Morrison, 2009; Yan et al., 2006; Wilson and Gladman, 1988; Uggowitzer et al., 1996; Baker, 2009; Ghosh et al., 2014; Isheim et al., 2006; Ye et al., 2012; Adabavazeh et al., 2017; Guo et al., 2008) covering the effects of alloying elements on strengths (YS and UTS), ductility (ELON), and impact toughness (Charpy impact energy absorption at different test temperatures) in a variety of low-carbon steels used for structural, linepipe, automotive, naval, and pressure vessel applications.

The DMs concluded that the metallurgical theory and literature back a majority of ICE plot results. The models can be deemed valid and be used to study steels' mechanical behaviour. We can see in Table 2 that many elements have conflicting effects on YS, UTS, Elongation, and Charpy energy. This is ultimately the source of trade-offs in MOPs that consider such objectives.

3.4. Problem formulation

Based on the consideration described in the previous subsection, the selected surrogate models (ExTR for YS and ELON, XGBoost for UTS, and GradBoost for Charpy energy), can predict the values of YS, UTS, ELON, and Charpy energy value of vanadium microalloyed steels

Table 3
The upper and lower bound for the concentrations (in percentage weight) of the alloying elements and their cost as used in (MOP-I) and MOP-II.

Alloying element	Lower bound (MOP-I)	Upper bound (MOP-I)	Lower bound (MOP-II)	Upper bound (MOP-II)	Cost (USD per Kg)
Carbon	0.009	0.34	0.006	0.4	0
Silicon	0.01	0.55	0	0.6	0.122
Manganese	0.28	1.63	0.28	1.63	1.7
Phosphorus	0	0.035	0	0.035	1.82
Sulphur	0	0.035	0	0.042	2.69
Molybdenum	0	0.53	0	1.0	0.0926
Nickel	0	3.12	0	3.12	40.1
Aluminium	0	0.06	0	0.06	13.9
Nitrogen	0	0.024	0	0.024	1.79
Niobium	0	0.086	0	0.45	0.140
Vanadium	0	0.2	0	0.2	72
Boron	0	0.001	0	0.001	3.68
Titanium	0	0.18	0	0.31	11.5
Chromium	0	1.24	0	1.5	9.4
Cerium	0	0.015	0	0.015	4.6
Copper	0	0.312	0	0.312	6
Zirconium	0	0.008	0	0.008	237

as functions of their compositions (and additionally the temperature for the Charpy energy). Thus, the metallurgical properties form the objectives of an MOP to be optimized with the alloy composition as the decision variables. Besides the surrogate models, two more objectives were added to the problem formulation: carbon equivalent value and the cost of materials. These objectives can be numerically calculated as functions of the steel composition. Eq. (2) is used to calculate the carbon equivalent value. The cost of materials is calculated as a linear combination of the alloy composition values weighted by the cost of the elemental components as stated in Table 3.

We formally define the MOP (MOP-I) as:

$$\begin{array}{rl} \text{maximize} & \texttt{YS}(\mathbf{comp}) \\ \text{maximize} & \texttt{UTS}(\mathbf{comp}) \\ \text{maximize} & \texttt{ELON}(\mathbf{comp}) \\ \text{maximize} & \texttt{Charpy}(\mathbf{comp}, temp) \\ \text{minimize} & C_{eq}(\mathbf{comp}) \\ \text{minimize} & Cost(\mathbf{comp}) \\ \text{subject to} & \mathbf{LB} \leq \mathbf{comp} \leq \mathbf{UB}, \end{array} \tag{MOP-I}$$

where YS, UTS, ELON, and Charpy are the surrogate models for YS, UTS, ELON, and the Charpy energy, respectively. The decision variable **comp** is the vector of the concentration of 17 alloying elements in the steel bounded by lower and upper bounds **LB** and **UB**. The function C_{eq} refers to the carbon equivalent value, and the $Cost$ function refers to the cost of materials. The temperature $temp$ input for the Charpy surrogate model is kept constant at $-80\ {}^{\circ}C$. The bounds **LB** and **UB** are calculated as the bounds of the intersection of the datasets used for all four surrogate models and the values are given in Table 3. Expanding this range makes the search space larger, which increases the likelihood of finding suitable alloy compositions. This is why removing unimportant or noisy columns was a crucial step in surrogate modelling, as described in Section 3.2.

We create a simpler version of (MOP-I) (named MOP-II) by removing the Charpy energy objective from (MOP-I). We consider this version since, as mentioned, the Charpy surrogate model had a worse accuracy than the other surrogate models. Removing this objective also led to an expansion of the bounds of the decision variables (calculated as the bounds of the intersection of the remaining datasets). The upper and lower bound values for the various alloying elements can be seen in Table 3.

4. Optimization and decision making

The third and fourth steps of the seamless chain are multiobjective optimization and decision making. The open-source software framework DESDEO (Misitano et al., 2021) provides a variety of interactive and some non-interactive methods to solve multiobjective optimization problems. We begin by solving the two MOPs formulated in the previous section with two non-interactive evolutionary algorithms: RVEA and NSGA-III, to generate an approximate representation of Pareto optimal solutions. These method have been reported to work well with MOPs with more than four objectives. The details of the optimization and the results are discussed in Section 4.1. We conducted optimization using non-interactive MOEAs and presented visualizations of the results to the DMs to help them form their initial preferences.

In Section 4.2, we outline the details of the interactive MOEA called MultiDM/IOPIS, a method developed to solve the MOP interactively with more than one DM. We then describe the interactive optimization process itself and the results obtained in Section 4.3.

4.1. Non-interactive optimization

We ran the RVEA and NSGA-III algorithms for 150 generations each to solve MOP-I and MOP-II. There was no improvement in the objective values attained by the solutions after around 120 generations. The other parameters of the algorithms were otherwise unchanged from the values suggested in original publications which proposed RVEA and NSGA-III. We combined the final solutions from both RVEA and NSGA-III and removed the ones where some objective function value was worse and other values not better than in some other solution. Figs. 3 and 4 show the remaining solutions for MOP-I and MOP-II, respectively, in parallel coordinates plots. Interactive versions of these plots can be seen at https://desdeo.it.jyu.fi.

The two methods found 1055 solutions for MOP-I and 1599 solutions for MOP-II. The ideal, i.e., best possible, values for the objectives of MOP-I were (726 MPa, 1577 MPa, 34%, 103 J, 0.087, 43 USD per kg). The ideal values for the objectives of MOP-II were (752 MPa, 1593 MPa, 34%, 0.055, 42.9 USD per kg). The methods were able to find slightly better solutions for MOP-II than for MOP-I. This may be because optimization becomes exponentially more difficult with an increasing number of objectives. As the parameters of the methods were the same for solving MOP-I and MOP-II, the results were slightly worse in the version with one more objective.

Despite the differing performance, certain similarities can be visually observed in the solutions of MOP-I and MOP-II. For example, in both cases, there is a cluster of solutions with very high UTS values, followed by a discontinuity, and then a different cluster of medium and low UTS values. There are also three clusters in the $Cost$ objective. In both problem formulations, $Cost$ did not have significant trade-offs with the ELON objective. The solutions represented high values for ELON for both low $Cost$ and high $Cost$ alloys. However, $Cost$ was very strongly correlated with UTS. Low $Cost$ alloys could only achieve UTS values of up to 1200 MPa. Only the highest $Cost$ alloys could achieve the best UTS values. Increasing $Cost$ also generally increased the YS values.

