Published in *Journal of Mechanical Design*, Copyright (c) 2019 by ASME
The final publication is available at ASME Digital Collection via
https://doi.org/10.1115/1.4043203

OPTIMAL FLOW CONTROL AND SINGLE SPLIT ARCHITECTURE EXPLORATION FOR FLUID-BASED THERMAL MANAGEMENT*

Satya R. T. Peddada[1], **Daniel R. Herber**[1], **Herschel C. Pangborn**[2], **Andrew G. Alleyne**[2], **James T. Allison**[1]
[1]Department of Industrial and Enterprise Systems Engineering
[2]Department of Mechanical Science and Engineering
University of Illinois at Urbana-Champaign, Urbana, IL 61801
Email: {speddad2,herber1,pangbor2,alleyne,jtalliso}@illinois.edu

Abstract

High-performance cooling is often necessary for thermal management of high power density systems. However, human intuition and experience may not be adequate to identify optimal thermal management designs as systems increase in size and complexity. This article presents an architecture exploration framework for a class of single-phase cooling systems. This class is specified as architectures with multiple cold plates in series or parallel and a single fluid split and junction. Candidate architectures are represented using labeled rooted tree graphs. Dynamic models are automatically generated from these trees using a graph-based thermal modeling framework. Optimal performance is determined by solving an appropriate fluid flow distribution problem, handling temperature constraints in the presence of exogenous heat loads. Rigorous case studies are performed in simulation, with components subject to heterogeneous heat loads and temperature constraints. Results include optimization of thermal endurance for an enumerated set of 4,051 architectures. The framework is also applied to identify cooling system architectures capable of steady-state operation under a given loading.

1 INTRODUCTION

The 21st century is seeing a continuing trend toward greater electrification of systems in industrial, transportation, agricultural, and consumer applications [2–4]. Such systems are often characterized by high power-to-weight and/or power-to-volume ratios for electrified components, such as motors (with densities 2-3 kW/kg), power converters (with densities 120-200 W/in^3), battery systems, and other electronics. These components are used in a wide range of applications, including hybrid electric vehicles, more electric aircraft, data centers, electric ships, etc. Inefficiencies in these high-power electrical systems result in the generation of large quantities of waste thermal energy. In turn, their performance, efficiency, safety, and life cycle can depend strongly on their temperature. Therefore, thermal management systems are tasked with maintaining operating temperatures within specified limits.

While waste thermal energy must ultimately be rejected to the environment, it is often first transferred within thermal management systems from heat-generating components to a liquid coolant circulating through a network of pipes, referred to as the "cooling system architecture" in this article. These

*This work was presented in part at the 44th ASME Design Automation Conference, Quebec City, Canada, August 26–29, 2018 [1].

architectures must not only be energy efficient and reliable, but also help reduce capital cost in terms of size, complexity, and fabrication cost. Mobile systems have additional considerations, such as size and weight limitations, and possibly more limited options for cooling mechanisms.

Controlling fluid-based thermal management systems requires determining actuator inputs to all valves and pumps in the fluid network. The control design must balance maintaining the temperature constraints of individual components with actuator considerations, such as rate limits and pumping efficiency. Pertinent performance metrics for both architecture and control design can vary widely. For example, aircraft thermal management systems can be characterized by their thermal endurance, defined as the duration of operation before any temperature constraint is violated, to ensure safe operation and maximum range [5]. Other applications, such as hybrid-electric vehicles [6] or server farms [7] may have a different set of requirements. As new cooling system applications and requirements emerge, engineers must learn how best to meet new needs, sometimes without the benefit of design heritage or the associated expert knowledge for particular systems.

Engineers designing cooling system architectures must select from among many possible designs. A conventional design process would often begin with a pure series or pure parallel configuration, with possible modifications selected by intuition and analyzed in terms of steady-state performance. However, as electro-thermal systems become increasingly transient in nature, achieving efficient and high-performance thermal management requires supplanting steady-state methods with approaches that consider transient behavior [8]. In this article, this is achieved by an approach to evaluating candidate architectures that directly considers dynamic behavior and can accommodate heterogeneous temperature constraints and time-varying heat loads. Furthermore, while it may be relatively easy to make and evaluate architecture design decisions in simple cases (e.g., a small number of components with uniform component temperature limits and/or constant thermal loads), traditional design strategies fall short as the size and complexity of thermal management systems increases, commensurate with that of the systems that they cool. The resulting system architecture design problems can have a vast design space that is cognitively difficult to navigate, motivating the need for efficient and systematic design methods with the flexibility to explore and assess many candidate configurations. In this article, architecture design optimization is employed to meet this need. Given a predefined set of heat generating components, temperature constraints, and objectives, a framework for identifying the highest-performing candidate within a class of architectures is proposed.

Recent work has illustrated the sub-optimality of conventional thermal management architectures. For example, Ref. [5] demonstrates that a dual fuel tank architecture for an aircraft can achieve a 35% improvement in thermal endurance over a conventional single fuel tank architecture under closed-loop control. In the dual configuration, one tank is used in a standard recirculation loop, while the second tank acts as an auxiliary reservoir. This reduces the thermal capacitance of the recirculation loop as compared to the single tank architecture, allowing the temperature of the fuel feeding the engine to increase faster and therefore remove more thermal energy from the system when that fuel is burned early in flight. The decision to analyze this dual-tank architecture aligns with the concept of minimizing exergy destruction in thermal systems [9]. However, in the absence of exhaustive or other systematic design studies, it is left to engineering intuition to conceive of unconventional architectures such as this. This begs the question of whether additional modifications to conventional configurations for this class of systems could improve performance, efficiency, or robustness.

Many past efforts have focused on improving individual components in a cooling system, but few have addressed overall system design. Examples of the latter include Refs. [10, 11]. In Ref. [10] a design methodology was developed to predict cooling tower performance considering heat load distributions. In Ref. [11] interactions between a heat exchanger network and cooling tower were considered. Cooling system architecture studies typically focus on a single application such as process utilities in petrochemical plants [12] or multi-chip processors [13], as opposed to a general design framework.

Existing architecture studies have often been limited in scope to improve tractability. For example, rearranging components in a given topology (i.e., an unlabeled graph compatible with the given components/labels) restricts problem complexity, but limits broad exploration of new designs [14].

In this article, the architecture exploration is supported by a first-principles approach to modeling fluid-thermal dynamics. Simplifying model assumptions are made that support tractability while retaining important physical effects and applicability to a wide range of thermal management system applications. Many of these assumptions have been validated in previous experimental work [15], and are discussed in detail in later sections where appropriate. The design methodology presented here:

- Uses dynamic graph-based modeling to provide flexibility in assessing a variety of different cooling system types.
- Supports systems with multiple temperature-sensitive components having high heat loads and different maximum allowable temperatures.
- Is applicable to small-scale (up to five heat generating components) and medium-scale (up to eight heat generating components) systems, whereas extension to large-scale systems is a topic of ongoing work. Here we define small and medium-scale architecture design problems to be those that are solvable via efficient enumeration techniques.
- Uses flexible solution methods such as variable-horizon direct optimal control.
- Determines optimal coolant flow distribution across the system, balancing competing objectives.

The contributions of this article are 1) a new method for enumerating system architectures modeled using graphs as labeled rooted trees, and 2) a rigorous method for comparison between candidate architectures. The first contribution builds on recent work in efficient methods for architecture enumeration [16]. The second contribution is enabled by recent advancements in direct optimal control [17, 18].

For each architecture, flow is distributed optimally across the system such that thermal endurance is maximized across components operating under heterogeneous heat loads and temperature constraints. The highest performing architectures are then determined as those with the greatest thermal endurance. As an initial step, here we limit the design scope to a fluid-based thermal management system with a single junction and single split. However, the class of architectures considered in this article can be used as canonical architectures that can be built upon to generate other complex architectures. Hence, future work will expand this to include architectures with greater degrees of freedom in topological design beyond the single junction and single split assumption.

The remainder of this article is organized as follows. Section 2 describes the class of thermal management architectures considered in this article, as well as the graph-based approach used to generate dynamic models of these systems. Section 3 describes how to enumerate architectures as labeled rooted tree graphs for cooling system design. Section 4 explains the variable time horizon dynamic optimization problem formulation. Section 5 presents four case studies performed with different architectures at various sets of operating conditions, heat loads, and individual temperature constraints to demonstrate the efficacy of the design method introduced here. Finally, Section 6 concludes by summarizing the design methodology, outlining guidelines for thermal management system design derived from case study results, and suggesting topics for future work.

2 SYSTEM DESCRIPTION AND MODELING

The class of thermal management system architectures considered in this article is depicted in Fig. 1. The primary purpose of the system is to manage the operating temperature of a number of heat-generating electrical or mechanical components, each mounted to a cold plate heat exchanger (CPHX) through which a coolant flows. This fluid is stored in a tank and driven by a pump through a set of N_f *parallel* flows, each of which includes a variable-aperture valve v. The fluid in each parallel flow passes through a number of CPHXs in *series*, absorbing thermal energy from the CPHX walls. Thermal

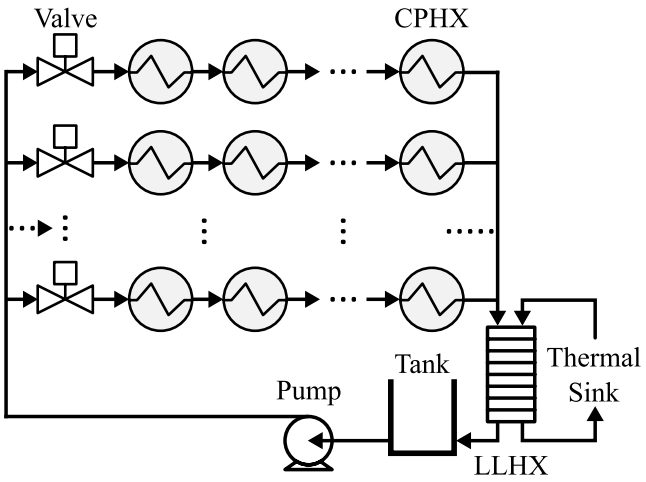


Fig. 1 Illustration of the class of cooling system architectures considered, consisting of cold plate heat exchangers (CPHXs) in series and parallel, and constrained to have a single split/junction in the coolant flow.

energy is transferred from the fluid to a thermal sink (e.g., a secondary loop regulated by a vapor compression cycle system [15]) via a liquid-to-liquid heat exchanger (LLHX). This class of architectures is representative of many single-phase fluid-based thermal management systems, such as those found in aircraft [19, 20], electrified automobiles [21], and server farms [7].

