

# An Automated Design Tool for Generation and Selection of Optimal Aircraft Thermal Management System Architectures

Robert Buettner\*

*UES, Inc., Dayton, OH 45432, United States*

Daniel R. Herber†

*Colorado State University, Fort Collins, CO 80523, United States*

Philip Abolmoali‡ and Soumya S. Patnaik§

*Air Force Research Laboratory, Wright-Patterson AFB, OH 45433, United States*

**This paper introduces an automated graph theory-based aircraft thermal management system (TMS) architecture design tool. The tool is designed to generate and facilitate the comparison of possible TMS architectures within a combination of built-in and user-specified constraints. The tool features the ability to generate architectures including open-loop air cycle machines and closed and semi-closed liquid cooling circuits. A description of the tool's important features is provided. A case study using the tool for the design of an optimal TMS for an aircraft using hybrid-electric propulsion is presented as an example of the tool's capabilities. Results of the case study are discussed as well as ongoing and planned improvements to the tool.**

## I. Introduction

As mission systems in military aircraft continue to grow in complexity, electrical power requirements and the thermal waste energy that must be managed are also increasing [1,2,3,4]. At the same time, the available thermal sinks are shrinking due to the reduction of ram air inlet sizes, the use of more efficient engines, and the employment of composite skins with low thermal conductivity [1,2,3,4]. Historically, TMS architectures have been designed using the experience and intuition of the designer, which can sometimes fail to identify non-intuitive, but optimal solutions. The architecture of a system is defined here as the set of all components in the system and the physical connections between them. These facts have led to the development of numerous methodologies and tools to design more effective and efficient thermal management systems (TMS) for aircraft, such as those discussed in [1,2,4]. Few, if any, such tools provide the designer with assistance in creating multiple possible architectures from which an optimal TMS design can be chosen.

Recently, the authors used an enumerative graph generation technique described in [5,6] to generate all feasible air cycle machine (ACM) based TMS architectures within a design space [3]. Graph theory is especially useful for this task as it enables the representation of the systems-of-interest as a set of vertices and edges. The vertices represent

---

\* Research Scientist, Aerospace Power & Propulsion Division, AIAA Non-member.

† Assistant Professor, Department of Systems Engineering, AIAA Member.

‡ Aerospace Engineer, Aerospace Systems Directorate, AIAA Non-Member.

§ Principal Engineer, Aerospace Systems Directorate, AIAA Senior Member.

the discrete components that make up the system, such as compressors and valves. The edges represent connections between the components where energy or information can be transferred. With an appropriate method of representing these vertices and edges, it is possible to generate all feasible architectures within a set of constraints known as network structure constraints (NSCs). Choosing appropriate NSCs was found to be very important to limit a study to a physically meaningful and computationally tractable design space.

In the present paper, the current state of the architecting tool and its ongoing development are discussed, along with results from a case study designing an optimal TMS for a hybrid electric aircraft. Thermal management challenges associated with aircraft hybrid electric propulsion have drawn the attention of many researchers in recent years [7,8]. These challenges are the result of greater electric power demands when compared to more conventional propulsion and power systems. The additional thermal waste energy dissipated by these systems adds to already challenging thermal management issues aboard military aircraft. TMS performance across a notional mission was evaluated by multipoint optimization. The present study is limited to TMS architectures featuring closed and semi-closed liquid cooling systems and ram air.

## II. Tool Overview

### A. Architecture Generation

We use a modified combinatorial approach to efficiently generate all potentially useful architectures within a set of NSCs [3]. The NSCs include the types and numbers of individual components in the system as well as constraints on the connections between them [3,5,6]. These must be carefully defined to avoid generating large numbers of infeasible architectures (from a practical implementation point of view) while at the same time not overly constraining the design space. For this purpose, a predefined list of components are modeled and provided within the tool. The remainder of the NSCs are built-in and, in general, define which components can, must, or must not be connected to each other. More complex NSCs are possible and the reader is referred to [3,5,6] for a more in-depth exploration of the topic. The NSCs used by the tool are based on the thermodynamic processes that occur within the component as well as other practical considerations. A more detailed discussion of the constraints is provided in [3,5,6]. With the NSCs defined, the tool can efficiently generate multiple TMS architectures that meet the user's requirements. With a large number of TMS within the reduced design space generated and evaluated, the likelihood of identifying an optimal system is increased.

