A Framework for Explainable Knowledge Generation with Expensive Sample Evaluations

Alberto Costa[a], Riccardo Talami[a,b]

[a]*Future Resilient Systems, Singapore-ETH Centre, Singapore*
[b]*National University of Singapore, Singapore*

Abstract

Real world problems often require complex modeling and computation efforts to be effectively addressed. Relying solely on data-driven approaches without integrating physics-based models can result in limited predictive capabilities. Even advanced techniques such as deep learning may be impractical for decision-makers due to the lack of data and challenges in justifying and explaining results. In this paper, we propose INFERNO (INference Framework for Efficient Rule-based kNOwledge generation), a framework designed to address these challenges. Our methodology integrates derivative-free optimization, bipartite network clustering, and a novel procedure to derive explainable inference rules. These rules enable decision-makers to easily identify high-quality solutions. We introduce a new metric, called Price of Explainability (PoX), to quantify the trade-off between quality and explainability. The framework was validated on two building design problems. Results show that INFERNO achieves PoX values that are, overall, 4.5 to 10 times lower than those of a classification tree.

Keywords: clustering, derivative-free optimization, explainability, inference rules

1. Introduction

In recent years, advances in artificial intelligence (AI), coupled with improved hardware capabilities, have led to significant innovations. One notable example is deep learning, where neural networks with multiple layers of nodes are trained on high-performance GPU systems with large amounts of data. These methods are widely used for regression and classification tasks in various domains, including computer vision, speech recognition, supply chain, healthcare, and natural language processing (Dong, Wang, & Abbas, 2021). A related example is the popular field of Large-Language Models (LLMs), which are trained on a large amount of data and commonly used for text recognition and generation. Leading technology companies such as Google, Meta, Microsoft and OpenAI, continuously introduce updated models to improve performance and expand capabilities in handling and generating text, images, audio, and video, while also addressing substantial hardware demands (Vaswani et al., 2017; Minaee et al., 2024).

Although data-driven techniques are increasingly widespread and can have significant impacts, they also introduce challenges related to security, privacy, fairness, bias, and explainability (Yao et al., 2024; Gallegos et al., 2024; Angelov, Soares, Jiang, Arnold, & Atkinson, 2021; Ding, Abdel-Basset, Hawash, & Ali, 2022). In complex systems where erroneous decisions may have severe consequences, explainability is particularly critical. For example, a misdiagnosis in healthcare care or a failure in critical infrastructure can be difficult to justify if the decision-making process is based on a black-box algorithm. Addressing this issue, often framed within the domain of Explainable AI (XAI) (Angelov et al., 2021; Dwivedi et al., 2023), is essential. A potential solution is to integrate data-driven methodologies with physics-based models when available. In principle, this allows decisions to be supported by the theoretical foundations of the underlying model (Wang, Li, Gao, & Zhang, 2022; Proverbio, Costa, & Smith, 2018a). In the field of LLMs, the challenge of explainability is often associated with the so-called *hallucination* phenomenon: as the task of an LLM is to probabilistically predict the next output, the generated answers could present false statements and inaccuracies that cannot be easily explained. To mitigate this phenomenon, various techniques have been proposed. For example, Retrieval Augmented Generation (RAG) allows an LLM to access information from an existing dataset, thus grounding the answers on specific information sources (Gao et al., 2024). Another example is agentic AI (e.g., LangChain, HuggingFace Smolagents), where LLMs can rely on predetermined functions to generate the information required to answer users' queries. In addition, prompting techniques such as Chain-of-Thought can mitigate hallucinations and inaccuracies by asking the LLM for reasoning *step-by-step* instead of directly answering (Wei et al., 2022). However, explainability challenges may still persist in different forms at various levels of the decision-making process.

Given that AI is becoming so popular, it is no surprise that even the field of Operations Research (OR) has been exposed to both AI techniques and, as a consequence, challenges. With reference to explainability, the concept of XAIOR (Explainable AI for OR) has been put forward as the "*conceptualization and application of advanced methods for transforming data into insights that are simul-*

taneously performant, attributable, and responsible for solving OR problems and enhancing decision-making" (De Bock et al., 2024).

In this paper, we consider OR problems in applied fields like architecture and engineering, where often expensive simulators are used to evaluate solution quality. This is sometimes referred to as black-box or derivative-free optimization. Even though the term black-box may suggest a situation incompatible with explainability, it is actually quite the opposite given the presence of physics-based models to perform simulations. Instead, the main challenge is that decision makers in these fields often prefer a range of solutions rather than a single optimal option. When solutions are high-dimensional and cannot be easily visualized, selecting among them is not trivial. Dimension-reduction techniques such as PCA are also not ideal, since results lack easy explainability. Consequently, justifying a decision in such scenarios can be difficult (Chakraborty, Kiefer, & Raubal, 2024; Proverbio, Costa, & Smith, 2018b). Further complicating the process, decision-makers may be unable or unwilling to explicitly define the criteria guiding their selection, leaving this information unavailable to algorithm designers. In this context, explainability is not about understanding how good solutions were generated. Rather, being able to explain how, starting from such solutions, decision-makers can derive meaningful actions and policies.

This paper proposes a framework to address the abovementioned challenges. To illustrate the approach, we consider a scenario in which a decision maker should identify high-quality solutions to a building design problem with two objectives: minimizing energy consumption and maximizing thermal comfort. A derivative-free optimization algorithm, coupled with a physics-based model, is used to identify a pool of high-quality candidate solutions. This set of solutions is then represented as a bipartite graph, which allows visualization independently of the dimensionality of the problem. Then a clustering algorithm is applied to extract meaningful patterns, and the resulting cluster information is used to derive *inference rules*, that is, guiding and easy-to-interpret principles that help decision makers identify suitable solutions. This way, this novel approach to rule generation supports decision making by offering flexibility in selecting high-quality solutions that better align with specific design requirements.

The rest of the paper is organized as follows. Section 2 introduces the necessary background information on optimization, clustering, and building design. The proposed framework, called INFERNO (INference Framework for Efficient Rule-based kNOwledge generation) is presented in Section 3. An explanation of the setting and evaluation procedure used for the experiments is provided in Section 4, where the Price of Explainability (PoX) metric is also introduced. The results of two case studies of building design are discussed in Section 5, including a comparison with a classification tree. Conclusions and future work are drawn in Section 6.

2. Background

This section introduces the necessary background to understand the methodology of this work. It focuses on three main topics: derivative-free optimization, clustering in bipartite networks, and building design, which is the field where the framework was tested.

Notation. In the rest of the paper, bold symbols indicate vectors and matrices. The i-th component of the vector $\boldsymbol{x}$ is indicated by $\boldsymbol{x}_i$, while $\boldsymbol{A}_{i,j}$ is the element located in the i-th row and the j-th column of matrix $\boldsymbol{A}$.

2.1. Derivative-free Optimization

Derivative-free (or black-box) optimization addresses problems in the following form:

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) && (1)\\ \text{s.t.} \quad & \boldsymbol{x} \in [\boldsymbol{x}^L, \boldsymbol{x}^U] && (2)\\ & \boldsymbol{x} \in \mathbb{Z}^q \times \mathbb{R}^{n-q}, && (3) \end{aligned}$$

where $\boldsymbol{x}$ is the n-dimensional vector of variables with lower an upper bounds $\boldsymbol{x}^L$ and $\boldsymbol{x}^U$, respectively, and where $q \leq n$ of such variables are integer while the remaining $n - q$ are continuous. In the derivative-free optimization setting analyzed here, the analytical form of the function $f(\boldsymbol{x})$ is unknown and can only be evaluated by expensive simulations. Therefore, methods for estimating partial derivatives using finite differences are impractical. For a similar reason, heuristic approaches such as simulated annealing and genetic algorithms are not suitable in this case, as they often require a large number of function evaluations (Regis & Shoemaker, 2007; Gutmann, 2001).

A popular approach to deal with the challenge of expensive simulations is to build a surrogate model (also called the response surface) of the unknown function $f(\boldsymbol{x})$. In practice, the surrogate model is an approximation of $f(\boldsymbol{x})$ obtained by interpolation and is used to guide the search for potentially better solutions. Once a good-quality surrogate model is obtained, it can also be used for a quick exploration of the domain without the need to perform additional simulations.

Some examples of this approach are the Radial Basis Function (RBF) method (Gutmann, 2001) and the Efficient Global Optimization (EGO) based on Kriging interpolation (Jones, Schonlau, & Welch, 1998).

