Journal of Industrial and Management Optimization

https://www.aimspress.com/journal/jimo

JIMO, 22(1): 256–281.
DOI:10.3934/jimo.2026010
Received: 30 July 2025
Revised: 19 September 2025
Accepted: 09 October 2025
Published: 21 November 2025

Research article

A multi-objective framework based on estimation of distribution algorithms for data-driven fuel experimental design

Vicente P. Soloviev[1,*], Pedro Larrañaga[1], Marco Bernabei[2], Marina A. Chirita[2], Jose M. Seoane[2], Pedro Fontán[2] and Concha Bielza[1]

[1] Universidad Politécnica de Madrid, Madrid, Spain

[2] Repsol Technology Lab., Madrid, Spain

* **Correspondence:** Email: vicente.perez.soloviev@upm.es.

Abstract: Fuel design is a critical field of energy production, demanding precision, efficiency, and safety. Hence, laboratory experimentation is a fundamental step for finding the best formula that can tackle further constraints. Given that laboratory experimentation is costly and time-consuming, there is a compelling need for an efficient approach to undertake it This challenge has led to the development of intelligent methods, instead of using the traditional alternatives in sampling methods over the cost function. In this article, we propose a pioneering approach based on multi-objective optimization using estimation of distribution algorithms to find the optimal fuel formula. This approach suggests the set of experiments that should be carried out in the laboratory. We went one step further allowing experts to set those ingredients that should take a fixed value in the formulation and also controlling the exploration/exploitation trade-off of the optimizer. The resulting output for the experts is a diverse set of experiments to carry out. Our findings unveiled a spectrum of results across varying levels of exploration. When compared with some state-of-the-art optimizers, our estimation of distribution algorithm approach outperforms them. Moreover, we identified several ingredients, traditionally not considered by the experts, whose presence in the total mixture has a major impact on the fuel performance. This enriches the overall interpretability of both our methodology and the underlying fuel design challenge.

Keywords: estimation of distribution algorithms; data-driven; experimental design; Bayesian network; environment variables
Mathematics Subject Classification: 90C29, 62K05

1. Introduction

The United Nations' global goal of net zero emissions by 2050 [1, 2] is demanding an increase in the production of high-performance fuels from sustainable feed-stocks in order to reduce dependence on petroleum-based products and mitigate the environmental footprint. Considering the small time-window available to achieve the net-zero emission goal, the design of a new generation of fuels that can improve sustainability by reducing carbon dioxide emissions can benefit from the speed-up provided by a holistic approach that integrates artificial intelligence and laboratory experiments. Fuel design is a black-box optimization task, where the result of the target properties of a proposed fuel formulation are most of the time unknown until experiments are performed, and can seldom be estimated (predicted or analytically imputed, depending on the problem). Thus, smart fuel design relies on methodologies that propose promising formulations and analyze past experiments to improve future proposals, minimizing the number of experiments in the real laboratory. Traditional experimental design methods involve modeling and sampling with the aim of exploring unknown regions of the design landscape. Some examples use factorial design [3], in which linearity between variables is assumed and samplings are located at the orthogonal corners of the landscape. There are more complex methods such as Latin hypercubes sampling with general purpose space-filling designs [4, 5], and response surface methodologies [6], where polynomials are used to model the search space. However, the main drawback of these approaches is that once sampled, most of the solutions are unnecessarily evaluated, and the knowledge extracted from the promising ones is not incorporated into future samplings, as these are not iterative processes. Data-driven design of experiments (DoE) is a powerful approach that leverages statistical and machine learning techniques to optimize the design and execution of experiments. By leveraging previous data provided by experts and computational models, data-driven DoE aims at identifying the most informative and efficient experimental configurations, thus minimizing the number of experiments required to obtain a fuel with the desired properties while maximizing the insights gained. This methodology allows researchers and scientists to strategically select experimental conditions, variables, and sample sizes, leading to more accurate and robust conclusions from the available data. It has become instrumental in a wide range of scientific disciplines, including chemistry, biology, physics, and engineering, ultimately accelerating the pace of knowledge discovery and innovation. Data-driven DoE becomes even more complex when: (i) some of the variables involved in the historical data must be set to a specific value, as determined by the experts [7]; (ii) there are several constraints on the variables; or (iii) there is no historical data available from which to learn. Recent approaches include complex simulation methodologies such as adaptive sampling [8] in which promising regions of the search space are detected and sampled iteratively. Adaptive sampling has been combined with Bayesian optimization and widely used as a data-driven DoE approach. This leverages a surrogate model that can be sampled efficiently and an acquisition function which is used to select promising solutions from the search space [9–11]. Other approaches found in the literature face the task as a multi-level optimization problem, where evaluating a given solution involves solving further optimization tasks [12], although this design decision is problem dependent. However, all these approaches act as gray boxes where all variables are of equal importance and it is not possible to find knowledge of the relationships between variables. Additionally, some research has advanced multi-objective optimization through hybrid and learning-enhanced methods. Knowledge-based and reinforcement

learning approaches have been applied to scheduling problems [13–15], while novel metaheuristics such as the multi-objective chimp optimizer and its variants have been explored in diverse engineering domains including energy systems, structural design, and truss optimization [16–21]. These studies emphasize the potential of combining probabilistic modeling, domain knowledge, and adaptive mechanisms to improve efficiency and scalability, providing a foundation for extending such ideas to data-driven fuel design. These types of problems are often constrained by the need to balance physical feasibility, computational cost, and conflicting objectives. Several studies have demonstrated these challenges across diverse applications, including ammonia-fueled micro-planar combustors [22, 23], liquid cooling plates for battery thermal management [24–26], and proton exchange membrane fuel cells [27]. Collectively, these works highlight how limitations such as manufacturability, stability, flow distribution, and computational efficiency shape the optimization domain and the effectiveness of DOE-based procedures. Estimation of distribution algorithms (EDAs) [28] belong to the family of evolutionary algorithms (EAs) [29] but, unlike traditional EAs that use genetic operators, EDAs focus on building and updating a probabilistic model that represents the current selected individuals of the search space and captures the dependencies between variables. By iteratively sampling new candidate solutions from the learned model, EDAs effectively explore the search space and guide the search toward promising regions. Different variants of this algorithm depending on the probabilistic model embedded in the optimization are available in the literature, some of which include Bayesian networks (BNs) [30]. EDAs have been successfully applied in a wide range of problems [7, 31–33] where they have demonstrated competitive performance. Continuous multi-objective optimization deals with the simultaneous optimization of multiple objectives $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})$ in the context of continuous decision variables, where some of them might be conflicting. Multi-objective optimization aims at identifying a set of solutions that represent the trade-offs between different objectives, i.e., the best Pareto frontier approximation. Then, a continuous multi-objective optimization problem is defined as

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & F(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \\ & \text{subject to } \boldsymbol{x} \in \mathbb{R}^n \end{aligned} \tag{1.1}$$

where m and n are the number of objectives and variables involved in the problem, respectively, and the optimization criterion is maximization. Multi-objective optimization has been widely researched using EAs in the literature [29, 34] for different applications such as portfolio management [35], data mining [36], or feature subset selection [37], among others.

In this paper, we present a new data-driven DoE approach aimed at obtaining an optimal multi-ingredient fuel exploiting previous experimentation provided by experts. The design of the fuel is modeled as a black-box multi-objective optimization problem with $m = 3$, in which one of the objectives ($f_1(\boldsymbol{x})$) is estimated using an embedded regression model, while others ($f_2(\boldsymbol{x}), f_3(\boldsymbol{x})$) are computed analytically. Moreover, several bounds are imposed regarding the fuel descriptors which lead to a constrained optimization problem. Starting from an initial set of experiments, provided by the experts, the EDA approach converges to a new set of experiments to be evaluated in a laboratory, where the real properties are experimentally measured. The new data points are fed back in the framework loop and the EDA is newly executed to further explore the Pareto front in search of the optimal solutions. By the use of Gaussian Bayesian networks, our approach is able to restrict the search space and propose feasible solutions for the optimization problem. The level of

exploration/exploitation over the initial set of experiments is regulated by the introduction of a new parameter, which is experimentally analyzed. A posteriori analysis is performed, in which relevant variables during the optimization process are identified. EDAs were employed in this problem because they explicitly model the probabilistic distribution of promising solutions, which enhance search efficiency and diversity preservation compared to conventional evolutionary operators.

Thus, the significant contributions of this work are:

- A novel integration of multi-objective EDAs with Gaussian Bayesian networks for data-driven DoE in fuel design.
- The enforcement of the probabilistic model with an embedded regression model, improving the accuracy of optimization under partial knowledge.
- The introduction and systematic analysis of a parameter controlling the exploration–exploitation trade-off, enabling flexible adaptation to different data scenarios.
- An a posteriori interpretability analysis that identifies dependencies between fuel variables, offering additional domain knowledge beyond the optimization results.

Compared to previous studies that treat fuel design as a black-box optimization task or apply Bayesian optimization without interpretability, our approach provides both more efficient convergence to high-performing solutions and valuable knowledge discovery about the underlying variable relationships.

The rest of the paper is organized as follows. In Section 2, we present the formal formulation of the black-box multi-objective optimization problem. In Section 3, we provide a theoretical background for EDA approaches. The proposed methodology for the optimization task and further results are presented in Section 4. Section 5 rounds out the paper with conclusions and future research lines.

2. Optimizing the design of fuel

Fuel is a physical mixture of several ingredients of different chemical nature and is normally defined in terms of the relative weight percentage of each ingredient in the formulation.

By starting from an initial pool of n available ingredients, the design of a fuel for a specific application requires one to efficiently explore an n-dimensional search space in order to find optimal combinations of ingredients, whose physical properties must meet a set of constraints.

In this work we are dealing with $n = 24$ ingredients $I_1, \ldots, I_{24}$ that can be grouped into six different categories $\{A, B, C, D, E, F\}$ and each ingredient is characterized by a set of physical properties $P1, \ldots, P14$, as shown in Supplementary materials. Supplementary materials shows a histogram for each of the properties.

