Saeid Bayat[1]
Department of Industrial and Enterprise
Systems Engineering,
University of Illinois at Urbana-Champaign,
Urbana, IL, USA
email: bayat2@illinois.edu

Nastaran Shahmansouri[1]
Autodesk Research,
661 University Ave,
Toronto, Canada
email:
nastaran.shahmansouri@autodesk.com

Satya RT Peddada
Department of Industrial and Enterprise
Systems Engineering,
University of Illinois at Urbana-Champaign,
Urbana, IL, USA
email: speddad2@illinois.edu

Alexander Tessier
Autodesk Research,
661 University Ave, Toronto, Canada
email: alex.tessier@autodesk.com

Adrian Butscher
Autodesk Research,
661 University Ave,
Toronto, Canada
email: adrian.butscher@autodesk.com

James T Allison
Department of Industrial and Enterprise
Systems Engineering,
University of Illinois at Urbana-Champaign,
Urbana, IL, USA
email: jtalliso@illinois.edu

Multi-split configuration design for fluid-based thermal management systems

High power density systems require efficient cooling to maintain their thermal performance. Despite this, as systems get larger and more complex, human expertise and insight may not suffice to determine the desired thermal management system designs. To this end, a framework for automatic architecture exploration is presented in this article for a class of single-phase, multi-split cooling systems. For this class of systems, heat generation devices are clustered based on their spatial information, and flow splits are added only when required and at the location of heat devices. To generate different architectures, candidate architectures are represented as graphs. From these graphs, dynamic physics models are created automatically using a graph-based thermal modeling framework. Then, an optimal fluid flow distribution problem is solved by addressing temperature constraints in the presence of exogenous heat loads to achieve optimal performance. The focus in this work is on the design of general multi-split heat management systems. The methods presented here can be used for diverse applications in the domain of configuration design. The multi-split algorithm can produce configurations where splitting can occur at any of the vertices. The results presented include three categories of problems and are discussed in detail.

Keywords: Design Synthesis, Graph Generation, Optimization, Optimal Flow Control, Thermal Management System

1 Introduction

Electrification of many systems in various domains has increased over the last few decades [1–3]. These electronic devices are experiencing intensifying miniaturization [4–9], resulting in higher power densities. Currently, microprocessors generate heat fluxes of over 102 W/cm^2. Hot spots on microelectronic chips can also generate heat fluxes of 1 kW/cm^2 or more, which can result in excessive temperatures in local regions [4]. When devices are operated at excessive temperatures, their performance and reliability are negatively affected, resulting in eventual malfunction [10,11]. The automated optimal design of thermal management systems (TMSs) is essential for expediting design processes and achieving ambitious performance goals [12,13].

Throughout the literature, the design of TMSs has focused largely on improving individual components [14,15]. Yet, optimizing components in isolation can result in a sub-optimal overall TMS design. Additionally, the design of a whole TMS sometimes is only relevant to a particular application [16], disincentivizing investment in effective tailored design. Peddada et al. [14] took an important step towards generalized design automation of a class of TMSs that are not limited to a specific application. However, the systems considered in that work are restricted to single-split system architectures, i.e., configurations with a single source where thermal-devices are located in branches that split only at the source. There are, nevertheless, many instances in which multi-split systems are required [17]. In a multi-split architecture, branching can occur at the main source or any system thermal-device. Figure 1 illustrates the difference between single-split and multi-split systems. Figure 1(a) represents a general single-split system where branching occurs only at the source (S). In contrast, branching in the multi-split system (Fig. 1(b)) can occur either at the source or at the junctions (J). Multi-split systems represent a much richer set of candidate design configurations compared to single-split.

After determining a design configuration, system dynamics must be modeled for performance comparison. System dynamics models can range from low-fidelity (e.g., regression of empirical models) to high-fidelity (e.g., finite element methods). At early system design stages, ideal models are 1) computationally efficient, 2) sufficiently comprehensive to capture key physics that drive design decisions, and 3) can be generated automatically for new configurations. These attributes support effective exploration of a large design space with reasonable computational expense. To achieve these objectives, the method presented here utilizes graph modeling for automated construction of TMS lumped-parameter dynamics models. In this model, vertices represent capacitive energy

[1] Corresponding Author.
Version 1.18, September 26, 2024

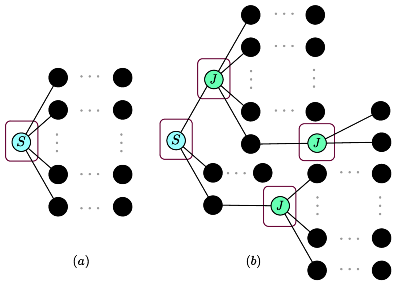


Fig. 1 Comparison of the configuration between (a) Single-Split and (b) Multi-Split Cases. The location of branching is indicated by a box encompassing source or junction nodes.

storage, while edges represent pathways for energy transfer between vertices [14,18–21]. In TMS graphs, vertices and edges are characterized by temperature and the rate of thermal energy transfer, respectively. Candidate TMS configurations identified based on lumped parameter model results can then be analyzed in the next phase using higher fidelity models to ensure they satisfy all conditions and constraints.

Designs of general multi-split heat management systems should meet transient response conditions [22] as many electronic devices work in applications with time-varying workloads [4]. Accurate modeling of the design problem as an optimization problem, therefore, requires treatment of system dynamics, including bounds and constraints on dynamic behavior, as well as optimal control for active systems. Continuous optimization can be applied to the combined design of physical and control system design, a well-established problem in Control Co-Design (CCD) [23–26], for a given system configuration. Distinct configurations can have fundamentally different dynamics, design variables, and constraints, so each configuration must be treated as independent CCD optimization problems. Fair comparison of configurations requires CCD optimization of each, which, for architecture design problems of practical scale, requires automatic generation and solution of the optimization problem for each candidate configuration. In this paper, we introduce a strategy for the generation and the optimal design of general multi-split heat management systems using graphs. For every generated unique and feasible multi-split system configuration, a continuous optimization problem is formulated and solved [14,27,28].

The thermal management system configuration design considered in this paper belongs to an especially challenging class of optimization problems where discrete decisions change the set of continuous decisions to be made. This necessitates a nested discrete/continuous approach. Furthermore, the very general nature of these problems prevent the use of established integer programming methods that can be used for efficient solution of other problem classes with spatial properties [29]. At least three possible approaches for solving general physical systems configuration design problems exist. The first approach, as described above and used in the studies presented here, entails enumerating all unique and feasible configurations for a given design space, solving the continuous optimization problem for each one, and then producing a ranked set of design candidates. This is the only approach that can produce a result that is a confirmed global optimum, as it explores the entire design space. A second approach employs a population-based optimization algorithm, such as a Genetic Algorithm (GA), or other gradient-free search strategy to navigate the configuration space while still solving the continuous optimization problem for each configuration. A third approach is to utilize machine learning or artificial intelligence techniques to restructure the system configuration design space [30,31] such that it is more tractable, while again nesting continuous optimization within this search. Some recent machine learning strategies have proven to be only marginally better than random search [31], although design synthesis using machine learning does appear to have significant potential. General solution approaches for configuration problems are too large for enumeration and without special problem structure remains elusive, and is truly an engineering design grand challenge. Data sets generated via targeted enumeration may serve as an important basis for scaling to larger problems in the future using data-driven techniques.

In this article, multi-split TMS spatial considerations are also incorporated into the design problem; the spatial data of the system heat-devices are used for clustering and defining junctions where branching starts. After forming the clusters, we find the Euclidean distance of heat-device from the centroid of the cluster that each heat-device belongs to. The heat-device closest to the centroid is defined as the cluster's junction; refer to Sec. 4.2 for more details. This architecture design strategy is generalizable across a wide range of applications. It also applies to both single-split and multi-split architectures, supporting the quantification of trade-offs between multi-split system performance improvements and increases in cost and complexity.

The main contributions of the work presented in this article are:

1 Introduction of a new automated modeling strategy for multi-split thermal management systems. This strategy has applications for configuration generation across a wide range of systems.

2 Employing spatial information to cluster data and to define junctions.

3 Comparison of multi-split versus single-split configurations, providing insight into the trade-offs between system performance and system complexity.

4 In-depth study of multi-split system optimal results and analysis of the system signals such as flow rates and flow temperatures at heat-device locations that led to this optimal solution.

The analysis of the optimization study results produced evidence-based design guidelines for active thermal management system configuration design practice, also presented in this article.

This article continues as follows: In Sec. 2, we discuss the thermal management system architectures studied in this work. The dynamic graph based models of the thermal architectures are presented in Sec. 3. Section 4 describes the graph-based representation of multi-split architectures. The formulation of the variable time horizon dynamic optimization problem is explained in Sec. 5. Section 6 presents case studies using different architectures and heat loads. Section 7 concludes with a summary of the design methodology, guidelines for thermal management system design, and potential future research topics.

2 System Description and Modeling

Figure 2 illustrates the class of problems considered in this article. The purpose of the systems presented in the figure are to manage the temperature of various heat generating devices mounted on Cold Plate Heat Exchangers (CPHXs) through which a liquid coolant flows. The coolant is stored in a tank and is transferred to each branch by a pump. Each branch has several valves and can divide the flow it receives into its sub-branches. The coolant that passes through heat exchangers absorbs heat and transfers it to the thermal sink through a Liquid-to-Liquid Heat Exchanger (LLHX).

The class of architectures considered in this work are multi-split configurations as opposed to the single-split configurations generated in the study by Peddada et al. [14]. A single-split assumption limits the configuration search space; expanding the search space to include multi-split configurations may enhance system performance. The multi-split enumeration algorithm created in this work produces configurations where splits may be made at the pump (source) or at any of the CPHX locations. Sections 3 and 4 describe the graph-based physical modeling of the thermal systems

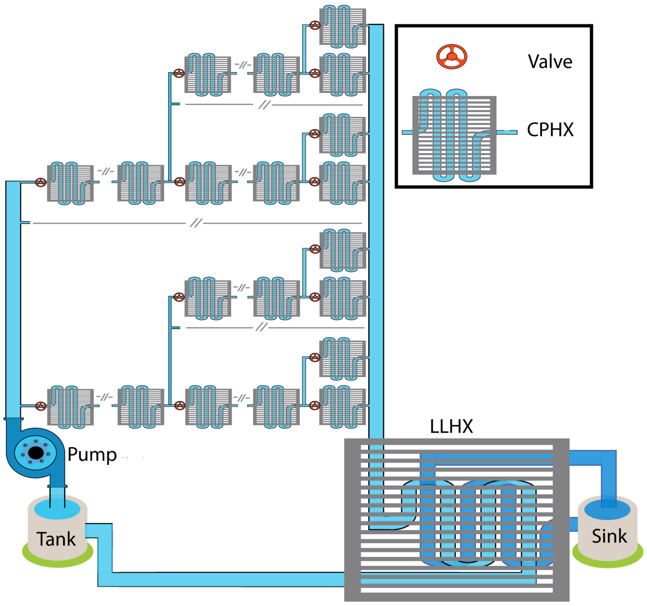


Fig. 2 Class of problems considered in this paper. The systems include a tank, a pump, valve(s), CPHXs in parallel and series, a LLHX, and a sink.

and the representation and generation of multi-split architectures. Here the heat load applied to each CPHX, as well as the inlet temperature and mass flow rate of the thermal sink, have been assumed to be known exogenous inputs. In addition, heat generating devices are assumed to have the same temperature as the wall of the CPHX on which they are mounted. It is assumed that heat loss through pipes is zero.