In this paper, we used RBFOpt, an open source code based on the RBF approach (Costa & Nannicini, 2018; Nannicini, 2021). The reasons for this choice are two: i) the generation of a surrogate model can provide to decision-makers a quick way to assess the quality of alternative solutions, if needed, and ii) RBFOpt has demonstrated good performance in a variety of applications (Wortmann, Costa, Nannicini, & Schroepfer, 2015; Diaz, Fokoue-Nkoutche, Nannicini, & Samulowitz, 2017; Costa, Buccio, Melucci, & Nannicini, 2018), also highlighted by the results of the 2015 GECCO Black-Box competition, where an early version of RBFOpt ranked seventh overall (there were 28 participants) and first among open source software.[1] In the next section, we present the main idea of RBFOpt.

2.1.1. RBFOpt

RBFOpt builds a surrogate model of the unknown objective function $f(\boldsymbol{x})$ by iteratively selecting new samples. Each sample is often obtained by running an expensive simulation and is used to refine the surrogate model. The sample selection method balances the *exploration* of unknown areas of the domain and the search for good quality solutions according to the current surrogate model (*exploitation*).

The surrogate model is built using *radial basis functions*, and it assumes the following form:

$$s_k(\boldsymbol{x}) = \sum_{i=1}^{k} \lambda_i \phi(||\boldsymbol{x} - \bar{\boldsymbol{x}}_i||) + p(\boldsymbol{x}), \tag{4}$$

where k is the number of samples, $\phi \to \mathbb{R}^+ : \mathbb{R}$ is the radial basis function, $||\cdot||$ denotes the Euclidean distance operator, $\lambda_i \in \mathbb{R}$ is the coefficient associated with the $i-$th sample $\bar{\boldsymbol{x}}_{\boldsymbol{i}}$, $p(\boldsymbol{x})$ is a polynomial that may be necessary to ensure the existence of the interpolant. There are a few choices for the radial basis function $\phi(r)$, for example, cubic ($\phi(r) = r^3$) and thin plate spline ($\phi(r) = r^2 \log_2 r$). The choice of function determines the shape of the interpolant, that is, the coefficients $\boldsymbol{\lambda}$ and the polynomial $p(\boldsymbol{x})$. In practice, such values are derived by solving a linear system. This system imposes that for each sample $\boldsymbol{x}_i$ the interpolant is forced to

[1] For more details, see `https://www.ini.rub.de/PEOPLE/glasmtbl/projects/bbcomp/results/BBComp2015GECCO/summary.html` (accessed on 03/2025).

assume the same value of the unknown objective function, i.e., $s_k(\boldsymbol{x}_i) = f(\boldsymbol{x}_i)$, and some additional conditions. The interested reader can find more details in (Costa & Nannicini, 2018; Nannicini, 2021).

The summary of the basic functioning of RBFOpt is explained in the following:

1. select and evaluate m starting samples, using strategies like Latin Hypercube Design;
2. create the set S including the samples obtained so far, and set $k \leftarrow m$;
3. compute the RBF interpolant of the samples in S;
4. select a target objective function value f^*_{k+1};
5. find $\boldsymbol{x}_{k+1}$ such that the RBF interpolant to the samples obtained so far plus the sample $(\boldsymbol{x}_{k+1}, f^*_{k+1})$ is the least *bumpy*;
6. evaluate f in $\boldsymbol{x}_{k+1}$ and add the sample $(\boldsymbol{x}_{k+1}, f(\boldsymbol{x}_{k+1}))$ to S;
7. if $k \geq$ maximum number of evaluations, stop and return the best sample in S. Otherwise, set $k \leftarrow k + 1$ and go to Step 3.

The choice of f^*_{k+1} in Step 4 determines the balance between exploration and exploitation. When the value is very small, the resulting operation is a global search for solution potential much better than those evaluated so far. On the other hand, when f^*_{k+1} is equal to $\min_{\boldsymbol{x}} s_k(\boldsymbol{x})$ a local search is implemented on the current surrogate model, i.e., it is considered an accurate representation of the unknown function f. By cycling between these strategies, the algorithm allows one to find good solutions and avoid to be stuck in a local optimum. Once the value of f^*_{k+1} is determined, the problem of finding the point $\boldsymbol{x}_{k+1}$ that minimizes the surrogate bumpiness in Step 5 is a nonlinear, nonconvex optimization problem for which all elements are known. Therefore, despite being in general difficult, it is at least solvable either with mixed-integer nonlinear solvers or heuristics.

Note that this is the basic implementation of RBFOpt. Updated versions of the software include other features, e.g., more efficient solving procedures for intermediate steps and the managing of categorical variables (i.e., discrete, unordered variables) without introducing a bias. The reader should refer to the manual in `https://github.com/coin-or/rbfopt` for an exhaustive list of features.

2.2. Clustering in Bipartite Networks

Let $G = (V, E)$ be a network (also called graph), where V is the set of vertices (or nodes), and E is the set of unweighted undirected edges connecting pairs of vertices. A bipartite network is a network where $V = V_R \cup V_B$, $V_R \cap V_B = \emptyset$, and edge (v_i, v_j) joins a *red* vertex $v_i \in V_R$ with a *blue* vertex $v_i \in V_B$. Therefore, no edge exist between vertices of the same color. This structure is well-suited to represent relationships between entities of two different nature. For example, users and products they purchased (e.g., Netflix, Amazon) or authors and articles they wrote.

Clustering refers in general to the identification of groups of vertices, called clusters, communities, or modules, that are *similar* and share some common features. The formalization of the concept of similarity produced several approaches and it depends on the nature of the entities being clustered. For example, points in the space can be grouped together based on their distance and a popular algorithm to do so is K-means (MacQueen, 1967; Lloyd, 1982). When dealing with nodes connected by edges, different clustering methods can be used. Some of them do not require a function to be optimized, as the heuristic of Girvan and Newman where edges with the highest betweenness (the number of shortest paths including that edge) are removed (Girvan & Newman, 2002). Others are based on rules that clusters must respect. Some examples are the strong, weak, semi-strong, extra-weak, and almost strong definitions (Radicchi, Castellano, Cecconi, Loreto, & Parisi, 2004; Hu et al., 2008; Cafieri, Caporossi, Hansen, Perron, & Costa, 2012). In practice, they impose constraints for each vertex on the number of neighbors (i.e, the amount of vertices connected to that vertex) within and outside their cluster. For example, the strong definition requires each vertex in a cluster to have more neighbors within its own cluster than neighbors in other clusters. Finally, clusters can be identified by optimizing a function. One of the most famous of such functions is modularity, which can be defined as the fraction of edges within clusters minus the expected fraction of such edges in a random graph with the same degree distribution (Girvan & Newman, 2002; Newman & Girvan, 2004). To find clusters, modularity should be maximized. An extension of modularity to bipartite graphs has been proposed (Barber, 2007). In this work, we employ bipartite graphs and bipartite modularity optimization as a clustering method. This choice allows to interpret and visualize results independently of the problem dimension, as illustrated in the case studies. Details on the bipartite modularity formulation are provided in the next section.

2.2.1. Bipartite Modularity Optimization

Bipartite modularity was introduced in (Barber, 2007) as an extension of modularity to bipartite networks. The bipartite modularity optimization (BMO) problem can be formulated as follows:

$$\max \quad \frac{1}{m}\sum_{v_i\in V_R}\sum_{v_j\in V_B}\left(\boldsymbol{A}_{i,j}-\frac{k_ik_j}{m}\right)x_{i,j} \tag{5}$$

$$\text{s.t.} \quad \forall i<j<l\in N \quad -x_{i,j}+x_{i,l}+x_{j,l}\leq 1 \tag{6}$$

$$\forall i<j<l\in N \quad x_{i,j}+x_{i,l}-x_{j,l}\leq 1 \tag{7}$$

$$\forall i<j<l\in N \quad x_{i,j}-x_{i,l}+x_{j,l}\leq 1 \tag{8}$$

$$\forall i<j\in N \quad x_{i,j}\in\{0,1\}, \tag{9}$$

where $m = |E|$ is the number of edges of the graph G, $N = \{1, \dots, |V_R| + |V_B|\}$ is the set of vertex indices, $\boldsymbol{A}_{i,j}$ is an element of the adjacency matrix of G equal to 1 if vertices v_i and v_j are joined by an edge, 0 otherwise, k_i is the degree of vertex v_i (i.e., the number of its neighbors), $x_{i,j}$ is a binary variable equal to 1 if vertices v_i and v_j belong to the same cluster, 0 otherwise, and Constraints (6)-(8) represent the transitivity conditions imposing that when vertices v_i and v_j belong to the same clusters and vertices v_i and v_l belong to the same cluster, then vertices v_j and v_l also belong to the same cluster. The objective function (5) represents the bipartite modularity that should be maximized to identify clusters.