The total number of CPHXs, which is assumed to be fixed for a given design study, is denoted as N_c. We assume that the heat load applied to each CPHX, as well as the inlet temperature and mass flow rate of the thermal sink fluid, are known exogenous disturbances. We also assume that the temperature of each heat-generating component is the same as the temperature of the wall of the CPHX to which it is mounted. The thermal management system is controlled by commanding the rotational speed of the pump and the position of each variable-aperture valve.

2.1 Dynamic Graph-Based Modeling

One approach for the lumped parameter dynamic modeling of power flow systems, including the class of thermal management architectures considered in this article, is to apply conservation equations within a graph-based framework, where vertices represent the capacitive storage of energy and edges represent the paths along which power can flow between vertices [15, 22–25].

This graph-based approach is particularly well-suited to architecture exploration and optimization because system models can be programmatically generated from a library of component models and a structural mapping of their interconnections. This allows graph-theoretic techniques to be leveraged for rapidly generating all possible models within a class of architectures, as well as for both system analysis and control design [26]. The remainder of this subsection presents an overview of the graph-based modeling approach, which is then specified to the class of architectures considered in this article.

In graph-based modeling, the structure of interconnections of the system is described by the oriented graph $G = (\boldsymbol{v}, \boldsymbol{e})$ of order N_v with vertices $\boldsymbol{v} = [v_i], i \in \{1, 2, ..., N_v\}$, and size N_e with edges $\boldsymbol{e} = [e_j], j \in \{1, 2, .., N_e\}$ [27]. As shown in the notional graph example of Fig. 2, each edge e_j is incident to two vertices and indicates directionality from its *tail* vertex v_j^{tail} to its *head* vertex v_j^{head}. For example, in Fig. 2, v_1 is the tail vertex of e_1, while v_3 is the head vertex of e_1. The set of edges directed into vertex v_i is given by $e_i^{head} = \{e_j | v_j^{head} = v_i\}$. For v_3 in Fig. 2, this consists of e_1 and e_2. The set of edges directed out of vertex v_i is given by $e_i^{tail} = \{e_j | v_j^{tail} = v_i\}$. For v_3 in Fig. 2, this consists of e_4.

Each vertex has an associated dynamic state associated with energy storage. For thermal systems,

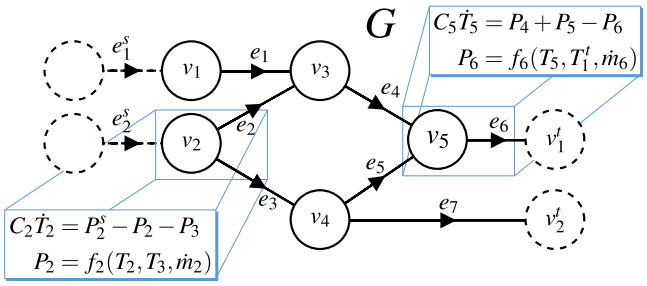


Fig. 2 Notional graph example to demonstrate key features of the modeling approach.

this state is the temperature of a thermal element, T_i. Each edge has an associated quantity P_j describing the rate of transfer of thermal energy (equivalently referred to as thermal power flow in this article) between adjacent vertices. The orientation of each edge indicates the convention assigned to positive power flow, from v_j^{tail} to v_j^{head}. Therefore, the dynamics of each state satisfy the conservation equation:

$$C_i\dot{T}_i = \sum_{\{j|e_j \in e_i^{head}\}} P_j - \sum_{\{j|e_j \in e_i^{tail}\}} P_j \tag{1}$$

where $C_i > 0$ is the thermal capacitance of the vertex. Equation (1) states that the rate of thermal energy storage in the vertex is equal to the total thermal power flow into the vertex minus the total thermal power flow out of the vertex.

For the fluid-based thermal system in this article, the power flow P_j along each edge is a function of the temperature states of the vertices to which it is incident and may also be a function of an associated mass flow rate $\dot{m}_j$ which can be treated as an input to the thermal system model [15,22–24]. The transfer rate along each edge is therefore given generically by:

$$P_j = f_j\left(T_j^{tail}, T_j^{head}, \dot{m}_j\right). \tag{2}$$

Figure 2 includes examples of Eq. (1) and Eq. (2) as applied to several vertices and edges.

In addition to capturing the exchange of energy within the graph, the modeling framework must account for exchange with entities external to the graph. Sources to graph G are modeled by source edges $\boldsymbol{e}^s = [e_j^s]$, $j \in \{1, 2, ..., N_s\}$ with associated power flows $\boldsymbol{P}^s = [P_j^s]$, which are treated as disturbances to the system that may come from neighboring systems or the environment. Therefore, edges belonging to $\boldsymbol{e}^s$ are not counted among the edges $\boldsymbol{e}$ of graph G, and transfer rates in $\boldsymbol{P}^s$ are not counted among the internal transfer rates $\boldsymbol{P}$ of the system.

Sinks of graph G are modeled by sink vertices $\boldsymbol{v}^t = [v_j^t]$, $j \in \{1, 2, ..., N_t\}$ with associated states $\boldsymbol{T}^t = [T_j^t]$. The sink vertices are counted among the vertices $\boldsymbol{v}$ of graph G, but the sink states $\boldsymbol{T}^t$ are not included in the state vector $\boldsymbol{T}$ of the system. Instead, the sink states $\boldsymbol{T}^t$ are treated as disturbances to the system associated with neighboring systems or the environment.

To describe the structure of edge and vertex interconnections of a graph, the incidence matrix $\boldsymbol{M} = [m_{i,j}] \in \mathbb{R}^{N_v \times N_e}$ is defined as:

$$m_{i,j} = \begin{cases} +1 & \text{if } v_i \text{ is the tail of } e_j, \\ -1 & \text{if } v_i \text{ is the head of } e_j, \\ 0 & \text{else.} \end{cases} \tag{3}$$

$\boldsymbol{M}$ can then be partitioned as:

$$\boldsymbol{M} = \begin{bmatrix} \bar{\boldsymbol{M}} \\ \underline{\boldsymbol{M}} \end{bmatrix} \text{ with } \bar{\boldsymbol{M}} \in \mathbb{R}^{(N_v - N_t) \times N_e} \tag{4}$$

where the indexing of edges is assumed to be ordered such that $\bar{\boldsymbol{M}}$ is a structural mapping from power flows $\boldsymbol{P}$ to states $\boldsymbol{T}$, and $\underline{\boldsymbol{M}}$ is a structural mapping from $\boldsymbol{P}$ to sink states $\boldsymbol{T}^t$.

Similarly, the connection of external sources to the system is given by $\boldsymbol{D} = [d_{i,j}] \in \mathbb{R}^{(N_v - N_t) \times N_s}$ where:

$$d_{i,j} = \begin{cases} 1 & \text{if } v_i \text{ is the head of } e^s_j, \\ 0 & \text{else.} \end{cases} \tag{5}$$

Following from the conservation equation for each vertex in Eq. (1) and the above definitions of $\bar{\boldsymbol{M}}$ and $\boldsymbol{D}$, the dynamics of all states in a system are given by:

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

where $\boldsymbol{C} = \text{diag}([C_i])$ is the diagonal matrix of capacitances.

Following from Eq. (2), the vector of all power flows $\boldsymbol{P}$ in a system is given by:

$$\boldsymbol{P} = \boldsymbol{F}(\boldsymbol{T}, \boldsymbol{T}^s, \dot{\boldsymbol{m}}) = [f_j(T_j^{tail}, T_j^{head}, \dot{m}_j)]. \tag{7}$$

2.2 Graph-Based Model for the Class of Architectures

Figure 3 shows the graph corresponding to the class of system architectures considered in this article. To model this class using the graph-based approach of Section 2.1, vertices are assigned to represent the temperature of the fluid in the tank, the temperature of the fluid in each CPHX and the wall of each CPHX, and the temperature of the fluid in each side of the LLHX and the wall of the LLHX. The thermal capacitance associated with states representing a fluid temperature in Eq. (1) is given by $C_i = \rho V_i c_p$, where V is the volume of the stored fluid, c_p is the specific heat capacitance of the fluid, and ρ is the density of the fluid. The thermal capacitance associated with states representing a wall temperature is given by $C_i = M_{w,i} c_{p,w,i}$, where M_w is the mass of the wall and $c_{p,w}$ is the specific heat capacitance of the wall material. All thermal capacitances in this article are assumed to be constant.

In this article, the modeling of two types of thermal power flow is required to capture the exchange of thermal energy between temperature states of the graph. The first is power flow due to convective heat transfer, given by $P_j = h_j A_{s,j}(T_j^{tail} - T_j^{head})$, where A_s is the convective surface area and h is the heat transfer coefficient, assumed to be constant in this article. This type of power flow occurs between the wall and the fluid of each CPHX, and between the wall and the fluid on each side of the LLHX.

The second type of thermal power flow is advection. This corresponds to heat transfer that occurs when fluid flows through the system transporting thermal energy through its bulk motion, and is given by $P_j = \dot{m}_j c_p T_j^{tail}$, where $\dot{m}$ is the mass flow rate of the fluid and c_p is the specific heat capacitance of the fluid, assumed to be constant in this article. When fluid flows in a loop between two thermal elements, as between the secondary side of the LLHX and the thermal sink, the advective power flow in each direction can be combined into a single "bidirectional advection" power flow, given by $P_j = \dot{m}_j c_p (T_j^{tail} - T_j^{head})$.