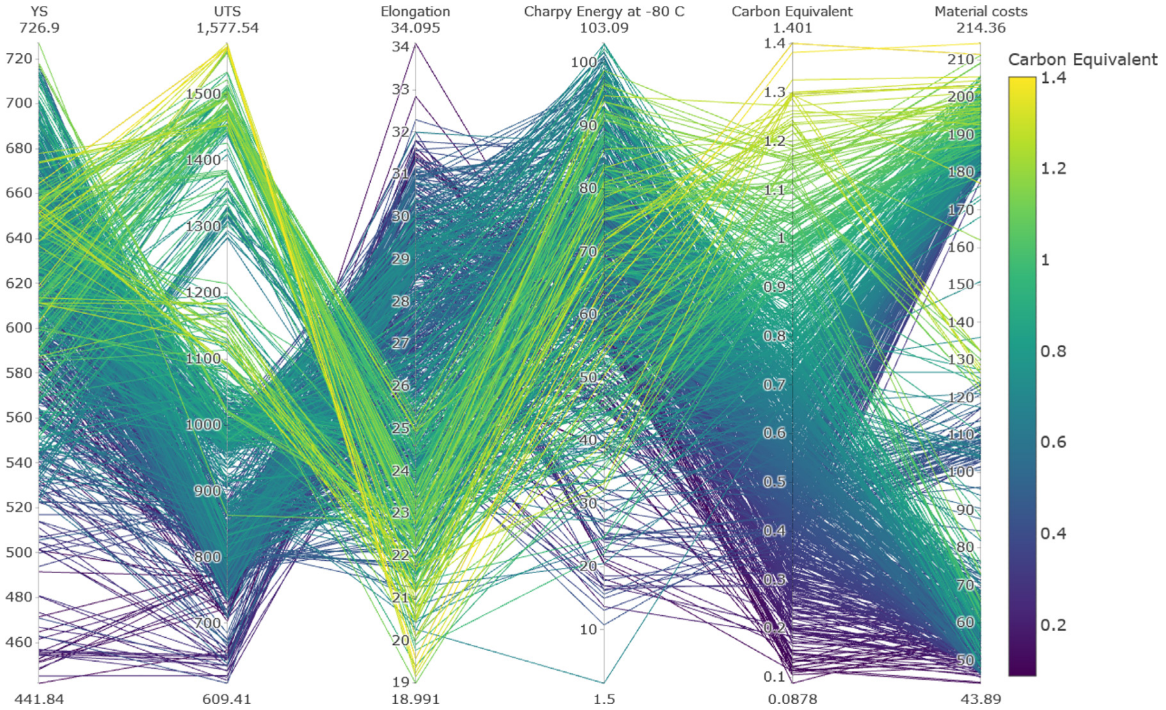


Fig. 3. Solutions of MOP-I presented as a parallel coordinates plot. The solution traces are coloured based on the Carbon equivalent value.

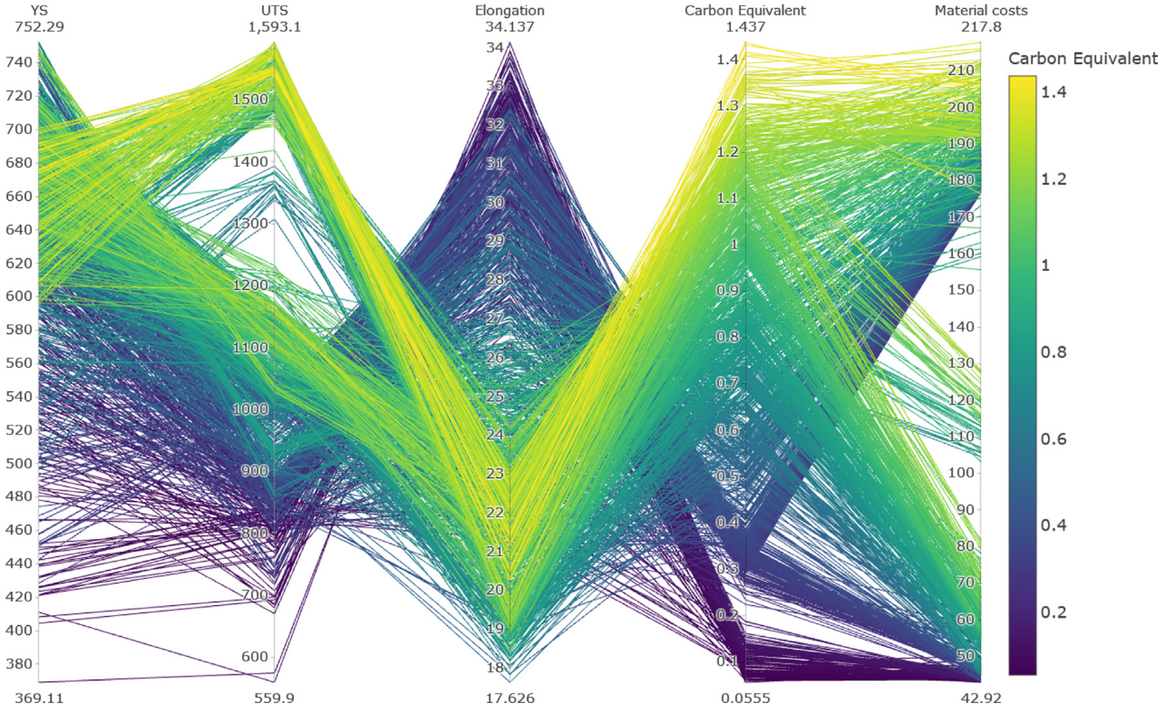


Fig. 4. Solutions of MOP-II presented as a parallel coordinates plot. The solution traces are coloured based on the Carbon equivalent value.

However, the low *Cost* alloys could still achieve the best YS values. Among the solutions of MOP-I specifically, the Charpy objective was weakly conflicting with all other objectives.

We compared the solutions of MOP-I and MOP-II against structural and pipeline steel grades. We enumerate the solutions that match various steel grades in Table 4. The MOP-II formulation resulted in many more feasible alloy compositions for structural steels than the MOP-I formulation. This is because, as mentioned earlier, MOP-II is easier to optimize than MOP-I. However, for linepipe steels, the MOP-I formulation discovered alloy compositions to satisfy a more diverse range of grades than MOP-II. The inclusion of the Charpy energy objective in MOP-I allowed a more diverse search space. Hence, the formulation was able to satisfy more grades. Neither formulation was able to discover solutions that matched the naval steel grades. This

Table 4
Number of solutions of MOP-I and MOP-II that match structural (BS EN), linepipe (API), and naval (MIL) steel standards.

Standard	Grade	Solutions (MOP-I)	Solutions (MOP-II)
BS EN-10025-4	S275ML	2	47
BS EN-10025-4	S355ML	2	47
BS EN-10025-4	S420ML	5	45
BS EN-10025-4	S460ML	5	17
BS EN-10149-2	S315MC	7	132
BS EN-10149-2	S355MC	8	149
BS EN-10149-2	S420MC	9	226
BS EN-10149-2	S460MC	7	182
BS EN-10149-2	S500MC	7	175
BS EN-10149-2	S550MC	3	133
BS EN-10149-2	S600MC	2	84
BS EN-10149-2	S650MC	1	58
BS EN-10149-2	S700MC	0	22
BS EN-10025-6	S460QL1	5	39
BS EN-10025-6	S500QL1	1	28
BS EN-10025-6	S550QL1	1	12
BS EN-10025-6	S620QL1	0	1
API - 5L - 2018	X42M	1	2
API - 5L - 2018	X46M	1	2
API - 5L - 2018	X52M	1	2
API - 5L - 2018	X56M	1	1
API - 5L - 2018	X60M	1	0
API - 5L - 2018	X65M	1	0
API - 5L - 2018	X70M	1	0
API - 5L - 2018	X80M	1	0
MIL-S-24645A(1984)	HSLA-80	0	0
MIL-S-24645A(1989)	HSLA-100	0	0

signifies that RVEA and NSGA-III were not able to find solutions in all regions of the Pareto front.