There is an alternative way to express bipartite modularity, but it has the disadvantage of requiring to specify the optimal number of clusters a-priori (while this information is a byproduct of solving Problem (5)-(9)). However, it has been shown that maximizing bipartite modularity is an **NP**-hard problem (Miyauchi & Sukegawa, 2015). Therefore, even though small and medium size instances can be addressed directly with Mixed-Integer Linear Programming (MILP) solvers like CPLEX (cplex126, 2015) and GUROBI (Gurobi Optimization, LLC, 2024), larger instances may be challenging. For the sake of this work, directly solving the MILP was feasible, so we employed that approach. Even though the bottleneck of the procedure is represented by the expensive simulations, the scalability of the clustering computation can be achieved with heuristic approaches (see e.g. (Barber & Clark, 2009; Liu & Murata, 2010; Costa & Hansen, 2014)).

2.3. Building Design

The adoption of a performance-based approach for building design is increasingly essential to reduce energy consumption and advance decarbonization (Mollaoglu-Korkmaz, Swarup, & Riley, 2013). The widespread use of Building Performance Simulation (BPS) tools, such as EnergyPlus, has significantly improved the assessment of design options (Mourshed, Kelliher, & Keane, 2003). This process involves evaluating design alternatives to meet specific performance objectives while ensuring compliance with building regulations. However, traditional iterative trial-and-error approaches to exploring designs can be time-consuming and can overlook viable alternatives (Touloupaki & Theodosiou, 2017). In this context, the integration of mathematical optimization with numerical simulation represents a promising approach to enhance the exploration and evaluation of candidate design solutions (Talami, Wright, & Howard, 2025).

Performance-driven building design spans multiple domains, including architectural configurations (e.g., building geometry and fabric), HVAC system design, and operational strategies. This inherently multidisciplinary task requires the collaboration of various stakeholders, such as architects, engineers, building managers, and owners. As a result, the design process is often complex, as it involves multiple decision-makers dealing with interdependent variables, strict constraints, and possibly competing objectives (Talami, Wright, & Howard, 2021). In addition, the high dimensionality of real-world design problems requires the evaluation of a large number of potential solutions. The large volume of output data can overwhelm stakeholders, which can affect understanding of the relationships between design parameters and performance outcomes (Talami & Jakubiec, 2019). Finally, simulation outputs are often not easily interpretable, lacking actionable insights and interactivity, hindering their practical use in decision-making.

These challenges highlight the necessity of developing computational frameworks that help identify well-performing solutions, mitigating the computational demand associated with extensive performance evaluations while enhancing the interpretability of the results.

3. Methodology

The proposed framework, INFERNO, is characterized by the following process:

- Step 1: an efficient algorithm for simulation-based (derivative-free) optimization should be applied to generate a first pool of high-quality solutions. The selected one is RBFOpt: in addition to providing excellent performance, it allows to model categorical variables without bias and provide as a byproduct a surrogate model that can be used for fast what-if scenario evaluation, if needed;

- Step 2: an unsupervised learning method is used to identify similar patters in the best solutions found at the previous step. To achieve that, the solution space is transformed into a bipartite network and then clustering is performed through bipartite modularity optimization. This ensure results to be visualized regardless of the dimension of the problem;

- Step 3: a set of *inference* rules is generated. These help produce useful insights from the clusters identified in the previous step and support decision-makers in selecting solutions aligned with their requirements. Such rules will be discussed in detail in this section.

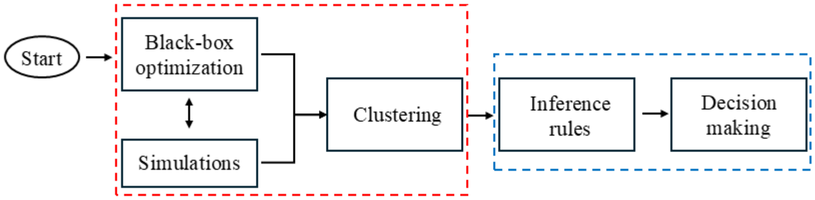

Figure 1: The main components of the INFERNO framework. The initial step (red dashed rectangle) including black-box optimization and bipartite network clustering produces the input for the novel inference rules generation procedure. Such rules are then used to support decision-makers by reducing the explorable domain while providing explainability (blue dashed rectangle).

Figure 1 illustrates the functioning of INFERNO. To explain how the method works, we consider in the following a toy example including three variables, i.e., $\boldsymbol{x} \in \{0,1\} \times [2.5, 10.5] \times \{1,2\}$, and an objective function to be optimized. As explained earlier, the first step is to find high-quality solutions. This is the most computationally intensive part of the process, as each sample needs to be evaluated through an expensive simulation to obtain the corresponding objective function value. Afterwards, the best solutions (according to the objective function) among those identified by the optimization process are selected. The number of solutions depends on the application and the dimension of the problem. For the sake of clarity, in this toy example, we assume that we have identified a set S^* of four good solutions.

In the next step, the selected solutions are transformed into a bipartite graph to run the bipartite modularity optimization algorithm. The transformation is as follows. Let the elements of S^* be characterized by $\forall i \in \{1, \ldots, |S^*|\}\, \boldsymbol{x}_i \in \mathbb{Z}^q \times \mathbb{R}^{n-q}$. For each of these solutions, a blue node is created. On the other hand, for each $j \in \{1, \ldots, q\}$ (that is, the discrete or categorical variables), if the number of possible choices associated with the variable j is d_j, then d_j red vertices are created. In addition, for each $j \in \{i+1, \ldots, n\}$ (i.e., the continuous variables), the corresponding variable domain is split into d_j intervals (whose size depends on the application) and again d_j red vertices are created. In practice, red vertices are associated with variable values, while blue vertices are associated with solutions found. Finally, each blue vertex is connected through the edges to the corresponding red vertices according to the values of $\boldsymbol{x}_{i,j}$. To illustrate the procedure, let $S^* = \{(0, 3.3, 2), (0, 5.5, 2), (1, 8.0, 2), (1, 9.5, 2)\}$, so we have four blue vertices labelled “sol 1”, “sol 2”, “sol 3”, and “sol 4”, respectively. If the domain of the continuous variable is divided into three sets of low values ($[2.5, 5[$), medium values ($[5, 8[$), and high values ($[8, 10.5]$), there are seven red vertices corresponding to the possible values of the three variables. The corresponding bipartite graph is represented in Figure 2. Note that some red vertices are placed on the right to indicate the values of variables always selected by the solutions (“x_3 2”) or never selected (“x_3 1”).

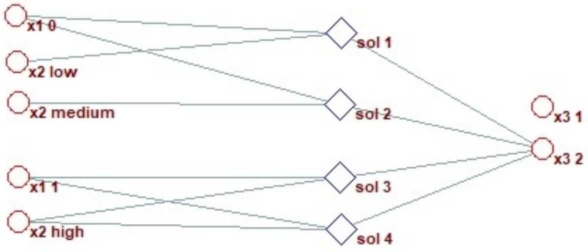

Figure 2: Bipartite graph obtained from the toy example where $S^* = \{(0, 3.3, 2), (0, 5.5, 2), (1, 8.0, 2), (1, 9.5, 2)\}$. Red vertices are circle-shaped, while blue vertices are diamond-shaped.

Given the bipartite network of Figure 2, the bipartite modularity optimization Problem (5)-(9) can be solved. The optimal solution, i.e., which pairs of vertices belong to the same cluster, allows to identify three clusters, represented in Figure 3 by yellow, green, and red colors.

The second cluster, including nodes with labels “sol 3” and “sol 4”, does not have any common vertex with the first one, except for node “x_3 2”. The third cluster associated with variable x_3 is also interesting because it includes a red vertex connected to all the blue vertices and another red vertex without connections. However, it may not be straightforward to identify even these simple patterns, especially when the size of the network increases.

The third step is the derivation of the inference rules from the analysis of the clusters. Each cluster that includes at least one red and one blue vertex yields

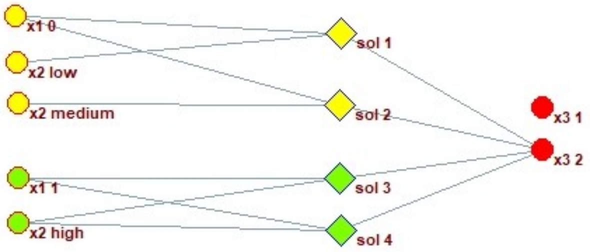

Figure 3: The three clusters obtained by applying the bipartite modularity optimization to the network of Figure 2. The first one (yellow color online) includes vertices with labels "x_1 0", "x_2 low", "x_2 medium", "sol 1", and "sol 2". The second one (green color online) includes vertices with labels "x_1 1", "x_2 high", "sol 3", and "sol 4". The third cluster is the special case with only red vertices, i.e., "x_3 0" and "x_3 1". Red vertices are circle-shaped and blue vertices are diamond-shaped.

one inference rule. It is important to highlight this fact, since a cluster with only red vertices that are linked to none or all blue vertices represents variable values that are never (or always) chosen. Such a cluster will not be used to generate an inference rule, but to make more specific all the others.

