Methodology for Automatic Architecture Generation and Assessment of Thermal Management System in Electric Vehicles

***Rabih Al Haddad*[a], *Maroun Nemer*[b]**

*[a] Ecole des Mines de Paris, Center for Energy, Environment, and Processes, 21 All. des Marronniers,
78000, Versailles, France, rabih.al_haddad@minesparis.psl.eu (CA)*
*[b] Ecole des Mines de Paris, Center for Energy, Environment, and Processes, 21 All. des Marronniers,
78000, Versailles, France, maroun.nemer@minesparis.psl.eu*

Abstract:

The rapid shift toward electric motorization has introduced complex thermal management system (TMS) requirements for electric vehicles (EVs). Beyond optimizing battery temperatures for performance, the motor and power electronics demand precise regulation, while cabin conditioning adds operational complexity. Because TMS performance directly impacts EV efficiency, potentially reducing driving range by up to 50%, designing an optimal architecture that manages numerous heat sources and sinks is critical. Despite this importance, a notable lack of methodologies exists for early-stage design space exploration, often leading to over-engineered or suboptimal solutions based on traditional "worst-case" design approaches. This research introduces a novel, systematic methodology to fill this gap by automating the generation and assessing EV TMS architecture. The approach frames the TMS design as a connection problem, utilizing Constraint Satisfaction Problem (CSP) techniques to identify feasible solutions from a predefined set of components and circulating fluids. The methodology operates in two distinct phases. First, the system automatically generates feasible primary refrigerant loops and secondary loops based on specified operating modes. These configurations are refined through "Nodes Law" and "Connections Law" filters to ensure they adhere to physical and thermodynamic principles. Second, an innovative transformation code converts these graphical representations into numerical physical models in Dymola. This enables rapid simulation and comparison of architectures under diverse scenarios using the Coefficient of Performance (COP) as the primary performance metric. The methodology was validated through two case studies. The first successfully generated a standard vapor compression cycle, validating the fundamental solver logic. The second case study uses the tool to discover the design rational to the complex Tesla Model Y "Octovalve" system. The methodology successfully replicated Tesla's primary refrigerant architecture and identified high-performing secondary circuits for both summer and winter conditions that align with industry-leading strategies. This systematic framework enables manufacturers to explore a vast design space efficiently, bridging the gap between component selection and control strategy to foster high-performance, energy-efficient EV thermal management solutions.

Keywords:
Thermal Management System, Electric Vehicles, Architecture Generation, Constraint Satisfaction Problem, Graph Theory

1. Introduction

1.1. Context and Background

The shift toward electric motorization has introduced unique TMS requirements for EVs. Not only must the battery system be cooled or heated to function optimally across varied temperatures, but the motor and power electronics demand thermal control to handle the heat generated during operation. Cabin cooling and heating add further complexity, especially in winter, where thermal deficits can impact passenger comfort. TMS are essential for regulating component temperatures to maintain safety, comfort, and performance.

TMS performance directly affects EV efficiency, with the potential to reduce the driving range by as much as 50% under specific conditions[1–5]. Furthermore, the TMS architecture must be robust enough to accommodate a variety of operational modes and thermal demands[6]. Designing an efficient TMS that can operate across all necessary modes, maintain optimal temperatures and limit its impact on driving range is both complex and critical. The numerous heat sources, sinks, and available technologies expand the physical design space and introduce additional challenges for the control algorithms that govern the system[6].

1.2. Literature Review

Prior studies on thermal management systems (TMS) for EVs have largely focused on individual component optimization, often relying on "worst-case scenario" design within sequential approaches [7–9]. While this ensures reliability, it leads to over-engineered, costly systems and limits architectural innovation.

Building a system that can extract heat from all the above sources and manage the heat could be both technically and financially challenging. Jeffs et al. (2018) tested all combinations of potential heat sources across multiple drive cycles and ambient temperatures to minimize electrical energy consumption, while preserving passenger comfort. However, this approach overlooks the thermal and energetic history of each mode and the interactions between heat sources. [10]. The design methodology for the TMS in EVs outlined in [11] focuses on an analysis-driven approach to optimize system performance. A decision tree is introduced to systematically explore design options, considering various configurations for cabin conditioning, heating methods, and waste heat recovery strategies[11]. The methodology leverages a versatile simulation framework developed in MATLAB-Simulink with Simscape, allowing for dynamic modeling of coolant and refrigerant loops. This framework supports early-stage design space exploration, enabling the evaluation of different TMS configurations and control strategies across various driving conditions. The main gap in this approach is that the decision tree approach is based on common and current technologies which limits the horizon of possibilities and innovative architecture introducing new fluids or combination of fluids. Li et al. [12] introduced a fresh perspective, outlining a methodology to assess different TMSs. This approach encompasses defining the thermal management requirements of each major component within an EV, considering the scope of ambient temperature variations and the corresponding operational modes that cater to the distinct needs of each component. They identified 27 working modes for the TMS though without clearly justifying architecture choices. In a separate study, Weustenfield et al. [13] introduced an optimization approach to the secondary coolant loop by linking thermal management concepts to the flow of thermal energy. This methodology divides the coolant loop into heating, cooling, and low-temperature cycles. The criterion is set based on energy demand and temperature constraints linking the coolant cycles to each component. Meanwhile, [14] employed a multi-objective optimization model to optimize both compressor energy consumption and battery aging losses. Using a genetic algorithm, they determined the optimal compressor speed based on battery temperature, aiming for energy efficiency.

In summary, the authors note that to the best of their knowledge, while literature provides valuable contributions in optimizing TMS components, modes, and isolated thermal strategies, it lacks a cohesive, integrated methodology for early-stage design exploration and comprehensive optimization of TMS architectures in EVs. The primary gaps in the literature include the absence of a robust framework for exploring the full design space of TMS configurations, a limited capacity for handling multi-modal, real-time operational demands, and an insufficiently integrated approach that simultaneously optimizes both component design and control strategies. In addition, listing and modelling manually all the architecture that we can imagine is infeasible. These limitations constrain the potential for novel, efficient TMS designs that can meet the complex requirements of modern EVs.