4.2. MultiDM/IOPIS

As mentioned earlier, optimization with MOEAs to approximate the entire Pareto front becomes increasingly more challenging with more objectives. Interactive MOEAs resolve this issue by using the preferences of a DM to narrow down the search to solutions that are of interest to the DM, thus, focusing computational resources in a smaller region. In this study, there were two domain experts who acted as DMs, whereas one of the authors acted as an analyst to guide them through the interactive decision making process. As the number of objectives is not an issue for the interactive MOEAs, only MOP-I was solved.

IOPIS was originally proposed for a single DM. We developed and implemented a new variant of the IOPIS algorithm (Saini et al., 2020), called MultiDM/IOPIS, to support collaborative interaction with multiple DMs simultaneously, that is, group decision making. The new algorithm provides this support while maintaining the beneficial properties of the original IOPIS algorithm, i.e., the guarantees of optimality, preferability, and searchability. The original IOPIS algorithm uses multiple scalarization functions, which take the same preference information, to convert an MOP to a new MOP with a lower number of objectives. The new MOP, with the preference information as a fundamental building block, allows non-interactive MOEAs such as RVEA and NSGA-III to focus on the region of interest of the DM and, thus, converts these non-interactive methods as interactive ones.

MultiDM/IOPIS reverses this concept by using copies of the same scalarization function, taking different preferences as input. The multiple preferences can come, for example, from multiple DMs, as in our case. MultiDM/IOPIS, thus, creates a new MOP with the same number of objectives as the number of DMs, regardless of the number of objectives in the original MOP. In our case, the number of DMs (two) is much smaller than the number of objectives (six), making the new MOP formed by MultiDM/IOPIS much easier to solve.

In brief, the interactive optimization process with multiple DMs applying MultiDM/IOPIS involves the following steps: (1) The DMs provide individually their own reference points reflecting their desired values for the objectives. (2) The new MOP is formed, where the objective functions are scalarization functions containing the preferences of one of the DMs. (3) The MOP is solved with an MOEA to get a set of solutions to be shown to the DMs. (4) The DMs analyze the solutions to see how their own and the preferences of the others are reflected. If a satisfactory solution is found, stop. Otherwise, go to (1).

Let us elaborate the behaviour of MultiDM/IOPIS in the case of two DMs (for simplicity). It has the following modes of operation:

1. Case 1. The DMs provide extremely different preferences: The MOEA converges towards the two solutions that best satisfy the two preferences individually, as well as many solutions that present a compromise between the preferences of the DMs. The inclusion of compromise solutions provides necessary information to the DMs to collaborate on a compromise.
2. Case 2. One DM explores the objective space by changing preferences but the other DM stays anchored: The MOEA converges further and provides better solutions near the region of interest of the DM that did not change preferences. At the same time, the MOEA discovers new solutions to reflect the new preferences given by the first DM. The MOEA also provides compromise solutions between the two regions of interest. This mode allows for quick but controlled exploration of potential solutions: discovery of new solutions in new regions of interest and the corresponding trade-offs with the anchored preferences. Thus, even the anchored DM can find new, potentially favourable solutions.
3. Case 3. The DMs provide preferences close to each other: The DMs are expected to arrive at a favourable compromise during the interactive optimization process. The MOEA returns solutions in a narrower region as the preferences draw closer, providing many solutions with minor variations in the objective values. This variety allows the DMs to choose a finely-tuned solution for their application.

To solve our problem, we applied MultiDM/IOPIS with the scalarization function from the STOM method (Nakayama and Sawaragi, 1984) in our implementation of MultiDM/IOPIS[6] (it was already available in DESDEO) and NSGA-III to solve the resulting biobjective optimization problem. NSGA-III generally performs better than RVEA in MOPs with a lower number of objectives (Li et al., 2018). The STOM scalarization function takes a DM's preferences as a reference point: a vector of objective values (for all objectives) that the DM aspires to achieve. In the interactive optimization process, the DMs can change the reference points at any time, resulting in MultiDM/IOPIS creating a new MOP which reflects the new preferences. However, the population from the MOEA continues its evolution from the previous MOP. Thus, no progress on optimization is lost.

We set the population size of NSGA-III to be 50. While this is an inadequate size for a problem with six objectives, we found that it is sufficient for the modified MOP generated by MultiDM/IOPIS. Moreover, we only ran the MOEA for 30 generations between iterations, i.e., asking the DMs for preferences and providing near-Pareto optimal solutions to analyze and update their preferences. We arrived at these parameter settings by first running the algorithm with the default population size and number of generations (as suggested in Deb and Jain (2014)), and then lowering the parameter values as long as there was no change in the quality of the solutions found by the algorithm. In our experiments, the small number of generations and population size led to a near-instantaneous return of solutions and high computational

[6] The DESDEO framework was designed with modularity and ease-of-use as core concepts. As such, our implementation of MultiDM/IOPIS in DESDEO can be used in exactly the same way as any other MOEA implemented in DESDEO. We provide examples of such usage in Misitano et al. (2021) and in the documentation of the framework.

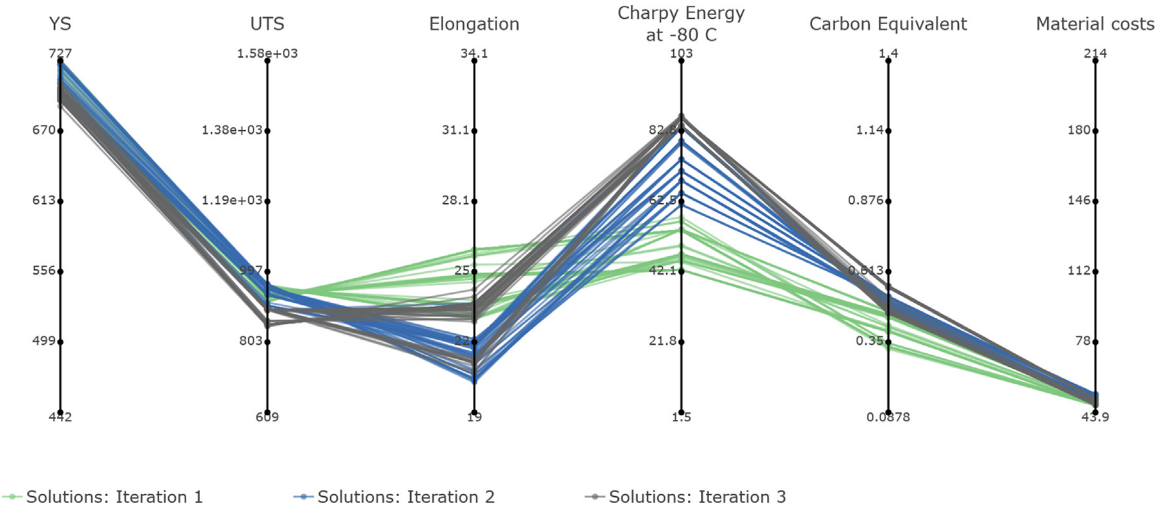


Fig. 5. Results of interactive optimization using MultiDM/IOPIS. The results from three iterations are presented in green (iteration 1), blue (iteration 2), and grey (final iteration).

Table 5
Preferences given by the two DMs during interactive optimization using MultiDM/IOPIS in the form of reference points. The values of components of each reference point represents the preferences for objectives in the order (YS, UTS, ELON, Charpy, C_{eq}, $Cost$).