Additionally, the total fuel mixture is characterized by a set of descriptors, namely $W = (W_{calc}, W_{lab})$. The former (W_{calc}) includes the percentage of some of the ingredients, and certain physical properties that can be computed analytically as weighted sums of the individual properties of the ingredients. The latter (W_{lab}) refers to physical properties that cannot be computed analytically, and have to be evaluated experimentally in the laboratory. Both types of descriptors are bounded by a lower (LB) and upper (UB) bound, respectively, leading to optimization constraints. Supplementary materials describes W_{calc} formulas, and bounds for all the descriptors.

Although the properties of the ingredients are known, their interaction as well as how they affect the laboratory descriptors (W_{lab}) is not known a priori. The optimizer proposed in this work not only allows one to efficiently navigate the search space to accelerate the design of optimal fuels but also provides useful insights into the effect of each ingredient over the descriptors of interest.

From the set of descriptors, W^1_{lab}, W^1_{calc}, and W^2_{calc} are desired to be maximized in the multi-objective optimization problem.

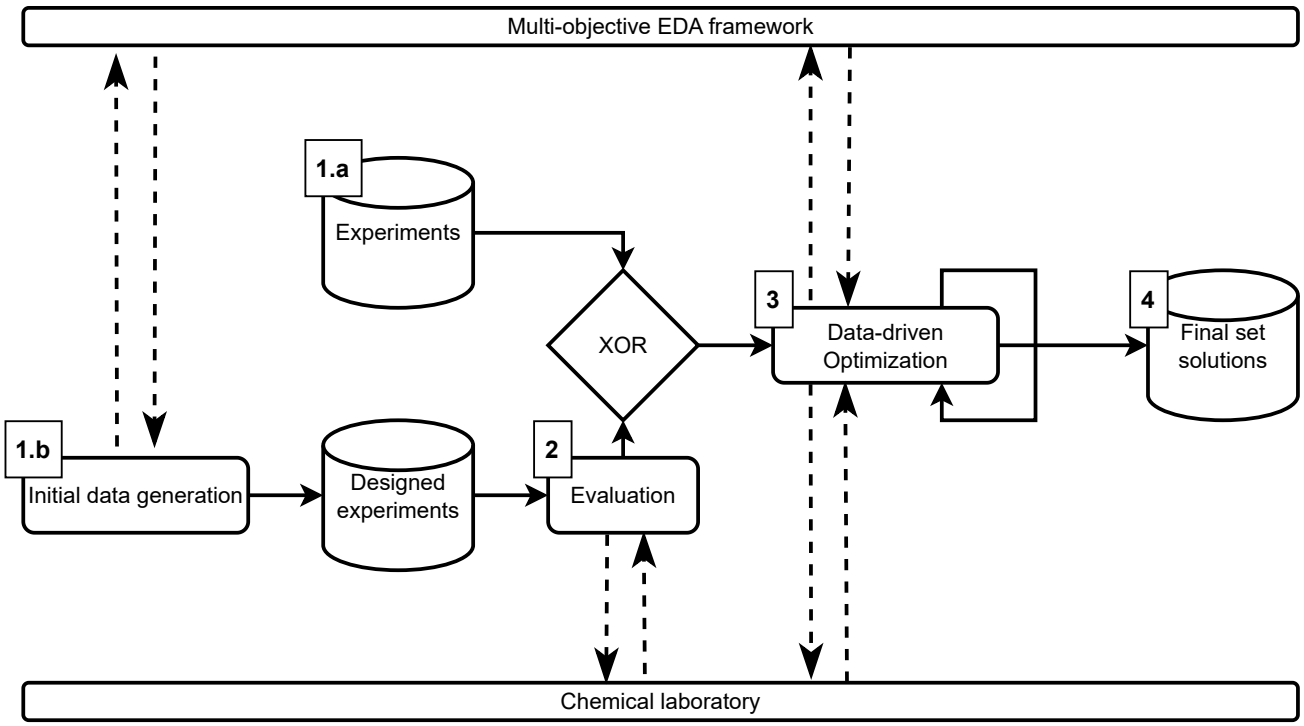

Figure 1. General flow chart of the multi-objective proposed approach for data-driven DoE.

Figure 1 depicts a schematic view of the multi-objective sequential learning framework for fuel design presented in this paper. Considering the high dimension of the search space and that laboratory experiments are time consuming and subjected to budget restriction, it is advantageous to assist the fuel design task with a multi-objective optimization algorithm that efficiently and iteratively explores the Pareto frontier guiding the search toward an optimal fuel that meets all the previously mentioned constraints.

By leveraging past experiments (Step 1.a), our methodology aims at finding a sweet spot between testing new fuels with different compositions from those already tested (exploration) and exploiting the knowledge enclosed in the existing data points to suggest a batch of new formulations to be tested in the laboratory. Once these new fuels are tested in the laboratory, if none of the suggestions meets the constraints, the optimization procedure is run again to suggest a new batch of experiments (Step 3) until convergence to a final set of optimal solutions is achieved (Step 4). Additionally, if no past experiments are provided, we propose a method based on a Latin hypercube sampling for proposing an initial set of experiments (Step 1b, Section 4.4) to be tested in the laboratory (Step 2). The data-driven optimization process is carried out starting from either a set of historical data or the set of proposed experiments (XOR, only one input is chosen).

The proposed approach also allows including domain knowledge to further guide or restrict the search for optimal solutions. For example, experimentalists can be interested in fixing the amount of certain ingredients to either 0 or a finite value between 0 and 100, and no solutions should suggest different values for those ingredients whose amount has been fixed. Previous research in this line has referred to these variables as environment variables [7].

2.1. Problem formulation

The problem involves $n = p + c$ variables, $\boldsymbol{X} = (\boldsymbol{Y}, \boldsymbol{Z}) = (Y_1, \ldots, Y_p, Z_1, \ldots, Z_c) = (X_1, X_2, \ldots, X_n)$, which represent percentages of each ingredient in the total mixture of the fuel, where $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are the environment variables whose values ($\boldsymbol{y}$) are predefined by the experts and the decision variables ($\boldsymbol{Z}$) that are to be tuned by the algorithm, respectively. The optimization problem is defined as

$$
\begin{aligned}
\max_{\boldsymbol{x}} \quad & F(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), f_3(\boldsymbol{x})) \\
& f_1(\boldsymbol{x}) = W^1_{lab}(\boldsymbol{x}) - i(\boldsymbol{x}) \\
& f_2(\boldsymbol{x}) = W^1_{calc}(\boldsymbol{x}) - i(\boldsymbol{x}) \\
& f_3(\boldsymbol{x}) = W^2_{calc}(\boldsymbol{x}) - i(\boldsymbol{x}) \\
& i(\boldsymbol{x}) = \sum_{i=1}^{4} j(\boldsymbol{x}, W^i_{lab}(\boldsymbol{x})) + \sum_{i=1}^{7} j(\boldsymbol{x}, W^i_{calc}(\boldsymbol{x})) \\
& j(\boldsymbol{x}, W^i_{lab}(\boldsymbol{x})) = \begin{cases} 0 & \text{if } \mathrm{LB}^i_{lab} \le W^i_{lab}(\boldsymbol{x}) \le \mathrm{UB}^i_{lab}, \\ \delta & \text{otherwise,} \end{cases} \\
& j(\boldsymbol{x}, W^i_{calc}(\boldsymbol{x})) = \begin{cases} 0 & \text{if } \mathrm{LB}^i_{calc} \le W^i_{calc}(\boldsymbol{x}) \le \mathrm{UB}^i_{calc}, \\ \delta & \text{otherwise,} \end{cases} \\
\text{subject to} \quad & \boldsymbol{x} \in \mathbb{R}^n_+, \\
& L(\boldsymbol{x}|\mathbb{M}) + \alpha \le 0, \\
& \sum_{i=1}^{n} x_i = 100,
\end{aligned}
$$

where $L(\boldsymbol{x}|\mathbb{M})$ is the log-likelihood of a solution $\boldsymbol{x}$ in the probabilistic model $\mathbb{M}$ learned from the initial data; $\alpha \ge 0$ represents the level of exploration-exploitation, with a higher value representing a more exploratory environment; $\delta > 0$ represents the penalization term for the constraint set in the problem; $\mathbb{M}$ is the model pre-trained with the historical data; and $(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), f_3(\boldsymbol{x}))$ are the three objectives related to the $(W^1_{lab}(\boldsymbol{x}), W^1_{calc}(\boldsymbol{x}), W^2_{calc}(\boldsymbol{x}))$ functions to optimize, respectively.

3. Background

3.1. Estimation of distribution algorithms

EDAs differ from traditional EAs in the fact that they do not use mutation and crossover operators to generate future promising solutions for a given task. Instead, they update the probabilistic model in each iteration, learned from the selected individuals and used for generating valid solutions in each iteration.

Algorithm 1 EDA baseline

Input: Population size λ, selection ratio τ, and cost function $F(\cdot)$
Output: Best individual $\boldsymbol{x}'$ and cost found $F(\boldsymbol{x}')$

1: $G_0 \leftarrow \lambda$ individuals randomly sampled
2: **for** $t = 1, 2, ...$ until stopping criterion is met **do**
3: Evaluate G_{t-1} according to $F(\cdot)$
4: $G^S_{t-1} \leftarrow$ Select $\lfloor \tau\lambda \rfloor$ individuals from G_{t-1}
5: $p_{t-1}(\cdot) \leftarrow$ Learn a probabilistic model from G^S_{t-1}
6: $G_t \leftarrow$ Sample λ individuals from $p_{t-1}(\cdot)$
7: **end for**

Algorithm 1 describes the EDA baseline, where an initial set of solutions is randomly sampled (Step 1) or can be provided by the user. Then, the EDA iteratively evaluates the set of solutions according to a cost function to be optimized ($F(\cdot)$) given by the user (Step 3), and selects the top $\lfloor \tau\lambda \rfloor$ individuals according to $F(\cdot)$ maximization (Step 4). A probabilistic model is learned in each iteration (Step 5), from which new solutions are sampled (Step 6). The algorithm iterates over the loop until the stopping criterion is met.

This paper focuses on the continuous domain application of EDAs. Thus, we can distinguish between single-objective and multi-objective approaches in the literature.