1.3. Novelty and Contribution

To address these gaps, this work introduces a novel decision tool that applies a systematic methodology tailored specifically to EV TMS design, combining automatic architecture generation with evaluation strategies to assess configurations across a wide range of operating conditions. This approach leverages CSP-based architecture generation to explore diverse design possibilities from an initial pool of components, ensuring that the generated architectures are feasible and aligned with thermal management needs. Using graph-theory approach those feasible architectures through graphs are translated automatically into physical based – architecture models in Dymola and simulated to assess their performance given a certain objective function.

In this context, the aim of this work is not only to fill this gap but also to guide future design processes by introducing a structured methodology that can design and evaluate TMS architectures. This approach automates the generation of potential TMS configurations from a set of predefined components and evaluates their performance across different operating scenarios to identify the most effective setups. The work presented is novel as follows:

- Automatic generation of a wide range of architectures instead of focusing on a few known designs. This is crucial for innovative thinking and the introduction of new components or multiple refrigerant circuits.
- Development of an automatic method that transforms generated architecture from graph form to physics – based architectures models in Dymola.
- Presentation of a complete design methodology for generations and evaluation of thermal management system architecture.

The proposed thermal architecture design methodology is discussed in the coming sections starting with a framework of the method in section 2.1. Furthermore, the proposed methodology is explained in detail in section 2.2 and up. Finally, the proposed methodology is tested on different case studies to show its applicability and implementation on real scenarios

2. Methodology

2.1. Proposed Framework:

The design of a TMS architecture can be effectively linked to a connection problem because, at its core, the process involves determining how different components should be interconnected to transfer heat efficiently across the system. Each component in the TMS has multiple ports (or nodes), which must be connected to other components or subsystems to facilitate the flow of fluids such as refrigerants, coolants, and air. The challenge then becomes identifying which nodes should be connected to one another in a way that meets the thermal requirements of the system while maintaining efficiency and minimizing energy consumption.

The proposed methodology for designing electric vehicle TMS, shown in Figure 1 utilizes a constraint-based approach and graph theory formulation to automatically generate and evaluate diverse architectural configurations. By first defining physical components, circulating mediums, and performance requirements, the system transforms these inputs into a nodal representation that follows strict thermodynamic and physical laws. This allows for the automated creation of complex architectures across refrigerant, coolant, and air loops, which are then filtered for redundancy and physical viability. The most promising graph-based designs are converted into numerical models for simulation in Dymola, where they are rigorously assessed under various operating scenarios based on their Coefficient of Performance (COP) and ability to maintain optimal temperatures for the cabin, battery, and powertrain.

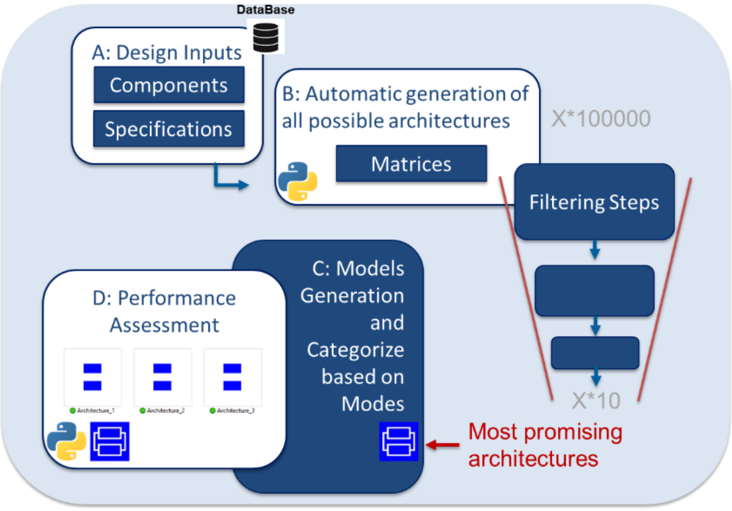


Figure 1. *Framework of proposed methodology*

2.1. Proposed presentation:

The methodology proposes to generate all the possible architectures that can be made when connecting a certain number of defined thermal components and defined connecting elements. The first step in this methodology involves defining the components and operating fluids to be used in the TMS. At this stage, the user selects the components that will form the possible TMS architecture. Traditionally, both manufacturers and researchers are often constrained by a fixed system architecture when designing or optimizing a TMS. However, this methodology introduces a significant degree of flexibility, allowing users to freely choose from a wide array of components and fluids. This freedom is crucial, as it enables the exploration of a larger design space and the possibility of discovering more efficient or innovative configurations. To guide users in making their selections, an example of a comprehensive list of potential components but not limited to that can be considered in a TMS design problem has been compiled based on an extensive review of the literature and the expertise developed throughout this project. The components cover all major areas of a TMS, such as heat exchangers, pumps, valves, compressors, splitters, mixers, and others and are detailed in the accompanying table. Users can choose any of, all, or add to these components depending on the specific requirements of their EV system.

2.1.1. Overview of the graphical representation:

The most effective way to represent the different components in this connection problem is by using nodes. Each node corresponds to a physical port of the selected components[15]. Nodes can either be connected or left unconnected, and this relationship can be mathematically defined by assigning a value of one for a connection and zero for no connection. ***Figure 2*** provides an example of an evaporator component, which has four physical ports, each represented by a node.