### B. Architecture Comparison

Generating every potentially useful architecture is of little use without a means to compare them to each other. For this purpose, each proposed architecture must be simulated and evaluated against some metric. The simulation process will be covered in a later section. This section will describe the multipoint optimization procedure used to evaluate candidate systems. It is also possible to perform single-point optimizations with the tool, but this process is omitted for brevity, and interested readers are referred to [3] for more details.

#### 1. User Inputs

In order to fairly compare a set of TMS architectures, they must all be analyzed with the same boundary conditions and objectives. The user is therefore required to enter in objectives and operational constraints based on the components that are allowed in the study. Often, these will include minimizing pumping power for liquid circuits, minimizing air flow rates, and/or maintaining thermal load temperatures within acceptable bounds. The user must also define discrete sets of boundary conditions that the system must contend with. There are two types of boundary conditions that may be specified. The first type, referred to in this paper as *constraining points*, are sets of boundary conditions in which the aircraft must operate within all operating constraints, but the objective function values are not significant. It is possible to run studies without constraining points if the objective function values are deemed important at all points. The second type, referred to in this paper as *important points*, are sets of boundary conditions where the objective function values are of primary importance. At least one important point must be specified for a study. For each important point, the user must assign a weighting factor that establishes the relative importance of the point in question.

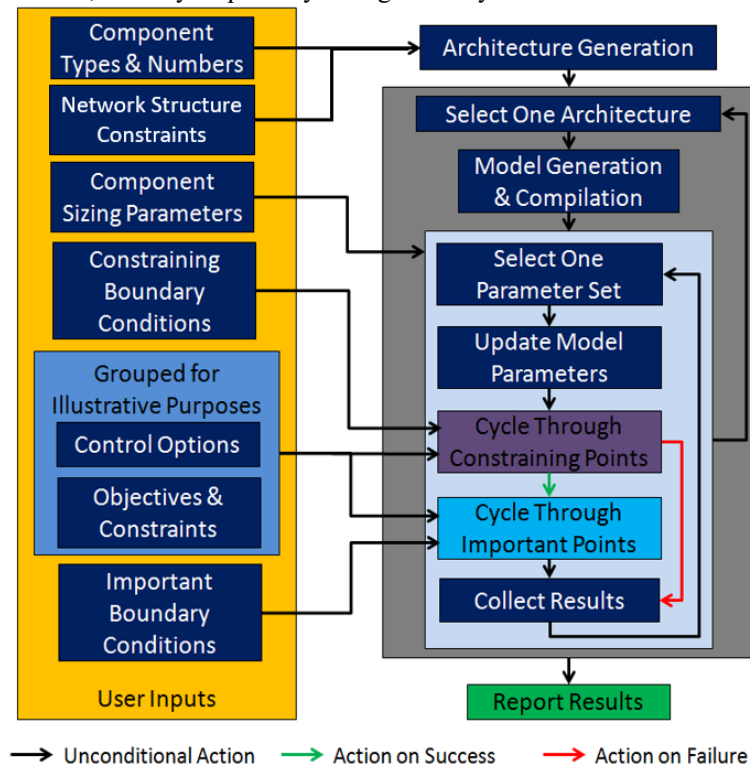
Additionally, the user must specify at least one set of component sizing parameters for each component type. Heat loads must be sized individually since their sizing is linked to the thermal load they are expected to dissipate. Giving more than one option for one or more sizing parameters will result in another layer of the optimization routine being added. Similarly, the user may specify one or more discrete values for each control parameter available in the system. Specifying more than one control option will result in an additional layer of the optimization routine. With each additional option, the number of simulations that must be performed will increase. For this reason, a tradeoff between solution space granularity and speed must be considered.

## 2. Comparison Procedure

The tool utilizes a nested co-design strategy as in [6] but replaces the inner-loop dynamic controls optimization with a discrete, multipoint controls optimization strategy based on the steady-state performance of the architectures. The outer layer of the optimization strategy analyzes each architecture and finds the optimal combination of user-defined discrete sizing parameter options by a brute force search if more than one set is specified. Nested within the sizing parameter optimization layer is the controls optimization layer.