Based on the above discussion, for every power flow in the graph-based model of the class of architectures, Eq. (2) can be put in the form:

$$P_j = a_{1,j} T_j^{tail} + a_{2,j} T_j^{head} + b_{1,j} \dot{m} T_j^{tail} + b_{2,j} \dot{m} T_j^{head} \tag{8}$$

where the coefficients $a_{i,j}$ and $b_{i,j}$ are constants. Specifically, for convective power flow, $a_{1,j} = h_j A_{s,j}$, $a_{2,j} = -h_j A_{s,j}$, and $b_{1,j} = b_{2,j} = 0$. For advective power flow, $b_{1,j} = \dot{m}_j c_p$ and $a_{1,j} = a_{2,j} = b_{2,j} = 0$. For bidirectional advective power flow, $b_{1,j} = \dot{m}_j c_p$, $b_{2,j} = -\dot{m}_j c_p$, and $a_{1,j} = a_{2,j} = 0$. The source power flows P^s of the graph-based model for the class of system architectures consist of the heat load to each

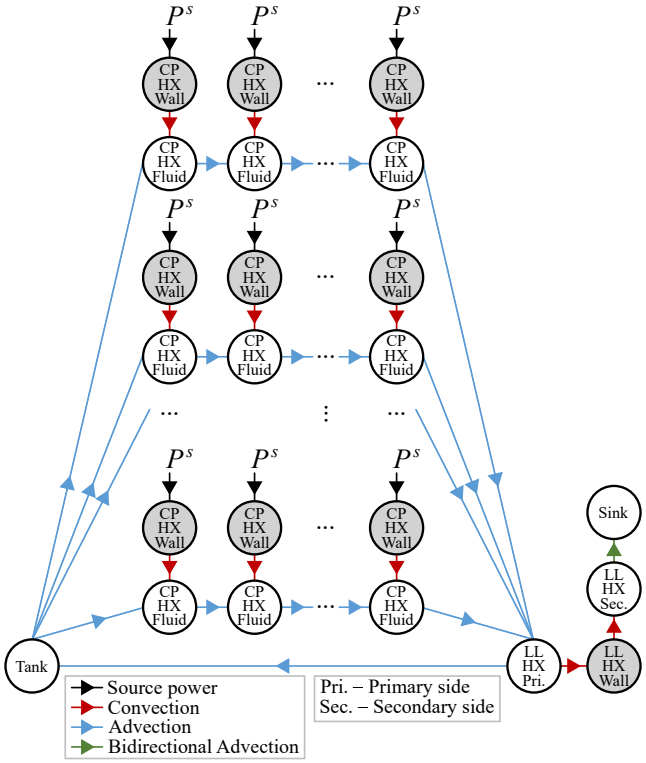


Fig. 3 Graph for the class of thermal management architectures considered in this article. Vertices representing fluid temperatures are colored white, while vertices representing wall temperatures are colored gray.

CPHX from the heat-generating device to which it is mounted. The lone sink state of the system T^t is the temperature of the thermal sink.

In this article, as in Ref. [5], it is assumed that the valves can be controlled such that the flow through the pump can be split with any desired proportioning into each of the parallel flows. It is also assumed that the mass flow rates achievable by the pump are independent of the number of parallel flows, number of CPHXs in each parallel flow, and valve positions. This assumption can be removed by employing the hydraulic graph-based models of Refs. [15, 22], which follow the same general modeling approach as the thermal graph-based models of Section 2.1. In this previous work, it is demonstrated that hydraulic and thermal graph-based models of a system can be interconnected to simultaneously capture both of these domains. However, while this hydraulic modeling falls outside the scope of this study, the roles that pressure drops and pump sizing play in the design of these systems are being addressed in ongoing complementary work.

We also assume that any friction heating or other thermal energy transfer to the fluid by the pump is negligible. However, this assumption can easily be removed when this is not the case, such as for systems requiring high-power pumps. In the graph-based model, this would be done by adding a source power flow representing pump heating incident to a vertex associated with the pump fluid temperature, similar to how source power flows are used to represent exogenous heat loads applied to the CPHX walls.

Experimental validation of the graph-based modeling approach for similar fluid-thermal architectures can be found in [15, 22, 24]. This has shown the modeling approach to be consistently accurate within approximately 2 °C, including under transient behavior. The dynamic model equations can be simulated orders of magnitude faster than real time on modern desktop computers. Therefore, this approach provides both sufficient accuracy and sufficient computational simplicity to support studies in

system and control design. The model parameters used in Section 5 were nominally sized in accordance with the parameters used for experimental validation in the previous work [15, 22, 24]. Several of the key parameters are listed in Table 1. Each CPHX is assumed to be identical, although the heat load applied to each CPHX may be different. The working fluid is assumed to be an equal parts mixture of propylene glycol and water.

2.3 State Equations for the Graph-Based Model

Let $\boldsymbol{M}_a = [\tilde{m}_{i,j}]$ be a weighted incidence matrix defined by:

$$\tilde{m}_{i,j} = \begin{cases} a_{1,j} & \text{if } v_i \text{ is the tail of } e_j, \\ a_{2,j} & \text{if } v_i \text{ is the head of } e_j, \\ 0 & \text{else} \end{cases} \tag{9}$$

where $a_{i,j}$ are the coefficients in Eq. (8). $\boldsymbol{M}_b$ can be defined similarly using the coefficients $b_{i,j}$ in Eq. (8).

Note that there is not a one-to-one relationship between the mass flow rates of the system architecture and the edges of its thermal graph. For example, the mass flow rate through each CPHX of a given parallel flow is the same. Therefore, the set of unique mass flow rates of the system can be mapped to the set of edges of the graph by:

$$\dot{\boldsymbol{m}}_e = \boldsymbol{Z} \begin{bmatrix} \dot{m}_p \\ \dot{\boldsymbol{m}}_f \\ \dot{m}_t \end{bmatrix}, \qquad \boldsymbol{Z} \in \{0,1\}^{N_e \times (2+N_f)} \tag{10}$$

where $\dot{m}_p$ is the mass flow rate rate through the pump, $\dot{\boldsymbol{m}}_f$ is the vector of mass flow rates through each parallel flow, and $\dot{m}_t$ is the mass flow rate of the thermal sink. We note that, by conservation of mass, $\sum_{i=1}^{N_f} \dot{\boldsymbol{m}}_{f,i} = \dot{m}_p$. From Eq. (6) and Eqs. (8)-(10), the state-space equation for the system architecture can be written as:

$$\dot{\boldsymbol{T}} = \boldsymbol{A} \begin{bmatrix} \boldsymbol{T} \\ T^t \end{bmatrix} + \boldsymbol{B}_1 \text{diag}\left(\boldsymbol{Z} \begin{bmatrix} \dot{m}_p \\ \dot{\boldsymbol{m}}_f \\ \dot{m}_t \end{bmatrix} \right) \boldsymbol{B}_2 \begin{bmatrix} \boldsymbol{T} \\ T^t \end{bmatrix} + \boldsymbol{D}\boldsymbol{P}^s \tag{11}$$

where:

$$\boldsymbol{A} = -\boldsymbol{C}^{-1} \bar{\boldsymbol{M}} \bar{\boldsymbol{M}}_a^T \tag{12a}$$

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

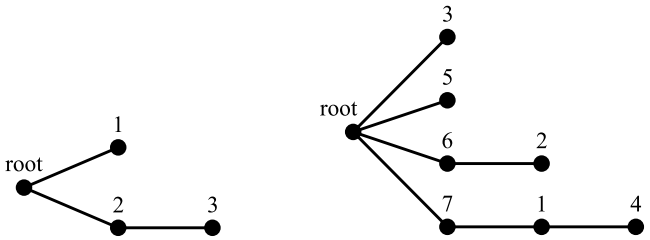
$$\boldsymbol{B}_2 = \bar{\boldsymbol{M}}_b^T \tag{12c}$$

and $\boldsymbol{D}$ is as defined in Eq. (5). For a given number of parallel flows and number of CPHXs in each flow, the corresponding state-space model in the form of Eq. (11) can be programmatically generated.

3 GENERATING NEW ARCHITECTURES WITH LABELED ROOTED TREE GRAPHS

3.1 Representation as Labeled Rooted Tree Graphs

The class of architectures $\mathcal{A}$ can be represented as labeled rooted tree graphs. In graph theory, a tree is a undirected graph in which any two vertices are connected by unique simple path [28, 29]. An equivalent



(a) $1-23$. **(b)** $3-5-62-714$.

Fig. 4 Two labeled rooted tree graphs in the class of graphs of interest.

definition is an undirected graph that is connected and has no cycles. Here, a labeled rooted tree is a tree where the root is labeled with 0 and each other vertex is assigned a unique integer value from 1 to N_c [29]. The architectures in $\mathcal{A}$ are representable by labeled rooted trees because the labels in the tree (other than the root) correspond to specific CPHXs in a predefined list and the fluid is defined to flow away from the root. Finally, to completely define $\mathcal{A}$, we require all the vertices in the tree except the root to have at most one child. Such graphs are called labeled rooted skinny-tree forests [30]. Two example trees are shown in Fig. 4 along with a unique direct representation. This direct representation, specific to single-split architectures, utilizes "–" to indicate a connection to the root (split) and neighboring CPHX numbers that are connected in series, with the leftmost number connected to the root.

The number of unique architectures in $\mathcal{A}$ can be determined by the following formula [30]:

$$|\mathcal{A}(N_c)| = \sum_{k=1}^{N_c} \binom{N_c}{k} \binom{N_c - 1}{k - 1} (N_c - k)! \tag{13}$$

To derive this formula, consider k such that $1 \leq k \leq N_c$, which represents the number of CPHXs connected to the root. Then there are $\binom{N_c}{k}$ ways to choose which CPHXs are attached to the root. For each set of k CPHXs connected to the root, there are $\binom{N_c-1}{k-1}$ compositions of N_c into exactly k parts [31]. Then for each of these compositions, we can permute the ordering of the nonroot CPHXs in $(N_c - k)!$ ways. Finally, we sum over all possible values for k to obtain Eq. (13).

The first few values for the number of unique architectures for increasing N_c are:

$$S = 1, 3, 13, 73, 501, 4051, 37633, 394353, \ldots \tag{14}$$

which is integer sequence number OEIS A000262 representing the number of "sets of lists" [30]. The 13 unique cooling system architectures for $N_c = 3$ are shown in Fig. 5. This information will be used when generating each of the trees.

3.2 Generating the Labeled Rooted Tree Graphs

Here we will utilize the parent pointers representation of a tree graph where the kth child (vertex labeled with k) of vector V has the parent node $V(k)$ [28]. For example, $V = [0\ 3\ 0]$ defines the labeled rooted tree in Fig. 5(*c*). To enumerate all possible labeled rooted trees with all the vertices except the root having at most one child, a recursive algorithm was developed[1].