Regarding single-objective optimization and depending on the complexity of the probabilistic model embedded in the framework, there exist different variants. Some of them are the: (i) univariate marginal distribution algorithm [38], which in each iteration fits an independent Gaussian for each variable, and thus does not consider dependencies between the variables; (ii) estimation of Gaussian distribution algorithm [39], which in each iteration learns a Gaussian Bayesian network [30]; (iii) semiparametric estimation of distribution algorithm [40], which in each iteration allows the algorithm itself to decide whether to use parametric or non-parametric probability distributions; and (iv) multivariate kernel estimation of distribution algorithm [40], which in each iteration learns non-parametric probability distributions.

Regarding multi-objective EDAs, it is worth mentioning the multi-objective Bayesian optimization algorithm [41] in which a multivariate EDA, called the Bayesian optimization algorithm (BOA) [42], and the non-dominated sorting genetic algorithm (NSGA-II) [43] are combined; the multi-objective hierarchical BOA [44], which modifies the previous algorithm by identifying promising solutions using clustering; and the multi-objective estimation of distribution algorithm [34], in which BNs are used to capture the dependencies between the decision variables and the objectives to be optimized.

This type of algorithm provides many advantages such as interpretability and ease of implementation. The probabilistic model learned at each iteration represents the landscape of solutions, where its uncertainty is reduced along the runtime, converging to a region where the best solution is found. Depending on the probabilistic model chosen, it can incorporate expert knowledge considered when generating new potential solutions. The EDA is able to return a set of feasible solutions with similar cost, so that the experts can analyze different alternatives for the given task.

3.2. Gaussian Bayesian networks

BNs [30] are a type of probabilistic graphical model that jointly represents the probability distribution of a set of random variables. A wide range of applications in the area of machine learning [45] are reviewed in the literature [46, 47] with different advantages such as their ability to represent uncertain knowledge underlying in the data, and being able to easily add knowledge from experts in a specific field. Moreover, they can be used as interpretable models [48] to perform sensitivity analysis and probabilistic reasoning.

Mathematically, BNs are defined as a pair $(G, \boldsymbol{\Theta})$ over a set of random variables $\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$, composed of a directed acyclic graph (DAG) $G = (V, A)$, where V are the variables in $\boldsymbol{X}$ represented as nodes in the DAG, and A are the arcs between the nodes. $\boldsymbol{\Theta}$ represents the dependency between variables defined as the conditional probability distribution of each variable X_i given its parents $\boldsymbol{Pa}_i$ in G, where the parents of a variable X_i are those which in the DAG are directly connected to X_i and pointing to it.

Regarding continuous variables, nodes are usually assumed to follow a Gaussian distribution. In Gaussian BNs, a conditional probability density of a variable is defined as a linear Gaussian model:

$$p(X_i|\boldsymbol{Pa}_i) = \mathcal{N}(\beta_{i0} + \beta_{i1} Pa_{i_1} + \cdots + \beta_{ik(i)} Pa_{i_{k(i)}};\ \sigma_i^2), \tag{3.1}$$

where β_{i0} is the intercept, $\beta_{i1}, \ldots, \beta_{ik(i)}$ are the weights associated to each of the variables in $\boldsymbol{Pa}_i$ with size $k(i)$, and σ_i^2 is the variance of the model.

Then, the joint distribution over $\boldsymbol{X}$ is factorized as

$$p(\boldsymbol{X}) = p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(X_i|\boldsymbol{Pa}_i) = \prod_{i=1}^{n} \mathcal{N}(\beta_{i0} + \sum_{j=1}^{k(i)} \beta_{ij} Pa_{i_j};\ \sigma_i^2)$$

Gaussian BNs are usually chosen for continuous data as they provide speed for fitting the data, ease of implementation, and the existence of closed formulas for performing the inference in the model.

3.3. Markov blanket

The Markov blanket $\mathbf{MB}(X_i)$ of a variable X_i in the BN is the minimal set of variables such that X_i is conditionally independent of all the other variables given $\mathbf{MB}(X_i)$. $\mathbf{MB}(X_i)$ represents the strongly related variables to X_i when data follow a stable distribution [49], and includes parents, children, and spouses (parents of children) of X_i found in G. Figure 2 shows an illustrative example of the Markov blanket of a variable in a BN structure.

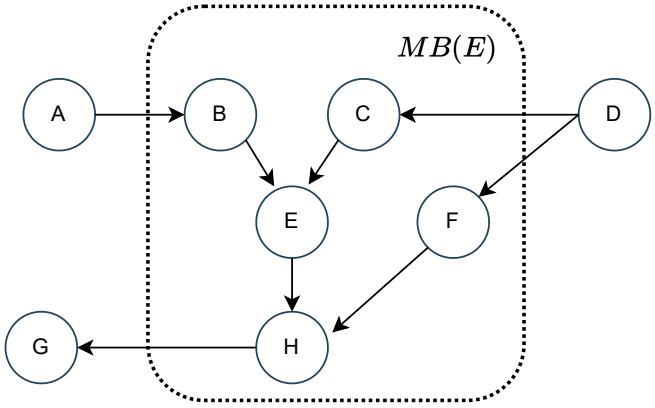

Figure 2. Illustrative example of the set of variables included in the Markov blanket of variable E, which includes the parents, children, and parents of children of E.

3.4. Multi-objective optimization

Optimizing a multi-objective problem (Equation 1.1) involves finding the sweet spot between the m objectives to be optimized. Identifying those promising solutions involves approximating the Pareto frontier. A solution is part of the Pareto frontier (or a solution is Pareto optimal) if there is no other feasible solution that can improve any objective without sacrificing the performance of at least one other objective.

For a maximization problem and a set $\mathcal{X}$ of solutions $\boldsymbol{x}'$, a solution $\boldsymbol{x}^*$ is part of the Pareto frontier if

$$f_i(\boldsymbol{x}^*) \geq f_i(\boldsymbol{x}') \text{ for all } i = 1, \ldots, m \text{ and for all } \boldsymbol{x}' \in \mathcal{X},$$

where $f_i(\boldsymbol{x})$ is the evaluation of solution $\boldsymbol{x}$ in the objective function i. Thus, the condition essentially states that no other solution from $\mathcal{X}$ can make all objectives better than those of Pareto frontier solutions $\boldsymbol{x}^*$.

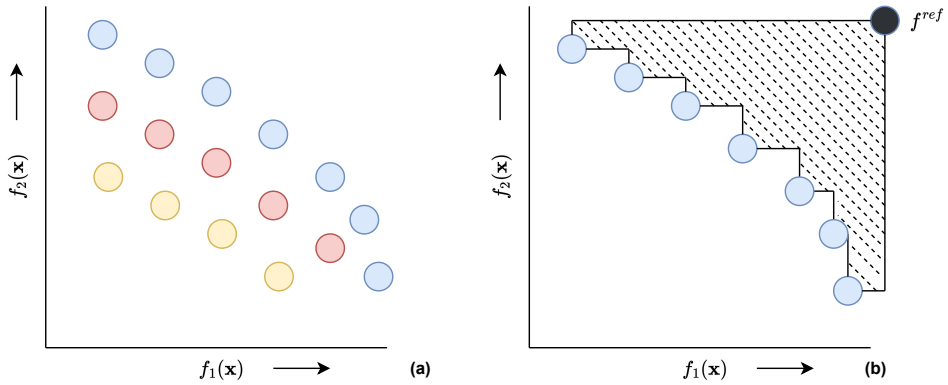

Figure 3. Panel (a) shows first, second and third Pareto frontiers represented as blue, red and yellow circles, respectively, for maximizing $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ objectives simultaneously. Panel (b) shows the hypervolume (HV) indicator computation for the first Pareto frontier and the reference point f^{ref}.

The Pareto frontier is identified in Figure 3(a) with blue circles, among all the possible solutions, where $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ are to be maximized. The second Pareto frontier is identified with red circles,

and defined as the Pareto frontier if the first Pareto frontier (blue circles) was removed. None of the red samples maximizes both objectives better than any of the blue set of samples. The same happens with the third frontier, represented with yellow circles, and defined as the Pareto frontier if red and blue samples are removed.

Quality indicators in multi-objective optimization are quantitative measures used to assess the performance and characteristics of solutions generated by multi-objective optimization algorithms [50].

For the optimization problem to be solved in this article, we use the hypervolume (HV) and diversity metric (DM) to evaluate the final results and compare against other approaches.

The HV measures the volume of the objective space that is dominated by a set of solutions. It quantifies how well a set of solutions covers the entire Pareto front. A higher hypervolume indicates a better spread of solutions. The HV of a set $\boldsymbol{S}$, given a reference point $f^{ref} = (f_1^{ref}, f_2^{ref}, \ldots, f_m^{ref})$, is the volume of the union of the hypercubes determined by each of its solutions $\boldsymbol{s} \in \boldsymbol{S}$ and f^{ref},

$$HV(\boldsymbol{S}, f^{ref}) = \Lambda(\bigcup_{\boldsymbol{S}} \{[f_1(\boldsymbol{s}), f_1^{ref}] \times \cdots \times [f_m(\boldsymbol{s}), f_m^{ref}]\}), \tag{3.2}$$

where f_i^{ref} refers to the reference ideal point for objective function f_i and $\Lambda(\cdot)$ refers to the Lebesgue measure. Figure 3(b) illustrates an example of the HV computation over the first Pareto frontier for $m = 2$ objectives.

Minimizing hypervolume relative to a reference point is equivalent to maximizing the dominated volume from the origin, and using a reference point near expected objectives focuses the search on relevant solutions, improves convergence, and enables more meaningful algorithm comparisons [51].

The DM quantifies the dissimilarity or spacing between solutions in a given set. It can be measured with different geometric distances or using the crowding distance (CD), which measures how densely solutions are distributed along the Pareto front in terms of the objectives. This metric characterizes a set of solutions and, depending on the optimization task to be solved, its optimization may vary. The CD of two solutions $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is defined as

$$CD(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sum_{k=1}^{m} (f_k(\boldsymbol{x_i}) - f_k(\boldsymbol{x_j}))^2, \tag{3.3}$$

where $f_k(\boldsymbol{x_i})$ and $f_k(\boldsymbol{x_j})$ are the objective function values of $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ for the objective f_k, respectively.

Then, we define the DM over a set of solutions $\mathcal{X}$ as

$$DM(\mathcal{X}) = frac1|\mathcal{X}|(|\mathcal{X}| - 1)sum_{\substack{\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X} \\ \boldsymbol{x}_i \neq \boldsymbol{x}_j}} CD(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{3.4}$$

where $|\mathcal{X}|$ is the number of solutions in $\mathcal{X}$.