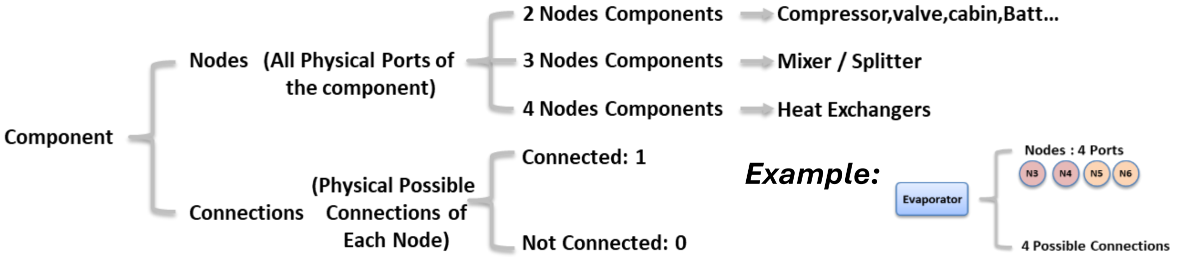


Figure 2. *The proposed presentation*

2.1.2: Definition of the characteristics of the nodes:

To apply constraints on nodes and connections for a connection problem, a node is given a set of features that clearly define its type, role, and function. Each node in this problem is given a set of characteristics such as node number, port type, fluid type, energetic type, and pressure medium. The port type of the node can be inlet or outlet. For the fluid type, the port can carry refrigerant, coolant, or air. The fluid type is given by the information the user gave for the mediums and the components. The energetic characteristic of the node can be whether it is work, energy, cooling, heating, or just a transporter. Finally, for the pressure medium, the nodes are classified into three categories: high, low, and medium pressure.

For instance, let’s consider chiller as an example. The chiller has four distinct nodes: two that circulate refrigerants and two that circulate coolant. While the connection problem is solved independently for each fluid category, refrigerants in one sub-problem and coolants in another, the connections across these sub-problems are interconnected through the shared nodes of the chiller. This approach allows for a more manageable and systematic way to handle the overall connection problem without losing sight of the system's integrated nature.

In addition to selecting the components, another critical aspect of this step is defining the circulating fluids that transfer heat between different components of the system. In a typical TMS, heat is transferred across various loops: the primary loops, the secondary loops (if present), and a default air loop. Multiple loops of each category can be present depending on the user’s selection. The user must specify the operating refrigerant or refrigerants for the primary loop. If a secondary loop is included in the architecture, the user must also define the type of coolant used in this loop. The air loop, which is defined by default, represents the flow of air used to either cool or heat the vehicle’s cabin or assist in other cooling and heating tasks, forming a crucial third loop within the TMS architecture.

2.2. Automatic Generation of Architectures

2.2.1 Problem Definition:

i) Variables and Domains:

This connection problem can be formulated and solved using CSP approach because the design process is constrained by several physical and operational factors. In a CSP, the goal is to find a set of variable assignments that satisfy all imposed constraints. In the context of TMS architecture design, the variables are the connections between nodes (whether two nodes should be connected or not), and the constraints are the physical and operational limitations.

A CSP is a problem that can be stated as follows:

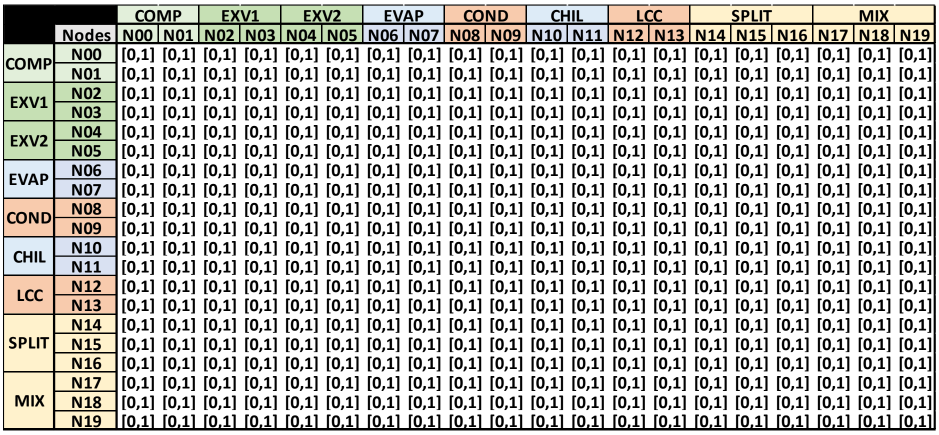
- A set of variables X = {x1,…,xn}
- Each xi has a domain of possible values Di
- A set of constraints Ci,j,k restricting the values that the variables xi,xj,xk, can take simultaneously.

The connections in a TMS architecture can be visualized as a graph: a set of nodes (components) connected by a set of edges (connections). Let us consider 9 components with a graph composed of a set of nodes 20 and connected by a set of edges 20 (this example for 9 components that are available in a thermal management system usually shown in Table 1):

- Adjacency matrix: n by n matrix where the columns are the nodes V0,…,Vn-1, the rows are the nodes V0,…,Vn-1 shown in Table 1. The rows and columns correspond to the nodes of all the components ports N00 to N19. Each color represents a component type. The light green corresponds to the compressor. The dark green corresponds to the valves. The light blue corresponds to the evaporator and chiller. The light orange corresponds to the LCC and Condenser.
- Number of variables: 20x20 = 400
- Domain of all variables: {0,1}
- Target: get all feasible solutions (Initial number of solutions without constraints = $2^{number\ of\ variables}$)

The initial number of solutions without constraints is: 2^{400}

Table 1. *Adjacency matrix before the addition of constraints*



		COMP		EXV1		EXV2		EVAP		COND		CHIL		LCC		SPLIT			MIX		
	Nodes	N00	N01	N02	N03	N04	N05	N06	N07	N08	N09	N10	N11	N12	N13	N14	N15	N16	N17	N18	N19
COMP	N00	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
	N01	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
EXV1	N02	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
	N03	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
EXV2	N04	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
	N05	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
EVAP	N06	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
	N07	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
COND	N08	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
	N09	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
CHIL	N10	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
	N11	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
LCC	N12	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
	N13	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
SPLIT	N14	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
	N15	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
	N16	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
MIX	N17	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
	N18	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
	N19	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]

ii) Constraints:

Any solution to this CSP problem should respect some solid constraints. The constraints for this problem were divided into two main categories: Nodes Law and Loops Law. The nodes law represents the physical constraints applied to the nodes themselves. The connections law represents the thermodynamic (operational) and physical constraints that apply on the connections within a loop.