Some key features of the approach are:

1. Recursively generating all vectors of length N_c with integer entries from 0 to N_c by adding one child-parent entry at a time.

[1]The MATLAB code for generating labeled rooted trees is available in Ref. [32].

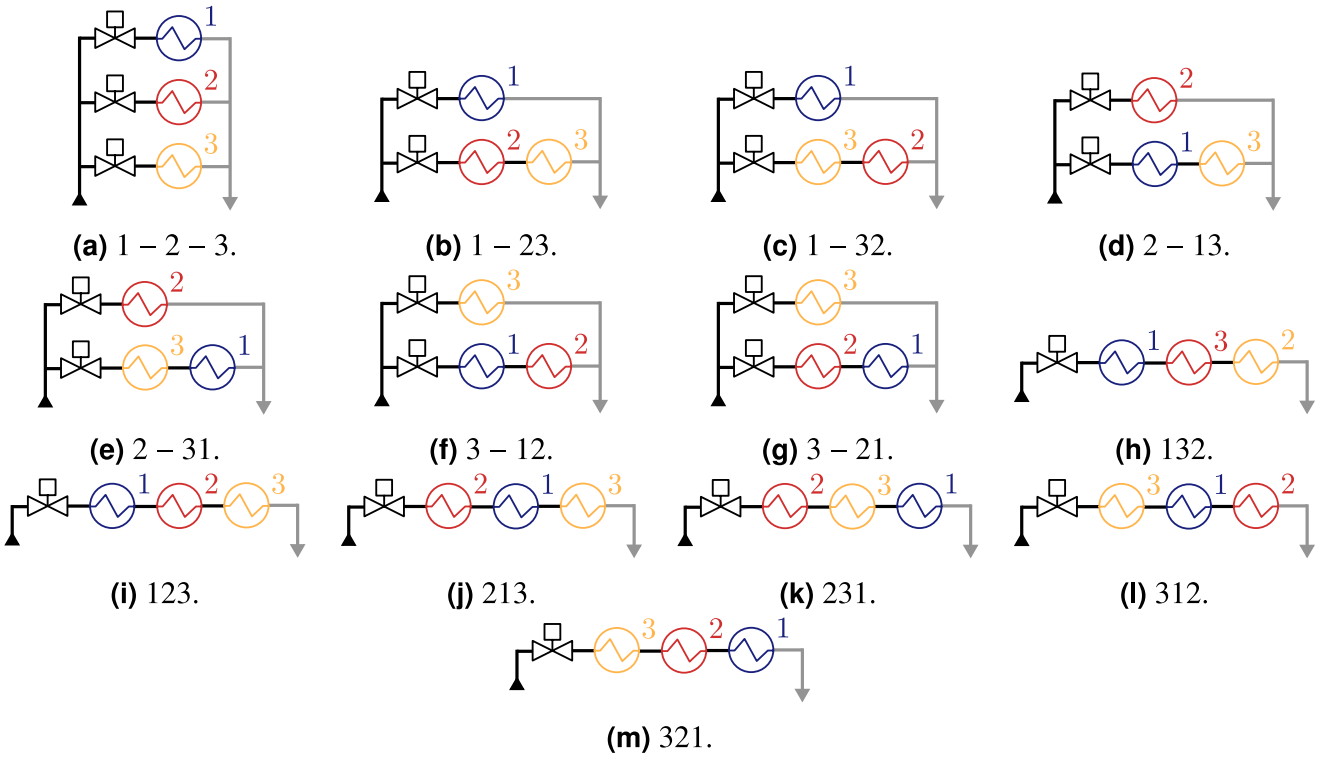


Fig. 5 The 13 unique architectures when $N_c = 3$.

2. Keeping a list of potential parents and removing vertices that already have a child from this list during the recursion. The root node is always a potential parent. This ensures the single-junction structure of the trees.
3. Removing any graphs that have a cycle added during the recursion (since they are not trees).

Due to the special structure of the parent pointers representation, all of the generated trees are unique. We can readily generate the listing of all the trees up to $N_c = 10$, where we start to have data storage issues due to the combinatorial nature of the problem. The computational costs for $N_c = [4, 5, 6, 7, 8]$ are $[0.00050, 0.0035, 0.030, 0.31, 3.6]$ seconds, respectively, using the workstation described in Section 5. These listings will serve as a basis for evaluating all potential cooling system architectures for a given problem.

4 OPTIMAL FLOW CONTROL PROBLEM

To study the thermal performance of the candidate architectures, a dynamic optimization problem is posed. The goal of this optimization problem is to maximize the thermal endurance of a candidate architecture, defined as the length of time before any temperature state violates its upper bound under a given heat load profile. In practice, to continue operation past this time, the system must risk failure or degradation due to operating beyond its thermal limits, or the devices generating thermal energy must be throttled to reduce the heat loads applied to the cooling system. Increasing thermal endurance therefore often facilitates improved life cycle, performance, and/or capability of the systems being cooled. In many cases, heat load profiles may resemble square waves, pulsing in accordance with the duty cycle of electrical or mechanical devices. Thermal endurance under a constant heat load is then representative of the maximum pulse duration that can be tolerated from a given initial temperature condition.

The variable time optimal control problem provides an upper limit on length of the time horizon t_{end}

such that the constraints of the problem are still satisfied:

$$\min_{\ddot{\boldsymbol{m}}_f, t_{\text{end}}} \quad -t_{\text{end}} \tag{15}$$

where $\ddot{\boldsymbol{m}}_f$ are the open-loop control variables used to regulate the rate of change of the valve flow rates.

The dynamics for this problem are then a combination of Eq. (11) and additional states to capture $\ddot{\boldsymbol{m}}_f$:

$$\dot{\boldsymbol{\xi}} = \begin{bmatrix} \dot{\boldsymbol{T}} \\ \ddot{\boldsymbol{m}}_f \end{bmatrix} \tag{16}$$

where $\boldsymbol{T}$ is the vector of temperature states, $\dot{\boldsymbol{T}}$ is given in Eq. (11), T^t is the temperature of the thermal sink (the chiller), and there are $n_\xi = 4 + 2N_c + N_f$ states.

The next constraint initializes the temperature states of the system:

$$\boldsymbol{T}_w(0) = T_{w,0}, \quad \boldsymbol{T}_f(0) = T_{f,0}, \quad \boldsymbol{T}_l(0) = T_{l,0} \tag{17}$$

where $T_{w,0}$ is the initial temperature of the cold plate walls, $T_{f,0}$ is the initial temperature of the cold plate fluids, and $T_{l,0}$ is the initial temperature of the tank and LLHX states.

To ensure that each component remains within an upper bound on its operating temperature, we include the following linear inequality path constraints on the temperature states:

$$\boldsymbol{T}_w(t) \leq \boldsymbol{T}_{w,\max}, \quad \boldsymbol{T}_f(t) \leq \boldsymbol{T}_{f,\max}, \quad \boldsymbol{T}_l(t) \leq \boldsymbol{T}_{l,\max} \tag{18}$$

where $\boldsymbol{T}_{w,\max} \in \mathbb{R}^{N_c}$ are the maximum allowable temperatures for the CPHX walls, $\boldsymbol{T}_{f,\max} \in \mathbb{R}^{N_c}$ are the maximum allowable temperatures for the CPHX fluids, and $\boldsymbol{T}_{l,\max} \in \mathbb{R}^4$ are the maximum allowable temperatures for the tank and LLHX states.

We also ensure nonnegative flow through each of the valves (i.e, no fluid flow in the reverse direction) with the following linear inequality path constraint:

$$0 \leq \dot{\boldsymbol{m}}_f. \tag{19}$$

The following linear equality path constraint enforces an algebraic form of conservation of mass to ensure that the total mass flow rate through all the parallel flows is equal to the mass flow rate through the pump:

$$\sum_{i=1}^{N_f} \dot{\boldsymbol{m}}_{f,i} = \dot{m}_p \tag{20}$$

where $\dot{m}_p$ is the mass flow rate of the pump, assumed to be constant in this article. We note that this algebraic form of conservation of mass can be used because the mass flow dynamics for this class of systems evolve over a timescale orders of magnitude faster than the thermal dynamics [22]. Therefore, the former can assumed to always be at steady-state when modeling the latter, with no significant loss of accuracy. This timescale separation is also leveraged, for example, in Ref. [5].

In physical systems, actuators are often subject to limits on the maximum rate of change by which they can modulate the system. In particular, fluid pumps and valves may be subject to an upper bound on the rate of change by which the fluid mass flow rate can be adjusted. In general, components with more restrictive rate limits can be obtained at lower cost, for example due to employing less powerful motors. Rate limits may also be imposed on a hydraulic system to reduce the wear and tear on components that

can be caused by sudden pressure spikes. In practice, these rate limits are often determined through experimentation. For example, in Ref. [33], a rate limit of approximately 0.05 kg/s^2 was determined experimentally for the pump of a fluid-thermal system similar to those studied in this article. To capture this physical limitation, we enforce constraints on the derivative of the mass flow rates:

$$|\ddot{\boldsymbol{m}}_f(t)| \leq \ddot{m}_{f,\max} \tag{21}$$

where $\ddot{m}_{f,\max}$ is the rate limit.

Finally, we impose a small quadratic penalty term on the controls to both smooth the solution trajectories and improve convergence, modifying Eq. (15) to:

$$\min_{\ddot{\boldsymbol{m}}_f, t_{\text{end}}} \quad -t_{\text{end}} + \lambda \int_0^{t_{\text{end}}} \|\ddot{\boldsymbol{m}}_f(t)\|_2^2 dt \tag{22}$$

where λ is the penalty parameter. Larger values of λ help regularize the problem, but extremely large values push the solution away from the desired unpenalized value for t_{end}. We ensure that the additional penalty term has a limited overall effect on the objective function value by setting $\lambda \ll 1/(N_f \ddot{m}_{f,\max}^2)$, noting that $\lambda t_{\text{end}} N_f \ddot{m}_{f,\max}^2$ is the maximum value for the penalty term. Therefore, selecting $\lambda = 0.01/(N_f \ddot{m}_{f,\max}^2)$ ensures that the total penalty cost is always at most 1% of the final value for t_{end}.