Preferences in the first iteration	DM 1: (460, 550, 17, 27, 0.47, 50) Similar to structural grade S460QL1 DM 2: (690, 770, 14, 27, 0.65, 80) Similar to structural grade S690QL1
Preferences in the second iteration	DM 1: (552, 600, 20, 81, 0.6, 50) Similar to naval grade HSLA-80 DM 2: (690, 770, 14, 27, 0.65, 80) Similar to structural grade S690QL1
Preferences in the third iteration	DM 1: (552, 600, 20, 81, 0.6, 50) Similar to naval grade HSLA-80 DM 2: (690, 770, 18, 81, 1, 80) Similar to naval grade HSLA-100

efficiency. The other parameters (such as the crossover and mutation operators) of NSGA-III were otherwise unchanged from the settings suggested in the original publication.

4.3. Interactive optimization process

We started the process of interactive optimization by first showing the solutions of the non-interactive optimization methods to the DMs (Figs. 3 and 4) in interactive plots that allowed brushing and filtering of solutions. This process enabled them to explore the approximated Pareto optimal solutions to learn about the trade-offs and form their initial preferences. The analyst introduced MultiDM/IOPIS to the DMs by describing how they can steer the interactive optimization process with their preference information (and the three modes of operation).

We show the first reference points given by the two DMs in Table 5. The first DM set the reference point to target solutions to match the structural steel grade S460QL1 and the second DM did the same for S690QL1. During non-interactive optimization, the MOP-I formulation resulted in only five solutions to match the S460QL1 grade. The MOP-II formulation resulted in 39 solutions that matched that grade. Neither formulation had resulted in any solutions for the S690QL1 grade, as it is a stricter grade.

The analyst visualized the solutions generated based on the two reference points to the DMs in a parallel coordinate plot, see green lines in Fig. 5. We combine the solutions of all iterations in Fig. 5 for the sake of brevity. The solutions of the first iteration vastly exceeded the first DM's preferences. The solutions closely matched the second DM's much stricter preferences, while still giving much better values for the Charpy objective compared to the reference point. No solution discovered during the non-interactive optimization could match the solutions found in the first iteration of the interactive optimization in all six objectives.

For the second iteration, the presence of solutions with very good Charpy values led to the first DM changing preferences to a much stricter naval steel grade: HSLA-80. The second DM chose not to change preferences to preserve the newly discovered solutions for further consideration. The exact values of the two reference points are presented in the second row of Table 5.

We visualize the solutions of the second iteration in Fig. 5 in blue. The most apparent differences between the solutions of the two iterations were in the ELON and Charpy objective values. The Charpy values were much better for solutions of the second iteration, which came at the cost of worse ELON values (although still better than the given preferences). This change was because of the much higher value for the Charpy objective in the reference point given by the first DM. The equivalent carbon content C_{eq} for the solutions of the second iteration was also higher but still well within the grade specifications. There was a small number of solutions that satisfy the preferences given by the first DM (higher Charpy and ELON), and a small number that met the preferences given by the second DM (higher YS and UTS). A majority of solutions represented the trade-offs between the two preferences.

Based on the results of the second iteration, the second DM decided to change the preferences to a much stricter naval steel grade HSLA-100. The reference points for the third iteration (in the third row of Table 5) were very close to each other: the two DMs were coming into agreement regarding their preferences.

The solutions of the third iteration, shown in Fig. 5 in grey, had a narrower spread compared to the previous iterations. A significant trade-off this time was present between the UTS and ELON objectives. This trade-off originated from the differences in the two reference points. Even though the second DM allowed for a much higher $Cost$ value of 80 USD/kg, MultiDM/IOPIS achieved satisfactory values for all other objectives at close to half of the $Cost$ value. The DMs were satisfied with the solutions obtained and decided to end the interactive optimization process. The solutions of the third iteration were very similar and the solution with the objective values (700, 846, 24.2, 87, 0.470, 47) was chosen. This solution is better than the preferences provided by either DM as seen in Table 5, which is why it was chosen. The complete set of all solutions found by MultiDM/IOPIS in all three iterations can be found via the DESDEO framework.

5. Discussion

MOPs and their solution processes are often presented straightforwardly in the literature: problem formulation (often given without details of modelling), followed by optimization, visualization and discussion of results. It provides a straightforward approach to give an account of the problem and the steps taken to solve it. However, it hides many complexities of the challenges faced in solving real MOPs, particularly data-driven MOPs. In Sections 3 and 4, we provided a detailed account of all the steps taken to formulate and solve MOPs in the case considered. However, we did not discuss why some of those steps were necessary. This is because, for example, we took the measures taken during the first steps of solving the MOP (i.e., data preprocessing) to solve issues that arose during the last steps (i.e., optimization). We discovered many fundamental issues while solving the MOP, which required us to rethink our approach, apply fixes to earlier steps, and restart the solution process. An explanation of such issues and their resolutions requires the knowledge of the entire solution process.

After Sections 3 and 4, having discussed the entire solution process, we can now elaborate on the challenges we faced and justify the choices we made to solve them. In this section, we also talk about approaches that lead to dead-ends and, therefore, did not warrant a mention in the previous sections.

The first issue we faced was converting the data into a usable format. The naive approach of simply training surrogate models on the raw datasets (mostly empty cells) would lead to issues related to extrapolation (models giving bad predictions in areas outside the bound of a given objective but within the bound of the dataset). Sometimes, the models trained in such a manner also led to run-time errors. Thus, cleaning and preprocessing the data were needed. Dividing the dataset into individual datasets for each objective also allowed us to calculate each objective's domains (ranges of decision variable values). This information helped us define the constraints of the MOP during the later steps.

The dataset involving the Charpy energy and ITT experiments required additional treatment as it required combining multiple columns (Charpy energy measured at various temperatures/ITT measured at various energy levels) into just two columns (temperature and Charpy energy). No individual column in the original dataset contained enough information for proper training of corresponding surrogate models. Therefore, initially, we chose not to include Charpy energy in this study. However, the MOEAs could not find solutions to match steel grades requiring high performance at very cold temperatures because of the lack of Charpy energy in our initial problem formulations. The Charpy dataset we created enabled us to train reasonably well-performing models, which solved the issue.

Creating the MOP required us to find the intersection of the domains of all the objectives. We discovered in our initial calculations that the intersection was a null set. Thus, the models trained on the individual datasets would constantly be extrapolating if used together. This issue was mildly resolved by removing some extraneous columns that were almost entirely empty. This demanded further additions to the data preprocessing steps. However, the intersection was still very restricting. Expanding on the previous idea, we posited that we could remove even more columns to increase the intersection area. Therefore, it was essential to identify which decision variables had little or no impact on the objective values. We discuss the tests conducted for this task at the beginning of Section 3.2.

The choice and verification of models are crucial in data-driven optimization and help build trust in the solutions generated by the methods. Ideally, such models should be verified by confirming their predictions from experiments (creating and testing alloys, in our case). However, those resources were not available to us during this study. Therefore, we conducted rigorous testing of many surrogate modelling techniques to determine the best choice for each objective. We also confirmed that the predictions of the models matched current metallurgical literature. For modelling Charpy energy specifically, we tried some novel approaches of combining metallurgical knowledge (such as approximate functions which predict Charpy energy based on temperature) and surrogate modelling techniques. However, because the dataset contained many different kinds of steels, this approach did not work as well as simply using the surrogate model with the Charpy dataset.