In the early works on the generation tool, some basic constraints were defined. Then based on the generated graphs and by logical thinking and engineering knowledge these constraints were enhanced, and others were added. At the final stage, the following constraints were the ones considered. For each constraint, an example of the evolution of the adjacent matrix and number of solutions will be presented.

The constraints that were used in this methodology are listed in ***Table 2***.

Table 2. *Constraints on Nodes and Connections*

Constraint #	Category	Description
C000	Nodes Law	Symmetric matrix (undirected graph)
C001	Nodes Law	A node cannot be connected to itself
C002	Nodes Law	Nodes of same component cannot be connected to each other
C003	Nodes Law	Inlets nodes can only be connected to outlet nodes
C004	Connections Law	Nodes of the same pressure medium can be connected to each other
C005	Connections Law	Nodes with the same energetic type Example: nodes that refer to work components cannot be Connected to Each other
C006	Connections Law	Max number of connections for one node is equal to 1

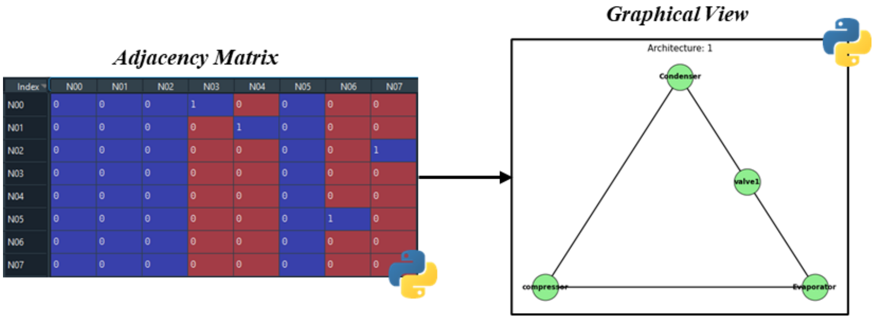
C007	Connections Law	Nodes of different fluids are filtered and cannot be connected
C008	Connections Law	Various Constraints on connections that intuitively cannot be connected

2.2.2 Problem Implementation and Solving

Now that the problem variables, domains and constraints have been defined, the next step is to choose an environment where this CSP can be implemented and a solver that can solve the problem.

Various techniques are used in Constraint Programming (CP) to solve problems efficiently. These include backtracking, constraint propagation methods such as arc and path consistency, and local search algorithms that incrementally refine solutions through minor adjustments. Additionally, hybrid approaches, like combining CP with Linear Programming (CP+LP) or SAT Solving (CP+SAT), are commonly employed to address both combinatorial and optimization aspects of a problem. Decomposition methods can also be utilized to break large, complex problems into smaller, more manageable subproblems. After evaluating a range of tools for solving CSP, the Python-Constraint module was selected as the most suitable solution. This choice was driven by its ease of implementation within the Python environment and the accessibility of additional Python libraries, such as those for graph visualization, which are freely available and widely supported. Python's flexibility and the extensive range of open-source modules further reinforced the decision to use this tool. The default solver within this module, known as “backtracking”, was chosen for its ability to perform constraint propagation and forward checking, making it capable of generating all possible solutions.

The automatic generation of architectures consists now in solving this problem to get all the solutions of adjacency matrices and then in transforming these matrices into graphs and visualizing them. These solutions are in form of adjacency matrices. NetworkX module on python is used to transform these adjacency matrices into graphs that are visualized to the user as a first step. Example of the generated adjacency matrix and the visualization through a graph is shown in Figure 3.



Index	N00	N01	N02	N03	N04	N05	N06	N07
N00	0	0	0	1	0	0	0	0
N01	0	0	0	0	1	0	0	0
N02	0	0	0	0	0	0	0	1
N03	0	0	0	0	0	0	0	0
N04	0	0	0	0	0	0	0	0
N05	0	0	0	0	0	0	1	0
N06	0	0	0	0	0	0	0	0
N07	0	0	0	0	0	0	0	0

***Figure 3**. Generation of architectures in graph form*

After examining the generated graphs, further filtering is mandatory to reduce the pool of solutions which is explored in the coming section.

It is important that the nodes are categorized based on the fluid type and the first solving setup and generation of architectures will be on the primary fluid. For each promising architecture generated of the primary fluid the same methodology will be applied to generate the remaining circulating fluids architectures.

i) Filtration Steps:

The first filtration step addresses the elimination of architectures that contain open loops within their fluid circuits. From a thermodynamic perspective, a closed loop is essential for the conservation of mass ($\dot{m}_{in} = \dot{m}_{out}$) and energy ($\dot{E}_{in} = \dot{E}_{out}$) within the system. Moreover, the fundamental thermodynamic cycles (e.g., Carnot, Rankine, or refrigeration cycles) require closed loops to function correctly, ensuring that the working fluid undergoes a series of processes that return it to its initial state. If we consider an architecture where the refrigerant exits the evaporator but does not have a defined path back to the compressor, In the adjacency matrix, this would manifest as a missing edge connecting the evaporator's outlet to the compressor's inlet. Graphically, the refrigerant circuit graph lacks a closed loop, indicating an open loop. This configuration would fail to sustain the refrigeration cycle, as the compressor cannot receive the low-pressure refrigerant vapor from the evaporator, preventing the continuation of the cycle.