The inference rules generation can be summarized as follows. Let the domain of each of the n variables of the problem be divided into intervals, as explained above, where d_j is the number of intervals of the variable with index j. To make the discussion clearer, we refer to $l_{j,k}$, $k \in \{1, \ldots, d_j\}$, as the red vertex that indicates the k -th interval of the variable j. For example, in the network of Figure 2, the vertex with the label "x_1 0" would be associated with $l_{1,1}$. The vertex with the label "x_2 low" would be identified by $l_{2,1}$ instead. For categorical variables, any order can be selected.

- *Uniformity Principle*: if there exists a cluster including only red vertices, and in such a cluster there is a vertex $l_{j,k}$ connected to all blue vertices, then for all inference rules the only valid value of variable j is its k-th interval;

- *Exclusion Principle*: if there exists a cluster including only red vertices, and in such a cluster there is a vertex $l_{j,k}$ connected to no blue vertex where variable j was not previously affected by the *Uniformity Principle*, then for all inference rules the k-th interval of variable j is removed from its domain;

- *Generation Principle*: for each cluster including at least a blue vertex, the valid values for each variable j are those associated with nodes $l_{j,k}$ in the cluster. If for some variable j there are no red vertices, then the valid values are those that remain after domain reduction due to the *Uniformity Principle* and the *Exclusion Principle*.

This way, each cluster with at least one blue vertex produces a set of conditions, i.e., domain-reduction constraints for the problem variables, that allows identifying a subset of solutions. We call them *inference* rules. We can see how these rules are derived from the example in Figure 3.

- *Uniformity Principle*: as the vertex "x_3 2" is linked to all blue vertices, then in all inference rules the value of x_3 is set to 2.
- *Exclusion Principle*: this does not apply because the domain of x_3 has already been set in the previous step.
- *Generation Principle*: The first cluster is associated with $x_1 = 0$, x_2 low or medium, and $x_3 = 2$. The second cluster is associated with $x_1 = 1$, x_2 set to high and $x_3 = 2$. The corresponding inference rules can be written as follows:
  - $\mathcal{I}_1 := \{(x_1 = 0) \wedge (x_2 \in [2.5, 8[) \wedge (x_3 = 2)\}$
  - $\mathcal{I}_2 := \{(x_1 = 1) \wedge (x_2 \in [8, 10.5]) \wedge (x_3 = 2)\}$.

The rules can be interpreted this way. When looking for good solutions according to the objective function of the derivative-free problem in Step 1, one should restrict the search space. In this example, either looking for solutions with $x_1 = 0$, $x_2 < 8$, and $x_3 = 2$, or solutions with $x_1 = 1$, $x_2 \geq 8$, and $x_3 = 2$.

4. Experimental Setting

In this section, we discuss how the experiments were designed. INFERNO was compared with a classification tree in terms of the Price of Explainability (PoX). These are discussed in this section, while a detailed evaluation of the results is presented in the next section.

4.1. Problems Description

We consider two problems, based on a case study building. This is a simulated 5-zone open-plan single-story office with a total floor area of 100 m^2 and a floor-to-ceiling height of 2.7 meters. It is nominally located in Nottingham, UK. The façades incorporate operable ribbon windows without internal or external shading, allowing natural ventilation and free cooling during the summer period. A drawing of the building is presented in Figure 4

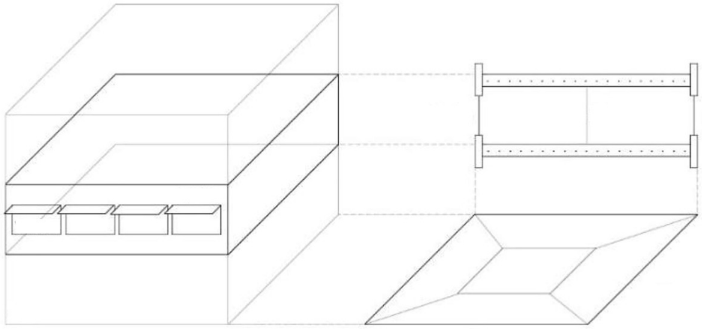

Figure 4: Axonometric, plan, and section view of the open-plan office floor subdivided into five thermal zones.

The two problems evaluate the performance of candidate design solutions based on two performance objectives. The total heating energy demand, expressed in kWh/year, represents the energy consumption of the Heating, Ventilation and Air Conditioning (HVAC) system required to meet the heating needs of the space during occupied periods. Cooling is not considered, as it is typically negligible in this climate. The total discomfort hours are defined as the number of occupied hours during which the Predicted Mean Vote (PMV) index, based on Fanger's comfort model (Fanger, 1970), exceeds 0.5 or falls below -0.5. The goal of the optimization process is to minimize the heating energy demand and the number of occupied hours that fall outside the defined comfort zone.

The design space comprises six key variables, summarized in Table 1, selected due to their impact on the performance criteria.

The aspect ratio controls the building geometry and defines the overall shape and compactness. Each of the selected options increases the envelope wall area by 10% from a baseline square shape with a ratio of 1:1 while fixing the building volume, resulting in progressively higher heat gains and losses due to the larger area of the exterior wall. The window-to-wall ratio (WWR) determines the proportion between opaque and transparent surfaces and varies from a predominantly opaque facade (20% WWR) to a fully glazed facade (95% WWR). An interme-

Element	Variables	Domain	Short form
Geometry	Aspect ratio (AR)	1:1, 1:2.5, 1: 3.7, 1:4.8	1, 2.5, 3.7, 4.8
	Window-to-wall ratio (WWR)	20%, 50%, 95%	0.2, 0.5, 0.95
Fabric	Construction (FAB)	Lightweight Part L Lightweight Passivhaus Heavyweight Part L Heavyweight Passivhaus	L, LP, H, HP
HVAC	System (HVAC)	All-air, Radiant	A, R
Controls	Setpoint air temperature (SPT)	19 °C, 21 °C, 23 °C	19, 21, 23
	Setback air temperature (SBK)	10 °C, 13 °C ,16 °C	10, 13, 16

Table 1: Variables and domain for the building design case studies. The short version of variable names (in parenthesis) and the values listed under the "Short form" column are employed in the pictures of the networks and when deriving the inference rules.

diate value of 50% is also analyzed. The fabric of the building represents different material configurations and four types are selected based on (Talami, Wright, & Howard, 2020). Each option includes heavy-weight concrete and lightweight timber constructions, allowing for an assessment of thermal mass impact. Two baseline options comply with the Approved Document L2A of the UK Building Regulations, while two options display higher thermal performance and meet the Passivhaus Standard. In this study, each option includes glazing, walls, and roof, as a complete system.

Two HVAC systems are analyzed. A radiant system consisting of a Floor Embedded Surface System (ESS) which covers over 50% of the space heating load through thermal radiation. The "all-air" system instead is a Constant Air Volume (CAV) system which uses convection-based heating. Heating setpoint and setback room air temperatures constitute the operational controls of the HVAC system, which influence energy use and comfort levels. Based on (Lush, Butcher, & Appleby, 2006), three scenarios are defined: cooler, average, and warmer, with corresponding values for the setpoint and setback variables to 19°C, 21°C, 23°C and 10°C, 13°C, 16°C, respectively.

4.2. Derivative-free Optimization Setting

The considered problems are associated with two objectives, i.e., energy consumption and thermal comfort. The INFERNO framework was applied to each problem separately.

The total number of possible solutions is 864, according to the variable domains in Table 1. The following RBFOpt settings were used. First, the maximum number of evaluations was set at 87, that is, approximately 10% of the size of the feasible solution space. Of the six problem variables, the third and fourth (FAB, HVAC) were considered categorical (i.e., discrete unordered), while the others were considered integer. After running RBFOpt, the top 25% of the solutions (according to the objective function) were kept, that is, 22 of them. This value was chosen to balance the quality of the solutions with the generation of a bipartite network of a reasonable size.

4.3. Comparison with a Classification Tree

To evaluate the effectiveness of the INFERNO inference rules, we made a comparison with another methodology. A possible choice was to employ K-means, as it was also used as a benchmark in (Proverbio et al., 2018b). However, this choice was discarded for a number of reasons. First, the features considered are discrete/categorical, making K-means not suitable for the task. Moreover, K-means did not allow for data visualization, as the number of features is 6. Some dimensionality reduction techniques, such as PCA, could have been used, but interpretability would have been affected.