4. Methods and results

In this section we deeply explain the methodology implemented to solve the task described in Section 2. Figure 1 outlines the framework used in this task. The pseudocode in Algorithm 2 shows the algorithm behind the multi-objective EDA framework described in Figure 1.

Algorithm 2 Multi-objective data-driven EDA

Input: Population size λ, selection ratio τ, cost functions (f_1, f_2, f_3), exploration ratio α, and historical data $\mathcal{X}_{train}$

Output: Best individual $\boldsymbol{x}'$ and cost found $F(\boldsymbol{x}')$

1: $G_0 \leftarrow \mathcal{X}_{train}$ historical data
2: **for** $t = 1, 2, ...$ until stopping criterion is met **do**
3: Evaluate G_{t-1} according to (f_1, f_2, f_3)
4: Rank G_{t-1} according to Pareto dominance for (f_1, f_2, f_3)
5: $G^S_{t-1} \leftarrow$ Select $\lfloor \tau\lambda \rfloor$ individuals from G_{t-1}
6: $p_{t-1}(\cdot) \leftarrow$ Learn a Gaussian BN from G^S_{t-1}
7: $p_{t-1}(\cdot) \leftarrow$ Set environment variables $\boldsymbol{Y} = \boldsymbol{y}$
8: $G_t \leftarrow$ Sample λ individuals from $p_{t-1}(\boldsymbol{Z}|\boldsymbol{Y} = \boldsymbol{y})$
9: **end for**

In each iteration of the general workflow (Figure 1), the EDA approach (MO_EDA) is executed and receives new historical data $\mathcal{X}_{train}$, which includes the historical data used in the previous iteration as well as the new batch of solutions suggested and validated in the laboratory. The exploration ratio α is tuned in each iteration of the workflow as desired by the experts.

The EDA approach is an iterative algorithm that evaluates a set of individuals (line 3) in each iteration, starting from the historical data, according to (f_1, f_2, f_3) (Section 2 and Section 4.1). The solutions are ranked based on the Pareto dominance criteria for the cost functions (line 4 and Section 4.3). From this ranking, $\lfloor \tau\lambda \rfloor$ solutions are selected (line 5) to train the posterior conditional probability described in Section 4.2 (lines 6–7) and sample λ new solutions (line 8).

Table1 describes the chosen parameters used for the experimentation setup.

Table 1. Parameters and settings for the proposed strategy.

Parameter	Value
Population size (λ)	500
Maximum iterations	100
Stagnation threshold	20
Learning rate (α)	0.6

The validity of the proposed problem formulation and the corresponding algorithm presented are ensured through comparison with benchmark cases reported in the literature and by verifying consistency between predicted trends and established physical principles.

The following sections describe a single iteration of the proposed approach illustrated in Figure 1. In this process, a subset of experiments is first defined by the experts, after which one iteration of Step 3 is performed. The multi-objective optimization is then executed to identify a new subset of promising candidate solutions for subsequent analysis in the chemical laboratory.

4.1. Prediction of W^1_{lab}

As described in Section 2, one of the objectives to be maximized (W^1_{lab}) cannot be computed analytically, and thus, has to be predicted based on the rest of the variables. Therefore, a regression

model has been implemented to predict W^1_{lab}. Note that this regression becomes part of a bigger framework, and improving the accuracy of the regression model is out of the scope of this paper; however, several models have been tested in order to select the best performance so far. The sampling provided from the Gaussian BN is replaced by the prediction made by the regression model.

Considering the target variable W^1_{lab} found in the historical data provided by the experts, we train different regression models, where the input variables are the decision variables $\boldsymbol{Z}$ (the set of fuel ingredients in the feature space to be tuned by the EDA approach). We prepare a leaving-one-out cross validation experiment to find the model that most accurately predicts W^1_{lab}. The tested models [52] are Bayesian ridge regression, Lasso regression, ridge regression, XGBoost regression, and kernel ridge regression, all of which are available at scikit-learn-1.3.0 [52]. Hyperparameter configurations for these algorithms can be found in Supplementary materials.

Table 2 shows the *R2*[*], mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE). It can be observed that Bayesian ridge regression obtains the best results, while XGBoost performs the worst.

The regression models are compared not only in terms of accuracy metrics, but also regarding their relative strengths and limitations. The selected model was chosen because it achieved the best trade-off between predictive accuracy and generalization capability, providing stable results across the tested scenarios while avoiding overfitting observed in other candidates.

In order to improve the performance of the regression model, we have carried out a hyperparameter tuning using Bayesian optimization [53], and *R2* results have increased to 60%. The final configuration is available in 5. Analyzing all the computed metrics, the model has shown to detect the relations between the variables in order to suggest good predictions and improve the EDA performance.

Table 2. *R2*, MAE, MSE, and RMSE results of leaving-one-out cross validation experiments after predicting W^1_{lab} using $\boldsymbol{Z}$ as input variables.

	BRidge	Lasso	XGBoost	Ridge	KRidge
R2 (%)	0.57	0.55	0.39	0.53	0.47
MAE	1.47	1.51	1.95	1.58	1.75
MSE	1.48	1.53	2.07	1.61	1.79
RMSE	1.22	1.24	1.44	1.27	1.34

For each sample generated from the probabilistic model, we predict W^1_{lab} with the regression model and use $\hat{W}^1_{lab}$ to evaluate the solution in the cost function $F(\boldsymbol{x})$ (Section 2). W^1_{calc} and W^2_{calc} are computed analytically, and the rest of the W^i_{calc} are sampled from the probabilistic model.

[*] *R2* denotes the coefficient of determination which is upper bounded by 1, and can be negative when the model performs arbitrarily worse. *R2* is defined as $R2 = 1 - \frac{SS_{res}}{SS_{tot}}$, where SS_{res} and SS_{tot} denote the sum of squares of residuals, and the total sum of squares, respectively. When $SS_{res} = 0$, the predicted values perfectly match the observed values.

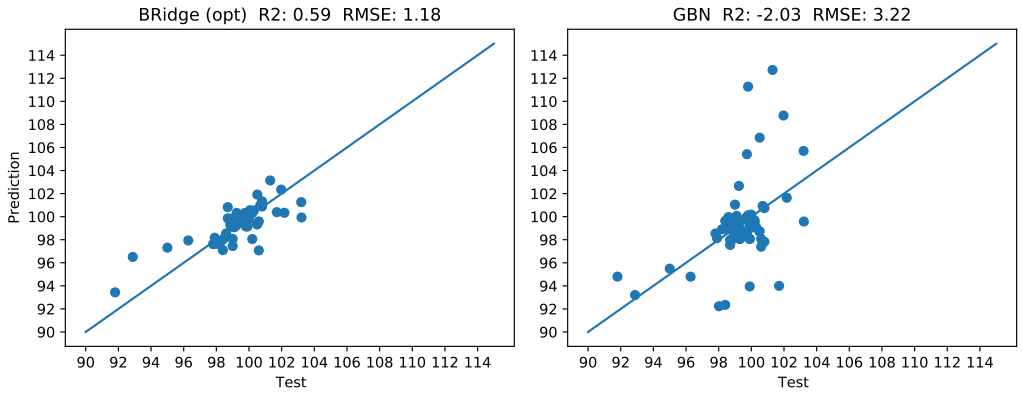

Figure 4. Comparison of real versus predicted values for the Bayesian ridge model after parameter tuning, and the GBN approach.

Figure 4(a) shows a comparison between the real and predicted values of BRidge using leaving-one-out cross validation. Note that the diagonal line shows the perfect case in which the real values match the predicted ones. Figure 4(b) shows the same situation using the GBN approach where the target variable W^1_{lab} is estimated using the maximum a posteriori, that is, the most probable value for $p(W^1_{lab}|\boldsymbol{X} \setminus \{W^1_{lab}\})$. A negative *R2* metric and a much higher RMSE have been obtained as it seems to overestimate the target variable in some instances from the dataset. Thus, the use of an external regressor is justified in order to improve the predictions made for W^1_{lab} during runtime.

4.2. Probabilistic model

The algorithm will utilize the embedded probabilistic model to learn patterns identified during the optimization process and to sample new feasible solutions. Gaussian BNs have been chosen for the following reasons: (i) they allow the use of evidence in the model, causing all the samplings to have the environment variables fixed to a specific value [7]; (ii) they represent uncertain knowledge as a graph; and (iii) we are dealing with continuous variables.

In the historical data, we find fuel formulas with ingredients that traditionally belonged to the set of decision variables $\boldsymbol{Z}$. However, due to current governmental restrictions on fuels, they must be restricted to a specific value or be entirely banned. Thus, we must set these variables to a fixed value, i.e., environment variables [7]. The environment variables $\boldsymbol{Y}$ are evidence in the learned Gaussian BN, so that future solutions are sampled from the posterior conditional probability $p(\boldsymbol{Z}|\boldsymbol{Y} = \boldsymbol{y})$.

The sampling process has been parameterized so that the degree of exploration and exploitation is controlled by the log-likelihood ($L(\dot{)}$) of each sample $\boldsymbol{x}$ in $\mathbb{M} = p(\mathcal{X}_{train})$, where $\mathcal{X}_{train}$ is the historical data. A sample is only valid if $L(\boldsymbol{x}|\mathbb{M}) + \alpha \leq 0$. As a reference, we have computed the log-likelihood of each sample in $\mathcal{X}_{train}$ under $\mathbb{M}$. A box plot is shown in Figure 5, where the mean and median are approximately –46 and –42, respectively. Thus, for increasing values of exploratory parameter ($\alpha \to \infty$), the sampling process becomes more exploratory, and for α values close to zero, the sampling process tends to an exploitation behavior. Further analyses on the performance of the EDA approach depending on α parameter are analyzed in Section 4.5.