The second filtration step targets architectures with unnecessary redundancy. Redundancy in a TMS architecture refers to the inclusion of duplicate components or parallel paths that do not contribute additional functionality or improve system reliability beyond the required specifications. We analyze the adjacency matrices of each architecture to detect redundant components or connections. Mathematically, redundancy can be identified by multiple edges or parallel paths between the same set of nodes, indicating duplicate components performing the same function. We employ algorithms to detect subgraphs that are isomorphic, structurally identical to each other within the larger graph, representing redundant pathways.

ii) *Categorization* based on modes

The generated architectures are categorized based on a table of modes formulated by the author and based and inspired by the literature review. These modes are the states that the TMS in EV needs to cover. The modes are selected based on the external weather temperature, vehicle state (start or driving), charging state (fast or normal), and driver presence (in car or not). As a results 27 modes cover all the possible TMS functions needed in EV. For example, one of the modes is at external temperature of -15°C, vehicle state is in transient (start of the trip). In this case, the cabin needs heating, the battery needs also heating, the power electronics and motors are in equilibrium state. The criterion of classification is based on the minimum components required for the mode to be present. As such, the architecture having those components is considered potential. The architecture that does not meet any mode will be filtered out.

We begin by defining the set of required operating modes ($M = \{m_1, m_2, .., m_k\}$). Each architecture is then evaluated to determine its capability to support these modes. This involves checking for the presence of necessary components and connections associated with each mode.

Mathematically, we define a function $f(A, m_i)$ that maps architecture to a binary outcome for each mode m_i:

$$f(A, m_i) = \begin{cases} 1 \; if \; architecture \; A \; supports \; mode \; m_i \\ 0 \; otherwise \end{cases} \tag{1}$$

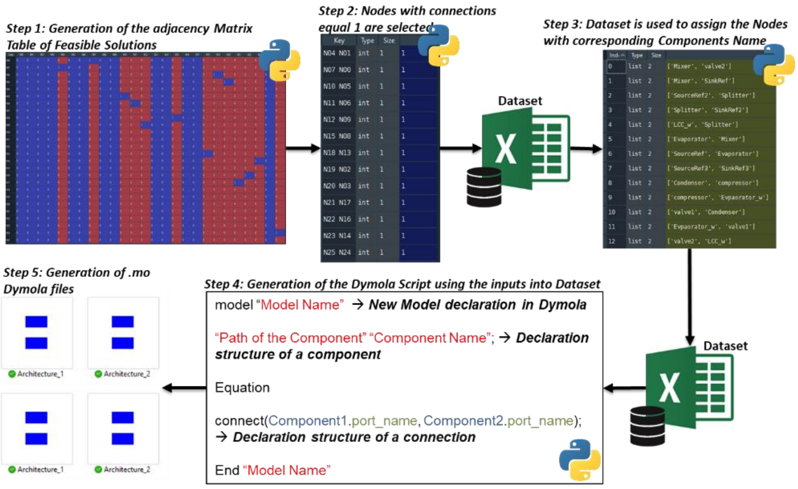
Architecture is retained if $f(A, m_i) = 1$ for the required mode. This evaluation considers the thermodynamic processes needed for each mode and whether the architecture's components and connections facilitate these processes.

Each operating mode imposes specific thermal demands and requires certain thermodynamic cycles or processes. For example, battery heating might require a heat pump mode where the refrigerant circuit operates in reverse to absorb heat from the environment and deliver it to the battery. The physical feasibility of architecture to support this mode depends on the availability of a condenser. Thus, as mentioned earlier, now for each surviving architecture assigned to a mode the other fluids circuits undergo the same steps and complete architecture is generated.

By integrating these mathematical and physical considerations into the filtration methods, we enhance the robustness of the design methodology and facilitate the development of TMS architectures that are both innovative and functionally sound.

iii) *Transforming* generated graphs to architectures in Dymola

An automatic models' formation of the feasible architectures resulting from the Solver is essential. The goal was to transform the adjacency matrices table to a physical architecture model automatically with a concrete illustration of the connections and the position of the components in Dymola. Figure 4 explains the logic behind the automatic model's generation in Dymola.



Key	Type	Size	
N04 N01	int	1	1
N07 N00	int	1	1
N10 N05	int	1	1
N11 N06	int	1	1
N12 N09	int	1	1
N15 N08	int	1	1
N18 N13	int	1	1
N19 N02	int	1	1
N20 N03	int	1	1
N21 N17	int	1	1
N22 N16	int	1	1
N23 N14	int	1	1
N25 N24	int	1	1

Ind	Type	Size	
0	list	2	['Mixer', 'valve2']
1	list	2	['Mixer', 'SinkRef']
2	list	2	['SourceRef2', 'Splitter']
3	list	2	['Splitter', 'SinkRef2']
4	list	2	['LCC_w', 'Splitter']
5	list	2	['Evaporator', 'Mixer']
6	list	2	['SourceRef', 'Evaporator']
7	list	2	['SourceRef3', 'SinkRef3']
8	list	2	['Condenser', 'compressor']
9	list	2	['compressor', 'Evpaorator_w']
10	list	2	['valve1', 'Condenser']
11	list	2	['Evpaorator_w', 'valve1']
12	list	2	['valve2', 'LCC_w']

model "Model Name" → ***New Model declaration in Dymola***

"Path of the Component" "Component Name"; → ***Declaration structure of a component***

Equation

connect(Component1.port_name, Component2.port_name);
→ ***Declaration structure of a connection***

End "Model Name"

Figure 4. *Transformation of Adjacency matrix of feasible solutions to Dymola Models Logic*

iv) *Performance* Assessment Criteria

The performance assessment criteria play a critical role in evaluating the feasibility and effectiveness of the generated TMS architectures. Thermal efficiency is often quantified through metrics such as the Coefficient of Performance (COP), which measures the ratio of thermal energy transfer to the energy consumed by the system. A high COP indicates efficient operation, which is particularly critical for EVs where the TMS directly impacts driving range.