An optimal control problem is defined for each fluid-based thermal management systems configuration generated using the multi-split algorithm. The control problem seeks the optimal flow rate trajectory for each pipe that maximizes system performance, while satisfying component temperature constraints. The flow rates are controlled by valves. Dynamic system models incorporate the thermal physics of advection, convection, and bi-directional advection. A unique model is generated for each configuration design. We assume that system performance is quantified by thermal endurance, i.e., the goal is to maximize the time that the device is on while ensuring that all temperature bounds are met. This is consistent with the approach in the study conducted by Peddada et al. [14], supporting direct comparison with earlier single-split studies using graph-based configuration representations. Thermal endurance is an important objective because it ensures the system can operate reliably and efficiently over time, meeting safety standards and reducing maintenance costs. However, it should be noted that while thermal endurance is used as the objective for the control problem, other important metrics, such as graph complexity, component cost, or energy efficiency can be utilized to rank the graphs or quantify conflicting tradeoffs. The best graph for the overall system needs can then be chosen as the optimal configuration based on this ranking or tradeoff data.

The workflow diagram illustrating an overview of the solution procedure and steps involved in the code execution is provided in Fig. 3. In the subsequent sections, each part will be discussed in detail. In this diagram, the variable *Data* represents the positions of all CPHXs in $[x, y, z]$ coordinates. The variable *Heat Load & Graph Config* specifies thermal properties, heat load (disturbances) of each node, the depth of the split generation in the graph, and the graph index to select from the total population of graphs. Using the given *Data* and *Heat Load & Graph Config*, a *Base Graph* is generated, representing the connections between the tank, junctions (denoted as j_2 and j_3 in Fig. 3), and other CPHX nodes. Subsequently, a *Physics Graph* is created by adding CPHX wall nodes, sink nodes, and source power (heat load) to the *Base Graph*, along with the underlying physics between these nodes, which include convection, advection, and bidirectional advection. Based on the *Physics Graph*, an Open Loop Optimal Control (OLOC) is defined and solved. Then the OLOC optimization result is reported as the objective function value (thermal endurance) of the chosen graph.

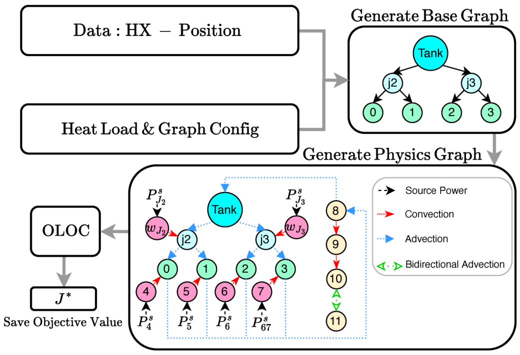


Fig. 3 Workflow diagram illustrating the steps involved in the code execution. Starting with the given *Data*, *Head Load* and *Config Number*, the base graph and physics graph are generated. The OLOC problem is then defined and solved to obtain the objective function value (thermal endurance with the incorporation of a penalty function for control signal smoothness and convergence). Refer to Sec. 5 for a comprehensive discussion.

3 Dynamic Graph-Based Modeling

This article uses the dynamic graph-based modeling framework discussed in Ref. [14,19]. Consider a graph with N_v vertices (nodes), N_e regular edges, and N_s source edges. The N_v nodes includes N_t sink nodes and $N_v - N_t$ regular nodes. A regular edge connects two regular vertices and a source edge connect a regular node to a source. Each regular node or sink node has an associated dynamic state. For the case of thermal management systems, the dynamic state of a regular vertex and sink vertex are the temperature T_i and the temperature T^t_i, respectively. Furthermore, each regular and source edge have associated rates of thermal energy transfer, P and P^s, respectively. For regular edges, P is a function of the temperature T of the corresponding two vertices (T_i and T_j) that make up this edge, and also a mass flow rate ($\dot{m}_{i-j}$) which can be treated as an input to the model. In addition, note that the sink vertex temperatures act as system disturbances, and the temperature of all regular vertices is obtained based on the thermal power transferred through edges and the thermal sink. In this system, the dynamics of each regular state must satisfy a conservation equation, as shown in Eq. (1). Here, i is the vertex index, $C_i > 0$ represents the thermal capacitance of the vertex, T_i is the temperature, P^{in}_{ij} represents the transfer of thermal energy from node j into node i, and P^{out}_{ij} represents the outgoing thermal energy of node i to node j. The summation Σ_j should be computed over all nodes that are connected (edged) to node i. Additionally, P^s_{i-e} represents the thermal energy from the source, and the summation Σ_e should be computed over all edges that are from sources to node i.

$$C_i \dot{T}_i = \Sigma_j \left(P^{\text{in}}_{ij} - P^{\text{out}}_{ij} \right) + \Sigma_e P^s_{i-e} \tag{1}$$

For fluid-based systems, the power flow through edge $(i - j)$ is a function of temperatures T_i and T_j, and $\dot{m}_{i-j}$, which is the mass flow rate through that edge. As a result we have:

$$P_{ij} = f\left(T_i, T_j, \dot{m}_{i-j}\right) \tag{2}$$

To describe the dynamics of regular vertices, we define a matrix $\mathbf{M}_{N_v \times N_e}$, where each element $m_{i,j}$ is defined as follows [14,18]:

$$m_{i,j} = \begin{cases} +1 & \text{if } P_{i,j} \text{ exits } i \text{ and enters } j \\ -1 & \text{if } P_{i,j} \text{ enters } i \text{ and exits } j \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Based on $\mathbf{M}$, we can define Eq. (4), where $\overline{\mathbf{M}}$ is a structural mapping from regular power flows P to regular temperatures T, and $\underline{\mathbf{M}}$ is a structural mapping from regular power flows P to vertex temperatures T^t.

$$\mathbf{M}_{N_v \times N_e} = \begin{bmatrix} \overline{\mathbf{M}}_{(N_v - N_t) \times N_e} \\ \underline{\mathbf{M}}_{N_t \times N_e} \end{bmatrix} \tag{4}$$

Similarly, we can define the matrix $\mathbf{D}_{(N_v - N_t) \times N_s}$ for the connection of external sources to the system, where each element $d_{i,e}$ is:

$$d_{i,e} = \begin{cases} +1 & \text{if } P^s_{i-e} \text{ enters } i \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Based on Eq. (1) and the above definitions, we can define the dynamics of all regular states as follows, where $\mathbf{C} = \text{diag}([C_i])$ is the diagonal matrix of capacitances.

$$\mathbf{C}\dot{T} = -\overline{\mathbf{M}}P + \mathbf{D}P^s \tag{6}$$

This section summarizes the equations employed in the automated dynamic model generation. For a more detailed exposition, see Ref. [14].

3.1 Graph-Based Model for Multi-split Architectures.

Figure 4 illustrates the graph corresponding to the class of system architectures considered in this article. To model this class using the graph-based approach, vertices correspond to temperatures of the fluid in the tank, fluid in the sink, fluid in each CPHX, wall of each CPHX, fluid in each side of the LLHX, and the wall of the LLHX. The thermal capacitance for fluid temperatures is $C_i = \rho V_i c_p$, where V is the fluid volume, c_p is the specific heat capacitance, and ρ is the fluid density. The wall thermal capacitance is $C_i = M_{w,i} c_{p,w,i}$, where M_w is the wall mass and $c_{p,w}$ is the specific heat capacitance of the wall material. Thermal capacitance values are assumed to be constant in all analyses.

In this article, two types of power flows are required: convection and advection. The convective power flows are governed by $P_{ij} = h_{ij} A_{ij} \left(T_j - T_i\right)$ and advective power flows are governed by $P_{ij} = \dot{m}_{i-j} c_p T_j$, where A_{ij} is the convective surface area, h_{ij} is the heat transfer coefficient, and $\dot{m}_j$ is the mass flow rate of the fluid. Additionally, when fluid circulates between two thermal elements, such as the LLHX's secondary side and the thermal sink, both directions' advective power flows can be combined into a single 'bidirectional advection' power flow: $P_{ij} = \dot{m}_{i-j} c_p (T_j - T_i)$.

Figure 5 depicts the governing dynamics of the main components for the problem class considered in this article. The model includes advection, convection, and bidirectional advection. The power-flow type in the tank is advection, which occurs between 1) the tank and the fluid node on the CPHX, and 2) the tank and the primary side of the LLHX. Advection and convection are present in the CPHX. Advection occurs between 1) the CPHX fluid node and tank and 2) the CPHX fluid node and primary side of the LLHX. Convection occurs between the CPHX fluid and the wall node. The LLHX involves advection, convection, and bi-directional advection. The advection occurs between 1) the primary side of the LLHX and tank, and 2) the primary side of the LLHX and the CPHX fluid node. The bi-directional advection takes place between the secondary side of the LLHX and the sink. The convection occurs between 1) the LLHX wall node and the primary side of LLHX and 2) the LLHX wall node and the secondary side of LLHX. These are building blocks of any larger system. Using these components, a model of the dynamics of any complex system of the form illustrated in Fig. 4 can be generated.

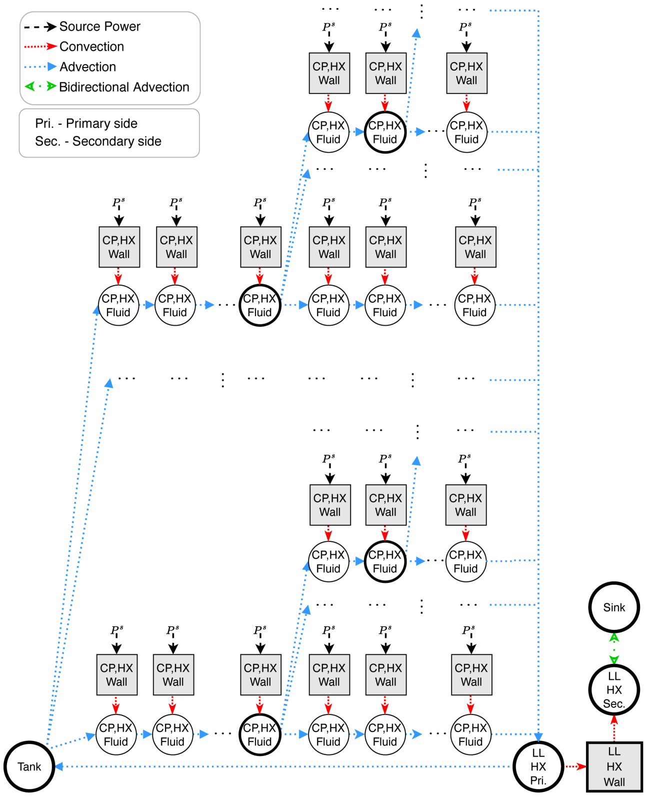


Fig. 4 Graph-based model for multi-split architectures studied in this article. Here, Fluid nodes are represented by circles, wall nodes by squares, and thermal power is indicated by directional lines (distinguished according to the legend).

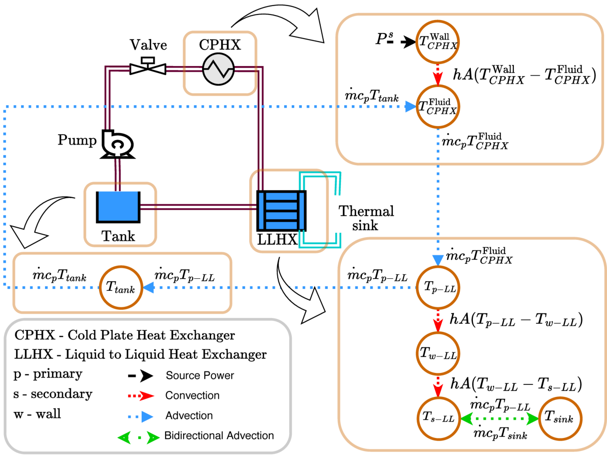


Fig. 5 Notional example to illustrate the elements of thermal physics included in this system model. Here, a node (circle) represents a temperature and an edge (directional line) represents a thermal power flow.