To address the aforementioned challenges, we performed a comparison using a classification tree trained on the same data used to represent bipartite networks. In practice, the samples are the 22 blue nodes that represent the solutions in the bipartite network, and the features are the six discrete/categorical variables of Table 1. The target variable to be predicted is the cluster to which each sample belongs, according to the results of the optimization of bipartite modularity. In this way, it is possible to compare the inference rules with the classification tree rules that describe each cluster.

To implement the classification tree, the Python Scikit-learn library was used. The input data, for both energy consumption and thermal comfort, was the whole dataset of the corresponding 22 samples without train/test split. This choice was motivated by the fact that the dataset is small and to have a more fair comparison with the inference rule generation procedure, where the full dataset was used to derive the rules. To obtain the simplest tree, the number of leaves was increased until the clusters produced by bipartite modularity optimization were identified.

4.4. Price of Explainability

Starting from the clusters generated by bipartite modularity optimization, INFERNO derives inference rules for each cluster. Such rules (and those produced by a classification tree) identify candidate solutions to the problem under study.

However, the fact that such rules can be easily stated and interpreted has a price: the quality of the solutions identified by applying them can be lower than that of the set of solutions from the corresponding cluster. To quantify this phenomenon, we introduce the concept of price of explainability. Other authors considered this topic. For example, (Laber & Murtinho, 2021) addressed the price of explainability (PoE) in the context of clustering problems. For a minimization problem, PoE was defined as the maximum ratio, over a set of instances, of the optimal cost of an *explainable* partition over the optimal cost of an *unrestricted* partition. For a maximization problem, the definition is similar, but the numerator and denominator are swapped.

In this work, we propose a different approach. After the decision-maker selects a cluster of interest, the set of (explainable) solutions generated by the inference or classification tree rules should be used instead of the (unrestricted) solutions of that cluster. The Price of Explainability (PoX) of the cluster is a robust estimation of the relative performance decrease when selecting an explainable solution versus one of the corresponding unrestricted solutions. In practice, this approach is based on the robust nonparametric Hodges-Lehmann Estimator (Jr. & Lehmann, 1963).Let B be the set of cost values of the (explainable) solutions in cluster c. Also, let W be the set of cost values of the solutions generated from the rule associated with the cluster c. After the median of all pairwise differences between the two sets is computed, PoX is obtained by dividing this value by the median of the values in B (assuming that it is nonzero). In this way, we can estimate how much we can expect to lose when using solutions generated by the explainable set W versus the baseline set B. The pseudocode of PoX for a minimization problem is summarized by Algorithm 1:

Algorithm 1: Relative Hodges-Lehmann Estimator for PoX Calculation

```
  Input:
  $W$: Set of values from group $W$
  $B$: Set of values from group $B$ with Median$(B) \neq 0$
  Output:
  PoX: Price of Explainability (relative HLE)
1 Step 1: Compute all pairwise differences
2 $D \leftarrow \{w - b \mid w \in W, b \in B\}$ ; // All pairwise differences
3 Step 2: Calculate the Hodges-Lehmann Estimator
4 $H \leftarrow$ Median$(D)$ ;             // Hodges-Lehmann Estimator
5 Step 3: Normalize by the median of $B$
6 $M_B \leftarrow$ Median$(B)$
7 PoX $\leftarrow \frac{H}{M_B}$
8 return PoX
```

It is important to note that PoX has some limitations. As mentioned, if the median of B is zero, we should instead use the absolute value and return the H value of Step 6 in the algorithm. However, this is quite unlikely to happen. In addition, there is a tendency to obtain lower values if the size of the set W is small, as can be observed in the results of the next section. Although this may not be a direct consequence of the main intuition behind the definition of PoX, the result can be explained as the decision-making task is more straightforward with a smaller number of options.

5. Results

The bipartite networks for the energy consumption and thermal comfort test cases were created according to the procedure described earlier. After that, the bipartite modularity optimization problem was solved to identify clusters. Rules were derived using both INFERNO and the classification tree. Finally, they were compared in terms of quality and PoX. We present separately the two cases in the following.

5.1. Test Case 1: Energy Consumption

The clusters obtained in the energy consumption case are shown in Figure 5.

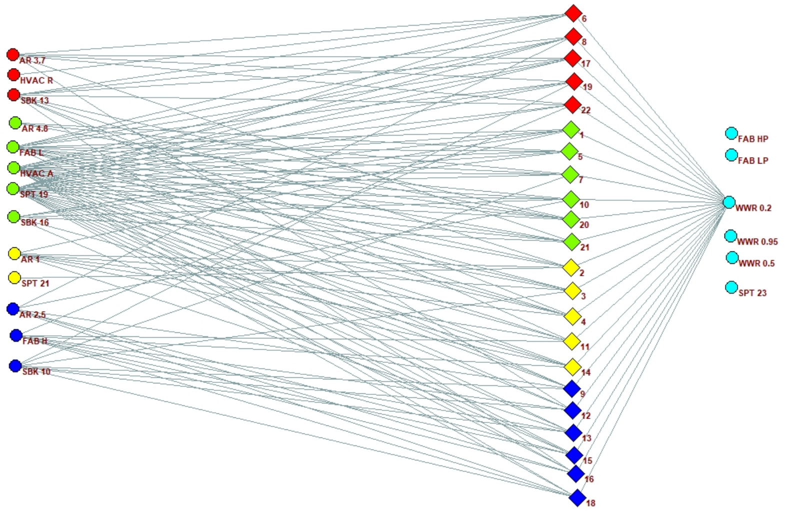

Figure 5: Clusters identified from the bipartite network representing the best energy consumption solutions. Vertices depicted with the same color belong to the same cluster.

It can be observed that five clusters were identified, one of them being the special cluster that included only vertices connected to all blue vertices or to none of them. Recall that red nodes, according to the bipartite network terminology, are circle shaped and represent variable values, whereas blue nodes are diamond shaped and represent solutions. The colors of the nodes in the figures refer instead to the clusters.

To evaluate the baseline results of the clusters, we computed the number of solutions in each cluster. We also checked the quality of the best and worst of them compared to the global best solution of the whole domain (obtained by simulating all the possible 864 solutions). The quality of such solutions is summarized in Table 2.

Next, we apply the INFERNO framework. According to the procedure described earlier, for each of the non-special clusters an inference rule is generated. The procedure is summarized in the following.

- *Uniformity Principle*: as the vertex WWR 0.2 is linked to all blue vertices, then in all inference rules the value of WWR is set to 0.2.

- *Exclusion Principle*: The vertices FAB HP, FAB LP, and SPT 23 are never linked to the blue nodes, so these values are removed from the domain of the corresponding variables. Vertices WWR 0.5 and WWR 0.95 do not need

Rule	# Solutions	Best Case %	Worst Case %
$\mathcal{I}_1$	5	99.7	97.3
$\mathcal{I}_2$	6	99.3	97.9
$\mathcal{I}_3$	5	99.9	97.6
$\mathcal{I}_4$	6	99.7	97.5

Table 2: Energy consumption results, original clusters. For each cluster, the number of solutions in that clusters. Also, among those solutions, how the best and worst of them compare versus the rest of the solutions of the domain (e.g., 99.7% means that the solution is better than 99.7% of the solutions in the domain).

to be considered here, as the variable WWR was already fixed to 0.2 by the previous principle. At this point, the domains of the variables WWR, FAB and SPT are: $\mathcal{D}(\text{WWR}) = \{0.2\}$, $\mathcal{D}(\text{FAB}) = \{\text{L}, \text{H}\}$, $\mathcal{D}(\text{SPT}) = \{19, 21\}$, respectively.

- *Generation Principle*:
  - the first cluster (red) has AR 3.7, HVAC R, SBK 13. By considering the updated domains of the other variables, we have $\mathcal{I}_1 := \{(\text{AR} = 3.7) \wedge (\text{WWR} = 0.2) \wedge (\text{FAB} \in \{\text{L}, \text{H}\}) \wedge (\text{HVAC} = \text{R}) \wedge (\text{SPT} \in \{19, 21\}) \wedge (\text{SBK} = 13)\}$;
  - the second cluster (green) fixes all variables, so the corresponding inference rule is $\mathcal{I}_2 := \{(\text{AR} = 4.6) \wedge (\text{WWR} = 0.2) \wedge (\text{FAB} = \text{L}) \wedge (\text{HVAC} = \text{A}) \wedge (\text{SPT} = 19) \wedge (\text{SBK} = 16)\}$;
  - the third cluster (yellow) has AR 1 and SPT 21, therefore $\mathcal{I}_3 := \{(\text{AR} = 1) \wedge (\text{WWR} = 0.2) \wedge (\text{FAB} \in \{\text{L}, \text{H}\}) \wedge (\text{HVAC} \in \{\text{A}, \text{R}\}) \wedge (\text{SPT} = 21) \wedge (\text{SBK} \in \{10, 13, 16\})\}$;
  - the fourth cluster (blue) has AR 2.5, FAB H, SBK 10, and its inference rule is $\mathcal{I}_4 := \{(\text{AR} = 2.5) \wedge (\text{WWR} = 0.2) \wedge (\text{FAB} = \text{H}) \wedge (\text{HVAC} \in \{\text{A}, \text{R}\}) \wedge (\text{SPT} \in \{19, 21\}) \wedge (\text{SBK} = 10)\}$.