5 CASE STUDIES

Here we present four case studies[2] to demonstrate both the efficacy and utility of the design methodology for cooling system architectures. The first case study (Sec. 5.1) contains three sub-cases in which a total heat load of 15 kW is divided in different ways among three CPHXs, while the temperature constraints on all CPHXs are the same. In the second case study (Sec. 5.2), both the heat loads and the temperature constraints vary among the CPHXs. To determine whether a steady-state feasible operating condition can be reached for a given set of temperature constraints and heat loads, a third case study (Sec. 5.3) was performed using three CPHXs. This case study also explores the sensitivity of thermal endurance to pump mass flow rate. Unlike the first three case studies, the fourth case study (Sec. 5.4) maximizes thermal endurance subject to time varying load profiles. The key model and optimization problem parameters used in the case studies are given in Table 1.

The nonlinear optimal control problem presented in Section 4 was solved using GPOPS-II, a commercial MATLAB-based software, which discretizes the dynamic optimization problem using pseudospectral methods into a finite-dimensional nonlinear program [17]. However, for the pure series architectures ($N_f = 1$), the flow rate through all cold plates must be equal to $\dot{m}_{\text{pump}}$. Therefore there are no control decisions to make, and we instead utilize a simple shooting-based, bisection method to determine t_{end} and solve the dynamic optimization problem more efficiently than with GPOPS-II. First, an initial interval $[\alpha, \beta]$ is defined such that the system at time α doesn't violate any of the temperature constraints, but the system at time β will, unless the system can reach steady state without violation. For example, the initial interval [0,10000] seconds is sufficient for the considered case studies. Thus, the interval either contains t_{end} or the current architecture can operate indefinitely. The interval endpoints are tested with a forward simulation using the dynamics in Eq. (16). Then the midpoint γ of the interval is tested and the interval is updated with the interval ($[\alpha, \gamma]$ or $[\gamma, \beta]$) that continues to bound t_{end}. This process continues until the interval is within a small tolerance (here we use 10^{-3} seconds). Typically, about 20 simulations are needed as the interval shrinks at a rate of 2^n, where n is the number of iterations.

[2]The MATLAB code for the test problems is available in Ref. [32].

Table 1 Key model and optimization problem parameters.

Parameter	Value
CPHX wall mass	1.15 kg
LLHX wall mass	1.2 kg
Tank fluid mass	2.01 kg
Thermal sink temperature T^t	15 °C
Tank/LLHX initial temperatures, $T_{l,0}$	15 °C
CPHX initial wall temperatures, $T_{w,0}$	20 °C
CPHX initial fluid temperatures, $T_{f,0}$	20 °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	$0.01/(N_f \ddot{m}^2_{f,\max})$

All reported computational costs were obtained using a workstation with an Intel Xeon E5-2660 CPU @ 2.00 GHz, 64 GB DDR4-2400 RAM, WINDOWS 10 64-bit, and MATLAB 2018a. Also note that the task of evaluating the nonlinear optimal control problems for each of the architectures is parallelizable, greatly reducing the absolute computational cost depending on the available resources. A maximum of 64 parallel threads have been used, where each parallel thread solves one architecture dynamic optimization problem at a time.

5.1 Case Study 1

Here we study cooling system architectures that have three CPHXs. As shown in Fig. 5 of Section 3, there are thirteen unique architectures possible. For these results, we assume that all temperature constraints are equal to 45 °C and that the total heat load applied to the cooling system is equal to 15 kW.

5.1.1 Identical Heat Loads

The first set of results has an identical heat load to each CPHX, resulting in only three unique architectures. The results are shown in Fig. 6. The optimal values for t_{end} were $(133, 129, 123)$ s, for the (pure parallel, hybrid, pure series) architectures, respectively. This result indicates that the pure parallel architecture is the best one under these conditions. Note that in each of the cases, at least one of the temperature constraints is active at the final time (a necessary stopping condition), and for both $1-2-3$ and $1-32$, all the wall temperatures are at their maximum allowable value. The average computational cost for each architecture was 127.4 Thread·seconds (T·s) (with a total cost of 1,655.9 T·s) within the interval [2.3, 460.6] T·s.

5.1.2 Unique Heat Loads

The next set of results, shown in Fig. 7, have a unique and linearly spaced heat load to each CPHX. Here the best architecture is a series topology, 321, with $t_{\text{end}} = 234$ s (in fact, there would only be a minor constraint violation of 0.25 °C were the constraint softened such that the system could reach steady-state). The pure parallel architecture was 7th best overall with $t_{\text{end}} = 119$ s. The worst architecture was 123, with $t_{\text{end}} = 71$ s. It is observed that the CPHX with the highest heat load (3) should be connected to the root to achieve good thermal endurance. In addition, many of the solutions have some of the

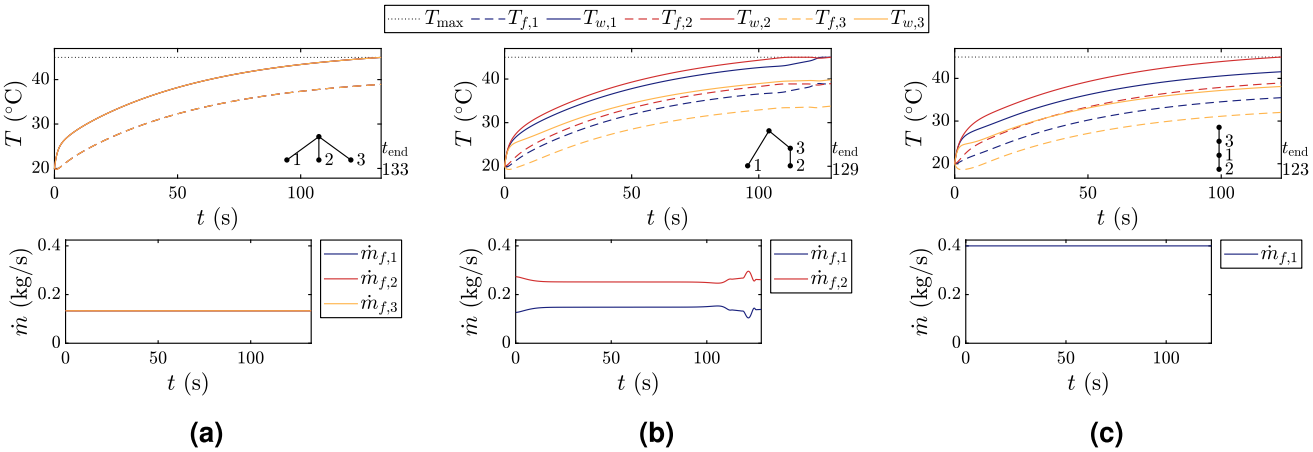


Fig. 6 Optimal temperature and flow trajectories for the architectures with $\boldsymbol{P}^s = [5, 5, 5]'$ kW and $T_{\max} = 45$ °C: (a) $N_f = 3$; (b) $N_f = 2$; (c) $N_f = 1$.

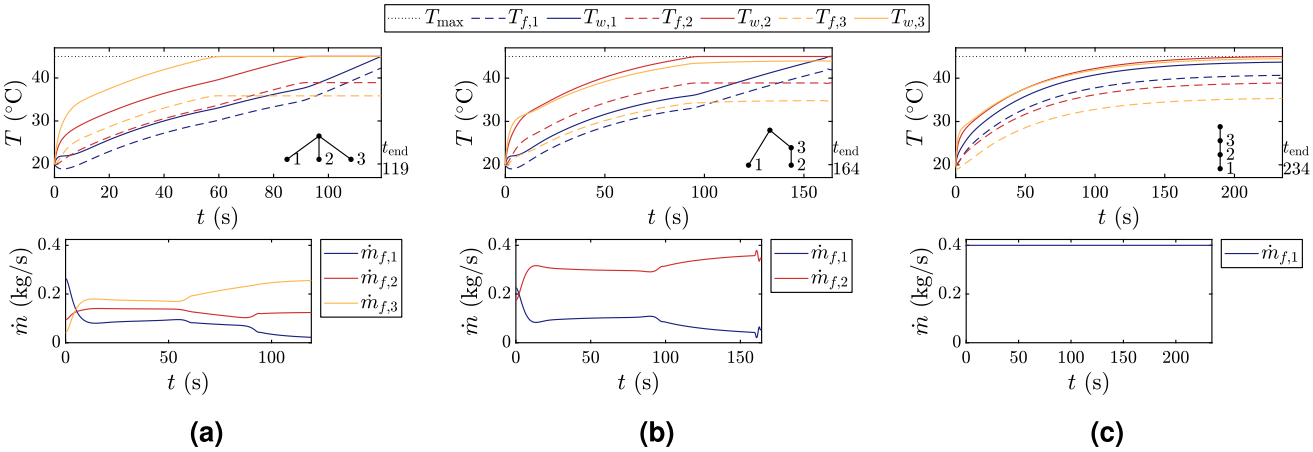


Fig. 7 Optimal temperature and flow trajectories for the best architectures with $\boldsymbol{P}^s = [2.5, 5, 7.5]'$ kW and $T_{\max} = 45$ °C: (a) best architecture with $N_f = 3$ $(1 - 2 - 3)$; (b) best architecture with $N_f = 2$ $(1 - 32)$; (c) best architecture with $N_f = 1$ (321).