In addition to thermal and energy metrics, the assessment criteria also evaluate the system's ability to meet specific operational constraints. These constraints may include maintaining battery temperatures within a safe and optimal range, ensuring cabin comfort within a target time frame, and adhering to predefined environmental or component limits. The assessment framework integrates these criteria into an objective function, enabling the comparison of architectures based on their overall performance. By systematically applying these performance criteria, the methodology ensures that only the most efficient and feasible architectures are selected for further analysis or implementation.

It is important to note that all the generated architectures are simulated in Dymola. The selection of the best architecture at a specific operating mode is based on equation and meeting the constraints in equations.

$$OF = \max_{n}(COP_n) \tag{2}$$

$$19°C \leq T_{cabin} \leq 23°C \tag{3}$$

$$15°C \leq T_{battery} \leq 40°C \tag{4}$$

Where: n is the architecture number

Defining the final architecture to be used in done by the user which will be decided after examining the best operating architecture at different modes.

3. Case Study: application of the proposed methodology

To enhance understanding of the proposed methodology for designing and assessing optimal TMS architectures, it is crucial to evaluate its performance through concrete case study. This case study is a foundational application of methodology, designed to assess its functionality in generating and evaluating a simple vapor compression cycle without imposing additional constraints on the assessment criteria. This step-by-step analysis highlights the methodology's systematic approach and validates its fundamental processes.

This case study aims to apply the methodology presented on a simple exercise of developing a vapor compression cycle and achieve its optimal performance. Although given the simplicity of such exercise, it will allow us to examine the methodology step by step. The coming section will be split into defining the problem, stating the objective function, and explaining the solution evolvement from one step to another.

i) Problem Definition:

The starting components that are introduced in the automatic generation tool are the components of a typical vapor compression cycle which are a compressor, condenser, evaporator, expansion Valve.

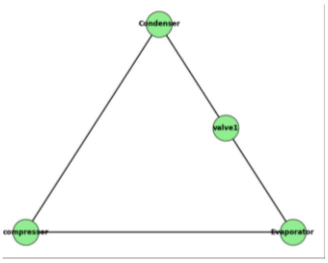
A connection problem can be formulated based on the given components and the CSP has the following specifications on the refrigerant side:

- Number of Nodes: 8
- Number of variables: 8x8
- Domain of all variables: [0,1]

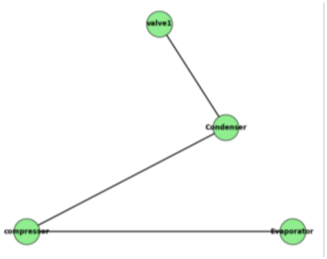
The problem constraints are already pre-defined inside the architecture generation tool. The CPS solving is launched, and the tool generates all the possible solutions that are visualized as graphs. The initial number of solutions for this connection problem is 2^{8x8}.

ii) Automatically generated architecture:

The tool generated 16 architectures based on the fed constraints. Examples of the generated architectures are shown in Figure 5. After filtering the architecture based on the open loop elimination and redundancy filtering, only one architecture survived which represents the only feasible architecture on the refrigerant side, shown in Figure 5 – Architecture 1.



Architecture 1



Architecture 2

Architecture 6 *Architecture 7*

Architecture 3 *Architecture 14*

Figure 5. *Example of the generated architectures of the refrigerant loop for the case study*

The solving steps of the methodology on the refrigerant side is shown in Figure 6.

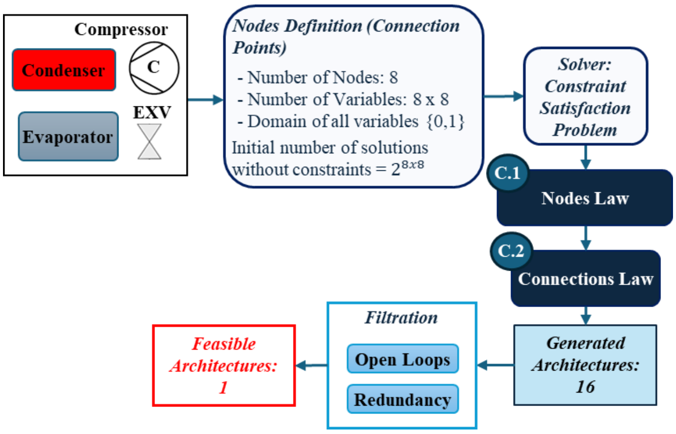
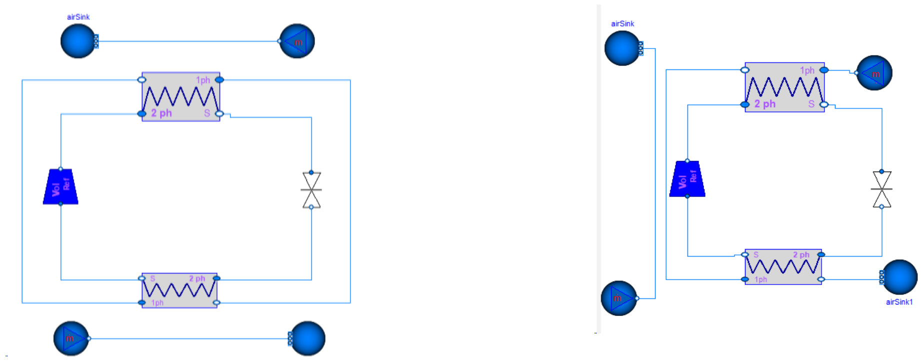
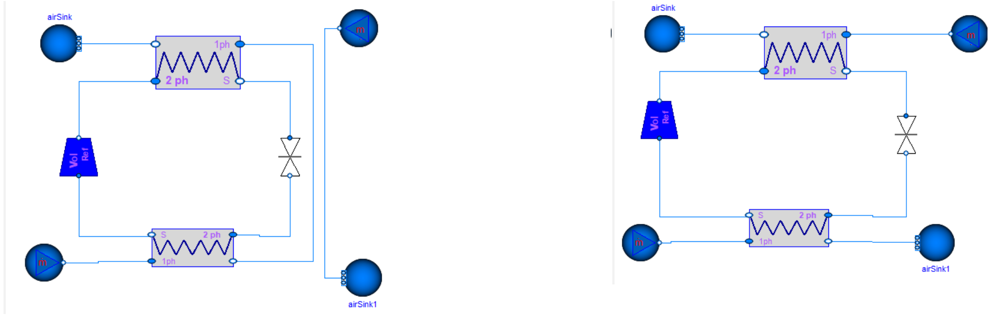