For a single set of sizing parameters, the simulations are performed for every set of boundary conditions supplied by the user. Evaluation begins with the constraining points if any are specified. At each constraining point, the evaluation algorithm tries every combination of control parameter values until one is found that results in the system meeting all operational constraints. If no such combination is found, the sizing parameter set is discarded for the architecture, and the next set is tested. If no sizing parameter set is found that can satisfy the operational constraints for all constraining points, then the architecture is reported as infeasible and removed from further consideration. Otherwise, the tool runs through every important point for every combination of control parameters. If no combination is found that satisfies all operational constraints for any important point, the sizing parameter set is discarded, and the process begins again. When a viable parameter set is found, the control decision set that results in the minimum overall objective function value (the sum of all individual objective function values at a point) is selected, and those results are kept. If constraints are met for all important points, the objective function values for each point are multiplied by the point's weighting factor and added. The resulting sum is the overall weighted objective function value for the sizing parameter set.

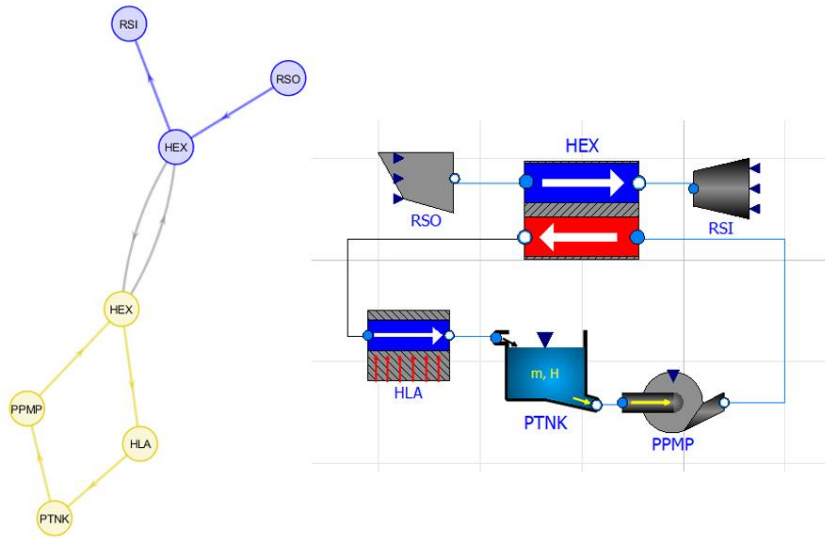
The overall weighted objective function value reported for the architecture is then the minimum value over all sizing parameter sets for that architecture. Once this procedure is completed for every architecture, the overall weighted objective function value for each architecture is reported to the user, as well as the sizing parameter values that achieved it. If no parameter sets are found that can meet the operational constraints for all boundary condition sets, then the architecture is reported as infeasible. The user can then select the most promising architectures for further, more detailed investigation. The multipoint optimization procedure is summarized graphically in Fig. 1. Note that the user inputs and the stages in which they are used by the tool are indicated as well. The NSCs are hard-coded in the current version of the tool, but may be partially configurable by the user in future versions.



**Fig. 1 Multipoint optimization procedure**

### C. Simulation Procedure and Component Models

Every architecture in a study must be simulated to determine its response to the boundary conditions, parameter sets, and control options provided by the user. For this purpose, system-level models are automatically generated from the graph representations. As previously done in [3], a set of custom component models created in the Modelica modeling language are used to generate the system-level models. Modelica is a component-oriented, physics-based modeling language [9] that lends itself well to graph-based architecture studies. This is because components are represented individually and therefore translate directly to vertices in a graph while connections between components are easily represented as edges. Fig. 2 shows an example of the graph representation of a simple ram air cooled liquid loop and the corresponding Modelica model illustrating the similarities between them. It should be noted that heat exchangers are represented in the graph as two vertices (both labeled HEX) connected to each other by a bidirectional edge while all other vertices and components have a one-to-one correspondence. This is done to facilitate generating architectures with multiple heat exchangers. A script is used to generate the system models in Modelica from the graph representations of the architectures. Additional scripts are used to compile and simulate the Modelica models, as was done in [3].



**Fig. 2 Example graph representation (left) and corresponding Modelica model (right)**

The custom Modelica component library included with the tool includes components for air cycle machines, ram air ducts, and liquid-cooling circuits. For brevity, only the components used in the example study will be presented here.