For the systems considered here, Eq. (2) can be written as [14]:

$$P_{ij} = a^1_{ij} T_j + a^2_{ij} T_i + b^1_{ij} \dot{m}_{i-j} T_j + b^2_{ij} \dot{m}_{i-j} T_i \tag{7}$$

where the system's coefficients a_{ij} and b_{ij} are constants for different power flows: for convection, $a^1_{ij} = h_{ij} A_{ij}$, $a^2_{ij} = -h_{ij} A_{ij}$, and $b^1ij = b^2_{ij} = 0$; for advection, $b^1_{ij} = \dot{m}_{i-j}c_p$ (with others zero); for bidirectional advection, $b^1_{ij} = \dot{m}_{i-j}c_p$ and b^2_{ij} is its negative. Two other external system inputs are considered: the source power (P^s) and the thermal sink temperature (T^t).

3.2 State Equations for the Graph-Based Model. The TMS state equations are obtained by using a methodology similar to the one presented in Refs. [14,19,20]. The methodology described there has been extended to accommodate multi-split configurations. The elements $m^a_{i,j}$ of matrix $\mathbf{M}^a_{N_v \times N_e}$ are defined in Eq. (8), where $a^1_{i,j}$ and $a^2_{i,j}$ are the coefficients in Eq. (7). $\mathbf{M}^b_{N_v \times N_e}$ can be defined similarly using the coefficients $b_{i,j}$ in Eq. (7).

$$m^a_{i,j} = \begin{cases} a^1_{i,j} & \text{if } P_{i,j} \text{ exits } i \text{ and enters } j \\ a^2_{i,j} & \text{if } P_{i,j} \text{ enters } i \text{ and exits } j \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Please note that since there are more edges in the system than independent flow rates, there is not a one-to-one relationship between the mass flow rates of the system architecture and the edges of its thermal graph. As a result, we should map the flow rates to their corresponding flow in each edge. To do so, first, based on the graph structure and known pump flow rate m_p and tank flow rate m_t, we define $\dot{m}_{\text{dp}}$ as the flow rate in independent branches. The derivative of $\dot{m}_{\text{dp}}$ is later defined as a control signal and is obtained through OLOC optimization. After that, we compute the flow rate for dependent branches ($\dot{m}_{\text{indp}}$) by using the graph structure and the pump, sink, and independent branches' flow rate. This algebraic computation can be represented through matrix multiplication. Please refer to Fig. 6 for a simple example. In the next step, we define an $N_e \times (n_{dp} + n_{\text{indp}} + 2)$ matrix $\boldsymbol{Z}$ that maps these flow rates to corresponding edges. Here, n_{dp} and n_{indp} are the number of independent and dependent flow rates, respectively, and the +2 accounts for the pump and sink flow rates. One main difference between single and multiple split TMS models is the $\boldsymbol{Z}$ matrix. Additionally, as we move from single to multi-split, the number of nodes and branches increases.

As a result, we have:

$$\boldsymbol{P} = [\boldsymbol{M}^a]^T \begin{bmatrix} \boldsymbol{T} \\ T^t \end{bmatrix} + \text{diag}\left(\boldsymbol{Z} \begin{bmatrix} \dot{m}_p \\ \dot{\boldsymbol{m}}_f \\ \dot{m}_t \end{bmatrix}\right) [\boldsymbol{M}^b]^T \begin{bmatrix} \boldsymbol{T} \\ T^t \end{bmatrix} \tag{9}$$

Hence, by using Eqs. (6) and (9), the state space equations can be written as:

$$\dot{\boldsymbol{T}}(t) = \boldsymbol{A} \begin{bmatrix} \boldsymbol{T}(t) \\ T^t(t) \end{bmatrix} + \boldsymbol{B}_1 \left(diag \left(\boldsymbol{Z} \begin{bmatrix} \dot{m}_p(t) \\ \dot{\boldsymbol{m}}_f(t) \\ \dot{m}_t(t) \end{bmatrix} \right) \right) \boldsymbol{B}_2 \begin{bmatrix} \boldsymbol{T}(t) \\ T^t(t) \end{bmatrix} + \boldsymbol{C}^{-1}\boldsymbol{D}\boldsymbol{P}^s(t) \tag{10}$$

where:

$$\boldsymbol{A} = -\boldsymbol{C}^{-1}\overline{\boldsymbol{M}}[\boldsymbol{M}^a]^T \tag{11}$$

$$\boldsymbol{B}_1 = -\boldsymbol{C}^{-1}\overline{\boldsymbol{M}} \tag{12}$$

$$\boldsymbol{B}_2 = [\boldsymbol{M}^b]^T \tag{13}$$

A simple example problem is shown in Fig. 6. Each of the two junctions has 3 associated nodes. The first junction (j_2) distributes the flow to 3 different branches, and the second junction (j_3) distributes the flow to two branches, where 2 nodes in a branch are in series, and the second branch has only 1 node. In this figure, *states* are denoted using the symbol ξ; here the states all correspond to node temperatures. For example, ξ_1 is the temperature of node 1 and ξ_{w-1} is the wall temperature of the CPHX connected to node 1. The system involves both independent and dependent flows, as depicted in Fig. 6. Independent flows can be controlled, whereas dependent flows can be calculated based on the independent flows (see Fig. 6 for these equations). In the figure, the edges that carry dependent power flows are shown with dashed lines.

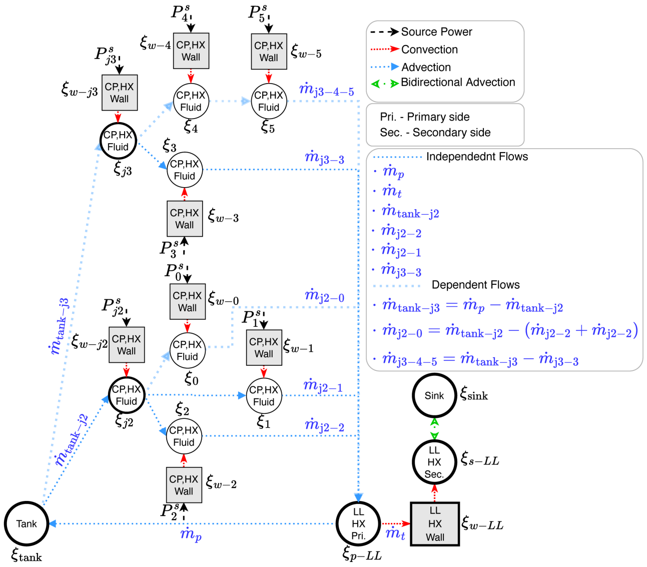
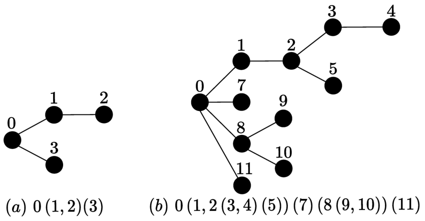


Fig. 6 Graph illustrating the variables for a simple example multi-split configuration. Types of power flow are illustrated, including the distinction between independent and dependent flows.

4 Generating Multi-split Spatial Graphs

4.1 Graph Representation. The system configurations here can be represented by acyclic undirected connected graphs where any two nodes are connected by only one simple path (i.e., a tree). In the current representation, the tank is always the root node and each CPHX is labeled with a number from 1 to N. Splittings occur at the root or at any other node. In the corresponding symbolic representation, branching is shown by parentheses and consequent CPHXs, i.e., nodes in one branch are separated by a comma. Figure 7 shows the representation for two configurations and their equivalent graphs.



(a) 0 (1, 2) (3) (b) 0 (1, 2 (3, 4) (5)) (7) (8 (9, 10)) (11)

Fig. 7 Two configuration representations and their equivalent graphs.

The number of single-split configurations increases rapidly as the number of heat-generating devices increases. The equation $G(N) = \sum_{k=1}^{N} \binom{N}{k}\binom{N-1}{k-1}(N-k)!$, adopted from Refs. [14,32], quantifies the number of generated configurations for this enumeration. In this equation, k and N represent the number of CPHXs connected to the root (tank) and total number of nodes (CPHXs). Here, $\binom{N}{k}$ is the number of ways to choose k CPHXs to connect to the root. $\binom{N-1}{k-1}$ is the number of k-compositions of an integer N, where a k-composition is a sum of k positive integers that add up to N. Here, $\binom{N-1}{k-1}$ can be viewed as a lower bound that illustrates how rapidly the number of single-split configurations increases with N. When splits (or *junctions*) are added, the total number of configurations increases more rapidly as the single-split graphs are the base case. Multi-split graph models can represent both all single-split graph scenarios (where branching can happen at the root) as well cases with branching at any node other than the root. As a result, the number of possible multi-split configurations grows more with the number of nodes. Derivation of bounds on multi-split configuration growth is an open question and left for future work.

The addition of junction nodes and layers leads to a rapid growth in the number of graphs. It is desirable to impose constraints in a way that meaningfully limits this growth. One strategy is to recognize that these graphs represent physical designs, and that spatial system information can be leveraged to reduce problem complexity. Here we employ a spatial clustering method to group the nodes in sub-domains and enumerate the sub-branches only in the nearby neighborhoods; this procedure is detailed in Sec. 4.2. This approach, however, cannot access all possible designs (as can enumeration). Therefore, we also examine a second strategy where the junction locations are enumerated. Exploring the trade-offs between performance improvement and reducing problem complexity is investigated in this article; in some cases enumeration may be worth the added computational cost.

4.2 Generation of Multi-split Spatial Graphs. Algorithms 1–4 describe the multi-split graph generation framework of this work. The generation procedure passively chooses junction nodes during run-time instead of employing all the nodes in enumerations from the beginning. It performs recursion for refinement. The enumerations can cover (i) tank-junctions, (ii) junction-CPHXs, and (iii) both, based on the specific requirements of the optimization problem; this study focuses on class (ii). In Algorithm 1, we utilized K-Means clustering and a threshold to define various clusters for nodes based on their 3-D space locations; these clusters are used in Algorithm 2 to form Super-Nodes. In Algorithm 1, the vertex data were grouped into a varying number of clusters, ranging from 1 to half of the total number of vertices (rounded down). Inertia, which measures how well a dataset was clustered by K-Means, is computed by measuring the distance between each data point and its centroid, squaring this distance, and summing these squares within a single cluster. The change in slope of a curve based on all inertia data was utilized to determine the number of clusters. Here, stable clustering is defined as when the slope change of inertia data falls below a specified threshold (0.1 here).

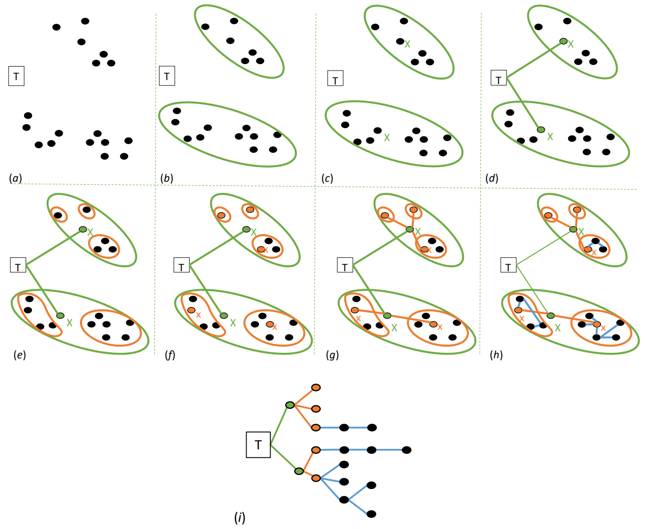


Fig. 8 Clustering and the creation of a sample configuration for a set of nodes. This figure shows (a) the spatial locations of a source (T) and 17 nodes, (b) first-level clusters, (c) centroid of each cluster, (d) chosen junctions and a routing from T to junctions, (e) second-level clusters, (f) the centroid and the corresponding junctions of the second-level, (g) a routing from first-level to second-level junctions, (h) a sample enumeration in the internal cluster with more than 1 node. Figure (i) presents the generated sample graph.