During interactive optimization, we encountered the issue of supporting multiple DMs. We could not find implementations of interactive optimization methods that could solve our MOP while incorporating the preferences of multiple DMs. Therefore, we created a new method using DESDEO, based on the IOPIS algorithm. We elaborated on MultiDM/IOPIS in Section 4.2, which uses reference points as the mode of receiving preferences. However, we had created multiple variants of IOPIS which use different forms of preferences. The DMs in question felt most comfortable giving preferences in the form of reference points, so we only discuss MultiDM/IOPIS in this paper. The creation of these new methods was greatly quickened by the modularity of the DESDEO framework, which enabled us to reuse the components already present in the framework instead of implementing everything from scratch.

The ease of use of the DESDEO framework also allowed the analyst to quickly switch between and demonstrate different MOP formulations to the DMs. This, along with being actively involved in the interactive optimization process, inspired the DMs to suggest the addition of the carbon equivalent value as one of the objectives. Thus, MultiDM/IOPIS and DESDEO led to the formulation of a better MOP. Note that the DESDEO framework provides many other interactive methods, including ones that are not MOEAs (Miettinen and Mäkelä, 2006; Ruiz et al., 2015). In MOPs with a single DM, any of these interactive methods (or, in fact, any interactive method from any other framework) can be applied.

To sum up, we faced many interesting challenges while solving (or even formulating) our MOP. We made additions to the DESDEO framework to resolve the challenges successfully. We implemented the novel MultiDM/IOPIS method that helped us arrive at a better problem formulation. It also provided better results than RVEA and NSGA-III while being computationally much more efficient, requiring fewer generations with a smaller population size.

As we have established, the linear schematic of a seamless chain presented in Fig. 1 does not capture the complexity of solving real-life data-driven problems entirely. We present a new depiction that reflects our experience of solving the MOPs presented in this study in Fig. 6. The decisions made to preprocess the data, model the objectives, formulate the MOP, and supporting one or multiple DMs are interconnected and interdependent. The data limits which objectives can be considered for the MOP. The modelling and MOP formulation steps can reveal further shortcomings and issues with the data, forcing the analyst to change the preprocessing strategy. The expertise of the DMs is crucial not just in decision making, but also in understanding the data, validating the surrogate models, and formulating meaningful MOPs. An analyst's role is to channel the expertise of DMs to successfully tackle the issues that arise during all steps of the seamless chain process.

To resolve such issues, an analyst needs to use a variety of tools, many of which we have described in this study. While other real-life data-driven problems may not have the same challenges we faced, we document the entire process as a general guideline of the seamless chain method to solve MOPs. We make all methods, procedures, implementations, data, and visualizations developed or created during this study openly available via the DESDEO framework (Misitano et al., 2021).

6. Conclusions

We considered many challenges in formulating and solving real-life data-driven multiobjective optimization problems. The challenges covered all steps of a seamless chain from data to decision-making

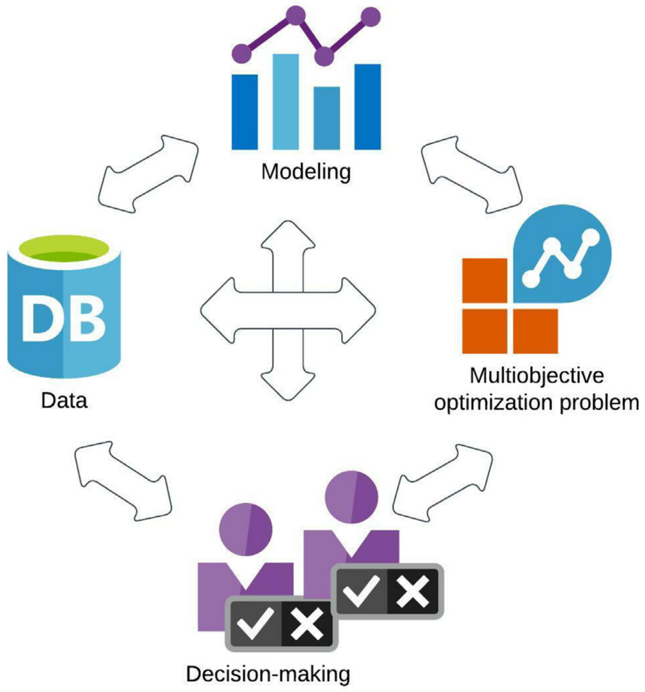


Fig. 6. A realistic depiction of the seamless chain.

that we introduced. The concrete steps related to microalloyed steels demonstrated the reasoning. We processed the raw dataset significantly to enable using it. We identified the essential decision variables, and trained and tested the best surrogate models to emulate the objectives derived from the data. We validated those models by comparing their response to changing decision variables to known metallurgical literature. With the surrogates as objectives, we formulated multiobjective optimization problems. We introduced interactive optimization to two domain experts who acted as DMs, which inspired them to create better problem formulations. We developed a novel interactive evolutionary method called MultiDM/IOPIS to simultaneously support the two DMs (who had different preferences) in interactive optimization. The solutions obtained were very satisfactory to the DMs and the solution process had a low computation cost.

We utilized many popular open-source tools in the process of formulating and solving our problems. The DESDEO framework was instrumental as it enabled us to experiment quickly with different versions of the problems. It also allowed us to implement the MultiDM/IOPIS method quickly by utilizing its modular implementations of scalarization functions (implemented via the GLIDE-II framework (Ruiz et al., 2012)) and MOEAs. We discussed the steps taken to solve the MOPs in great detail and established a methodology and guidelines for solving real-life data-driven MOPs. We provide all tools, data, and methods created or used openly via the DESDEO framework, enabling their usage by others.

MultiDM/IOPIS enables supporting multiple DMs simultaneously to solve multiobjective optimization problems. It worked very well in our study and found solutions that non-interactive evolutionary algorithms RVEA and NSGA-III could not find. While our study had only two DMs, MultiDM/IOPIS can support any number of DMs (by having as many scalarization functions in its formulation). The properties of such formulations and the application of MultiDM/IOPIS with more than two DMs need further studies. The algorithm is also limited by a lack of an intuitive UI, making providing preferences and visualizing results challenging for DMs. Designing UIs to facilitate group decision making is an important future research direction. Another interesting area of study is the application of MultiDM/IOPIS in online data-driven MOPs, i.e., data-driven MOPs with the option to conduct further function evaluations to increase the accuracy of the surrogates. Moreover, solving such problems requires tackling challenges specific to online data-driven problems, and a guideline to do so will be of great use.

CRediT authorship contribution statement

Bhupinder Singh Saini: Conceptualization, Methodology, Software, Writing a majority of the original draft, Revisions. **Debalay Chakrabarti:** Data gathering, Participation as a decision maker, Review. **Nirupam Chakraborti:** Data gathering, Participation as a decision maker, Review. **Babooshka Shavazipour:** Writing introduction in the original draft, Revisions. **Kaisa Miettinen:** Writing background in the original draft, Revisions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data will be made available via the DESDEO framework.

Acknowledgements

This research was partly funded by the Academy of Finland (grant 322221). The research is related to the thematic research area Decision Analytics utilizing Causal Models and Multiobjective Optimization (DEMO), jyu.fi/demo, at the University of Jyvaskyla.

Appendix A. Effect of concentrations of alloying elements on metallurgical properties of microalloyed steels

Carbon: Being the primary interstitial solute in steel, C atoms strongly interact with both edge- and screw-dislocations and provide significant solid solution strengthening. Besides, C increases hardenability and promotes the formation of harder phase constituents like pearlite, bainite and martensite which increases the strength. In addition, iron carbides and alloy carbides contribute to precipitation strengthening. An increase in strength by the increase in C content in general hampers ductility, toughness, formability and even weldability. The coarse and brittle carbide particles can act as the crack initiators (or void nucleation sites) and thus, affect impact toughness. On the other hand, fine carbide precipitates pin down the grain boundaries and restrict grain growth. The beneficial effect of microalloy carbide precipitates (say, NbC) on the retardation of austenite recrystallization and the consequent ferrite grain refinement in thermomechanical processed microalloyed steels is well known. Grain refinement is beneficial for impact toughness. As a result, C shows a positive effect on strength, a negative effect on ductility and a mixed response on impact toughness.