We then perform a similar analysis to that of Table 2 considering the solutions generated by the inference rules. Results are reported in Table 3.

From the results, it appears that the number of solutions for each inference rule is in the range of 1 to 12, where the smaller size refers to rules that restrict the domain of the variables more. The quality is also in general high, even for the worst-case solution. However, it is not as high as that of the original clusters -

Rule	# Solutions	Best Case %	Worst Case %
$\mathcal{I}_1$	4	97.3	86.9
$\mathcal{I}_2$	1	98.0	98.0
$\mathcal{I}_3$	12	97.6	91.1
$\mathcal{I}_4$	4	99.0	89.5

Table 3: Energy consumption results, inference rules. For each inference rule, the number of solutions it allows to identify. Also, among those identified solutions, how the best and worst of them compare versus the rest of the solutions of the domain.

as was expected since explainability is likely to affect quality. The distribution of the energy consumption values and the best and worst solutions for each inference rule are shown in Figure 6.

We then generated rules using a classification tree. The tree was trained with the objective of classifying the 22 solutions in Figure 5 in the four corresponding clusters. To find the easier-to-explain rules, we increased the number of leaves until the 4 clusters could be identified. To achieve this, 4 leaves were necessary. The identified rules are as follows:

- first cluster (red): $\mathcal{I}_1 := \{(\text{AR} \in \{2.5, 3.7\}) \wedge (\text{FAB} = \text{L})\}$;
- second cluster (green): $\mathcal{I}_2 := \{(\text{AR} = 4.8) \wedge (\text{FAB} = \text{L})\}$;
- third cluster (yellow): $\mathcal{I}_3 := \{(\text{AR} = 1)\}$;
- fourth cluster (blue): $\mathcal{I}_4 := \{(\text{AR} \in \{2.5, 3.7, 4.8\}) \wedge (\text{FAB} \in \{\text{LP}, \text{H}, \text{HP}\})\}$.

The first major difference between classification-tree-based rules and the INFERNO inference rules is that the former does not reduce the explorable domain, as shown in Table 4.

Rule	# Solutions
$\mathcal{I}_1$	108
$\mathcal{I}_2$	54
$\mathcal{I}_3$	216
$\mathcal{I}_4$	486

Table 4: Energy consumption results from the classification tree. For each classification tree rule, the number of solutions it allows to generate. In total, the whole domain (864 solutions) was identified.

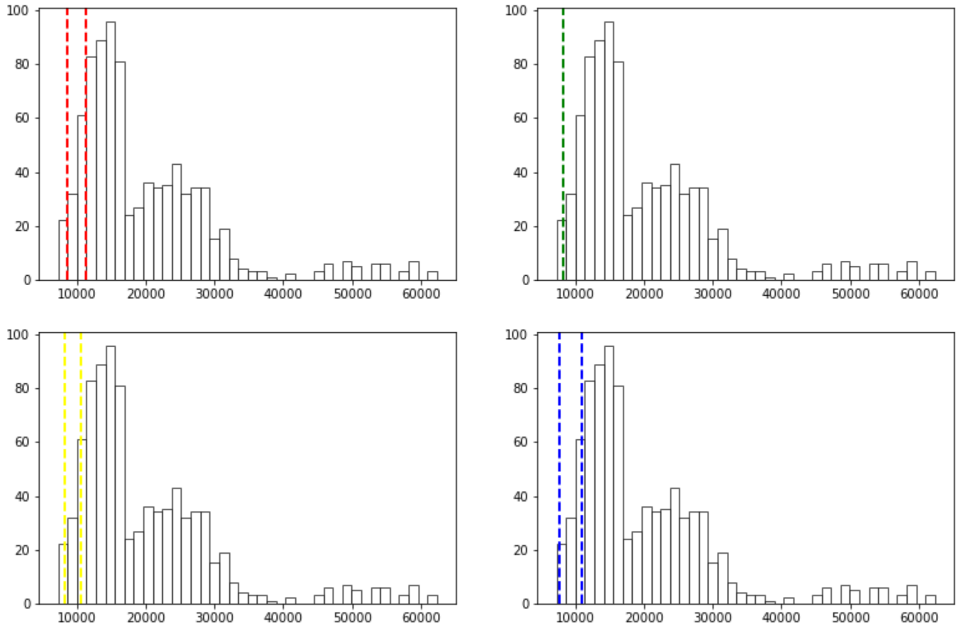

Figure 6: Energy consumption results. Distribution of energy values for the entire set of solutions, with indication for each cluster, in different colors, of the best and worst case solutions identified by the corresponding inference rules (cluster 1, red, is on the top-left; cluster 2, green, is on the top-right; cluster 3, yellow, is on the bottom-left; cluster 4, blue, is on the bottom-right).

To better and further assess the differences between the INFERNO inference rules and the classification tree rules, we computed the PoX. The reference group (B) is that of the original clusters. The results are shown in Table 5.

It is clear that the inference rules outperform the classification tree rules. The best PoX value for the inference rules is associated with $\mathcal{I}_2$. This may be related to the fact that the associated rule identified a single solution. The overall inference rules PoX is around 4.5 times smaller than that of the classification tree rules.

5.2. *Test Case 2: Thermal comfort*

Looking at the thermal comfort optimization case, the obtained clusters are those in Figure 7. The number and quality of the solutions associated with the clusters are presented in Table 6.

The inference rule derivation procedure for the 4 clusters is summarized in the following.

- *Uniformity Principle*: as the vertex WWR 0.2 is linked to all blue vertices,

Rule	PoX - Inference Rules	PoX - Classification Tree
$\mathcal{I}_1$	27.75	67.73
$\mathcal{I}_2$	0.49	77.16
$\mathcal{I}_3$	25.56	85.97
$\mathcal{I}_4$	29.12	143.46

Table 5: Comparison of Price of Explainability (PoX) for INFERNO inference rules and classification tree rules for the energy consumption case. The reference group for PoX computation is the set of solutions from the original clusters. The sum of PoX values of the INFERNO inference rules is approximately 4.5 times smaller than that of the classification tree rules.

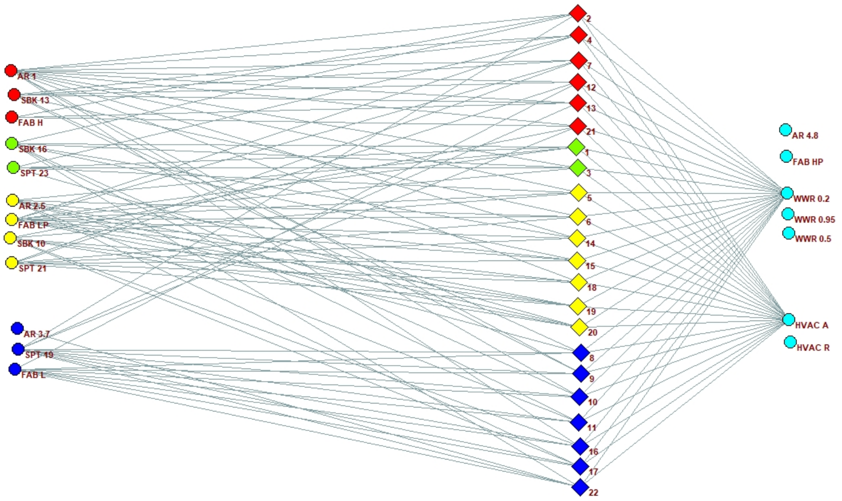

Figure 7: Clusters identified from the bipartite network representing the best thermal comfort solutions. Vertices depicted with the same color belong to the same cluster.

then in all inference rules the value of WWR is set to 0.2. For a similar reason, HVAC is set to A.

- *Exclusion Principle*: Vertices FAB HP and AR 4.6 are never linked to blue nodes, so these values are removed from the domain of the corresponding variables. Vertices WWR 0.5, WWR 0.95, and HVAC R do not need to be considered here, since variables WWR and HVAC were already fixed to 0.2 and A, respectively, by the previous principle. After that, the domains of the variables WWR, FAB, and HVAC are: $\mathcal{D}(\text{WWR}) = \{0.2\}$, $\mathcal{D}(\text{FAB}) = \{\text{L}, \text{LP}, \text{H}\}$, $\mathcal{D}(\text{HVAC}) = \{\text{R}\}$, respectively.