Algorithm 1 Selecting Number of Clusters to form Super-Nodes

```
$N \leftarrow$ number of nodes
for $K$ in range(1, $N$) do
    cluster nodes into $K$ clusters via K-means
    if the clustering is stable, stop
return $K$ and the corresponding nodes of each cluster
```

Algorithm 2 Generation of a Hierarchical Tree of Super-Nodes

```
$T \leftarrow$ tree
super-node[0]$\leftarrow \{P[0], D[0]\}$ here $P[0]$ is Tanks and $D[0]$ is the spatial data
    of all nodes
$L[0]$ includes super-node[0]
$numL \leftarrow$ number of levels in the tree
$m \leftarrow$ number of super-nodes in $T$
for $l$ in range(1, $numL$) do
    $Ids \leftarrow$ index of all super-nodes in $L[l-1]$
    for $i$ in $Ids$ do
        cluster super-node $i$ into $K[i]$ clusters using Algorithm 1
        for $k$ in range(1,K[i]) do
            $m = m + 1$
            $P_k \leftarrow$ the closest node to centroid of $D(k)$ ($P_k$ is junction.)
            $P[i] \leftarrow [P_1, P_2, ..., P_n, P_k]$
            remove $P_k$ from $D(k)$
            $T[m] \leftarrow \{P[k], D[k]\}$
            $L[l] \leftarrow$ add $T[m]$
return Tree T with numL levels
```

Algorithm 3 Graph Enumeration

```
$N \leftarrow$ number of nodes
$eG[0] \leftarrow [[0]]$          ▷ each sub-list represents the adjacency list of a graph
$eG[1] \leftarrow [[(0, 1)]]$
for $n$ in [2, $N$] do
    $Parents[n] \leftarrow [1, \dots, n-1]$
    for $g$ in $eG[n-1]$ do
        for each $Node$ in $g$ do
            if $Node$ belongs to $Parents[n]$ then
                add edge ($Node$, $n$) to $g$ and add the adjacency list of the new
                graph to $eG[n]$
return all enumerated graphs $eG[N]$
```

Algorithm 4 Graph Generation for a Selected Tree Level

```
$l \leftarrow$ Tree level selected for graph generation
$Ids \leftarrow$ index of all super-nodes in $L[l]$ for tree generated by Algorithm 2
for $i$ in $Ids$ do
    $Q \leftarrow$ parents of the super-node ($P[i]$)
    $g1[1:j] \leftarrow$ enumerate super-node[$i$] using Algorithm 3 with its junction
        as its root
    $g2 \leftarrow$ the circular graph of $Q$ with $(P[-1], P[0])$ edge removed
    $g[i, 1:j] \leftarrow$ merge $g2$ with every $g1[1:j]$
combine a sub-graph selected from each super-nodes of level l to generate all the
    graphs
```

Algorithm 2 initially creates trees using node spatial data. The node data are clustered to form *super-nodes* in a recursive manner. Here, a level represents the levels of splitting in an architecture tree. A junction node is defined for each super-node by choosing the node with the smallest Euclidean distance from the cluster's centroid; see Fig. 8 and Algorithm 2. Note that a junction represents a CPHX node where the coolant flow splits into branches. Next, we enumerate all sub-graphs of a selected tree level. This means, for a chosen super-node in that tree level, its nodes are enumerated to form all possible sub-graphs; Algorithm 3 describes how the enumeration algorithm works. Afterwards, we connect the root node (Tank) and the corresponding junction nodes of the super-node to its sub-graphs; see Algorithm 4. This process is performed for all super-nodes of the selected level. Finally, an architecture graph is created by choosing and merging one sub-graph from the pool of sub-graphs generated for each super-node in that level; refer to Algorithm 4. Note that various sub-clusters can be defined within a cluster to form multiple layers of junction nodes.

In addition to the above mentioned graph generation algorithm, we also generated graphs using a second strategy where the locations of junctions in the trees are enumerated. The main difference between the first and second graph generation strategies is in topological locations of junction nodes in the architectural graphs.

We generate all configurations within a class, evaluate the performance of each configuration, and choose the configuration with the best optimized performance. Figure 9 shows all 13 configurations made by the single-split algorithm for 3 nodes, depicting both the base graphs and the extended physics graphs generated for simulations. Figure 10 shows two sample configurations and their physics models for a multi-split system with 19 nodes (root, 6 junction CPHXs, and 12 CPHXs nodes). In the base graph, there are no dynamics; only the connections between tank and fluid nodes are determined using Algorithms 1–4. Then, physics graphs are generated based on the dynamic equations developed in Sec. 3.

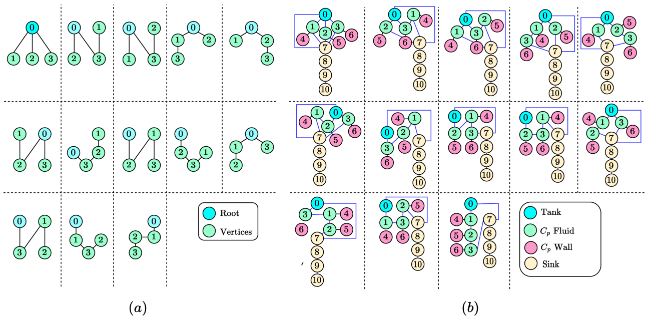


Fig. 9 The (a) base and (b) complete physics graphs for single-split systems with 3 CPHXs

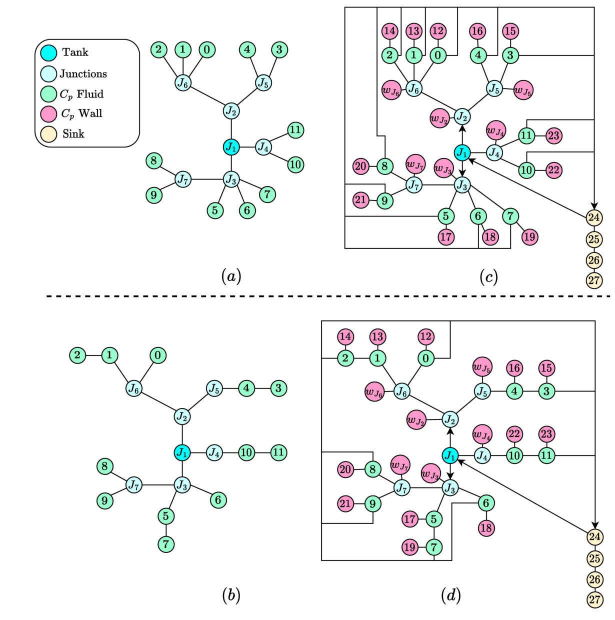


Fig. 10 Two sample configurations for a multi-split system with 18 CPHXs. Figures (a) and (b) show the base graphs, and Figs. (c) and (d) show the complete physics graphs for the base graphs (a) and (b), respectively.

5 Open-loop Optimal Control (OLOC) Problem

The objective of the optimal control problem is to maximize the thermal endurance while satisfying limits on temperatures and mass flow rates. When any of the node's temperatures (defined as states) reaches the upper bound, the OLOC terminates, and the final time is recorded as the thermal endurance. Figure 11 illustrates the OLOC implementation structure. Here, we seek to determine a control trajectory $u(t)$ for each independent flow that maximizes the objective function while satisfying constraints. Within this figure, Eq. (c_1) presents the system states ($\boldsymbol{\xi}$), encompassing the vector of temperature nodes ($\boldsymbol{T}$), and the flow rate of independent branches ($\dot{\boldsymbol{m}}_{\text{indp}}$). The dynamics that shows how these states evolve over time are shown in Eq. (c_8). Equation (c_2) defines the control signals as the rate of change of the valve flow rates in independent branches. The flow rate in dependent branches is determined by an algebraic equation that ensures that the input flow rate of each branch is equal to the output flow rate of that branch. This equation can be represented as a matrix multiplication, as shown in Eq. (c_3), where matrix $\mathbf{M}$ maps the flow rate of independent branches ($\dot{\boldsymbol{m}}_{\text{indp}}$) to the flow rate of dependent branches ($\dot{\boldsymbol{m}}_{\text{dp}}$). A practical illustration of this calculation is provided in Fig. 6. In Eq. (c_4), the total flow rate of all branches is presented, comprising both independent and dependent flow rates. It is worth mentioning that the order of combining independent and dependent flow rates depends on the graph structure. To achieve this, the code, developed for the analysis, categorizes nodes into parents and children, where children of a parent are nodes branching from it. If a parent node has N children, it results in $N-1$ independent flow rates. This process is illustrated in Fig. 12 for a graph with 9 nodes. Nodes are initially divided into child and parent groups based on their connections, then independent and dependent branches are identified. Here, independent branches are represented by solid lines, while dependent branches are shown as dashed lines.

Run OLOC

· Define States: All temperatures and and flow rate of independent branches

$$\xi(t) = [\mathbf{T}(t), \dot{\mathbf{m}}_{\text{indp}}(t)] \qquad (c_1)$$

· Define Controls: Derivative of flow rate of independent flows

$$\mathbf{u}(t) = [\ddot{\mathbf{m}}_{\text{indp}}(t)] \qquad (c_2)$$

· Define Dependent Flows: Flow rate of dependent branches

$$\dot{\mathbf{m}}_{\text{dp}}(t) = \mathbf{M} \times \dot{\mathbf{m}}_{\text{indp}}(t) \qquad (c_3)$$

· Define All Flows: Flow rate of all branches

$$\dot{\mathbf{m}}_f(t) = \begin{bmatrix} \dot{\mathbf{m}}_{\text{dp}}(t) \\ \dot{\mathbf{m}}_{\text{indp}}(t) \end{bmatrix} \qquad (c_4)$$

· Define Initial condition , Bound, Path vonstraint, and Dynamics

$$\mathbf{T}_w(0) = \mathbf{T}_{w,0} \qquad \mathbf{T}_f(0) = \mathbf{T}_{f,0} \qquad \mathbf{T}_l(0) = \mathbf{T}_{l,0} \qquad (c_5)$$

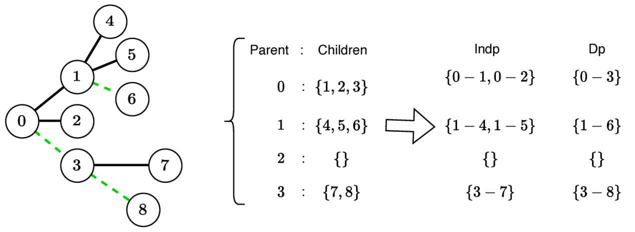
$$\mathbf{T}_w(t) \leq \mathbf{T}_{w,\max} \qquad \mathbf{T}_f(t) \leq \mathbf{T}_{f,\max} \qquad \mathbf{T}_l(t) \leq \mathbf{T}_{l,\max} \qquad (c_6)$$

$$0 \leq \dot{\mathbf{m}}_{\text{dp}}(t) \leq \dot{m}_p \qquad 0 \leq \dot{\mathbf{m}}_{\text{indp}}(t) \leq \dot{m}_p \qquad |\mathbf{u}(t)| \leq \ddot{m}_{f,\max} \qquad (c_7)$$

$$\begin{cases} \dot{\mathbf{T}}(t) = \mathbf{A} \begin{bmatrix} \mathbf{T}(t) \\ T^t(t) \end{bmatrix} + \mathbf{B}_1 \left(diag \left(\mathbf{Z} \begin{bmatrix} \dot{m}_p(t) \\ \dot{\mathbf{m}}_f(t) \\ \dot{m}_t(t) \end{bmatrix} \right) \right) \mathbf{B}_2 \begin{bmatrix} \mathbf{T}(t) \\ T^t(t) \end{bmatrix} + \mathbf{DP}^s(t) & (c_8) \\ \ddot{\mathbf{m}}_{\text{indp}}(t) = \mathbf{u}(t) & (c_9) \end{cases}$$

· Define Objective: $\max\ t_{\text{end}} - \lambda \int_0^{t_{end}} \|\ddot{\mathbf{m}}_f(t)\|_2^2 \, dt \qquad (c_{10})$

Fig. 11 OLOC components



Parent :	Children		Indp	Dp
0 :	$\{1,2,3\}$		$\{0-1, 0-2\}$	$\{0-3\}$
1 :	$\{4,5,6\}$	⇨	$\{1-4, 1-5\}$	$\{1-6\}$
2 :	$\{\}$		$\{\}$	$\{\}$
3 :	$\{7,8\}$		$\{3-7\}$	$\{3-8\}$

Fig. 12 Procedure to define dependent and independent branches in OLOC problem. Here, Independent branches are represented by solid lines, while dependent branches are shown as dashed lines.