Silicon: As a substitutional solute, Si provides solid solution strengthening and shows a positive effect on strength (particularly UTS) and a negative effect on ductility. Si is preferred in steels containing low levels of C and other strengthening elements like Mn. That could result in a mixed response of Si on YS. Si decreases the cohesive strength of the atomic planes helping cleavage crack propagation and renders steel brittle, affecting impact toughness. In contrast, Si restricts the formation of iron-carbides (detrimental to roughness) and contributes to carbide free bainitic microstructures. As a result, Si does not show a clear trend on Charpy energy.

Manganese: Being present at a considerable amount (0.5 to 2.0 wt%), Mn is a strong solid solution strengthener in steels, which can naturally affect ductility. Mn is maintained at a higher side typically in low-C steels, having high ductility and impact toughness. Besides, Mn stabilizes retained austenite which contributes to the TRIP (transformation induced plasticity) effect. Mn also suppresses pearlite formation and coarsen the interlamellar spacing in pearlite. Although MnS inclusion is detrimental to toughness, it is not as harmful as iron-sulfide. These combined factors resulted in a mixed response of Mn on ductility and a positive influence on impact toughness.

Phosphorous: Segregation of P at the grain boundaries reduces the boundaries' cohesive strength, and the associated embrittlement seriously affects ductility and impact toughness. Although P is a solid solution strengthener, its content is restricted in commercial grades of steel. It is only allowed when the levels of C and the other alloying elements are very low (like P strengthened interstitial free steel). That is possibly the reason behind P's negative effect on strength, as detected here.

Sulfur: S has a strong negative effect on steel properties, particularly on ductility and toughness. Coarse and elongated MnS inclusions initiate large voids and fissures (promoting ductile fracture), cause anisotropy in properties, and even act as the cleavage crack initiators. Segregation of S at grain boundaries and interdendritic regions and the formation of iron-sulfides can be even more detrimental for properties. Being softer than the steel matrix, MnS can reduce the strength when present at a high fraction. However, occasionally fine MnS particles can offer grain refinement by (i) pinning the grain boundaries, and (ii) VN precipitation on MnS can provide nucleation sites for intergranular ferrite within austenite grains. Hence, S showed a mixed response on the YS and a negative influence on other properties.

Molybdenum: Mo provides solid solution strengthening and precipitation strengthening by forming various precipitates such as Mo2C, (Ti, Mo)C and (V, Mo)C. Mo also increases hardenability and promotes bainite transformation in steels. It refines the interlamellar spacing of pearlite. These aspects can have a beneficial effect on strength but a detrimental impact on ductility. Besides, Mo significantly retards temper embrittlement. In modern structural and linepipe grades of steel (and their weld joints), acicular ferrite microstructure is preferred to achieve high strength and high toughness. Mo can promote such microstructure and improve the impact toughness.

Niobium: As a microalloying element in thermomechanically processed high-strength low-alloy (HSLA) steels, NbC, NbN, and Nb(C, N) precipitate contribute significant ferrite-grain refinement as well as precipitation strengthening. Nb is also a scavenger of C and N from solution, which can improve ductility. Besides, being a substitutional solute, Nb offers solid solution strengthening, increases hardenability and promotes the formation of bainite and acicular ferrite microstructures. Thus, Nb shows a clear trend, i.e., a positive effect on all the investigated properties.

Vanadium: As a microalloying element, the primary contribution of V is precipitation strengthening in steels by the formation of numerous fine VC and V(C, N) precipitates during austenite to ferritic transformation. However, strong precipitation strengthening from V negatively affects ductility and toughness. Although VN particles in austenite can act as nucleation sites for intragranular ferrite grains, V, in general, is not a strong grain refiner. Thus, V demonstrates a positive response on strength but a negative response on ductility and impact toughness.

Titanium: As a microalloying element, the primary contribution of Ti is to restrict austenite grain growth during soaking, welding, and even conventional hot-rolling by the formation of stable TiN and Ti(C, N) precipitates. Such a grain size control can be beneficial for impact toughness. Similar to Nb, dissolved Ti improves hardenability and provides solid solution strengthening. Fine-scale precipitation of TiC (at relatively lower temperatures) can also offer precipitation strengthening. Recently there has been an emphasis on nanometer-sized (Ti, Mo)C precipitation strengthened ferritic steels for automotive applications. Precipitation strengthening can hamper ductility and toughness. Ti and N contents should be controlled carefully in steels as course and brittle TiN particles are the potent sites for cleavage crack initiation that can seriously hamper impact toughness and ductility. Thus, Ti shows a positive effect on strength, a negative effect on ductility, and a mixed response on impact toughness.

Silicon: As a substitutional solute, Si provides solid solution strengthening and shows a positive effect on strength (particularly UTS) and a negative effect on ductility. Si is preferred in steels containing low levels of C and other strengthening elements like Mn. That could result in a mixed response of Si on YS. Si decreases the cohesive strength of the atomic planes helping cleavage crack propagation and renders steel brittle, affecting impact toughness. In contrast, Si restricts the formation of iron-carbides (detrimental to roughness) and contributes to carbide free bainitic microstructures. As a result, Si does not show a clear trend on Charpy energy.

Manganese: Being present at a considerable amount (0.5 to 2.0 wt%), Mn is a strong solid solution strengthener in steels, which can naturally affect ductility. Mn is maintained at a higher side typically in low-C steels, having high ductility and impact toughness. Besides, Mn

stabilizes retained austenite which contributes to the TRIP (transformation induced plasticity) effect. Mn also suppresses pearlite formation and coarsen the interlamellar spacing in pearlite. Although MnS inclusion is detrimental to toughness, it is not as harmful as iron-sulfide. These combined factors resulted in a mixed response of Mn on ductility and a positive influence on impact toughness.

Phosphorous: Segregation of P at the grain boundaries reduces the boundaries' cohesive strength, and the associated embrittlement seriously affects ductility and impact toughness. Although P is a solid solution strengthener, its content is restricted in commercial grades of steel. It is only allowed when the levels of C and the other alloying elements are very low (like P strengthened interstitial free steel). That is possibly the reason behind P's negative effect on strength, as detected here.

Sulfur: S has a strong negative effect on steel properties, particularly on ductility and toughness. Coarse and elongated MnS inclusions initiate large voids and fissures (promoting ductile fracture), cause anisotropy in properties, and even act as the cleavage crack initiators. Segregation of S at grain boundaries and interdendritic regions and the formation of iron-sulfides can be even more detrimental for properties. Being softer than the steel matrix, MnS can reduce the strength when present at a high fraction. However, occasionally fine MnS particles can offer grain refinement by (i) pinning the grain boundaries, and (ii) VN precipitation on MnS can provide nucleation sites for intergranular ferrite within austenite grains. Hence, S showed a mixed response on the YS and a negative influence on other properties.

Nickel: Being a solid solution strengthener Ni is expected to improve the strength of steel. Although strengthening can negatively affect ductility and toughness, Ni is particularly beneficial in improving the low-temperature impact toughness, preventing ductile-to-brittle transition in ferritic steels. Being an austenite stabilizer, Ni can also enhance impact toughness through the TRIP effect of retained austenite. Ni may not be as effective in improving room-temperature toughness as low-temperature toughness. Besides being an expensive alloying element, Ni is typically added in special grades of heavily alloyed high-strength steels, which inherently have a low ductility and impact toughness (to restore these properties). Therefore, Ni showed a negative response to ductility and a mixed response to impact toughness.