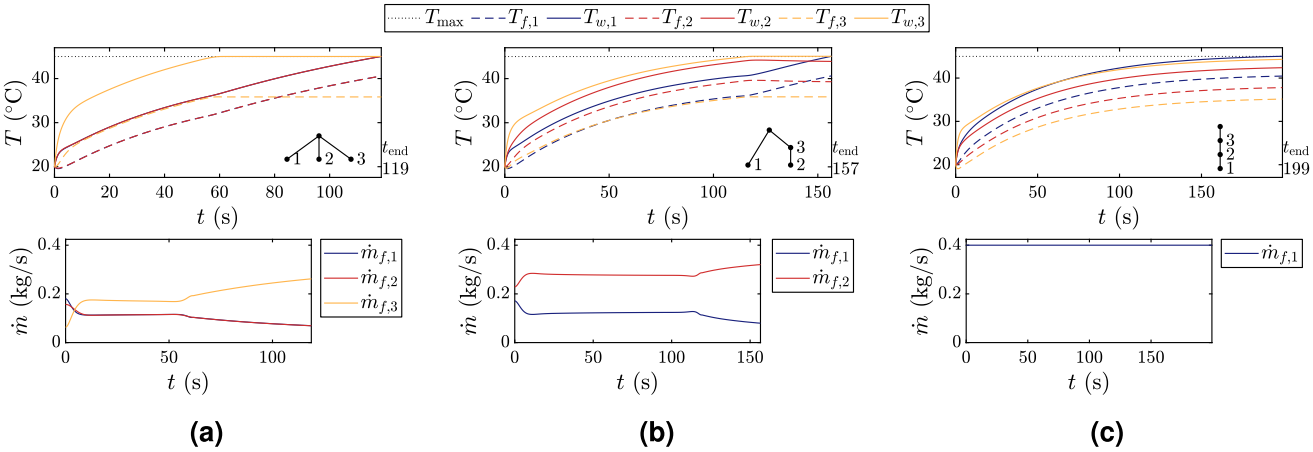


Fig. 8 Optimal temperature and flow trajectories for the best architectures with $\boldsymbol{P}^s = [3.75, 3.75, 7.5]'$ kW and $T_{\max} = 45$ °C: (a) best architecture with $N_f = 3$ $(1 - 2 - 3)$; (b) best architecture with $N_f = 2$ $(1 - 32)$; (c) best architecture with $N_f = 1$ (321).

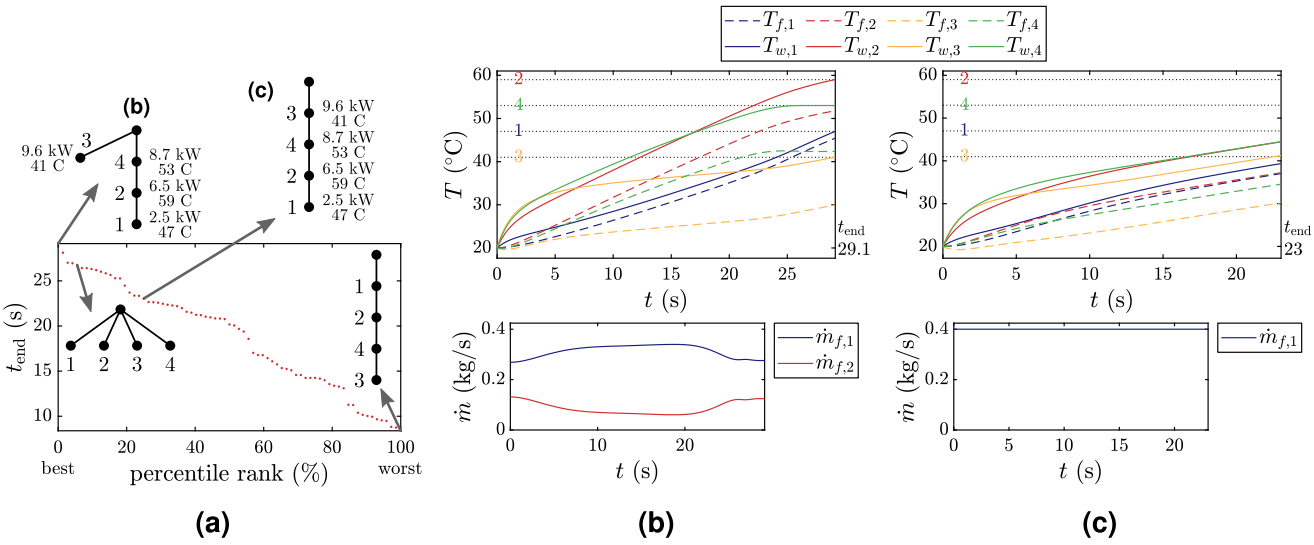


Fig. 9 Results from case study 2 for enumeration with four CPHXs: (a) sorted results for all 73 architectures; (b) best architecture (3 – 421); (c) best pure series architecture (3 – 4 – 2 – 1).

temperature path constraints active for long periods of time (up to 80 s). The average computational cost for each architecture was 401.8 T·s (with a total cost of 5223.5 T·s) within the interval [2.3, 2176.5] T·s.

5.1.3 Grouped Heat Loads

Here we consider the case when two CPHXs have the same heat load and the third is double the heat load of the first two. The results are shown in Fig. 8. Again, the series architectures 321 and 312 (identical models) performed the best with $t_{\text{end}} = 199$ s, while the pure parallel architecture was 5th with $t_{\text{end}} = 119$ s. Comparing these results to those under unique heat loads, we see that the best architecture performs worse with these grouped heat loads than in the unique case, even though the total heat load on the system is the same. The average computational cost for each architecture was 161.6 T·s (with a total cost of 2101.1 T·s) within the interval [2.3, 651.3] T·s.

5.2 Case Study 2

Unlike in case study 1, here we select different values for both the temperature constraints and heat loads among the CPHXs; thus, it is quite challenging to determine through intuition the optimal architecture.

5.2.1 Enumeration with Four CPHXs

For this first result, we consider architectures with four CPHXs and:

$$\boldsymbol{P}^s = [2.5, 6.5, 9.6, 8.7]' \text{ kW}, \quad \boldsymbol{T}_{w,\max} = [47, 59, 41, 53]' \text{ °C}$$

The results for all 73 architectures is summarized in Fig. 9(*a*). The best architecture was 3 – 421. This differs from the previous case study, where either pure series or pure parallel was determined to be the best. Observing the trajectories in Fig. 9(*b*) shows a large amount of constraint activity, with all four wall temperature constraints active at the end. The pure parallel architecture was the 5th best architecture, with marginally shorter thermal endurance than the best (26.8 s vs. 29.1 s). The average

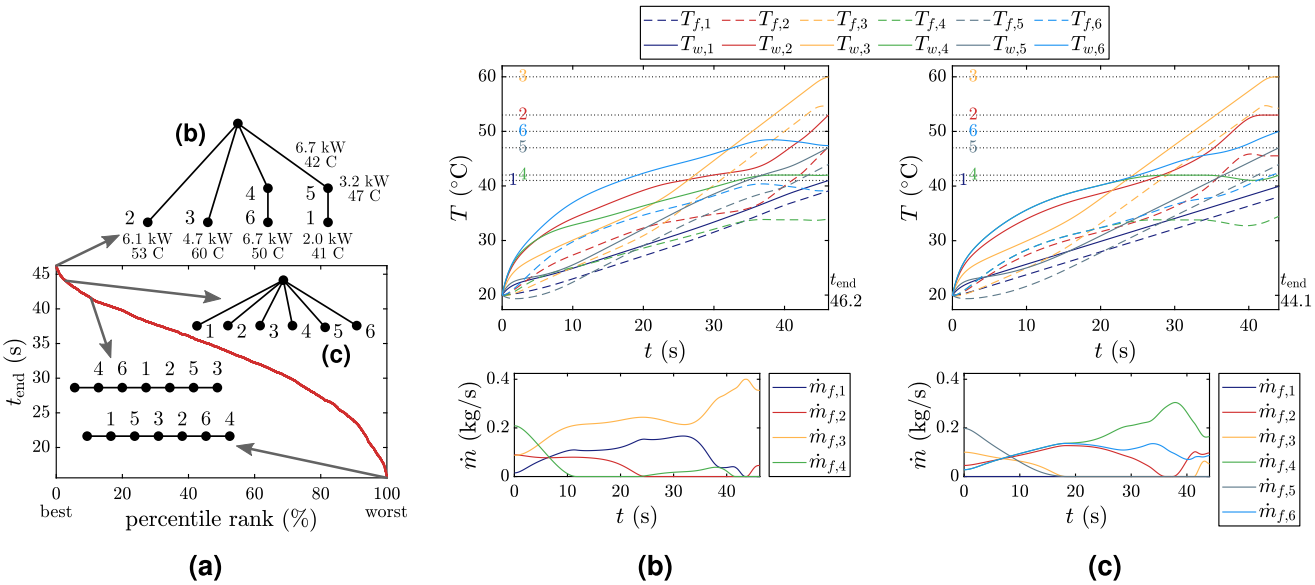


Fig. 10 Results from case study 2 for enumeration with six CPHXs: (a) sorted results for all 4051 architectures; (b) best architecture (2−3−46−51); (c) pure parallel architecture (1 − 2 − 3 − 4 − 5 − 6).

computational cost for each architecture was 33.3 T·s (with a total cost of 2428.4 T·s) within the interval [1.5, 295.5] T·s.

5.2.2 Enumeration with Six CPHXs

This study is structured similarly to the previous result but with six CPHXs:

$$\boldsymbol{P}^s = [2.0, 6.1, 4.7, 6.7, 3.2, 6.7]' \text{ kW}$$
$$\boldsymbol{T}_{w,\max} = [41, 53, 60, 42, 47, 50]' \text{ °C}$$

The results for all 4051 architectures are summarized in Fig. 10(*a*). Here, the best architecture was 2−3−46−51 with a thermal endurance of 46.2 s. The pure parallel architecture was the 116th best architecture and lasted for 44 s. The temperature and flow rate trajectories for both of these architectures are shown in Figs. 10(*b*) and 10(*c*), respectively. As with the previous result, the solution is not necessarily intuitive. Additionally, since we evaluated all 4051 architectures, we have a large amount of information to aid in choosing the architecture that is the best for the requirements in this case study. For example, patterns with desirable properties can be observed for the problem of interest, which is beyond what is possible when only generating a *single* optimal design. From these results, one clear pattern is improved thermal endurance when CPHX 4 is placed at the root. The average computational cost for each architecture was 155.0 T·s (with a total cost of 628060.0 T·s) within the interval [2.5, 5965.2] T·s.

5.3 Case Study 3

In the third case study, we consider the goal of determining whether a feasible steady-state operating condition can be reached for a given set of heat loads and temperature constraints. To test this property, we set an upper bound on t_{end} of 1000 s as a conservative estimate of the maximum duration that a feasible architecture could take to reach state-state. Here we used unique and linearly spaced heat loads with $N_c = 3$, similar to Section 5.1.2 except now with a slightly reduced total heat load of 13.5 kW.