Figure 6. *Example of solving the problem presented in the case study*

Based on the resulting refrigerant circuit, the same automatic generation tool is applied now on the air nodes found in the resulting architecture. The evaporator and condenser have four air nodes, and in addition, two identical sources and two identical sinks representing the ambient are added to the pool of components. The automatic generation tool is applied again. The resulting four architectures are feasible and meet the given constraints, which are shown in Figure 7. It is important to note that in this exercise here, no cabin or battery exists. The ambient sources and sink nodes are free to be connected in any sense. To further elaborate on that, we can see that for example, in Figure 7 architecture 1 the evaporator and condenser air nodes are connected to each other on one side and on another side the sources and sinks are connected to each other. This may not be very logical for design; however, this case study is a simple un-constrained application of the generation tool.



Architecture 1 *Architecture 3*



Architecture 7 *Architecture 11*

Figure 7. *Final resulting architectures for the case study*

iii) Performance Assessment:

In this section, the performance assessment criteria, objective function, operating conditions, modelling framework, and simulation platform are explained. Moreover, the results are valorized and discussed.

For this case study, the objective function is shown below:

$$OF = \max_{n}(COP_n) \qquad (5)$$

Where: n is the architecture number

The operating conditions for the simulations are presented in *Table 3*. The simulation platform used is Dymola. The models were already sized, and the initialization parameters were fed in each component. The generated feasible architectures are directly developed through python using the Modelica language and simulated in Dymola. The models used are from a thermal systems library developed at CEEP and were validated through the past years[16].

Table 3. *Operating parameters and model characteristics used for case study.*

Component	Parameter	Value
Ambient	Ambient Temperature	35°C
	Ambient RH	50%
Compressor	Compressor Displacement	28 cc
	Isentropic Efficiency	0.62
	Volumetric Efficiency	0.72
Evaporator	Type	Round-tube Finned Heat Exchanger
	Model	NTU
	SH	5°C
Condenser	Type	Mini-channel Finned Heat Exchanger
	Model	NTU
	SC	5°C

The model is simulated, and the COP was calculated which is the heat flow rate at the evaporator level divided by the compressor power. The results of the COP for the four resulting architectures are shown in Figure 8. The best performing architecture is Arch 7. The air is being cooled before entering the condenser. This will decrease the gap between the HP and the LP lowering the compressor work. It is important to note that Arch 1 did not simulate given that it is not possible to simulate a closed air loop without a physical fan. Thus, it was excluded from the comparison.

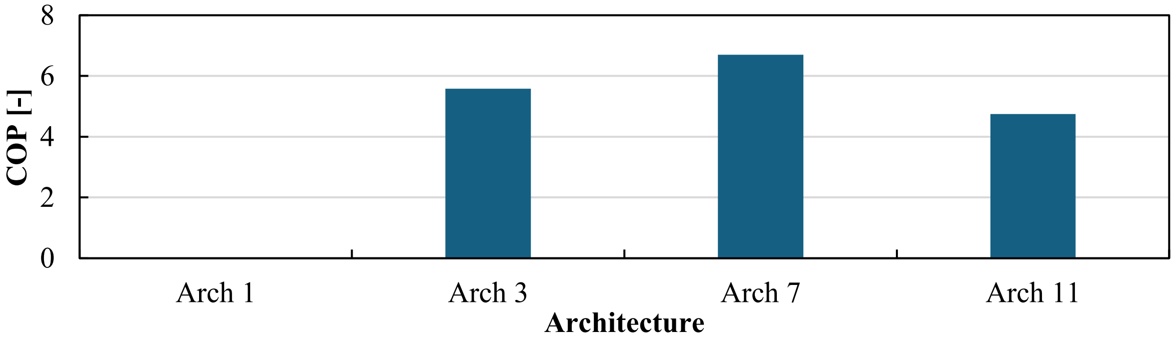


***Figure 8**. COP after simulation for each architecture*

Although case study 1 is a simple and straightforward application of the methodology, however, the potential of such tool is present, and the understanding of the different steps is deepened.

It is important to note that another case study is considered to test robustness by applying the methodology to the TMS components used in Tesla Model Y which increases the level of complexity of the design process. This case study evaluates the methodology's ability to replicate and assess a state-of-the-art system, comparing the resulting architectures with Tesla's existing design. Additionally, the evaluation includes a performance comparison of the resulting architectures under winter and summer conditions, benchmarking them against Tesla's operating modes to identify potential improvements or alternative configurations. The results are promising and the method was able to replicate the architectures used by the Tesla with their Octovalve and components[16].

4. Conclusion

The proposed methodology for the automatic generation and assessment of TMS architectures in EVs offers a transformative approach to the design of TMS. By systematically integrating CSP, multi-domain modeling, and performance assessment, the methodology addresses the key challenges in TMS design: the complexity of multi-medium systems, the need for adaptability to diverse operational modes, and the demand for energy-efficient configurations.