- *Generation Principle*: following the same procedure as in the energy case, we obtain the following rules.
  - for the first cluster (red), $\mathcal{I}_1 := \{(\text{AR} = 1) \wedge (\text{WWR} = 0.2) \wedge (\text{FAB} = \text{H}) \wedge (\text{HVAC} = \text{A}) \wedge (\text{SPT} \in \{19, 21, 23\}) \wedge (\text{SBK} = 13)\}$;
  - for the second cluster (green), $\mathcal{I}_2 := \{(\text{AR} =\in \{1, 2.5, 3.7\}) \wedge (\text{WWR} = 0.2) \wedge (\text{FAB} \in \{\text{L}, \text{H}, \text{LP}\}) \wedge (\text{HVAC} = \text{A}) \wedge (\text{SPT} = 23) \wedge (\text{SBK} = 16)\}$;
  - for the third cluster (yellow), $\mathcal{I}_3 := \{(\text{AR} = 2.5) \wedge (\text{WWR} = 0.2) \wedge (\text{FAB} = \text{LP}) \wedge (\text{HVAC} = \text{A}) \wedge (\text{SPT} = 21) \wedge (\text{SBK} = 10)\}$;
  - for the fourth cluster (blue), $\mathcal{I}_4 := \{(\text{AR} = 3.7) \wedge (\text{WWR} = 0.2) \wedge (\text{FAB} = \text{L}) \wedge (\text{HVAC} = \text{A}) \wedge (\text{SPT} = 19) \wedge (\text{SBK} \in \{10, 13, 16\})\}$.

Rule	# Solutions	Best Case %	Worst Case %
$\mathcal{I}_1$	6	99.4	97.5
$\mathcal{I}_2$	2	98.1	97.6
$\mathcal{I}_3$	7	99.5	98.0
$\mathcal{I}_4$	7	99.8	97.7

Table 6: Thermal comfort results, original clusters. For each cluster, the number of solutions in that clusters. Also, among those solutions, how the best and worst of them compare versus the rest of the solutions of the domain (e.g., 99.4% means that the solution is better than 99.4% of the solutions in the domain).

The results of the solutions identified by the inference rules associated with the clusters are depicted in Table 7.

Rule	# Solutions	Best Case %	Worst Case %
$\mathcal{I}_1$	3	98.0	85.1
$\mathcal{I}_2$	9	98.1	83.7
$\mathcal{I}_3$	1	98.4	98.4
$\mathcal{I}_4$	3	97.7	97.2

Table 7: Thermal comfort results, inference rules. For each inference rule, the number of solutions it allows to identify. Also, among those identified solutions, how the best and worst of them compare versus the rest of the solutions of the domain.

The results show that, as for the energy consumption case, the quality of the solutions when using the inference rules is slightly lower than that of the original cluster solutions. At the same time, the results seem to be worse than those obtained in the energy consumption case. This is also confirmed by comparing Figures 6 and 8.

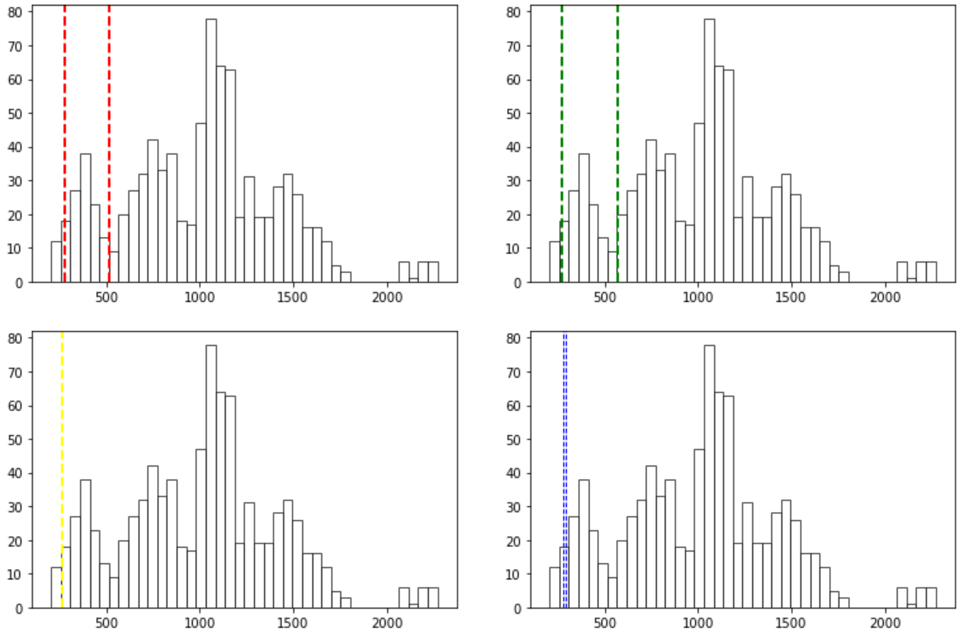

Figure 8: Thermal comfort results. Distribution of energy values for the entire set of solutions, with indication for each cluster, in different colors, of the best and worst case solutions identified by the corresponding inference rules (cluster 1, red, is on the top-left; cluster 2, green, is on the top-right; cluster 3, yellow, is on the bottom-left; cluster 4, blue, is on the bottom-right).

As in the previous case, a classification tree was also trained to classify the 22 solutions in Figure 7. In this case, 6 leaves were necessary. The associated rules to describe the clusters are as follows:

- first cluster (red): $\mathcal{I}_1 := \{(\text{FAB} = \text{L}) \wedge (\text{STP} \in \{21, 23\})\} \vee \{(\text{FAB} \in \{\text{LP}, \text{H}, \text{HP}\}) \wedge (\text{STP} = 19)\}$;
- second cluster (green): $\mathcal{I}_2 := \{(\text{FAB} \in \{\text{LP}, \text{H}, \text{HP}) \wedge (\text{STP} = 23) \wedge (\text{SBK} \in \{13, 16\})\}$;
- third cluster (yellow): $\mathcal{I}_3 := \{(\text{FAB} \in \{\text{LP}, \text{H}, \text{HP}) \wedge (\text{STP} \in \{21, 23\}) \wedge (\text{SBK} = 10)\} \vee \{(\text{FAB} \in \{\text{LP}, \text{H}, \text{HP}\}) \wedge (\text{STP} = 21) \wedge (\text{SBK} \in \{13, 16\})\}$;
- fourth cluster (blue): $\mathcal{I}_4 := \{(\text{FAB} = \text{L}) \wedge (\text{STP} = 19)\}$.

Note that, unlike in the case of energy consumption, there were sometimes multiple leaves of the tree associated with the same cluster (that is, the reason for the OR operator $\vee$ in clusters 1 and 3). This is already an element that can complicate the decision process because it introduces additional choices. Moreover, as shown in Table 8, the domain reduction ability of the inference rules is not a feature of the classification tree rules.

Rule	# Solutions
$\mathcal{I}_1$	144+216
$\mathcal{I}_2$	144
$\mathcal{I}_3$	144+144
$\mathcal{I}_4$	72

Table 8: Thermal comfort results from the classification tree. For each classification tree rule, the number of solutions it allows to generate. The ”+” in the thermal comfort column is used for clusters identified by two set of rules linked by the $\vee$ symbol. In total, the whole domain (864 solution) was identified.

Finally, we compare the Price of Explainability of the inference rules and the classification tree rules in Table 9.

Rule	PoX - Inference Rules	PoX - Classification Tree
$\mathcal{I}_1$	46.23	319.73
$\mathcal{I}_2$	43.99	295.87
$\mathcal{I}_3$	0.0	311.99
$\mathcal{I}_4$	23.40	254.04

Table 9: Comparison of Price of Explainability (PoX) for INFERNO inference rules and classification tree rules for the thermal comfort case. The reference group for PoX computation is the set of solutions from the original clusters. The sum of PoX values of the INFERNO inference rules si approximately 10 times smaller that that of the classification tree rules.

From the results, the PoX of the inference rules is worse than that of the energy consumption case. At the same time, the results of the classification tree are much worse. In fact, the overall PoX of the inference rules is more than an order of magnitude smaller than that of the classification tree rules. It is also interesting to observe that the PoX of the inference rules of cluster 3 is zero. This is compatible with the remark made in the case of energy consumption that rules associated with one solution provide very small PoX values.

6. Conclusions and Future Work

In this paper a new framework - referred to as INFERNO - based on derivative-free optimization and bipartite clustering is employed to address expensive simulation-based optimization problems. First, a set of high-quality solutions is computed with RBFOpt, an RBF-based optimization solver. The solutions are then used to generate a bipartite network, where bipartite modularity optimization is applied to identify clusters. Finally, a novel procedure is proposed to derive inference rules from the clusters. Such rules allow decision-makers to efficiently explore the domain of the problem while searching for alternative solutions.