In Fig. 11 the initial conditions for the temperature nodes are given by Eq. (c_5). In this equation, $\mathbf{T}_w$ represents the wall temperature of the CPHX states, $\mathbf{T}_f$ represents the fluid temperature of the CPHX states, and $\mathbf{T}_l$ represents the temperature of the Tank and LLHX states. Equation (c_6) presents inequality constraints that ensure the operating temperature of each component remains within an upper bound throughout the entire time horizon. The first two terms in Eq. (c_7) represent the inequality constraint that guarantees the flow rates of both independent and dependent branches remain within a specified bound. This equation, when combined with Eq. (c_3), ensures that the input flow rate of each branch is equal to its output flow rate. It also ensures that the maximum flow rate cannot exceed the pump flow rate ($\dot{m}_p$). The last term in Eq. (c_7) represents the limit on the derivative of the flow rate, capturing the physical limitations of the valves [14]. Equations (c_8) and (c_9) show how states defined in Eq. (c_1) evolve over time. In Eq. (c_{10}), the objective value is represented, aiming to maximize the thermal endurance. Similar to the approach in Ref. [14], a penalty term is incorporated to facilitate solution smoothness and enhance convergence, because if the control signal ($\ddot{\boldsymbol{m}}_f$) is not included in the objective, numerical issue often emerge involving bang-bang control and singular arcs [33], and the computation cost of OLOC increases [34]. The parameter λ is selected such that the total penalty cost remains below

1% of t_{end}. Table 1 shows the parameters used in the physical simulations for the studies in this article.

Table 1 Parameters used in the physics modeling of the thermal systems [14].

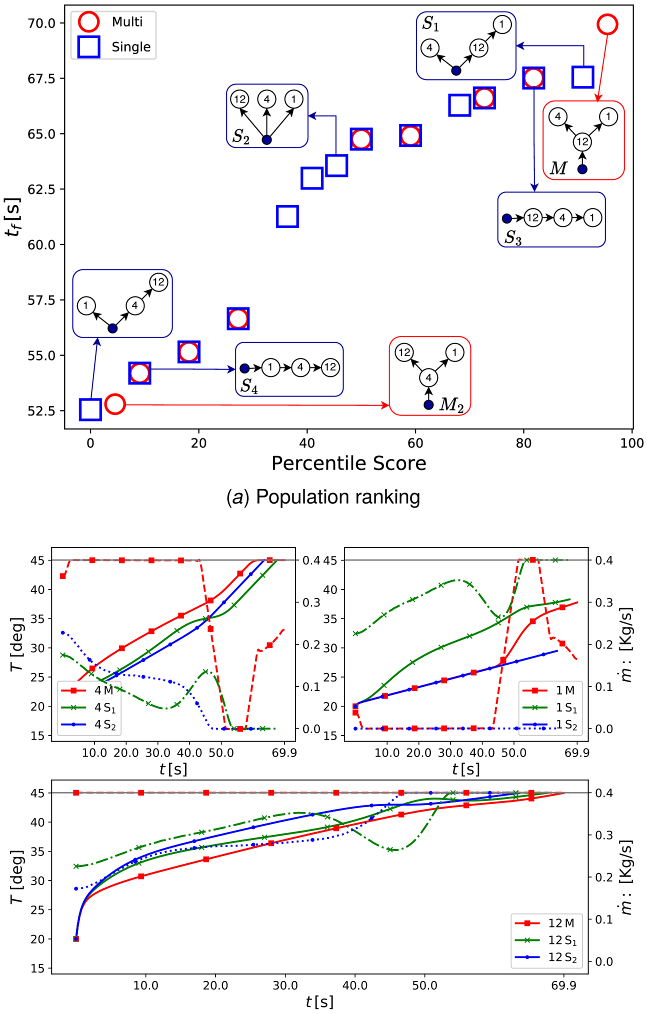
Parameter	Value
LLHX wall mass	1.2 kg
CPHX wall mass	1.15 kg
Tank fluid mass	2.01 kg
Thermal sink temperature T^t	15^o C
Tank/LLHX initial temperature , $T_{l,0}$	15^o C
CPHX initial wall temperature, $T_{w,0}$	20^o C
CPHX initial fluid temperature , $T_{f,0}$	20^o C
Thermal sink mass flow rate, $\dot{m}_t$	0.2 kg/s
Pump mass flow rate $\dot{m}_p$	0.4 kg/s
Valve rate limit $\ddot{m}_{f,\max}$	0.05 kg/s^2
Penalty parameter, λ	$0.01/(N_f \ddot{\boldsymbol{m}}^2_{\text{f,max}})$

There are two main approaches for solving OLOC problems: 1) Indirect (optimize then discretize) and 2) Direct (discretize then optimize). In the indirect method, a differential algebraic equation is derived using optimality conditions (the calculus of variations or the Pontryagin minimum principle). These equations are then discretized and solved numerically [35,36]. In contrast, the direct method first discretizes the problem so it can be transformed into a nonlinear program (NLP), which can then be solved by a nonlinear programming solver such as SNOPT [37] or IPOPT [38]. The indirect method provides more information about the structure of the problem, but has limited value for solving constrained problems. Alternatively, direct methods use NLP solvers to solve complex constrained problems successfully; some well-established OLOC software tools based on the direct method are available, for example GPOPS [34] and Dymos [39]. The studies presented here utilize Dymos to solve the OLOC problems, which is an open-source software tool developed in Python. The computational cost of solving the OLOC problem depends on the size of the graph, but on average, it takes approximately two minutes to solve the problem for each configuration using a workstation with an AMD EPYC 7502 32-Core Processor @ 2.5 GHz, 64 GB DDR4-3200 RAM, LINUX Ubuntu 20.04.1, and Python 3.8.10. Additionally, it is important to highlight that in these studies, the evaluation of the nonlinear optimal control problems for each of the architectures has been parallelized, which significantly reduces computational solution expense.

6 Case Studies

We present three case studies to illustrate how this work can help engineers design optimal thermal management systems. In the first case study (Sec. 6.1), the goal is to obtain the optimal structure for thermal management systems having 3 and 4 CPHXs. Here we use enumeration and compare both single-split and multi-split cases. For 3 and 4 CPHXs systems, the heat loads are $[12, 4, 1]$ kW and $[12, 4, 1, 1]$ kW, respectively. In Sec. 6.2, we show the results for the multiple-split case under two heat loads (disturbances): $[5, 5, 5, 5, 5, 5]$ kW and $[5, 7, 6, 4, 5, 5]$ kW. This example demonstrates that the optimal configuration can change when disturbances are different. OLOC signals are also compared for selected configurations. While the main focus of this article revolves around the results obtained from small graphs for detailed analysis and discussions, it is important to note that this methodology is applicable to graphs of any size. This versatility is demonstrated in Sec. 6.3, where a graph consisting of 18 nodes, with a Tank, 14 CPHXs, and 3 junctions CPHXs, is studied. It should be noted that in all these cases, the method illustrated in Fig. 8 is utilized for junction creation. Additionally, in all of these scenarios, the heat loads remain constant (not time-varying). However, the OLOC problem depicted in Fig. 11 is not limited to constant heat loads, and time-varying heat loads can be easily accommodated. The only drawback is that computational expense increases, as more discretization points in the OLOC problem are normally required when considering time-varying heat loads. It is worth noting that even when heat loads are constant, we are able to observe system dynamics and capture transient responses because the derivative of flow rate, which is a control signal, is not constant, as demonstrated in Sec. 6.1-6.3. Furthermore, as the OLOC problem is solved numerically, it requires an initial guess for all states and controls. While in many cases we have observed identical states and controls for various initial guesses, there may be some cases that are more sensitive to the initial guess. However, we have observed that even in those cases, the resulting objective value is almost identical.

6.1 Comparing single-split and multi-split cases with 3 and 4 CPHX-nodes. Here we aim to find the best architectures among single-split and multi-split configurations with 3 and 4 nodes. The result for the first scenario with 3 nodes is shown in Fig. 13. In this illustration, only the fluid nodes of CPHX and the tank are depicted, while the disturbance of each CPHX is displayed in its respective node. Here, *Multi* represents multi-split cases and *Single* represents single-split cases. The horizontal axis in Fig. 13(a) shows the percentile score and the vertical axis shows thermal endurance. Here, a percentile score represents the relative position of a value within a dataset by indicating the percentage of values that are strictly lower than it. Thus, the best case is at the top right, and the worst case is at the bottom left. As we see, the multi-split architecture yields the best result. It should be mentioned that the results obtained depend on the heat load. For example, here, we have an extreme load (12) that is much larger than other loads. Multi-split configurations usually produce a better result under large loads since the node with the maximum disturbance is connected to the tank and receives the maximum available flow-rate. However, in many other cases, the flow rate that this node receives is a smaller fraction of the pump flow rate. It is worth mentioning that, as shown in Fig. 13(a), there are instances where both multi-split and single-split cases achieve the same objective function value. This occurs in situations (such as S_4) where there is only one branch, resulting in zero splits. This type of configuration is generated by both single and multi-split algorithms, hence it is counted twice. In this analysis, the total number of single- and multi-split cases are $n_S = 13$, and $n_M = 9$, respectively.



(*a*) Population ranking

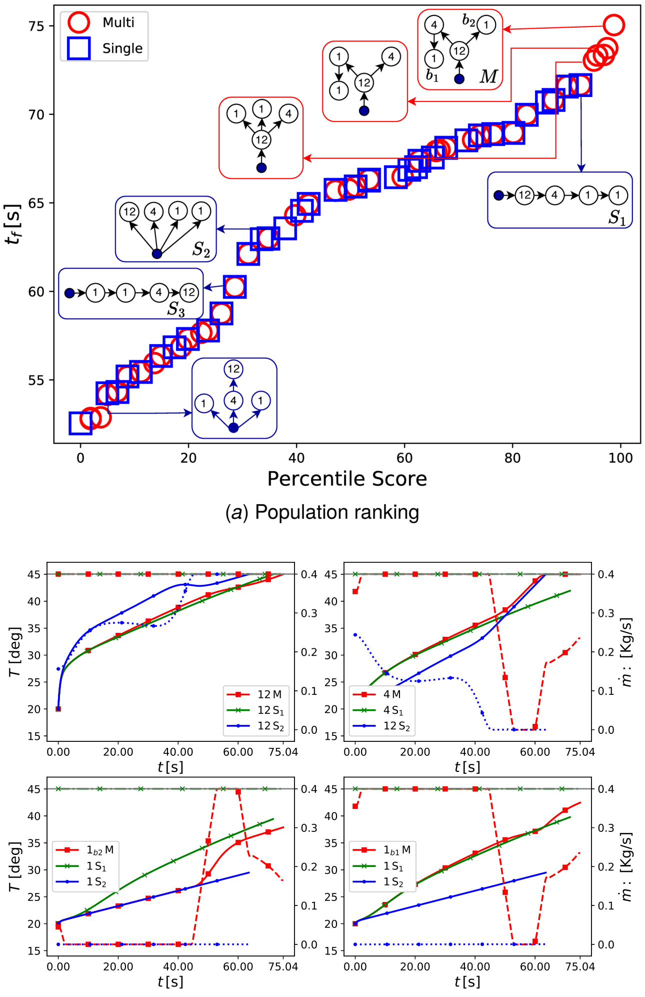
(*b*) Flow rate and wall temperature of the cases shown in Fig. 13(a)

Fig. 13 Comparison of multi- and single-split cases with 3 nodes having $dist = [12, 4, 1]$ kW. Number of cases: $n_S = 13$, $n_M = 9$. In Fig. 13(b), temperature signals are depicted with solid lines. Temperature and flow rate data corresponding to the same node are indicated by identical colors and symbols.