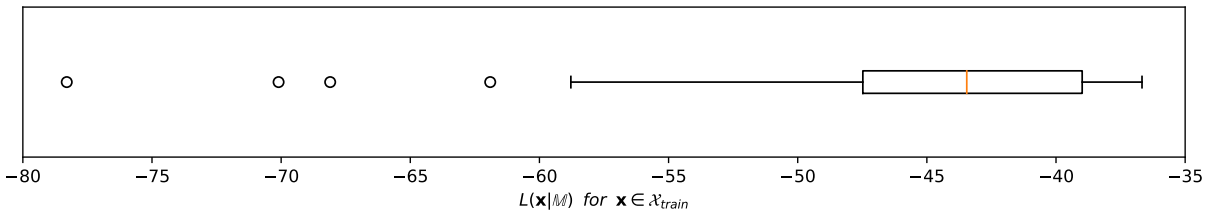

Figure 5. Box plot of the log-likelihood of each sample from $\mathcal{X}_{train}$ in $\mathbb{M} = p(\mathcal{X}_{train})$, where mean and median are approximately –46 and –42, respectively.

4.3. Truncation

The truncation process is performed in each iteration of the algorithm. Most multi-objective approaches use the hypervolume metric to select the top solutions that best approximate the Pareto frontier. However, in our approach we rank the solutions using the different frontiers as shown in Figure 6, where solutions from different frontiers (from the first one onward) are selected to train the probabilistic model of the respective generation. This way, we expect to avoid local convergence by including more diversity in the truncation set. Moreover, this would reduce the computation time by preventing to compute multi-objective quality indicators in each iteration. The best solutions are selected sequentially from successive Pareto frontiers until the required population size is reached, prioritizing those closest to the ideal reference point.

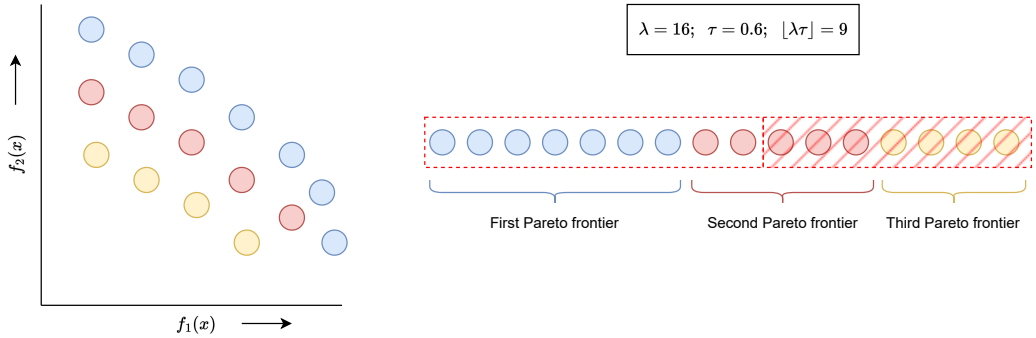

Figure 6. First, second, and third Pareto frontiers represented as blue, red, and yellow circles, respectively, are identified for maximizing $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ objectives, and ranked for the truncation process. An example for $\lambda = 16$ and $\tau = 0.6$ is shown.

4.4. Initial data generation

Although some historical data was provided for the problem we are addressing in this paper, we propose a methodology based on Latin hypercube sampling (LHS) to generate the initial set of experiments (Figure 1, Step 1.b). LHS has traditionally been used in a wide range of sampling designs [54].

LHS is incorporated only as an auxiliary alternative for cases where no expert-provided solutions are available. It is not employed in our experiments but its functionality is included to ensure applicability in the absence of prior knowledge, while its known shortcomings are therefore not impactful in the present study.

The overall performance of the optimization process heavily depends on the quality of the samplings of the initial data. A set of samplings that efficiently explores the landscape of solutions will facilitate, and most probably ensure, the discovery of good feasible solutions. However, a poor set of samplings may lead the optimizer to converge to local optima.

LHS is a statistical technique that efficiently explores a multi-dimensional parameter space while ensuring a representative sample of the input variables. LHS divides each variable range into equally likely intervals (number of samplings) and selects a single value from each interval, creating a stratified, non-repetitive sample that maximizes the coverage of the entire parameter space. A permutation between the samplings of each dimension is then run. We add a post-processing hill-climbing method to ensure equally spaced samples using the Lloyd-Max algorithm [55].

Although this intelligent sampling process ensures that all areas of the landscape are sampled, it is desired to execute in the laboratory those that are most promising for promoting the next generations of the algorithm. Thus, in Figure 1, the initial data process collaborates with our approach. The initial sampling is used to initialize the EDA, and the best solutions found are taken to the laboratory. The exploratory parameter α will be tuned by the user in such a way that the solutions are as exploratory as possible.

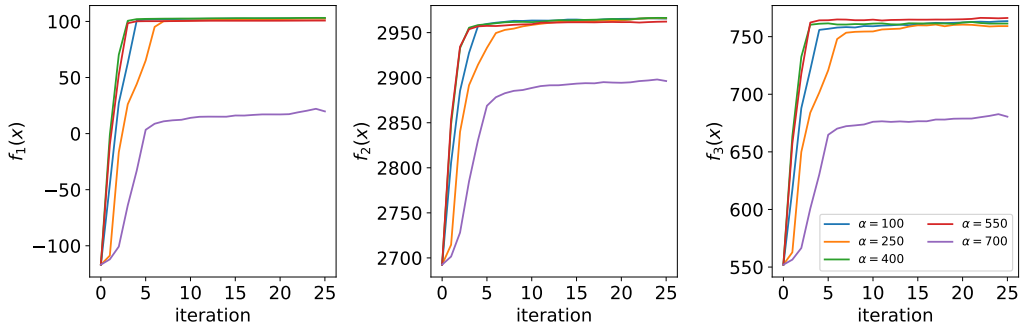

Figure 7. Convergence plot of the three objectives ($f_1(\boldsymbol{x})$, $f_2(\boldsymbol{x})$, $f_3(\boldsymbol{x})$) for different values of exploratory parameter α. Each line represents the mean objective value in each iteration after executing the EDA approach 25 times for each α configuration.

4.5. Performance analysis

Figure 7 shows a comparison of the three objectives for different values of exploratory parameter α after running the EDA approach 25 independent times for each configuration.

First, it can be observed that along the iteration process (x-axis) the objectives are maximized until they reach a constant value, and the algorithm is considered to have converged.

Second, depending on the values that the parameter takes, the algorithm reaches different optima. In the case of W^1_{lab} the best result is reached for almost all the values of the parameter. Parameters of $\alpha = 400$ and $\alpha = 580$ seem to be the best configurations, reaching the best values for the three objectives. Finally, $\alpha = 700$ returns the worst results for the three objectives.

To analyze the best parameter configuration we will use the HV quality indicator (Equation 3.2). In this case, the experts fix the reference ideal point to $f^{ref} = (102 + \text{RMSE}(\hat{W}^1_{lab}), 2980, 780)$ for the HV computation, where $\text{RMSE}(\hat{W}^1_{lab})$ is the root mean squared error obtained by the regression model, computed in Table 2.

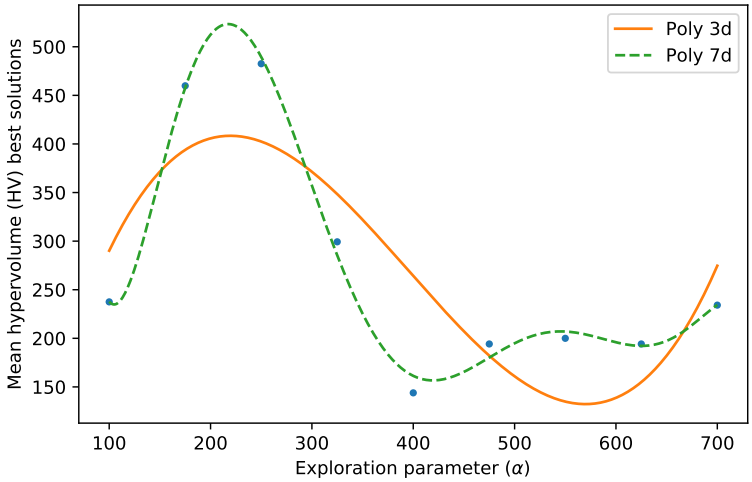

Figure 8. Mean HV after running the EDA approach 25 independent times for different values of exploratory parameter α. Blue dots represent the HV mean values, and orange and green dashed lines represent a function approximation using a polynomial with degree three and seven, respectively.

Figure 8 shows the mean HV achieved after 25 independent executions for each parameter configuration. It can be observed that there are no signs of increasing or decreasing monotony. In fact, we observe a non-monotonic line for each function approximation with a valley around values of $\alpha = 400$ and $\alpha = 580$. Note that the results obtained for low values of α, that is, low exploration and high exploitation with solutions similar to the ones provided by the experts in the historical data, achieve an HV of approximately 250 while the best results are obtained for $\alpha = 400$ with a mean HV of approximately 150. For greater values of parameter α, the mean HV is worse, but still better than the one achieved by the historical data. When $\alpha \to \infty$, an increasing tendency of the HV has been observed, which agrees with the results shown in Figure 7.

We conjecture that although the historical data contains good performing fuel formulas, if we move to another area of the search space within the landscape of solutions, we will find a fuel formula with better performance and objective values. Therefore, the experts were using a formulation corresponding to a local optimum. An overly exploratory approach can lead to very poor solutions, as can be seen for $\alpha = 700$ in Figure 7 and Figure 8.

Figure 9 shows the mean log-likelihood of the solutions in generation G_t along the optimization process. In general, a decreasing trend can be observed independently of the values of α, always respecting the threshold imposed by the exploration parameter.

Note that a common behavior among the different α values is that in the first iterations the log-likelihood decreases more sharply than in the following iterations. We conjecture that this behavior is justified by the fact that the EDA approach in the first iterations becomes more exploratory, but as the optimization advances it becomes more exploitative (within the range allowed by the α parameter), converging to already explored areas of the landscape.

Figure 10 shows an example of the new optimal proposals provided by the EDA approach for different values of exploration parameter α, analyzed in two dimensions of the landscape (Supplementary materials). Orange circles represent the traditional fuel formulas found in the

historical data presented by the experts, while blue symbols represent the best EDA solutions for different values of the exploration parameter.

The left panel illustrates a comparison between D4 and D3 ingredients, which belong to the same category of ingredients. Traditionally, the experts use either one of the two ingredients, however, the EDA proposes new formulas in which both ingredients are combined.