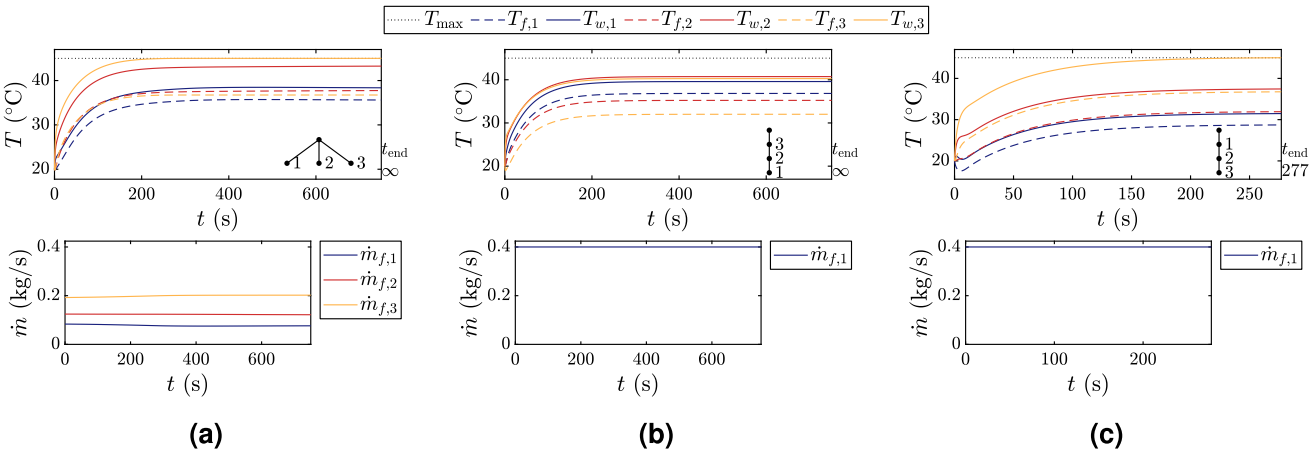


Fig. 11 Select results for $P^s = [2.25, 4.5, 6.75]$ kW and $T_{\max} = 45$ °C with steady-state solutions: (a) feasible architecture with $N_f = 3$ (1 − 2 − 3); (b) feasible architecture with $N_f = 1$ (321); (c) infeasible architecture with least t_{end} (123).

The results for several of the architectures are shown in Fig. 11. Of the 13 candidates, 11 were able to handle the heat loads for 1000 s. The two that were unable to satisfy the conditions were 123 and 213. The remaining candidates with feasible steady-state solutions are distinguished by differing numbers of CPHXs hitting their upper temperature bound. Architecture 321 in Fig. 11(*b*) had the lowest maximum temperature of 41 °C, so none of the constraints were active. These results demonstrate the sensitivity of thermal endurance with respect to the total heat load, as many architectures with 13.5 kW total heat load can operate indefinitely, while the best architecture with 15 kW could only last 234 s.

These results suggest several interesting design questions, such as which architecture achieves steady-state behavior with the lowest pump mass flow rate $\dot{m}_p$ (i.e., the smallest pump size), and is there a value for $\dot{m}_p$ such that all the architectures can achieve steady-state behavior? To investigate these questions, a pump sensitivity study was performed using the same conditions as Fig. 11. The results of the study are shown in Fig. 12 for all thirteen architectures, where each architecture is represented by a distinct curve, and each point on each curve represents maximal t_{end} for a given $\dot{m}_p$ value for the corresponding architecture. The pump flow rates increase from zero kg/s, where all architectures have a thermal endurance of 14.1 s. For each of the architectures, the thermal endurance increases as the pump flow rate increases. However, the rate of change can be different between the architectures and we can observe that the rankings of the architectures may be different at different flow rates. The pure series architecture (123), shown in Fig. 11(*c*), that was infeasible when $\dot{m}_p = 0.4$ kg/s is marked by the black dot in Fig. 12. We can observe that only a small increase above 0.4 kg/s is required to achieve indefinite operation, as all the architectures exhibit a steep increase in thermal endurance when near their steady-state asymptote. Overall, all the architectures do achieve steady-state behavior at or below 0.41 kg/s. To minimize the required pump size, the pure series architecture (321) is clearly the best architecture, achieving steady-state behavior with the lowest pump mass flow rate. Overall, 1296 problems were solved with an average computational cost of 391.6 T·s (with a total cost of 507510.9 T·s) within the interval [0.4, 21473.3] T·s.

5.4 Case Study 4

In contrast to the previous case studies with constant heat loads, time-varying heat load profiles are considered in this final case study. This is motivated by applications in which the heat loads are transient in nature, for which cooling system design becomes even more challenging and less intuitive than under

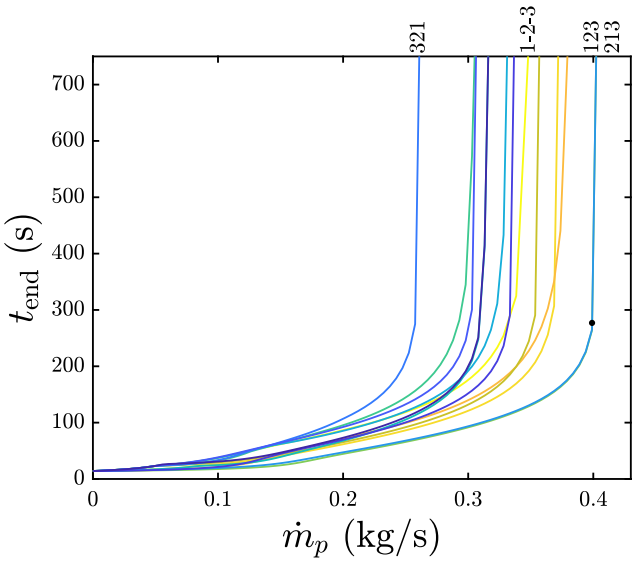


Fig. 12 Thermal endurance results for $\boldsymbol{P^s} = [2.25, 4.5, 6.75]$ kW and $T_{\max} = 45$ °C with varying values of the pump mass flow rate (select architectures are labeled).

constant loading. As will be seen, the computational cost of optimizing each architecture is typically higher as compared to the previous studies with constant heat loads. This is due to the more dynamic behavior of the solutions.

Here, three unique trapezoidal pulse profiles, shown in Fig. 13(*a*), were used with different peak powers and duty cycles. Under these conditions, the pure parallel architecture (1 – 2 – 3) had the maximum thermal endurance of 41.8 s, while the 312 pure series architecture was the worst, lasting for only 27.5 s. The temperature and flow rate trajectories for both of these architectures are shown in Figs. 13(*b*) and 13(*c*), respectively. For the pure parallel architecture, the fluid was dynamically distributed between the two more challenging heat load profiles, P_1^s and P_2^s. For P_3^s, there was no coolant flow across the entire time horizon, demonstrating that under certain conditions, some components should not be actively cooled in maximizing thermal endurance. As the thermal endurance is approached, we observe that the flow was controlled such that just enough cooling was provided to prevent both CPHX 1 and 2 from overheating at different times, although the net heating of the system over time eventually leads to constraint violation. Therefore, it seems that the pure parallel architecture was the best because flow could be directed to the necessary components on demand. The average computational cost for each architecture was 1073.0 T·s (with a total cost of 13949.0 T·s) within the interval [72.1, 3334.8] T·s.

6 CONCLUSION

This article serves as a preliminary study toward the design of novel fluid-based thermal management architectures with optimal coolant flow distribution. Candidate cooling system design architectures have been generated using labeled trees and a graph-based dynamic thermal modeling framework. A variable horizon optimal control problem is posed to identify cooling architectures having maximum thermal endurance. Four case studies are presented to demonstrate the effectiveness and potential of the proposed design methodology. The optimal solutions obtained show that intuition and experience may not be sufficient to identify the best-performing cooling architectures, especially as complexity increases. In particular, this methodology serves as a powerful tool to aid engineers in finding feasible and optimal architectures at different operating conditions. The identified optimal control trajectories can also serve as a basis for the realization of implementable controllers. Finally, this work highlights

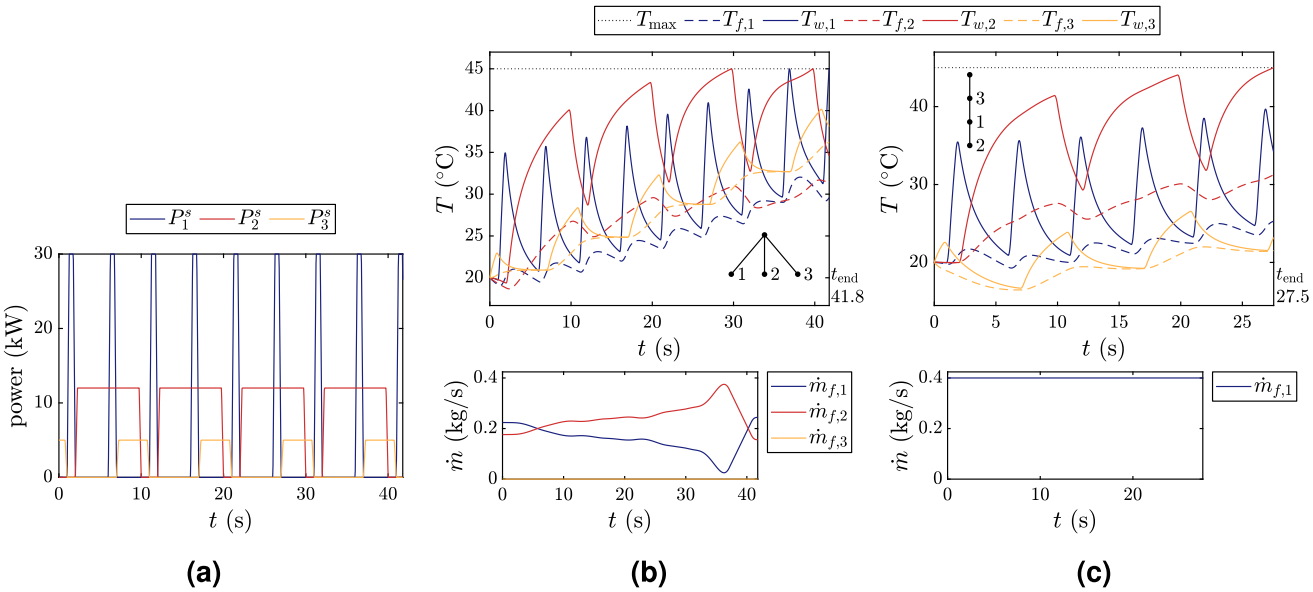


Fig. 13 Results from case study 4 with time-varying power flows: (a) time-varying power flows; (b) best architecture (1 – 2 – 3); (c) worst architecture (312).

some of the challenges associated with utilizing enumerative methods in architecture design, such as the combinatorial growth of the *restricted* class of architectures considered here, as well as the automated construction and solving of the corresponding dynamic optimization problems.