In Fig. 13, three cases are denoted as M (Multi-split), S_1 (single-split-1), and S_2 (single-split-2). The flow rate and wall temperature of these three cases are presented in Fig. 13(b). Here, the left vertical axis represents the temperature, while the right vertical axis displays the flow rate. Temperature signals are depicted with solid lines, whereas flow rate data are presented in dashed, dotted, and dash-dotted lines. Both temperature and flow rate data corresponding to the same node are indicated by identical colors and symbols. Additionally, a horizontal gray line indicates the maximum constraint values for both the flow rate and temperature. Among the three cases (M, S_1, S_2), S_2 exhibits the highest control authority (3) due to its maximum parallel flows. This suggests that the best results can be expected from this configuration. However, this assumption may not hold true universally. A significant difference between M and S_2 lies in how the node with a 12 kW heat load is handled. In the case of M, this node receives the maximum flow rate (pump flow rate) as it is connected directly to the pump. Conversely, in S_2, the flow received by this node is a fraction of the pump flow rate. As depicted in Fig. 13(b), the flow rates of these nodes differ across each graph. Consequently, this disparity impacts the objective function value and results in varying temperatures. Additionally, as shown in the left vertical of Fig. 13(b), the wall temperature of nodes 12 and 4 reached the upper bounds. Notably, the case labeled as 'M' achieves this upper bound at a later stage compared to the other cases, indicating a better objective function value (specifically, 69.9° C).

As mentioned earlier, one of the advantages of using a multi-split configuration in this case is that the node with the maximum heat load is directly connected to the pump and receives the maximum flow rate. Therefore, one might expect similar results if all nodes are connected in series directly to the pump. However, as depicted in Fig. 13(a), when these nodes are arranged in series with the pump and nodes with higher loads are positioned closer to the pump (S_3), the achieved result, although satisfactory, is inferior to the multi-split case. This disparity arises because the dynamics involved in this problem, such as convection, advection, and bidirectional advection, are also dependent on the graph's structure. Consequently, even though node 12 in both M and S_3 receives the full pump flow rate, their distinct dynamics lead to different objective function values. It should also be noted that even with a fixed graph structure, altering only the load locations yields different outcomes. For instance, as illustrated in Fig. 13(a), cases S_3 and S_4 possess the same structure but exhibit substantially different objective function values (t_f). Furthermore, simply changing the positions of the 12 and 4 heat loads results in significantly different objective function values for the multi-split cases (M and M_2). In these cases, we generally observe that when the structure remains fixed, the objective function value tends to improve when nodes with higher heat loads are positioned closer to the pump. This can be attributed to the fact that nodes with high heat loads require cooler fluid to dissipate the heat. When these nodes are closer to the pump, the fluid reaching them is relatively cooler. However, if these nodes are located far away from the pump, the fluid reaching them is already hot as it has absorbed heat from other nodes. Consequently, the objective function value (in this case, thermal endurance) decreases. Such studies provide valuable insights for engineering purposes, enabling us to extract knowledge from optimization data. In Refs. [40,41] we have expanded upon this idea to extract human-interpretable knowledge that is useful for practicing engineers.

Figure 14 presents a similar study, but this time for graphs with 4 CPHXs. In the case where we have two nodes with the same heat load (1), denoted as b_1 in the first branch and b_2 in the second branch of case M, we observe that the multi-split case (M) achieves a higher (better) objective function value compared to the single-split cases. The optimal solution is found in the multi-split case (M), where the node with the maximum load is connected to the pump and then divides into two branches. Among the single-split cases, the best solution is achieved when all nodes are arranged in series and the nodes with higher heat loads are positioned closer to the tank (S_1). On the other hand, if the order of heat loads is reversed (S_3), the objective value worsens. This is because in this scenario, the node with the highest heat load receives fluid that is already hot since it has absorbed heat from other nodes, resulting in a decrease in thermal endurance.



(*a*) Population ranking

(*b*) Flow rate and Wall temperature of the cases shown in Fig. 14(a)

Fig. 14 Comparison of multi- and single-split cases with 4 nodes, having $dist$ = [12,4,1,1] kW. Number of cases: n_S = 73, n_M = 88. In Fig. 14(b), temperature signals are depicted with solid lines. Temperature and flow rate data corresponding to the same node are indicated by identical colors and symbols.

The flow rate and wall temperature characteristics of three graphs, namely M, S_1, and S_2, are investigated in Figure 14(b). One notable difference between M and S_1 is that, in S_1, the flow rate in all nodes is the same as the pump flow rate, whereas in M, it is not. As depicted in the right vertical axis of Fig. 14(b), the flow rate in node 1_{b2} of M is nearly zero initially, increases, and then decreases again. An interesting observation is the change in the wall temperature of node 1_{b2}. As shown in the left vertical axis of Fig. 14(b), the wall temperature of this node increases as the flow rate in that branch increases. This phenomenon can be explained by the fact that node 1_{b2} has a lower heat load compared to other nodes, and the coolant fluid entering this node is already warmer than its wall. Consequently, the flow rate in this branch increases to allow the coolant fluid to dissipate some of its heat to the CPHX of this node. Furthermore, as the flow rate in 1_{b2} surpasses zero, the coolant flow gets cooler and the rate of change in the wall temperature of node 12 decreases, resulting in a delayed approach to the upper temperature bound (45 degrees Celsius) and thus increasing thermal endurance. As a result, an optimal coordination of coolant fluid flow rate is achieved, facilitating optimal heat transfer between different nodes and the coolant, ultimately leading to the best objective function value.

A similar situation was observed in the case of 3 nodes, for the node with heat load 1 kW, (see case 1M in Fig. 13) near the 40.0 second mark. At this point, the flow rate of the node with a 1 kW heat load increases and helps dissipate heat from the fluid flow, resulting in cooler fluid. This, in turn, allowed the other wall nodes to reach the upper bound at a later time, ultimately improving thermal endurance. Generating such strategies through human intuition gained from human experience alone can be challenging. However, optimization studies such as these assist engineers in discovering generalizable strategies to solve specific types of problems. By leveraging these studies, engineers can avoid the need for trial and error, thereby significantly reducing the time and resources required to achieve the desired outcome. Optimization studies may also be used to generate large design data sets from which insights can be derived, as opposed to relying upon results from implemented designs alone [40,41].

6.2 Multi-split configurations with 6 CPHX-nodes and investigation of the inner-loop results. In this section we define the 3D locations of the CPHXs, and then, based on spatial location, the junction nodes are produced. For this system, the location coordinates of the CPHXs are defined as: $[2, 0, 0]$, $[2, 1, 0]$, $[3, 1, 0]$, $[12, 12, 0]$, $[15, 10, 0]$, $[13, 13, 0]$, where $[x, y, z]$ are the coordinate locations of CPHXs in 3D space measured in meters. Two sets of heat loads are considered: case1 = $[5, 5, 5, 5, 5, 5]$ kW, case2 = $[5, 7, 6, 4, 5, 5]$ kW. In this problem, the locations of junctions and their heat-loads are fixed but all other nodes will vary. Therefore, we have nine different configurations in total, shown in Fig. 15. The optimal configuration for each of these cases depends on the disturbance values. For example, for the first scenario, configuration 0 produces the best result (highest thermal endurance). For the second scenario, however, the best result is obtained from configuration 6.

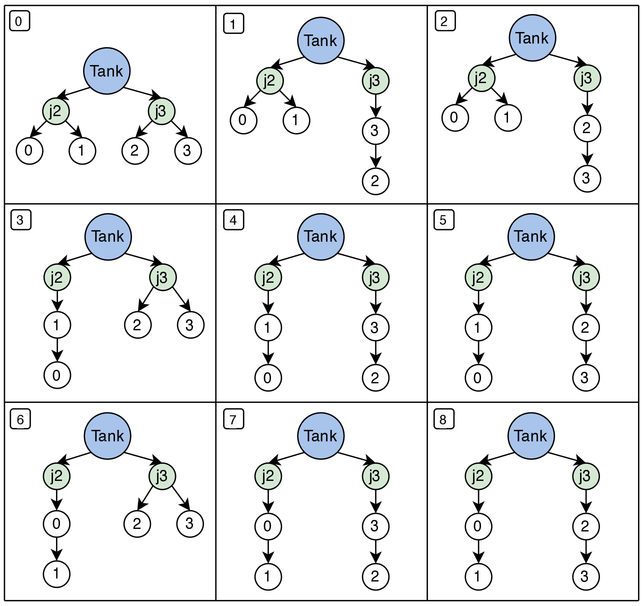


Fig. 15 Different architectures produced by the code with CPHXs 3D spatial locations defined as: $[2, 0, 0]$, $[2, 1, 0]$, $[3, 1, 0]$, $[12, 12, 0]$, $[15, 10, 0]$, $[13, 13, 0]$ m.

To understand what happens to each control and state signal when solving the OLOC problem for the *case2*, consider Fig. 16. In this visualization, the fluid nodes are represented by circles, and the corresponding wall temperatures are indicated in the plot legend using the notation w. For instance, the wall node linked to the fluid node 0 is denoted as $w - 0$. All nodes are constrained to a maximum temperature of 45 degrees Celsius; if temperature constraints are not active, it is possible to increase thermal endurance. Optimal thermal endurance often occurs when all wall nodes reach the upper-bound at the same time; if one of the nodes reaches the upper-bound sooner than the others, the thermal endurance could be increased (the capacity of the system has not been fully utilized).

In Fig. 16 the OLOC results are shown for case 6 (best), case 4 (worst), and case 3 (intermediate). The optimal configuration (case 6) has the maximum heat-load in series with nodes 0 and 1. Configuration 0 is the next best solution among the 9 cases. Figure 16 also illustrates the corresponding optimal control and state signals. For the best solution, we observe that all the wall nodes reach the upper-bound temperature at the same time, enabling this configuration to have the best thermal endurance. By comparing configurations 6 and 3 we notice that the only difference between these architectures is the order of nodes 0 and 1. In configuration 3, in the first branch where all nodes are in series, node 0 is the last node and has a larger disturbances than node 1. When the fluid reaches this node, it is already warmer than the fluid in node 1; therefore, the temperature for that node reaches the upper-bound temperature sooner than node 1. On the other hand, in the optimal configuration (6), node 0 is closer to the tank and receives cooler fluid than does node 1.

6.3 A multi-split configuration with 17 CPHX-nodes. In the previous section, small-scale graphs were studied to facilitate discussion of problem details and results. However, the code created for these studies has the potential of automatically generating and solving graphs of arbitrarily-large size (limited, of course, by computational resources). To demonstrate the ability to extend this modeling and design framework to a problem of larger scale, we considered a graph with 17 heat-exchanger nodes: 14 CPHXs, and 3 junction CPHXs which were added using the spatial locations of the CPHXs. Figure 17 displays the generated graph. Some CPHXs are connected directly to the tank, whereas others are connected to the created junctions. All C_p *Fluid* nodes have the same heat load of 4 kW, and the heat loads of junctions J_2, J_3, and J_4 are 3 kW, 4 kW, and 5 kW, respectively.