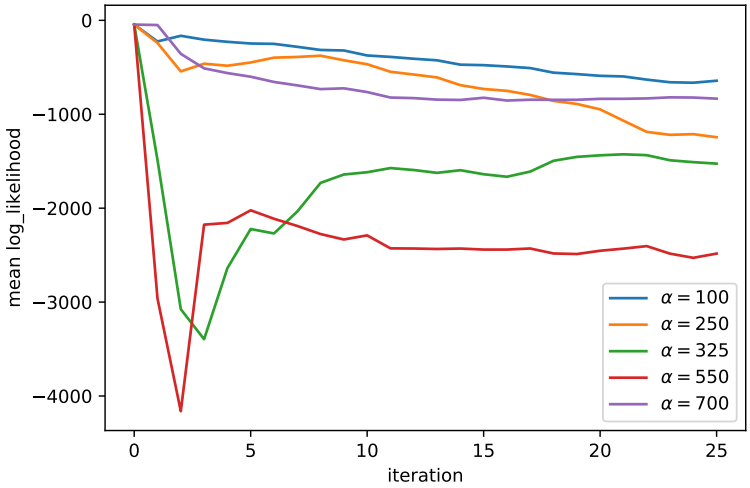

Figure 9. Mean log-likelihood of the set of solutions in each iteration along the optimization process for different values of the α parameter.

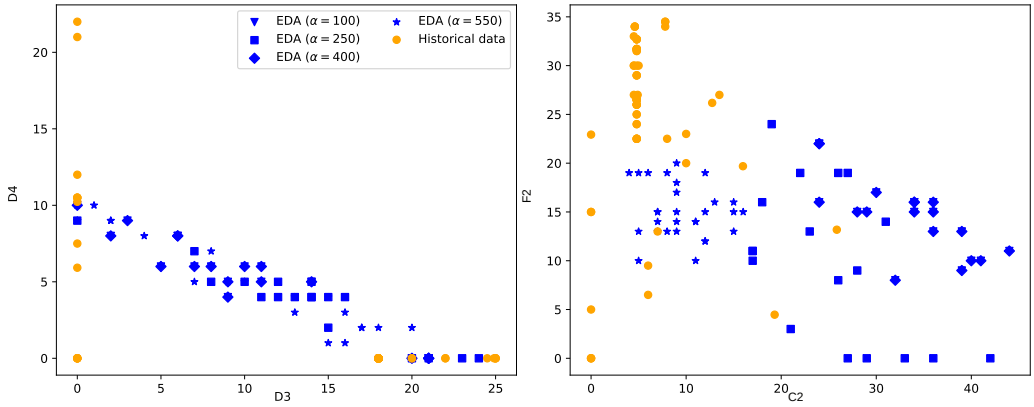

Figure 10. Comparison of two dimensions of the problem landscape where orange circles represent traditional formulas found in the historical data, and the rest of the symbols represent the new optimal formulas found by our approach. The left panel compares D4 and D3 ingredients and the right panel compares F2 and C2 ingredients.

The right panel shows a comparison between F2 and C2, which do not belong to the same category of ingredients. Typically, experts tend to maximize the first ingredient while minimizing the second one. It is observed that some of the EDA solutions try to imitate this knowledge learned from the historical data, but other solutions adopt a different perspective, maximizing the second ingredient while minimizing the other. This reflects the exploratory power of our approach.

4.6. *Knowledge discovery*

In this section, we analyze the set of variables that are strongly related to $f_1(\boldsymbol{x})$ in order to identify the dependencies between the fuel ingredients and the optimization objective, as this information is not known a priori.

It is worth mentioning that the Gaussian BN analyzed has been learned from the solutions found when the exploration parameter has been tuned to $\alpha = 400$ (where best results have been found). Thus, the strongly related variables found for different values of the parameter may vary.

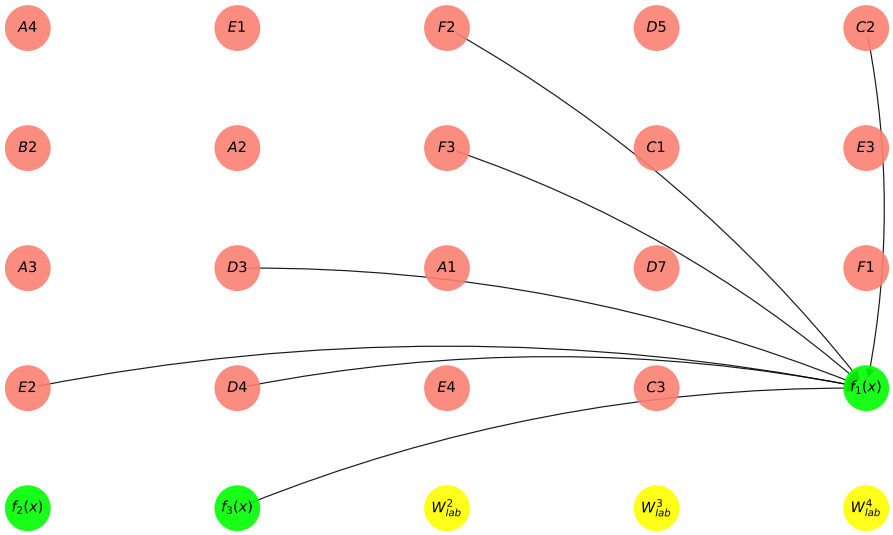

Figure 11. Gaussian BN structure learned from the best solutions found in the last 5 iterations of the EDA approach with $\alpha = 400$, where only the arcs contained in the Markov blanket of $f_1(\boldsymbol{x})$ are shown. Red, green, and yellow nodes represent the ingredients, optimization objectives, and W^2_{lab}, W^3_{lab}, W^4_{lab}, respectively.

Figure 11 represents the structure of the Gaussian BN learned with the best solutions found in the last 5 iterations of the EDA approach with $\alpha = 400$, where only the arcs contained in the Markov blanket of $f_1(\boldsymbol{x})$ are shown, i.e., $\mathbf{MB}(f_1(\boldsymbol{x}))$.

The Markov blanket of $f_1(\boldsymbol{x})$ includes ingredients from the categories $\{C, D, E, F\}$ and the objective $f_3(\boldsymbol{x})$. Remarkably, all the variables included in $\mathbf{MB}(f_1(\boldsymbol{x}))$ are the parents of $f_1(\boldsymbol{x})$ in the graph (Figure 11), and the conditional probability density (Equation 3.1) is expressed as:

$$\begin{aligned} p(f_1(\boldsymbol{x})|\boldsymbol{Pa}_{f_1(\boldsymbol{x})}) = \mathcal{N}(142.01 &- 0.239\, D4 - 0.158\, C2 - 0.105\, F3 + \\ &- 0.123\, D3 - 0.166\, F2 - 0.065\, f_3(\boldsymbol{x}) \\ &- 0.073\, E2;\ \sigma^2), \end{aligned} \tag{4.1}$$

where numbers represent the weights $(\beta_1, \ldots, \beta_7)$ of how the value of each variable influences the mean objective $f_1(\boldsymbol{x})$ in the conditional probability density, $\beta_0 = 142.01$ is the intercept coefficient, and σ^2 is the variance.

It is observed that the two variables with the highest weight associated, and thus, with more influence on the final objective are $D4$ and $C2$. Although the weight associated to $f_3(\boldsymbol{x})$ is low, the range of values of this objective is more than one magnitude higher than the values of the ingredients. Thus, we find a strong relation between both $f_1(\boldsymbol{x})$ and $f_3(\boldsymbol{x})$ objectives.

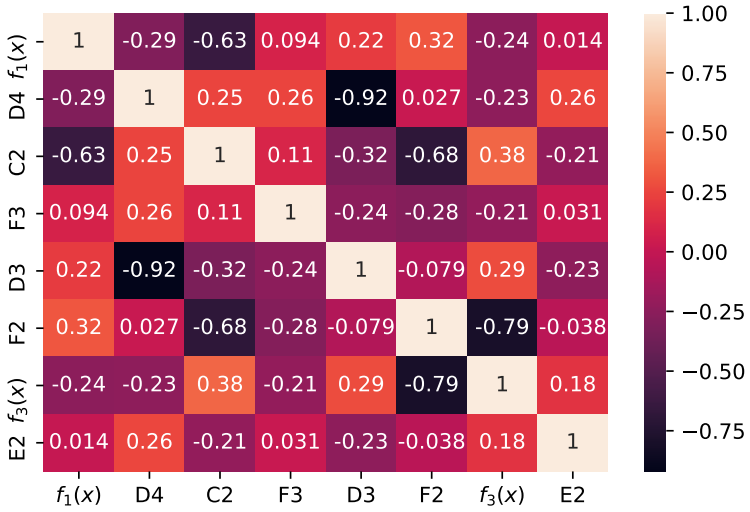

Figure 12. Heatmap that represents the Pearson correlation found between the variables in **MB**($f_1(\boldsymbol{x})$) and $f_1(\boldsymbol{x})$ in the set of best solutions obtained in the last 5 iterations of the EDA approach.

Figure 12 shows Pearson correlations computed over the set of best solutions found in the last iterations of the algorithm. Those variables more correlated with the objective $f_1(\boldsymbol{x})$ will denote a big influence between both variables. It can be observed that $C2$ and $F2$ are the two variables most correlated with $f_1(\boldsymbol{x})$.

Note that $f_2(\boldsymbol{x})$ and $f_3(\boldsymbol{x})$ heavily depend on the analytical descriptors, and thus the dependence between the ingredients and these objectives is already known.

4.7. Comparison

To perform a fair analysis of our approach (MO_EDA), it is compared to others, such as NSGA-II [43] and MOEAD [56]. The algorithms' hyperparameters tuning can be found in Supplementary materials. Hypervolume and diversity metrics described in Section 3.4 are used to evaluate the comparison.

Figure 13 represents the best results found by the NSGA-II, MOEAD, and MO_EDA approaches, plotted with blue, orange, and green markers, respectively. Note that, in the case of our approach, the two first Pareto frontiers are shown as two separate groups in the landscape of objective values.

Figure 13 shows that, while MOEAD seems to find an area of the search space in which all the solutions have a similar result (in terms of the three analyzed objectives), NSGA-II returns solutions with very varied results. With focus on the reference point ($f^{ref} = (102 + \text{RMSE}(\hat{W}^1_{lab}), 2980, 780)$), the solutions found by our approach seem to outperform the ones found by its competitors.