Aluminium: Al is used for deoxidation and grain refinement in steel by the formation of Al2O3 and AlN, respectively. Al has a weak effect on hardenability, and it stabilizes the soft and ductile ferrite phase. The ability of Al to scavenge N from solution reduces the strengthening effect of N, which can improve ductility, toughness, and formability. The steel also becomes resistant to strain-ageing and the associated yield point phenomenon as desired in formable automotive grades of steel. AlN particles restrict grain growth and help achieve a fine grain size, which is beneficial for impact toughness. In bainitic steels, Al also retards carbide precipitation (iron-carbides are detrimental to toughness) which stabilizes retained austenite and contributes TRIP effect. Hence, Al showed a negative effect on strength but a positive response on ductility and impact toughness.

Table 6
Details of the surrogate modelling algorithms considered in this study.

Acronym used in the article	Details
ExTR	Ensemble method in Python package `scikit-learn` named ExtraTreesRegressor.
Ada	Ensemble method in Python package `scikit-learn` named AdaBoostRegressor.
Bagging	Ensemble method in Python package `scikit-learn` named BaggingRegressor.
GradBoost	Ensemble method in Python package `scikit-learn` named GradientBoostRegressor.
XGBoost	Ensemble method in Python package `xgboost` named XGBRegressor.
XGBRF	Ensemble method in Python package `xgboost` named XGBRFRegressor.
LightGBM	Ensemble method in Python package `lightgbm` named LGBMRegressor.
RandomForest	Ensemble method in Python package `scikit-learn` named RandomForestRegressor.
Kriging	Kriging method in Python package `scikit-learn` named GaussianProcessRegressor.
SVM1	Support vector machine in Python package `scikit-learn` named SVR.
SVM2	Linear support vector machine in Python package `scikit-learn` named LinearSVR.
NN	Neural networks in Python package `scikit-learn` named MLPRegressor.

Nitrogen: Although N is a strong solid solution strengthener, it significantly hampers ductility, toughness, and formability when in solution. Therefore, N in solution is minimized by the scavenging action of strong nitride forming elements such as Ti, Al, Nb and V. The grain refinement of microalloy nitride and carbo-nitride precipitates, along with the N free matrix, can improve ductility and toughness. Therefore, N does not show any trend on strength, mixed response on ductility, and positive influence on impact toughness.

Boron: B is used at a controlled quantity to increase the hardenability of special grades of steels having bainite or tempered martensite microstructures. In this study, B does not show a clear trend with any of the properties possibly due to the following reasons. (i) B in solution at only tens of ppm can be effective for enhancing hardenability. However, higher addition of B can be detrimental due to the formation of hard and brittle particles like BN, metal-borides and boro-sulfides. (ii) To prevent BN formation, B is shielded by the addition of stronger nitride formers such as Ti, Al and Zr, and those elements also influence the steel's properties. (iii) Finally, the data available on B containing steels is limited.

Chromium: Cr enhances hardenability, refines interlamellar spacing of pearlite, and provides solid solution strengthening and precipitation strengthening. Hence, Cr has a positive effect on strength. Coarse Cr23C6 precipitates, however, can impose a negative effect on ductility and toughness.

Cerium: Being a rare earth metal, Ce is occasionally added in steels to control the shape and size of sulfide and oxide inclusions. Ce in solution and its grain boundary segregation may impart some strength apart from inclusion refinement, which can benefit toughness. However, Ce is usually added in high-strength steels with low ductility and toughness. Ce addition needs to be carefully controlled and a high Ce level ($>$ 0.03 wt%) can be detrimental. Data availability on Ce containing steels is also limited. Hence, Ce shows a positive effect on strength but no clear trend on the other properties.

Copper: Cu has low solubility in ferritic steels. It precipitates out as metastable BCC which is coherent with the matrix and then transforms to incoherent FCC, increasing strength. The negative effect of Cu on steels is difficult to explain apart from the fact that being softer than the matrix, Cu precipitates may soften the steel at the onset of plastic deformation when present at a high fraction. High addition of Cu can result in its segregation at grain boundaries and surface regions and can hamper the ductility, particularly at high temperatures, causing hot cracking. FCC-Cu precipitates can restrict the ductile-to-brittle transition at the low test temperature. However, the present study could not detect its beneficial effect on impact toughness.

Zirconium: Being a rare and expensive alloying element, Zr is occasionally used in steels. Its affinity for O, S, and N is the primary reason behind Zr addition: controlling the non-metallic inclusions and scavenging N from solution (say, to protect B). Zr's oxides, sulfides, and nitrides can prevent grain growth at high temperatures. However, the beneficial effects of Zr could not be identified here possibly due to limited data availability of Zr steels.

Some of the elements mentioned in Table 2 are added to steels to improve certain properties beyond the present study's scope. For example, Cr, N and Cu are beneficial for corrosion resistance.

Appendix B. Acronyms used in the article

See Table 6.

References

Abbaschian, R., Reed-Hill, R.E., 2009. Physical Metallurgy Principles-SI Version. Cengage Learning.

Adabavazeh, Z., Hwang, W., Su, Y., 2017. Effect of adding cerium on microstructure and morphology of Ce-based inclusions formed in low-carbon steel. Sci. Rep. 7 (1), 1–10.

Bae, C., Lee, C., Nam, W., 2002. Effect of carbon content on mechanical properties of fully pearlitic steels. Mater. Sci. Technol. 18 (11), 1317–1321.

Baker, T., 2009. Processes, microstructure and properties of vanadium microalloyed steels. Mater. Sci. Technol. 25 (9), 1083–1107.

Bhadeshia, H.K., Hansraj, D., 2019. Bainite in Steels: Theory and Practice. CRC Press.

Bhadeshia, H., Honeycombe, R., 2017. Steels: Microstructure and Properties. Butterworth-Heinemann.

Branke, J., Deb, K., Miettinen, K., Slowinski, R. (Eds.), 2008. Multiobjective Optimization: Interative and Evolutionary Approaches. Springer.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Chakraborti, N., 2022. Data-Driven Evolutionary Modeling in Materials Technology. CRC Press.

Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 785–794.

Chen, M.-Y., Linkens, D., Bannister, A., 2004. Numerical analysis of factors influencing charpy impact properties of TMCR structural steels using fuzzy modelling. Mater. Sci. Technol. 20 (5), 627–633.

Cheng, R., Jin, Y., Olhofer, M., Sendhoff, B., 2016. A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Trans. Evol. Comput. 20 (5), 773–791.

Chugh, T., Jin, Y., Miettinen, K., Hakanen, J., Sindhya, K., 2016. A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. IEEE Trans. Evol. Comput. 22 (1), 129–142.

Chugh, T., Sindhya, K., Hakanen, J., Miettinen, K., 2019. A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. Soft Comput. 23 (9), 3137–3166.

DeArdo, A., 2003. Niobium in modern steels. Int. Mater. Rev. 48 (6), 371–402.

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 Trans. Evol. Comput. 18 (4), 577–601.

Deb, K., Saxena, D., 2006. Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In: Proceedings of the World Congress on Computational Intelligence. pp. 3352–3360.

Emmerich, M., 2005. Single-and multi-objective evolutionary design optimization assisted by Gaussian random field metamodels. University of Dortmund.

Erhart, H., Grabke, H.-J., 1981. Equilibrium segregation of phosphorus at grain boundaries of Fe–P, Fe–C–P, Fe–Cr–P, and Fe–Cr–C–P alloys. Met. Sci. 15 (9), 401–408.

Friedman, J.H., Greedy function approximation: A gradient boosting machine. Ann. Statist. 29 (5), 1189–1232.