Figure 17 shows the wall temperature (solid red line), fluid temperature (dashed-dot green line), and flow rate (dotted-blue line) trajectories for each node. Additionally, the dashed gray line represents the range of these variables, with temperature ranging from 15 to 45 degrees and flow rate ranging from 0 to 0.4 kg/s. We can see that the wall temperature of all nodes at the end of each branch (nodes: 0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13) reaches the upper bound simultaneously. Similarly, the wall temperature of J_4, which has the highest heat load among all three junctions, also reaches the upper bound. While this might not always hold true (consider cases where the heat loads of the junctions are negligible), it demonstrates that compared to the other two junctions, this particular junction has a more significant impact on the objective function value of this graph. Moreover, as we see, the flow rate of J_4 is higher than that of J_2 and J_3. Additionally, since all nodes have the same heat load, nodes connected to the same junction (or to the tank) exhibit almost identical flow rate signals. As an example, the flow rate values of all nodes within the following three categories are identical: $\{6, 7, 8, 0, 1, 2\}$, $\{9, 10, 11\}$, and $\{12, 13\}$. The thermal endurance of this system under the given heat load is 48.68 seconds.

7 Conclusion

This article presents the optimal flow control of fluid-based thermal management systems with multi-split configurations. Graph-based modeling is used to generate different configurations and to automatically construct their dynamic equations. In addition, the spatial information of the heat-exchangers is used to define junctions and determine the level of system architecture complexity. The presented generative algorithm can be used for various applications in the domain of configuration design. Open-loop optimal control is used to quantify the best possible performance of each configuration. This allows a fair comparison of configurations and identification of the best overall system architecture.

The results include 3 parts. Part 1 compares the results of single and multi-split systems composed of 3 or 4 CPHXs and discusses the results in detail. Part 2 presents and compares the results for multi-split systems composed of 6 CPHXs and provides a comprehensive analysis of the inner-loop OLOC optimization. Finally, Part 3 studies a larger scale multi-split system made of 17 CPHXs; an analysis of the results obtained is presented. The results show that multi-split configurations result in a better configuration design in some cases. For many optimal cases, all the wall nodes positioned at the end of CPHX system branches reach the upper-bound temperature simultaneously. Additionally, we observed that the optimal configuration depends on the disturbance values. Finally, the results show the possibility of modeling and analyzing large system made of various CPHXs.

In our future work, prior research performed by some of the authors on holistic design for 3D spatial packaging and routing of interconnected systems [42–46] will be employed to capture the spatial aspects of the multi-split configurations via optimal placement of junction nodes and cooling circuit components such as the CPHXs, pumps, valves, and the tank with simultaneous optimization of 3D lengths of the branches (or pipe segments) while satisfying volume and multi-physics constraints. This can lead to a more complete definition of the real-world multi-split configuration fluid-based thermal system problem. Furthermore, in the next phase of these studies, we plan to use a population-based algorithm and compare its results with an enumeration-based framework. In addition, following the study conducted in this article, we have utilized machine learning techniques to extract generalizable design knowledge interpretable by humans from the design optimization data [40,41]. This approach enables us to reveal valuable insights and understand the underlying patterns and relationships in a more systematic and interpretable manner. By leveraging machine learning methods, we can enhance our understanding of patterns in optimal thermal system design and gain actionable knowledge that can be applied to further improve the design and performance of such systems. Some other future work items include using more sophisticated hydraulic and thermal models, performing 3D spatial optimization of the pipe network with simultaneous energy loss minimization, exploration of alternative objective functions, additional failure mode constraints, lifecycle considerations, and application of the proposed design framework to

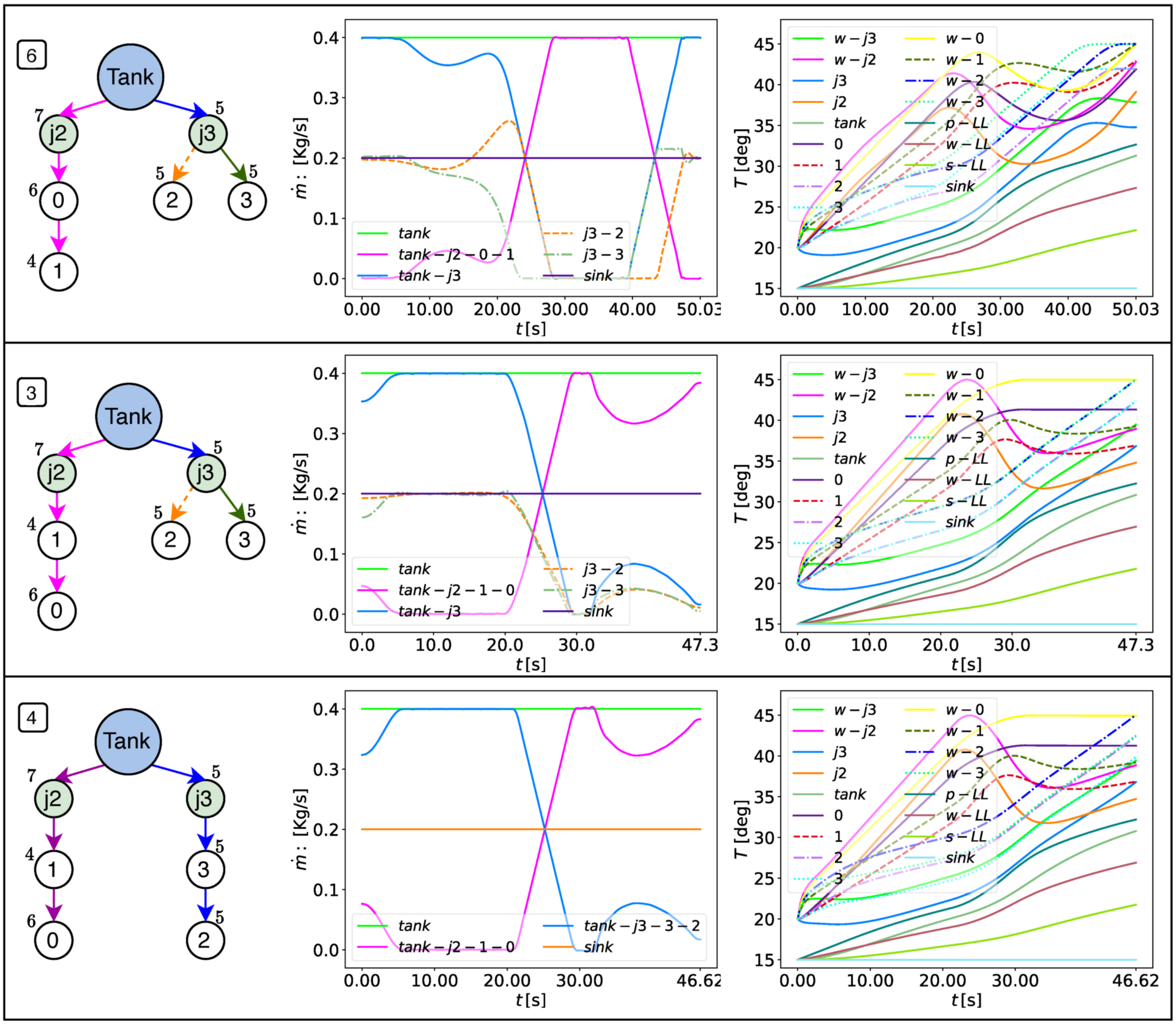


Fig. 16 Investigation of the OLOC signals for the three cases presented in Fig. 15, with heat loads of [5, 7, 6, 4, 5, 5] kW. The multi-split configuration reveals that case 6 exhibits the maximum thermal endurance, case 4 the minimum, and case 3 a value in the mid-range. In the left part of this figure, three configurations are displayed with labeled fluid nodes. The text inside each circle indicates the corresponding fluid node's label. Except for the Tank, each fluid node is paired with a corresponding wall (w) node. These wall nodes are not shown, but their respective heat loads are displayed beside each node in the leftmost images (in kW) and their temperatures are shown in the rightmost image, labeled as $w-i$, where i is the fluid node's label. Additionally, in the rightmost image, the nodes corresponding to the LLHX are labeled as $p-LL$, $w-LL$, and $s-LL$, representing the primary, secondary, and wall of the LLHX, respectively. The label for the sink node is shown as $sink$. In the middle figure, all flow rates of fluid nodes are shown, and on the right, all temperatures of fluid and wall nodes are depicted.

larger industry-relevant applications.

Acknowledgment

This material is based upon work supported by the National Science Foundation Engineering Research Center (NSF ERC) for Power Optimization of Electro-Thermal Systems (POETS) with cooperative agreement EEC-1449548.

Nomenclature

LLHX = Liquid-to-Liquid Heat Exchanger
CPHX = Cold Plate Heat Exchanger
CCD = Control Co-Design
OLOC = Optimal Open Loop Control
x, y, z = Three-dimensional coordinates of components
N_v = Number of all vertices
N_t = Number of sink vertices
N_e = Number of regular edges
N_s = Number of source edges
T = Temperature of regular node
T^t = Temperature of sink node
i, j = Index of graphs, vertices, etc
P = Rate of transfer of thermal energy in regular edge
P^s = Rate of transfer of thermal energy in source edge
$\dot{m}$ = Mass flow rate
C = Thermal capacitance
P_{ij}^{in} = The transfer of thermal energy from node j to node i
P_{ij}^{out} = The transfer of thermal energy from node i to node j
$i-j$ = The Edge that connects node i and j

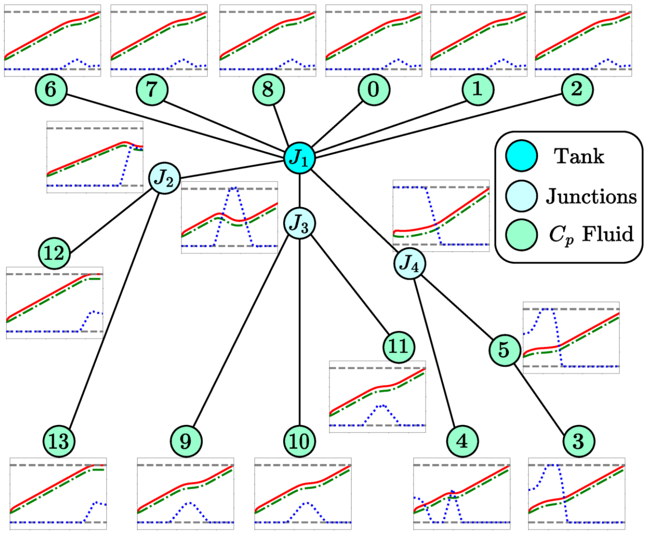


Fig. 17 A graph with 18 nodes: 1 tank, 14 CPHXs, and 3 junction CPHXs. Individual plots, adjacent to each node, display the wall temperature (solid red line), fluid temperature (dashed-dot green line), and flow rate (dotted blue line). The plots show temperature (left vertical axis) in °C and flow rate in Kg/s (right vertical axis) vs. time in S (horizontal axis). The dashed gray lines indicates the temperature values of 15°C and 45°C, and flow rates of 0 and 0.4 Kg/s.