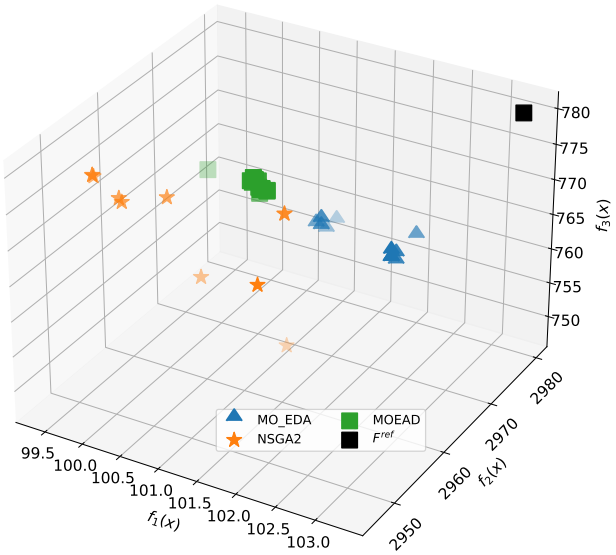

Figure 13. Blue, orange, and green markers represent the MO_EDA, NSGA-II, and MOEAD best solutions, respectively, in the landscape of the three objectives to be optimized. The black square represents the reference point (f^{ref}) used to compute the HV indicator. More than one frontier of solutions for MO_EDA were included. The x-, y- and z-axes represent $f_1(\boldsymbol{x})$, $f_2(\boldsymbol{x})$, and $f_3(\boldsymbol{x})$, respectively.

Table 3. Mean and standard deviation of hypervolume (Equation 3.2) and diversity (Equation 3.4) indicators for the best solutions obtained by the MO_EDA, NSGA-II, and MOEAD approaches.

	MO_EDA	NSGA-II	MOEAD
HV	124.63 ± 74.99	535.13 ± 265.8	260.7 ± 38.13
DM	7.27 ± 7.57	213.82 ± 304.0	4.38 ± 4.46

Results observed in Figure 13 are contrasted with the results of hypervolume and diversity metrics found in Table 3. Note that to compute those metrics, just the first frontier of solutions from the EDA approach was considered. The large variance among the results of NSGA-II is reflected in the DM indicator compared to the computed metric for the MO_EDA and MOEAD approaches. Also, Figure 13 places the solutions far from the reference point (f^{ref}), and the hypervolume computed for this algorithm is the worst one with a large standard deviation. Although the MO_EDA and MOEAD approaches achieve a similar diversity indicator, our approach outperforms its competitors in terms of hypervolume minimization.

An ANOVA test [57] has been computed to reject the null hypothesis of equal means for the hypervolume and diversity indicators, where statistically significant differences have been found (p-value = $3.07e-8$ and p-value = $3.82e-28$, respectively).

The effectiveness of the proposed algorithm is confirmed through multiple analyses. As shown in Table 3, the obtained solutions are consistently close to the imposed reference point. In addition,

Figure 13 provides a visual comparison with other competitors, demonstrating the superior accuracy of the proposed approach. Furthermore, by leveraging probabilistic models, the algorithm is able to uncover new patterns among the solutions, a capability not observed in the alternative strategies.

5. Conclusions

In this paper we proposed a new methodology for application in formulation laboratories for the experimental design of fuels. With our approach we expect to change the paradigm of traditional methods to a data-driven approach in which the number of real experiments is reduced leading to cost-cutting and time-saving.

The experimental design has been approached as an optimization problem where three properties of the fuel mixture have been maximized as a multi-objective task. Comparing our approach to some state-of-the-art algorithms, experimental results have found that the EDA-based optimizer outperforms its competitors.

Our approach includes an α parameter which controls the level of exploration and exploitation. Experimental results have shown that accurately tuning this parameter will lead to discovering better results than those already known by the experts. Moreover, the probabilistic model embedded by the EDA approach is able to learn patterns between fuel ingredients that were not considered by the experts in their experimental designs, thus enriching the interpretability of the problem.

We are aware that this is a first step toward the adoption of a new paradigm in the design of fuels, and there exist several ways to improve this approach. Among future steps, the following are included.

- The proposed approach samples $p(\boldsymbol{Z}|\boldsymbol{Y} = \boldsymbol{y})$ in each iteration, where $\boldsymbol{Z}$ and $\boldsymbol{Y}$ are the decision and environment variables of the problem, respectively, and uses $\hat{W}^1_{lab}$ to compute the multi-objective cost function. An improvement of the probabilistic model could happen if W^1_{lab} was introduced in the probabilistic model as an additional node, and the sampling of W^1_{lab} from the model was considered to compute the multi-objective cost function. The uncertainty of this sampling could also be analyzed.

- Exploration parameter (α) was tuned empirically, so finding a way to automatically tune it in the process would mean a more autonomous process.

- Although the control of exploration/exploitation of the algorithm is tuned through the log-likelihood, the use of geometric metrics in the landscape would better explore the optimization surface.

- Although in this optimization problem the historical data fits Gaussians, most of the industrial data does not fit normal distributions. Improving this approach with a semiparametric perspective [40], where assuming a parametric distribution is not a restriction, would enlarge the area of application of the data-driven approach.

- While the scalar penalty parameter δ used in the constraint handling was empirically tuned and yielded satisfactory results in this study, exploring more principled or adaptive approaches (e.g., feasibility rules or ϵ-constraint methods) remains an important direction for future work.

Author contributions

All authors contributed to the conception and design of the study. Vicente P. Soloviev led the research design, experimentation, and analysis. Marco Bernabei assisted with data interpretation, visualization, and drafting of the manuscript. Pedro Larrañaga and Concha Bielza provided critical revisions and supervised the overall project. All authors reviewed and approved the final version of the manuscript.

Use of Generative-AI Tools Declaration

The author(s) declare that Generative Artificial Intelligence (AI) tools were used in the preparation of this article. AI tools used: ChatGPT (OpenAI). How were the AI tools used: AI assistance was employed exclusively for English grammar revision, stylistic improvements, and clarity checking. Where in the article is the information located: AI was used throughout the manuscript for language polishing only; all scientific content, analyses, results, and conclusions were produced entirely by the authors.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Science and Innovation through the PID2022-139977NB-I00 project and TED2021-131310B-I00. This work has been also partially supported by Repsol through Agreement N.º6 UPM-Repsol "Self-driving lab project". Also, by the Autonomous Region of Madrid under Project484 ELLIS Unit Madrid and TEC-2024/COM-8. Vicente P. Soloviev has been supported by the predoctoral grant FPI PRE2020-094828 from the Spanish Ministry of Science and Innovation.

Conflict of interest

The authors declare no conflict of interest.

Code availability

EDA implementations are based in the ones available in the EDAspy Python project available at `https://pypi.org/project/EDAspy/` and can be downloaded from Pypi.

References

1. J. Deutch, Is net zero carbon 2050 possible?, *Joule* **4** (2020), 2237–2240. https://doi.org/10.1016/j.joule.2020.09.002
2. K. C. Seto, G. Churkina, A. Hsu, M. Keller, P. W. G. Newman, B. Qin, et al., From low-to net-zero carbon cities: The next global agenda, *Annu. Rev. Environ. Resour.*, **46** (2021), 377–415. https://doi.org/10.1146/annurev-environ-050120-113117

3. A. Jankovic, G. Chaudhary, F. Goia, Designing the design of experiments (doe)–an investigation on the influence of different factorial designs on the characterization of complex systems, *Energy Build.*, **250** (2021), 111298. https://doi.org/10.1016/j.enbuild.2021.111298
4. F. A. C. Viana, Things you wanted to know about the latin hypercube design and were afraid to ask, *10th World Congr. Struct. Multidiscip. Optim. vol.*, **19** (2013), 1–9.
5. H. Vieira, S. M. Sanchez, K. H. Kienitz, M. C. N. Belderrain, Efficient, nearly orthogonal-and-balanced, mixed designs: An effective way to conduct trade-off analyses via simulation, *J. Simul.*, **7** (2013), 264–275. https://doi.org/10.1057/jos.2013.14
6. R. H. Myers, D. C. Montgomery, C. M. Anderson-Cook, Response surface methodology: Process and product optimization using designed experiments, *John Wiley Sons*, 2016. https://doi.org/10.1080/00224065.2017.11917988
7. V. P. Soloviev, P. Larrañaga, C. Bielza, Estimation of distribution algorithms using gaussian bayesian networks to solve industrial optimization problems constrained by environment variables, *J. Combin. Optim.*, **44** (2022), 1077–1098. https://doi.org/10.1007/s10878-022-00879-6
8. H. Liu, Y. S. Ong, J. Cai, A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design, *Struct. Multidiscip. Optim.*, **57** (2018), 393–416. https://doi.org/10.1007/s00158-017-1739-8
9. S. Greenhill, S. Rana, S. Gupta, P. Vellanki, S. Venkatesh, Bayesian optimization for adaptive experimental design: A review, *IEEE Access*, **8** (2020), 13937–13948. https://doi.org/10.1109/ACCESS.2020.2966228
10. K. Hanaoka, Comparison of conceptually different multi-objective bayesian optimization methods for material design problems, *Mater. Today Commun.*, **31** (2022), 103440. https://doi.org/10.1016/j.mtcomm.2022.103440
11. K. Hanaoka, Bayesian optimization for goal-oriented multi-objective inverse material design, *iScience*, **24** (2021), 102781. https://doi.org/10.1016/j.isci.2021.102781
12. X. Sun, Z. Shi, G. Lei, Y. Guo, J. Zhu, Multi-objective design optimization of an ipmsm based on multilevel strategy, *IEEE Trans. Ind. Electron.*, **68** (2020), 139–148. https://doi.org/10.1109/TIE.2020.2965463
13. Z. Pan, D. Lei, L. Wang, A knowledge-based two-population optimization algorithm for distributed energy-efficient parallel machines scheduling, *IEEE Trans. Cybern.*, **52** (2020), 5051–5063. https://doi.org/10.1109/TCYB.2020.3026571
14. H. Wang, B. R. Sarker, J. Li, J. Li, Adaptive scheduling for assembly job shop with uncertain assembly times based on dual q-learning, *Int. J. Prod. Res.*, **59** (2021), 5867–5883. https://doi.org/10.1080/00207543.2020.1794075
15. F. Zhao, F. Yin, L. Wang, Y. Yu, A co-evolution algorithm with dueling reinforcement learning mechanism for the energy-aware distributed heterogeneous flexible flow-shop scheduling problem, *IEEE Trans. Syst. Man. Cybern. Syst.*, **55** (2024), 1794–1809. https://doi.org/10.1109/TSMC.2024.3510384
16. M. Khishe, N. Orouji, M. R. Mosavi, Multi-objective chimp optimizer: an innovative algorithm for multi-objective problems, *Expert Syst. Appl.*, **211** (2023), 118734. https://doi.org/10.1016/j.eswa.2022.118734