Gardner, M.W., Dorling, S., 1998. Artificial neural networks (the multilayer perceptron)-a review of applications in the atmospheric sciences. Atmos. Environ. 32 (14–15), 2627–2636.

Geurts, P., Ernst, D., Wehenkel, L., 2006. Extremely randomized trees. Mach. Learn. 63 (1), 3–42.

Ghosh, A., Sahoo, S., Ghosh, M., Ghosh, R., Chakrabarti, D., 2014. Effect of microstructural parameters, microtexture and matrix strain on the Charpy impact properties of low carbon HSLA steel containing MnS inclusions. Mater. Sci. Eng. A 613, 37–47.

Gladman, T., 1997. The Physical Metallurgy of Microalloyed Steels. Maney Pub.

Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E., 2015. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. J. Comput. Graph. Statist. 24 (1), 44–65.

Guo, A., Li, S., Guo, J., Li, P., Ding, Q., Wu, K., He, X., 2008. Effect of zirconium addition on the impact toughness of the heat affected zone in a high strength low alloy pipeline steel. Mater. Charact. 59 (2), 134–139.

Heikkinen, R., Sipila, J., Ojalehto, V., Miettinen, K., Flexible data driven inventory management with interactive multiobjective lot size optimization. Int. J. Logist. Syst. Manage..

Isheim, D., Gagliano, M.S., Fine, M.E., Seidman, D.N., 2006. Interfacial segregation at Cu-rich precipitates in a high-strength low-carbon steel studied on a sub-nanometer scale. Acta Mater. 54 (3), 841–849.

Ishibuchi, H., Tsukamoto, N., Nojima, Y., 2008. Evolutionary many-objective optimization: A short review. In: Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). pp. 2419–2426.

Jin, Y., Wang, H., Sun, C., 2021. Data-Driven Evolutionary Optimization. Springer.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y., 2017. LightGBM: A highly efficient gradient boosting decision tree. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc..

Kim, N.J., 1983. The physical metallurgy of HSLA linepipe steels—a review. JOM 35 (4), 21–27.

Kim, S.-H., Kang, C.-Y., Bang, K.-S., 2001. Weld metal impact toughness of electron beam welded 9% Ni steel. J. Mater. Sci. 36 (5), 1197–1200.

Lancaster, J.F., 1999. Metallurgy of Welding, sixth ed Elsevier.

Li, Y., Crowther, D.N., Green, M.J.W., Mitchell, P.S., Baker, T.N., 2001. The effect of vanadium and niobium on the properties and microstructure of the intercritically reheated coarse grained heat affected zone in low carbon microalloyed steels. ISIJ Int. 41 (1), 46–55.

Li, K., Wang, R., Zhang, T., Ishibuchi, H., 2018. Evolutionary many-objective optimization: A comparative study of the state-of-the-art. IEEE Access 6, 26194–26214.

Matheron, G., 1963. Principles of geostatistics. Econ. Geol. 58 (8), 1246–1266.

Mesquita, R., Kestenbach, H.-J., 2011. On the effect of silicon on toughness in recent high quality hot work steels. Mater. Sci. Eng. A 528 (13–14), 4856–4859.

Miettinen, K., 1999. Nonlinear Multiobjective Optimization. Kluwer Academic Publishers.

Miettinen, K., Mäkelä, M.M., 2002. On scalarizing functions in multiobjective optimization. OR Spectrum 24 (2), 193–213.

Miettinen, K., Mäkelä, M.M., 2006. Synchronous approach in interactive multiobjective optimization. European J. Oper. Res. 170 (3), 909–922.

Miettinen, K., Ruiz, F., Wierzbicki, A.P., 2008. Introduction to multiobjective optimization: Interactive approaches. In: Branke, J., Deb, K., Miettinen, K., Slowinski, R. (Eds.), Multiobjective Optimization: Interative and Evolutionary Approaches. Springer, pp. 27–57.

Misitano, G., Saini, B.S., Afsar, B., Shavazipour, B., Miettinen, K., 2021. DESDEO: The modular and open source framework for interactive multiobjective optimization. IEEE Access 9, 148277–148295.

Morrison, W., 2009. Microalloy steels–the beginning. Mater. Sci. Technol. 25 (9), 1066–1073.

Nakayama, H., Sawaragi, Y., 1984. Satisficing trade-off method for multiobjective programming. In: Grauer, M., Wierzbicki, A.P. (Eds.), Interactive Decision Analysis. Springer, Berlin, Heidelberg, pp. 113–122.

Norström, L.-Å., Vingsbo, O., 1979. Influence of nickel on toughness and ductile-brittle transition in low-carbon martensite steels. Met. Sci. 13 (12), 677–684.

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

Reback, J., et al., 2020. Pandas-dev/pandas: Pandas, version latest. http://dx.doi.org/10.5281/zenodo.3509134.

Roy, S., Saini, B.S., Chakrabarti, D., Chakraborti, N., 2020. Mechanical properties of micro-alloyed steels studied using a evolutionary deep neural network. Mater. Manuf. Process. 35 (6), 611–624.

Rozhkova, E., Garber, M., Tsypin, I., 1981. Effect of manganese on the transformation of austenite in white chromium cast irons. Metal Science and Heat Treatment 23 (1), 59–63.

Ruiz, F., Luque, M., Cabello, J.M., 2009. A classification of the weighting schemes in reference point procedures for multiobjective programming. J. Oper. Res. Soc. 60 (4), 544–553.

Ruiz, F., Luque, M., Miettinen, K., 2012. Improving the computational efficiency in a global formulation (GLIDE) for interactive multiobjective optimization. Ann. Oper. Res. 197 (1), 47–70.

Ruiz, A.B., Sindhya, K., Miettinen, K., Ruiz, F., Luque, M., 2015. E-NAUTILUS: a decision support system for complex multiobjective optimization problems based on the NAUTILUS method. European J. Oper. Res. 246 (1), 218–231.

Saini, B.S., Hakanen, J., Miettinen, K., 2020. A new paradigm in interactive evolutionary multiobjective optimization. In: Back, T., Preuss, M., Deutz, A., Wang, H., Doerr, M., Trautmann, H. (Eds.), Parallel Problem Solving from Nature, PPSN XVI, 16th International Conference, Part II. Springer, pp. 243–256.

Steinwart, I., Christmann, A., 2008. Support Vector Machines. Springer.

Tabatabaei, M., Hakanen, J., Hartikainen, M., Miettinen, K., Sindhya, K., 2015. A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods. Struct. Multidiscip. Optim. 52 (1), 1–25.

Uggowitzer, P.J., Magdowski, R., Speidel, M.O., 1996. Nickel free high nitrogen austenitic steels. ISIJ Int. 36 (7), 901–908.

Vervynckt, S., Verbeken, K., Lopez, B., Jonas, J., 2012. Modern HSLA steels and role of non-recrystallisation temperature. Int. Mater. Rev. 57 (4), 187–207.

Wang, H., Jin, Y., Sun, C., Doherty, J., 2018. Offline data-driven evolutionary optimization using selective surrogate ensembles. IEEE Trans. Evol. Comput. 23 (2), 203–216.

Wilson, F., Gladman, T., 1988. Aluminium nitride in steel. Int. Mater. Rev. 33 (1), 221–286.

Yan, W., Shan, Y., Yang, K., 2006. Effect of TiN inclusions on the impact toughness of low-carbon microalloyed steels. Metall. Mater. Trans. A 37 (7), 2147–2158.

Ye, D., Li, J., Jiang, W., Su, J., Zhao, K., 2012. Effect of Cu addition on microstructure and mechanical properties of 15% Cr super martensitic stainless steel. Mater. Des. 41, 16–22.