$\mathbf{M}$ = Structural mapping from powers to temperatures
$m_{i,j}$ = Element i, j of $\mathbf{M}$
$\overline{\mathbf{M}}$ = Structural mapping from regular power flows P to regular temperatures T
$\underline{\mathbf{M}}$ = Structural mapping from regular power flows P to vertex states T^t
$\mathbf{D}$ = Structural mapping from external sources to regular temperatures T
$d_{i,e}$ = Element i and e of $\mathbf{D}$
$\mathbf{C}$ = Diagonal matrix of capacitances.
V = Fluid volume
c_p = Specific heat capacitance
ρ = Fluid density
M_w = Wall mass
$c_{p,w,i}$ = The specific heat capacitance of the wall material
A = Convective surface area
h_{ij} = Heat transfer coefficient
a^1, a^2 = Coefficient for convection power flow
b^1, b^2 = Coefficient for advection power flow
p = primary
s = Secondary
w = Wall
$\mathbf{M}^a$, $\mathbf{M}^b$ = Structural mapping from temperatures to thermal energy flow
m^a, m^b = Element of $\mathbf{M}^a$ and $\mathbf{M}^b$
m_p = Pump flow rate
m_t = Tank flow rate
$\dot{m}_{\text{dp}}$ = Flow rate in dependent branches
$\dot{m}_{\text{indp}}$ = Flow rate in independent branches
n_{dp} = Number of dependent branches
n_{indp} = Number of independent branches
$\mathbf{Z}$ = Structural mapping from flow rates to all branches
$\mathbf{A}, \mathbf{B}_1, \mathbf{B}_2$ = Matrices to define state equations
ξ = States
N = Number of CPHXs
$G(N)$ = Number of Configurations
J = Number of junctions
$\boldsymbol{u}(t)$ = Control signal
λ = Penalty term in OLOC objective
NLP = Nonlinear program
kW = Kilo Watts
t_f = Final time
n_S = Number of single split cases
n_M = Number of multi split cases

References

[1] Rootzén, J., Wiertzema, H., Brolin, M., and Fahnestock, J., 2020, “Electrify everything! Challenges and opportunities associated with increased electrification of industrial processes,” .

[2] El-Refaie, A. M., 2016, “Growing role of electrical machines and drives in electrification,” 2016 XXII International Conference on Electrical Machines (ICEM), pp. 364–370.

[3] Nadel, S. M., 2019, “Electrification in the Transportation, Buildings, and Industrial Sectors: a Review of Opportunities, Barriers, and Policies,” Current Sustainable/Renewable Energy Reports.

[4] Mathew, J. and Krishnan, S., 2022, “A review on transient thermal management of electronic devices,” Journal of Electronic Packaging, **144**(1).

[5] Bayat, S., Nejat Pishkenari, H., and Salarieh, H., 2021, “Observation of Stage Position in a 2-Axis Nano-positioner Using Hybrid Kalman Filter,” Scientia Iranica, **28**(5), pp. 2628–2638.

[6] Smoyer, J. L. and Norris, P. M., 2019, “Brief historical perspective in thermal management and the shift toward management at the nanoscale,” Heat Transfer Engineering, **40**(3-4), pp. 269–282.

[7] Wong, H.-S. P., Akarvardar, K., Antoniadis, D., Bokor, J., Hu, C., King-Liu, T.-J., Mitra, S., Plummer, J. D., and Salahuddin, S., 2020, “A density metric for semiconductor technology [point of view],” Proceedings of the IEEE, **108**(4), pp. 478–482.

[8] Bayat, S., Pishkenari, H. N., and Salarieh, H., 2019, “Observer design for a nano-positioning system using neural, fuzzy and ANFIS networks,” Mechatronics, **59**, pp. 10–24.

[9] Ye, P., Ernst, T., and Khare, M. V., 2019, “The last silicon transistor: Nanosheet devices could be the final evolutionary step for Moore’s Law,” IEEE spectrum, **56**(8), pp. 30–35.

[10] He, Z., Yan, Y., and Zhang, Z., 2020, “Thermal management and temperature uniformity enhancement of electronic devices by micro heat sinks: A review,” Energy, p. 119223.

[11] Almubarak, A. A., 2017, “The Effects of Heat on Electronic Components,” International Journal of Engineering Research and Applications, **07**, pp. 52–57.

[12] Mathew, J. and Krishnan, S., 2021, “A Review On Transient Thermal Management of Electronic Devices,” Journal of Electronic Packaging.

[13] Feng, C.-P., Chen, L.-B., Tian, G.-L., Wan, S.-S., Bai, L., Bao, R., Liu, Z., Yang, M., and Yang, W., 2019, “Multifunctional Thermal Management Materials with Excellent Heat Dissipation and Generation Capability for Future Electronics.” ACS applied materials & interfaces, **11 20**, pp. 18739–18745.

[14] Peddada, S. R., Herber, D. R., Pangborn, H. C., Alleyne, A. G., and Allison, J. T., 2019, “Optimal flow control and single split architecture exploration for fluid-based thermal management,” Journal of Mechanical Design, **141**(8).

[15] Panjeshahi, M., Ataei, A., Gharaie, M., and Parand, R., 2009, “Optimum design of cooling water systems for energy and water conservation,” Chemical Engineering Research and Design, **87**(2), pp. 200–209.

[16] Muller, C. J. and Craig, I. K., 2016, “Energy reduction for a dual circuit cooling water system using advanced regulatory control,” Applied energy, **171**, pp. 287–295.

[17] Ling, L., Zhang, Q., Yu, Y., and Shuguang, L., 2016, “Experimental investigation on the thermal performance of water cooled multi-split heat pipe system (MSHPS) for space cooling in modular data centers,” Applied Thermal Engineering, **107**, pp. 591–601.

[18] Pangborn, H. C., Koeln, J. P., Williams, M. A., and Alleyne, A. G., 2018, “Experimental validation of graph-based hierarchical control for thermal management,” Journal of Dynamic Systems, Measurement, and Control, **140**(10), p. 101016.

[19] Pangborn, H. C., Koeln, J. P., and Alleyne, A. G., 2018, “Passivity and decentralized MPC of switched graph-based power flow systems,” *2018 Annual American Control Conference (ACC)*, IEEE, pp. 198–203.

[20] Koeln, J. P., Williams, M. A., Pangborn, H. C., and Alleyne, A. G., 2016, “Experimental validation of graph-based modeling for thermal fluid power flow systems,” *Dynamic Systems and Control Conference*, Vol. 50701, American Society of Mechanical Engineers, p. V002T21A008.

[21] Preisig, H. A., 2009, “A graph-theory-based approach to the analysis of large-scale plants,” Computers & Chemical Engineering, **33**(3), pp. 598–604.

[22] Bennett, G., Elwell, C. A., Lowe, R., and Oreszczyn, T., 2016, “The Importance of Heating System Transient Response in Domestic Energy Labelling,” Buildings, **6**, p. 29.

[23] Allison, J. T. and Herber, D. R., 2014, “Multidisciplinary Design Optimization of Dynamic Engineering Systems,” AIAA Journal, **52**(4), pp. 691–710.

[24] Allison, J. T., Guo, T., and Han, Z., 2014, “Co-design of an Active Suspension Using Simultaneous Dynamic Optimization,” Journal of Mechanical Design, **136**(8).

[25] Garcia-Sanz, M., 2019, “Control Co-Design: An Engineering Game Changer,” Advanced Control for Applications: Engineering and Industrial Systems, **1**(1), p. e18.

[26] Bayat, S. and Allison, J. T., 2023, “Control Co-Design with varying available information applied to vehicle suspensions,” *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 87301, American Society of Mechanical Engineers, p. V03AT03A002.

[27] Herber, D. R., Guo, T., and Allison, J. T., 2017, “Enumeration of Architectures with Perfect Matchings,” Journal of Mechanical Design, **139**(5).

[28] Herber, D. R. and Allison, J. T., 2019, “A Problem Class with Combined Architecture, Plant, and Control Design Applied to Vehicle Suspensions,” Journal of Mechanical Design, **141**(10).

[29] Peddada, S. R. T., 2023, "Automated interference-free layout generation methods for 2d interconnected engineering systems," .

[30] Allison, J. T., Cardin, M.-A., McComb, C., Ren, M. Y., Selva, D., Tucker, C., Witherell, P., and Zhao, Y. F., 2022, "Artificial Intelligence and Engineering Design," Journal of Mechanical Design, **144**(2).

[31] Guo, T., Herber, D., and Allison, J. T., 2019, "Circuit Synthesis Using Generative Adversarial Networks (GANs)," *AIAA Scitech 2019 Forum*, p. 2350.

[32] Sloane, N. J. et al., 2003, "The on-line encyclopedia of integer sequences," .

[33] Athans, M. and Falb, P. L., 2007, *Optimal control: an introduction to the theory and its applications*, Courier Corporation.

[34] Patterson, M. A. and Rao, A. V., 2014, "GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming," ACM Transactions on Mathematical Software (TOMS), **41**(1), pp. 1–37.

[35] Bayat, S. and Allison, J. T., 2023, "SS-MPC: A user-friendly software based on single shooting optimization to solve Model Predictive Control problems," Software Impacts, **17**, p. 100566.

[36] Bayat, S. and Allison, J. T., 2023, "LGR-MPC: A user-friendly software based on Legendre-Gauss-Radau pseudo spectral method for solving Model Predictive Control problems," arXiv preprint arXiv:2310.15960.

[37] Gill, P. E., Murray, W., and Saunders, M. A., 2005, "SNOPT: An SQP algorithm for large-scale constrained optimization," SIAM review, **47**(1), pp. 99–131.

[38] Biegler, L. T. and Zavala, V. M., 2009, "Large-scale nonlinear programming using IPOPT: An integrating framework for enterprise-wide dynamic optimization," Computers & Chemical Engineering, **33**(3), pp. 575–582.

[39] Falck, R., Gray, J. S., Ponnapalli, K., and Wright, T., 2021, "dymos: A Python package for optimal control of multidisciplinary systems," Journal of Open Source Software, **6**(59), p. 2809.

[40] Bayat, S., Shahmansouri, N., Peddada, S. R., Tessier, A., Butscher, A., and Allison, J. T., 2023, "Advancing Fluid-Based Thermal Management Systems Design: Leveraging Graph Neural Networks for Graph Regression and Efficient Enumeration Reduction," arXiv preprint arXiv:2311.14874.

[41] Bayat, S., Shahmansouri, N., Peddada, S. R., Tessier, A., Butscher, A., and Allison, J. T., 2023, "Extracting Design Knowledge from Optimization Data: Enhancing Engineering Design in Fluid Based Thermal Management Systems," arXiv preprint arXiv:2310.16324.

[42] Peddada, S. R. T., Rodriguez, S. B., James, K. A., and Allison, J. T., 2020, "Automated Layout Generation Methods for 2D Spatial Packing," *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. Volume 11B: 46th Design Automation Conference (DAC), doi: 10.1115/DETC2020-22627.

[43] Peddada, S. R. T., James, K. A., and Allison, J. T., 2020, "A Novel Two-Stage Design Framework for Two-Dimensional Spatial Packing of Interconnected Components," Journal of Mechanical Design, **143**(3), 031706.

[44] Peddada, S., Zeidner, L., Ilies, H. T., James, K., and Allison, J. T., 2022, "Toward Holistic Design of Spatial Packaging of Interconnected Systems with Physical Interactions (SPI2)," J. Mech. Des, pp. 1–26, https://doi.org/10.1115/1.4055055.

[45] Peddada, S. R. T., James, K. A., and Allison, J. T., 2020, "A Novel Two-Stage Design Framework for 2D Spatial Packing of Interconnected Components," *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. Volume 11B: 46th Design Automation Conference (DAC), doi: 10.1115/DETC2020-22695, V11BT11A032.

[46] Peddada, S. R. T., Dunfield, N. M., Zeidner, L. E., James, K. A., and Allison, J. T., 2021, "Systematic Enumeration and Identification of Unique Spatial Topologies of 3D Systems Using Spatial Graph Representations," *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. Volume 3A: 47th Design Automation Conference (DAC), doi: 10.1115/DETC2021-66900, V03AT03A042.