17. Q. Wang, G. Chen, M. Khishe, B. F. Ibrahim, S. Rashidi, Multi-objective optimization of iot-based green building energy system using binary metaheuristic algorithms, *J. Build. Eng.*, **68** (2023), 106031. https://doi.org/10.1016/j.jobe.2023.106031

18. N. Mashru, G. G. Tejani, P. Patel, M. Khishe, Optimal truss design with moho: A multi-objective optimization perspective, *PLoS One*, **19** (2024), e0308474. https://doi.org/10.1371/journal.pone.0308474

19. S. Liu, J. Li, H. Aghajanirefah, Reinforced concrete structures with damped seismic buckling-restrained bracing optimization using multi-objective evolutionary niching choa, *Steel Compos. Struct.*, **47** (2023), 147. https://doi.org/10.1017/ssh.2022.38

20. M. Aljaidi, N. Mashru, P. Patel, D. Adalja, P. Jangir, Arpita, et al., Morime: A multi-objective rime optimization framework for efficient truss design, *Results Eng.*, **25** (2025), 103933. https://doi.org/10.1016/j.rineng.2025.103933

21. P. Patel, D. Adalja, N. Mashru, Multi objective elk herd optimization for efficient structural design, *Sci. Rep.* **15** (2025), 11767. https://doi.org/10.1038/s41598-025-96263-5

22. X. Liu, W. Zuo, Q. Li, K. Zhou, Y. H. Huang, Y. W. Li, et al., Multi-objective optimization of an ammonia-fueled micro planar combustor with a secondary oxygen injection for thermophotovoltaic applications, *Energy*, **385** (2025), 138044. https://doi.org/10.1016/j.energy.2025.138044

23. W. Zuo, F. Li, Q. Li, Z. J. Chen, Y. H. Huang, H. Q. Chu, Multi-objective optimization of micro planar combustor with tube outlet by rsm and nsga-ii for thermophotovoltaic applications, *Energy*, **291** (2024), 130396. https://doi.org/10.1016/j.energy.2024.130396

24. Z. Fu, W. Zuo, Q. Li, K. Zhou, Y. H. Huang, Y. W. Li, Multi-objective optimization of liquid cooling plate partially filled with porous medium for thermal management of lithium-ion battery pack by rsm, nsga-ii and topsis, *Energy* **318** (2025), 134853. https://doi.org/10.1016/j.energy.2025.134853

25. W. Zuo, D. Li, Q. Li, Q. J. Cheng, K. Zhou, J. Qiang, Multi-objective optimization of multi-channel cold plate under intermittent pulsating flow by rsm and nsga-ii for thermal management of electric vehicle lithium-ion battery pack, *Energy*, **283** (2023), 129085. https://doi.org/10.1016/j.energy.2023.129085

26. J. Li, W. Zuo, Y. Zhang, Q. Q. Li, K. Sun, K. Zhou, et al., Multi-objective optimization of miniu-channel cold plate with sio2 nanofluid by rsm and nsga-ii, *Energy*, **242** (2022), 123039. https://doi.org/10.1016/j.energy.2021.123039

27. Z. Chen, W. Zuo, K. Zhou, Q. Q. Li, Y. H. Huang, J. Qiang, Multi-objective optimization of proton exchange membrane fuel cells by rsm and nsga-ii, *Energy Convers. Manage.*, **277** (2023), 116691. https://doi.org/10.1016/j.enconman.2023.116691

28. P. Larrañaga, J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*, Kluwer Academic Publishers, 2001.

29. C. A. Coello, *Evolutionary algorithms for solving multi-objective problems*, Springer, 2007.

30. D. Koller, N. *Friedman, Probabilistic graphical models: Principles and techniques*, The MIT. Press, 2009.

31. J. Ceberio, A. Mendiburu, J. A. Lozano, A roadmap for solving optimization problems with estimation of distribution algorithms, *Nat. Comput.*, **23** (2022), 1–15. https://doi.org/10.1007/s11047-022-09913-2

32. R. Armañanzas, I. Inza, R. Santana, Y. Saeys, J. L. Flores, J. A. Lozano, et al., A review of estimation of distribution algorithms in bioinformatics, *BioData Mining*, **1** (2008), 1–12. https://doi.org/10.1186/1756-0381-1-6

33. V. P. Soloviev, P. Larrañaga, C. Bielza, Quantum parametric circuit optimization with estimation of distribution algorithms, *Proc. Genet. Evol. Comput. Congr. Companion*, **4** (2022), 2247–2250. https://doi.org/10.1145/3520304.3533963

34. H. Karshenas, R. Santana, C. Bielza, P. Larrañaga, Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables, *IEEE Trans. Evol. Comput.*, **18** (2013), 519–542. https://doi.org/10.1109/TEVC.2013.2281524

35. K. Metaxiotis, K. Liagkouras, Multiobjective evolutionary algorithms for portfolio management: A comprehensive literature review, *Expert Syst. Appl.*, **39** (2012), 11685–11698. https://doi.org/10.1016/j.eswa.2012.04.053

36. A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C. A. Coello, A survey of multiobjective evolutionary algorithms for data mining: Part i, *IEEE Trans. Evol. Comput.*, **18** (2013), 4–19. https://doi.org/10.1109/TEVC.2013.2290086

37. Z. Wang, M. Li, J. Li, A multi-objective evolutionary algorithm for feature selection based on mutual information with a new redundancy measure, *Inf. Sci.*, **307** (2015), 73–88. https://doi.org/10.1016/j.ins.2015.02.031

38. H. Mühlenbein, J. Bendisch, H. M. Voigt, From recombination of genes to the estimation of distributions ii. continuous parameters, *Int. Conf. Parallel Problem Solving from Nature*, 1996, 188–197. https://doi.org/10.1007/3-540-61723-X_983

39. P. Larrañaga, R. Etxeberria, J. A. Lozano, J. M. Peña, Optimization in continuous domains by learning and simulation of gaussian networks, *Proc. Genet. Evol. Comput. Congr.*, 2000, 201–204. https://doi.org/10.1007/11903697_69

40. V. P. Soloviev, C. Bielza, P. Larrañaga, Semiparametric estimation of distribution algorithms for continuous optimization, *IEEE Trans. Evol. Comput.*, **4** (2023), 1069–1083. https://doi.org/10.1109/TEVC.2023.3290670

41. N. Khan, D. E. Goldberg, M. Pelikan, Multi-objective bayesian optimization algorithm, *Proc. 4th Annu. Conf. Genet. Evol. Comput.*, **8** (2022), 684–684. https://doi.org/10.1016/j.softx.2020.100520

42. M. Pelikan, D. E. Goldberg, E. Cantú-Paz, Boa: The bayesian optimization algorithm, *Proc. Genet. Evol. Comput. Congr.*, 1999, 525–532.

43. K. Deb, A. Pratap, S. Agarwal, T. A. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, *IEEE Trans. Evol. Comput.*, **6** (2002), 182–197. https://doi.org/10.1109/4235.996017

44. M. Pelikan, K. Sastry, D. E. Goldberg, Multiobjective hboa,clustering, and scalability, *Proc. 7th Annu. Conf. Genet. Evol. Comput.*, **8** (2005), 663–670. https://doi.org/10.1145/1068009.1068122

45. K. P. Murphy, *Machine learning: A probabilistic perspective*, The MIT. Press, 2012.

46. C. Bielza, P. Larrañaga, Bayesian networks in neuroscience: A survey, *Front. Comput. Neurosci.*, **8** (2014), 131. https://doi.org/10.7209/tanso.2014.131

47. C. Puerto-Santana, P. Larrañaga, C. Bielza, Autoregressive asymmetric linear gaussian hidden markov models, *IEEE Trans. Pattern Anal. Mach. Intell.*, **9** (2021), 4642–4658. https://doi.org/10.1109/TPAMI.2021.3068799

48. B. Mihaljević, C. Bielza, P. Larrañaga, Bayesian networks for interpretable machine learning and optimization, *Neurocomputing*, **456** (2021), 648–665. https://doi.org/10.1016/j.neucom.2021.01.138

49. I. Tsamardinos, C. F. Aliferis, Towards principled feature selection: Relevancy, filters and wrappers, *Int. Workshop Artif. Intell. Stat.*, 2003, 300–307.

50. M. Li, X. Yao, Quality evaluation of solution sets in multiobjective optimisation: A survey, *ACM Comput. Surv.*, **52** (2019), 1–38. https://doi.org/10.1145/3300148

51. E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. G. Da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, *IEEE Trans. Evol. Comput.*, **7** (2003), 117–132. https://doi.org/10.1109/TEVC.2003.810758

52. F. Pedregosa, G. Varoquaux, A. Gramfort, Scikit-learn: Machine learning in python, *J. Mach. Learn. Res.*, **12** (2011), 2825–2830.

53. J. Bergstra, D. Yamins, D. Cox, Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures, *Int. Conf. Mach. Learn.*, 2013, 115–123.

54. K. T. Fang, R. Li, A. Sudjianto, *Design and modeling for computer experiments*, CRC Press, 2005. https://doi.org/10.1201/9781420034899

55. S. Lloyd, Least squares quantization in pcm, *IEEE Trans. Inf. Theory*, **28** (1982), 129–137. https://doi.org/10.1109/TIT.1982.1056489

56. Q. Zhang, H. Li, Moea/d: A multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.*, **11** (2007), 712–731. https://doi.org/10.1109/TEVC.2007.892759

57. J. Kaufmann, A. Schering, Analysis of variance anova, *Wiley Encycl. Clin. Trials*, 2007. https://doi.org/10.1002/9781118445112.stat06938

AIMS Press