The case study demonstrated the methodology's robustness and versatility. The generation and evaluation of a simple VCC highlighted the step-by-step logic and systematic filtering process of the approach. Despite the simplicity of the problem, the results validated the methodology's ability to identify optimal architectures while ensuring adherence to thermodynamic principles.

One of the most significant contributions of this work lies in its capability to bridge the gap between theoretical design and practical implementation. The integration of automated architecture generation with Modelica-based simulation platforms ensures that the resulting designs are not only theoretically sound but also validated under realistic operating conditions. This approach eliminates the trial-and-error nature of traditional design processes, reducing development time and fostering innovation by enabling the exploration of unconventional solutions.

Given that the TMS and temperature of the different sub-systems are dynamic and changes throughout the trip and given that topology generation and optimization need to be simultaneous and not sequential to reach the optimal solution, certain limitations must be acknowledged. Achieving a global optimum remains challenging, particularly given the complexity and multitude of operating modes in TMS. The current methodology emphasizes steady-state and quasi-dynamic performance assessments, which may limit its applicability in scenarios requiring a detailed understanding of transient behavior, such as during trips or transitions between operating modes. Additionally, the optimization of control strategies is beyond the scope of this work. While the generated designs are thermodynamically efficient, their real-world performance is heavily influenced by control algorithms, which are addressed in subsequent chapters but remain separate from the current methodology. Using the proposed method, the user can generate all the possible architecture on the 27 modes and assessment. For a complex architecture like the Tesla Model Y, the total number of generated architectures for the 27 modes including the primary refrigerant loop, coolant refrigerant loop and air loop is 2700 architectures. However, the simulations done are still not optimized in each specific mode. A holistic approach is needed to optimize the system in terms of operating parameters, architecture used per mode, and architectures switch during a trip.

These limitations suggest that an architecture optimized for a single mode may not perform optimally across an entire trip. An additional step to the methodology should involve optimizing the connections within the secondary coolant loop over a complete trip, rather than relying solely on the listing method, to enhance the overall performance of the TMS.

References

[1] Al Haddad R, Mansour C, Kim N, Seo J, Nemer M. Comprehensive Thermal Modelinig and Analysis of a 2019 Nissan Leaf Plus for Enhanced Battery Electric Vehicle Performance. SAE WCX 2024 Congress, 2024.

[2] Haddad R Al, Mansour C, Kim N, Seo J, Stutenberg K, Nemer M. Comparative analysis of thermal management systems in electric vehicles at extreme weather conditions: Case study on Nissan Leaf 2019 Plus, Chevrolet Bolt 2020 and Tesla Model 3 2020. Energy Convers Manag 2025;332. https://doi.org/10.1016/j.enconman.2025.119706.

[3] Tunalı TE, Gözen E, Özgül E. Range improvement in BEV trucks using heat pump for cabin heating instead of PTC heater. Int J Heat Mass Transf 2026;256:127910. https://doi.org/10.1016/j.ijheatmasstransfer.2025.127910.

[4] Zhao C, Li Y, Yang Y, Wan S, Yu F, Yu C, et al. Research on electric vehicle range under cold condition. Advances in Mechanical Engineering 2022;14. https://doi.org/10.1177/16878132221087083.

[5] Wu J, Zhou G, Wang M. A comprehensive assessment of refrigerants for cabin heating and cooling on electric vehicles. Appl Therm Eng 2020;174. https://doi.org/10.1016/j.applthermaleng.2020.115258.

[6] Lemort V, Olivier Gerard, De pelsemaeker Georges. Thermal Energy Management in Vehicles. Wiley; 2023.

[7] Leoncini G, Mothier R, Michel B, Clausse M. A review on challenges concerning thermal management system design for medium duty electric vehicles. Appl Therm Eng 2024;236. https://doi.org/10.1016/j.applthermaleng.2023.121464.

[8] Singh S, Jennings M, Katragadda S, Che J, Miljkovic N. System design and analysis methods for optimal electric vehicle thermal management. Appl Therm Eng 2023;232. https://doi.org/10.1016/j.applthermaleng.2023.120990.

[9] Silvas E, Hofman T, Murgovski N, Etman LFP, Steinbuch M. Review of Optimization Strategies for System-Level Design in Hybrid Electric Vehicles. IEEE Trans Veh Technol 2016;66:57–70. https://doi.org/10.1109/TVT.2016.2547897.

[10] Jeffs J, McGordon A, Picarelli A, Robinson S, Tripathy Y, Widanage WD. Complex heat pump operational mode identification and comparison for use in electric vehicles. Energies (Basel) 2018;11. https://doi.org/10.3390/en11082000.

[11] Singh S, Jennings M, Katragadda S, Che J, Miljkovic N. System design and analysis methods for optimal electric vehicle thermal management. Appl Therm Eng 2023;232. https://doi.org/10.1016/j.applthermaleng.2023.120990.

[12] Li K, Chen H, Xia D, Zhang H, Dou B, Zhang H, et al. Assessment method of the integrated thermal management system for electric vehicles with related experimental validation. Energy Convers Manag 2023;276. https://doi.org/10.1016/j.enconman.2022.116571.

[13] Weustenfeld TA, Bauer-Kugelmann W, Menken JC, Strasser K, Ag A, Koehler GJ. Heat flow rate based thermal management for electric vehicles using a secondary loop heating and cooling system. n.d.

[14] Kuang X, Li K, Xie Y, Wu C, Wang P, Wang X, et al. Research on Control Strategy for a Battery Thermal Management System for Electric Vehicles Based on Secondary Loop Cooling. IEEE Access 2020;8:73475–93. https://doi.org/10.1109/ACCESS.2020.2986814.

[15] Kabalan B. Systematic methodology for generation and design of hybrid vehicle powertrains. n.d.

[16] Al Haddad R. Development of a design methodology of thermal management system in electric vehicles. Université Paris Sciences et Lettres, 2025.