Future work will incorporate the hydraulic graph-based models of Refs. [15, 22] in the optimization problem to capture the pressure drops and resulting mass flow rate constraints associated with each candidate architecture of the cooling system. In addition, an alternative fixed-horizon optimization problem in which the objective is to minimize violations of soft upper bounds on component temperatures, similar to that used for control in Ref. [15], will be considered. Furthermore, future work will leverage the modularity of the graph-based modeling approach to explore more general cooling system architectures having multiple fluid loops, tanks, pumps, and degrees of freedom in the topology beyond a single split. This freedom will greatly increase the number of architectures that must be considered in an enumerative solution strategy. To address some of the issues associated with the combinatorial growth intrinsic to enumerative methods, machine learning (using enumerated design data for training) [34] or heuristic-based methods [35] developed for architecture design problems could be utilized to identify high-performing candidates efficiently. The design problem in this article could serve as a good case study in the development of these methods, tailored towards architecture design of dynamic engineering systems. As an additional component of future work a subset of the best designs identified here will be experimentally validated on the reconfigurable fluid-thermal testbed described in Refs. [15, 22]. A long term goal of this work is to provide engineers with design guidelines supporting the development of unprecedented cooling systems that enable future power electronic 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, and the National Science Foundation Graduate Research Fellowship under Grant Number DGE-1144245.

REFERENCES

[1] Peddada, S. R. T., Herber, D. R., Pangborn, H. C., Alleyne, A. G., and Allison, J. T., 2018. "Optimal flow control and single split architecture exploration for fluid-based thermal management". In ASME 2018 International Design Engineering Technical Conferences, no. DETC2018-86148, p. V02AT03A005. doi: 10.1115/DETC2018-86148

[2] Kassakian, J. G., and Jahns, T. M., 2013. "Evolving and emerging applications of power electronics in systems". *IEEE Journal of Emerging and Selected Topics in Power Electronics,* ***1***(2), June, pp. 47–58. doi: 10.1109/JESTPE.2013.2271111

[3] Lequesne, B., 2015. "Automotive electrification: The nonhybrid story". *IEEE Transactions on Transportation Electrification,* ***1***(1), June, pp. 40–53. doi: 10.1109/TTE.2015.2426573

[4] Yu, X. E., Xue, Y., Sirouspour, S., and Emadi, A., 2012. "Microgrid and transportation electrification: A review". In IEEE Transportation Electrification Conference and Expo, pp. 1–6. doi: 10.1109/ITEC.2012.6243464

[5] Doman, D. B., 2017. "Fuel flow control for extending aircraft thermal endurance". *Journal of Thermophysics and Heat Transfer,* ***32***(1), Jan., pp. 35–50. doi: 10.2514/1.T5142

[6] Park, S., and Jung, D., 2010. "Design of vehicle cooling system architecture for a heavy duty series-hybrid electric vehicle using numerical system simulations". *Journal of Engineering for Gas Turbines and Power,* ***132***(9), Sept., p. 092802. doi: 10.1115/1.4000587

[7] Ouchi, M., Abe, Y., Fukagaya, M., Ohta, H., Shinmoto, Y., Sato, M., and Iimura, K.-i., 2012. "Thermal management systems for data centers with liquid cooling technique of CPU". In IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems, pp. 790–798. doi: 10.1109/ITHERM.2012.6231507

[8] Doty, J., Yerkes, K., Byrd, L., Murthy, J., Alleyne, A., Wolff, M., Heister, S., and Fisher, T. S., 2017. "Dynamic Thermal Management for Aerospace Technology: Review and Outlook". *Journal of Thermophysics and Heat Transfer,* ***31***(1), pp. 86–98. doi: 10.2514/1.T4701

[9] Jain, N., and Alleyne, A., 2015. "Exergy-based optimal control of a vapor compression system". *Energy Conversion and Management,* ***92***, Mar., pp. 353–365. doi: 10.1016/j.enconman.2014.12.014

[10] Kim, J.-K., and Smith, R., 2001. "Cooling water system design". *Chemical Engineering Science,* ***56***(12), June, pp. 3641–3658. doi: 10.1016/S0009-2509(01)00091-4

[11] Panjeshahi, M. H., 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), Feb., pp. 200–209. doi: 10.1016/j.cherd.2008.08.004

[12] Muller, C. J., and Craig, I. K., 2016. "Energy reduction for a dual circuit cooling water system using advanced regulatory control". *Applied Energy,* ***171***, June, pp. 287–295. doi: 10.1016/j.apenergy.2016.03.069

[13] Delia, D. J., Gilgert, T. C., Graham, N. H., Hwang, U., Ing, P. W., Kan, J. C., Kemink, R. G., Maling, G. C., Martin, R. F., Moran, K. P., Reyes, J. R., Schmidt, R. R., and Steinbrecher, R. A., 1992. "System cooling design for the water-cooled IBM enterprise system/9000 processors". *IBM Journal of Research and Development,* ***36***(4), July, pp. 791–803. doi: 10.1147/rd.364.0791

[14] Kim, J.-K., Lee, G. C., Zhu, F. X. X., and Smith, R., 2002. "Cooling system design". *Heat Transfer Engineering,* ***23***(6), pp. 49–61. doi: 10.1080/01457630290098754

[15] 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), Oct., p. 101016. doi: 10.1115/1.4040211

[16] Herber, D. R., Guo, T., and Allison, J. T., 2017. "Enumeration of architectures with perfect matchings". *Journal of Mechanical Design,* ***139***(5), May, p. 051403. doi: 10.1115/1.4036132

[17] 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,* ***41***(1), Oct. doi: 10.1145/2558904

[18] Betts, J. T., 2010. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming. SIAM.* doi: 10.1137/1.9780898718577.fm

[19] McCarthy, K., Walters, E., Heltzel, A., Elangovan, R., Roe, G., Vannice, W., Schemm, C., Dalton, J., Iden, S., Lamm, P., Miller, C., and Susainathan, A., 2008. "Dynamic thermal management system modeling of a more electric aircraft". In Power Systems Conference, no. Paper no. 2008-01-2886. doi: 10.4271/2008-01-2886

[20] Ganev, E., and Koerner, M., 2013. "Power and thermal management for future aircraft". In AeroTech Congress & Exhibition, no. 2013-01-2273. doi: 10.4271/2013-01-2273

[21] Kim, E., Shin, K. G., and Lee, J., 2014. "Real-time battery thermal management for electric vehicles". In ACM/IEEE International Conference on Cyber-Physical Systems, pp. 72–83. doi: 10.1109/ICCPS.2014.6843712

[22] 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". In ASME Dynamic Systems and Control Conference, no. DSCC2016-9782, p. V002T21A008. doi: 10.1115/DSCC2016-9782

[23] Moore, K. L., Vincent, T. L., Lashhab, F., and Liu, C., 2011. "Dynamic consensus networks with application to the analysis of building thermal processes". *IFAC Proceedings Volumes,* ***44***(1), Jan., pp. 3078–3083. doi: 10.3182/20110828-6-IT-1002.02549

[24] Williams, M. A., Koeln, J. P., Pangborn, H. C., and Alleyne, A. G., 2018. "Dynamical graph models of aircraft electrical, thermal, and turbomachinery components". *Journal of Dynamic Systems, Measurement, and Control,* ***140***(4), Apr., p. 041013. doi: 10.1115/1.4038341

[25] Preisig, H. A., 2009. "A graph-theory-based approach to the analysis of large-scale plants". *Computers & Chemical Engineering,* ***33***(3), Mar., pp. 598–604. doi: 10.1016/j.compchemeng.2008.10.016

[26] Pangborn, H. C., Koeln, J. P., and Alleyne, A. G., 2018. "Passivity and decentralized MPC of switched graph-based power flow systems". In American Control Conference, pp. 6658–6665. doi: 10.23919/ACC.2018.8431722

[27] West, D. B., 2001. *Introduction to Graph Theory*, 2nd ed. Pearson.

[28] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., 2009. *Introduction to Algorithms*, 3rd ed. MIT Press.

[29] Riordan, J., 1960. "The enumeration of trees by height and diameter". *IBM Journal of Research and Development,* ***4***(5), Nov., pp. 473–478. doi: 10.1147/rd.45.0473

[30] A000262. The on-line encyclopedia of integer sequences. url: https://oeis.org/A000262

[31] Stanley, R. P., 2011. *Enumerative Combinatorics: Volume 1*, 2nd ed. Cambridge University Press.

[32] Peddada, S. R. T., Herber, D. R., Pangborn, H. C., Alleyne, A. G., and Allison, J. T. Cooling-System Architecture Project. [GitHub], url: https://github.com/satyartpeddada/csap

[33] Pangborn, H., Hey, J. E., Deppen, T. O., Alleyne, A. G., and Fisher, T. S., 2017. "Hardware-in-the-loop validation of advanced fuel thermal management control". *Journal of Thermophysics and Heat Transfer,* ***31***(4), Oct., pp. 901–909. doi: 10.2514/1.T5055

[34] Guo, T., Herber, D. R., and Allison, J. T., 2018. "Reducing evaluation cost for circuit synthesis using active learning". In ASME 2018 International Design Engineering Technical Conferences, no. DETC2018-85654, p. V02AT03A011. doi: 10.1115/DETC2018-85654

[35] Chakrabarti, A., Shea, K., Stone, R., Cagan, J., Campbell, M., Hernandez, N. V., and Wood, K. L., 2011. "Computer-based design synthesis research: An overview". *Journal of Computing and Information Science in Engineering,* ***11***(2), June, p. 021003. doi: 10.1115/1.3593409