The INFERNO framework offers several advantages:

- Visualization of results: the use of bipartite networks and clustering enables effective visualization regardless of the problem's dimensionality;
- Improved explainability costs: the inference rules are easier to explain than classification-tree rules (for example, they do not have `OR` conditions inside) and achieve better PoX scores;
- Domain reduction: the generated inference rules effectively reduce the problem domain to a manageable size for decision-makers who want to explore alternative solutions with shared features. In contrast, the classification trees produced less specific rules and did not achieve domain reduction.

Nonetheless, the bipartite modularity optimization procedure could be time-consuming for larger networks. Even though the bottleneck is usually the expensive simulations in Step 1, the issue can be mitigated using heuristics like (Costa & Hansen, 2014).

Future work includes the extension to multiobjective optimization, where two or more performance metrics should be considered together, and the inclusion of uncertainty affecting some of the parameters of interest.

Acknowledgements

This work is an outcome of the Future Resilient Systems project at the Singapore-ETH Centre (SEC) supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. One of the authors (A.C.) would like to dedicate this article to the memory of Prof. Pierre Hansen, who passed away in January 2025 and introduced the author to the fascinating world of clustering.

References

Angelov, P. P., Soares, E. A., Jiang, R., Arnold, N. I., & Atkinson, P. M. (2021). Explainable artificial intelligence: an analytical review. *WIREs Data Mining and Knowledge Discovery*, *11*(5), e1424.

Barber, M. J. (2007). Modularity and community detection in bipartite networks. *Physical Review E*, *76*(6), 066102.

Barber, M. J., & Clark, J. W. (2009). Detecting network communities by propagating labels under constraints. *Phys. Rev. E*, *80*, 026129.

Cafieri, S., Caporossi, G., Hansen, P., Perron, S., & Costa, A. (2012). Finding communities in networks in the strong and almost-strong sense. *Physical Review E*, *85*(4), 046113.

Chakraborty, S., Kiefer, P., & Raubal, M. (2024). Estimating perceived mental workload from eye-tracking data based on benign anisocoria. *IEEE Transactions on Human-Machine Systems*, *54*(5), 499-507.

Costa, A., Buccio, E. D., Melucci, M., & Nannicini, G. (2018). Efficient parameter estimation for information retrieval using black-box optimization. *IEEE Transactions on Knowledge and Data Engineering*, *30*(7), 1240-1253.

Costa, A., & Hansen, P. (2014). A locally optimal hierarchical divisive heuristic for bipartite modularity maximization. *Optimization letters*, *8*, 903–917.

Costa, A., & Nannicini, G. (2018). RBFOpt: an open-source library for black-box optimization with costly function evaluations. *Mathematical Programming Computation*, *10*, 597 - 629.

De Bock, K. W., Coussement, K., Caigny, A. D., Słowiński, R., Baesens, B., Boute, R. N., ... Weber, R. (2024). Explainable ai for operational research: A defining framework, methods, applications, and a research agenda. *European Journal of Operational Research*, *317*(2), 249-272.

Diaz, G. I., Fokoue-Nkoutche, A., Nannicini, G., & Samulowitz, H. (2017). An effective algorithm for hyperparameter optimization of neural networks. *IBM Journal of Research and Development*, *61*(4/5), 9:1-9:11.

Ding, W., Abdel-Basset, M., Hawash, H., & Ali, A. M. (2022). Explainability of artificial intelligence methods, applications and challenges: A comprehensive survey. *Information Sciences*, *615*, 238-292.

Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. *Computer Science Review*, *40*, 100379.

Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., ... Ranjan, R. (2023). Explainable AI (XAI): Core Ideas, Techniques, and Solutions. *ACM Computing Surveys*, *55*(9).

Fanger, P. O. (1970). *Thermal comfort. analysis and applications in environmental engineering.* Copenhagen: Danish Technical Press.

Gallegos, I. O., Rossi, R. A., Barrow, J., Tanjim, M. M., Kim, S., Dernoncourt, F., ... Ahmed, N. K. (2024). Bias and fairness in large language models: A survey. *Computational Linguistics*, *50*(3), 1097–1179.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., ... Wang, H. (2024). *Retrieval-augmented generation for large language models: A survey.*

Girvan, M., & Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, *99*(12), 7821-7826.

Gurobi Optimization, LLC. (2024). *Gurobi Optimizer Reference Manual.*

Gutmann, H.-M. (2001). A radial basis function method for global optimization. *Journal of Global Optimization*, *19*, 201-227.

Hu, Y., Chen, H., Zhang, P., Li, M., Di, Z., & Fan, Y. (2008). Comparative definition of community and corresponding identifying algorithm. *Physical Review E*, *78*(2), 026121.

IBM ILOG. (2015). IBM ILOG CPLEX 12.6 User's Manual [Computer software manual]. Gentilly, France.

Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, *13*, 455-492.

Jr., J. L. H., & Lehmann, E. L. (1963). Estimates of Location Based on Rank Tests. *The Annals of Mathematical Statistics*, *34*(2), 598 – 611.

Laber, E. S., & Murtinho, L. (2021). On the price of explainability for some clustering problems. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning* (Vol. 139, pp. 5915–5925). PMLR.

Liu, X., & Murata, T. (2010). An efficient algorithm for optimizing bipartite modularity in bipartite networks. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, *14*(4), 408-415.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, *28*(2), 129-137.

Lush, D., Butcher, K., & Appleby, P. (2006). *Environmental design: CIBSE Guide A.*

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In L. M. L. Cam & J. Neyman (Eds.), *Proc. of the fifth berkeley symposium on mathematical statistics and probability* (Vol. 1, p. 281-297). University of California Press.

Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2024). *Large language models: A survey.*

Miyauchi, A., & Sukegawa, N. (2015). Maximizing barber's bipartite modularity is also hard. *Optimization Letters*, *9*(5), 897–913.

Mollaoglu-Korkmaz, S., Swarup, L., & Riley, D. (2013). Delivering sustainable, high-performance buildings: Influence of project delivery methods on integration and project outcomes. *Journal of Management in Engineering*, *29*, 71-78.

Mourshed, M., Kelliher, D., & Keane, M. (2003). Integrating building energy simulation in the design process. *IBPSA News*, *13*, 21-26.

Nannicini, G. (2021). On the implementation of a global optimization method for mixed-variable problems. *Open Journal of Mathematical Optimization*, *2*, 1–25.

Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E.*, *69*, 026113.

Proverbio, M., Costa, A., & Smith, I. F. C. (2018a). Adaptive sampling methodology for structural identification using radial-basis functions. *Journal of Computing in Civil Engineering*, *32*(3), 04018008.

Proverbio, M., Costa, A., & Smith, I. F. C. (2018b). Sensor data interpretation with clustering for interactive asset-management of urban systems. *Journal of Computing in Civil Engineering*, *32*(6), 04018050.

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., & Parisi, D. (2004). Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the U.S.A.*, *101*(9), 2658-2663.

Regis, R. G., & Shoemaker, C. A. (2007). A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing*, *19*(4), 497-509.

Talami, R., & Jakubiec, J. A. (2019). Sensitivity of design parameters on energy, system and comfort performances for radiant cooled office buildings in the tropics. In *Proceedings of building simulation 2019: 16th conference of ibpsa* (Vol. 16, pp. 1786–1793). Rome, Italy: IBPSA.

Talami, R., Wright, J., & Howard, B. (2020). A comparison between sequential and simultaneous whole-building design optimization for building performance. In *Proceedings of building simulation and optimization conference.* Loughborough, UK: IBPSA.

Talami, R., Wright, J., & Howard, B. (2021). Multi-criteria robustness assessment of a sequential whole-building design optimization. In *Proceedings of building simulation 2021: 17th conference of ibpsa* (Vol. 17, pp. 2015–

2022). Bruges, Belgium: IBPSA.

Talami, R., Wright, J., & Howard, B. (2025). Evaluating the effectiveness, reliability and efficiency of a multi-objective sequential optimization approach for building performance design. *Energy and Buildings*, *329*, 115242.

Touloupaki, E., & Theodosiou, T. (2017). Performance simulation integrated in parametric 3d modeling as a method for early stage design optimization—a review. *Energies*, *10*(5).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. In I. Guyon et al. (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc.

Wang, J., Li, Y., Gao, R. X., & Zhang, F. (2022). Hybrid physics-based and data-driven models for smart manufacturing: Modelling, simulation, and explainability. *Journal of Manufacturing Systems*, *63*, 381-391.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., … Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, *35*, 24824–24837.

Wortmann, T., Costa, A., Nannicini, G., & Schroepfer, T. (2015). Advantages of surrogate models for architectural design optimization. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *29*, 471-481.

Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., & Zhang, Y. (2024). A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, *4*(2), 100211.