#### 1. Ram Air Source (RSO)

The ram air source is modeled as an infinite reservoir with a single fluid outlet. For the purposes of the tool, the temperature, pressure, and composition of air in the ram air source are held constant at the specified boundary conditions.

#### 2. Ram Air Sink (RSI)

The ram air sink is identical to the ram air source except in that it has a single fluid inlet instead of an outlet.

#### 3. Ram Air Fan (FAN)

The ram air fan is modeled as a pressure rise based on the difference between the mass flow rate through the fan and the minimum flow rate specified in the user-defined sizing parameters. This is done using what is essentially a proportional-integral controller saturated at zero such that the fan can never act as a pressure drop. The saturation prevents the flow rate from being held at the minimum value where it would otherwise increase based on the flow boundary conditions. The fan must be coupled to a mechanical rotational source. For the purposes of the example study, the fan is connected to a constant speed source from the Modelica Standard Library.

#### 4. Heat Exchanger (HEX)

The heat exchanger is a simple effectiveness-based counter-flow model that does not account for thermal storage. The pressure drop across each side is directly proportional to the square of the mass flow rate through that side with the proportionality constant being determined by user-defined parameters for design mass flow rate and design pressure drop.

5. *Heat Load (HL<sub>x</sub>, where x denotes a single-letter identifier supplied by the user)*

The heat load is an idealized component that assumes the fluid outlet temperature is equal to the temperature of the thermal mass. For the purposes of the tool, the thermal mass is taken to be one kilogram of aluminum. The pressure drop across the heat load is proportional to the square of the mass flow rate with the proportionality constant being determined by user-defined parameters for design mass flow rate and design pressure drop.

6. *Tank (PTNK, FTKK)*

The tank is a simple component that acts as an infinite, uniform pressure, constant temperature source at the fluid outlet and an infinite, uniform pressure, constant temperature sink at the fluid inlet. The tank is only used for the liquid cooling circuits. The first letter in the label identifies the loop the tank belongs in.

7. *Pump (PPMP, FPMP)*

The pump is modeled as a mass flow source with the assumptions that the difference in height between inlet and outlet are negligible, the inlet and outlet flow areas are equal, and the pump is well insulated. The pump model also assumes that any flow rate can be met at any pressure rise. The required pressure rise across the pump is used to calculate the pumping power required. The pump is only used for the liquid cooling circuits.

8. *Engine (ENG)*

The engine is modeled as a flow sink that pulls the mass flow rate specified by the boundary conditions out of the fuel cooling circuit. The engine component is not intended to investigate TMS and propulsion system interactions beyond the removal of fuel at the specified rate from the TMS. For the purposes of the tool, the flow rate into the engine is taken as a constant at each boundary condition point.

9. *Three-port Node (ND3)*

The three-port node is not truly a component. It is instead a construct used by the tool in the architecture generation phase. In the model generation phase, the node is replaced with connections that directly link all three components that were meant to be connected to the node. This is because the Modelica language treats such connections as ideal mixing chambers with zero volume, which is precisely how the node would be handled if it were modeled as a component.

10. *Flow regulating valve (FFRV)*

The flow regulating valve, referred to simply as the valve for the remainder of this paper, is modeled as a pressure drop proportional to the mass flow rate through the valve and the valve position. The proportionality constant for this relationship is the nominal hydraulic conductance specified by the user as a sizing parameter. Flashing and cavitation are not considered. This valve is only used in liquid loops. There is another valve model available in the tool, but it is only used in the bleed loop and thus is not used in this example study.

## **D. Graphical User Interface**

A graphical user interface (GUI) created using MATLAB was developed to enhance user experience. The GUI is designed to guide the user through defining a study in an intuitive manner and without requiring in-depth knowledge of the tool's internal conventions and limitations. The GUI is split up into several tabs, most of which check the user's inputs as they are added to the study and warns the user if any obvious errors or omissions are detected.

The first tab of the GUI is shown in Fig. 3 and allows the user to load a previously saved file containing a GUI-generated problem setup. The tab also enables optionally loading a previously-generated set of graphs, also called architectures for the purposes of this paper, and analyzing a sub-set of the architectures in order to facilitate running large studies on computers with limited memory.

The second tab enables the user to set various options, including the name and location of the folder in which the Modelica models and their compilation files will be stored, the level of parallelization, and the level of debugging outputs to be displayed. This tab is shown in Fig. 4.

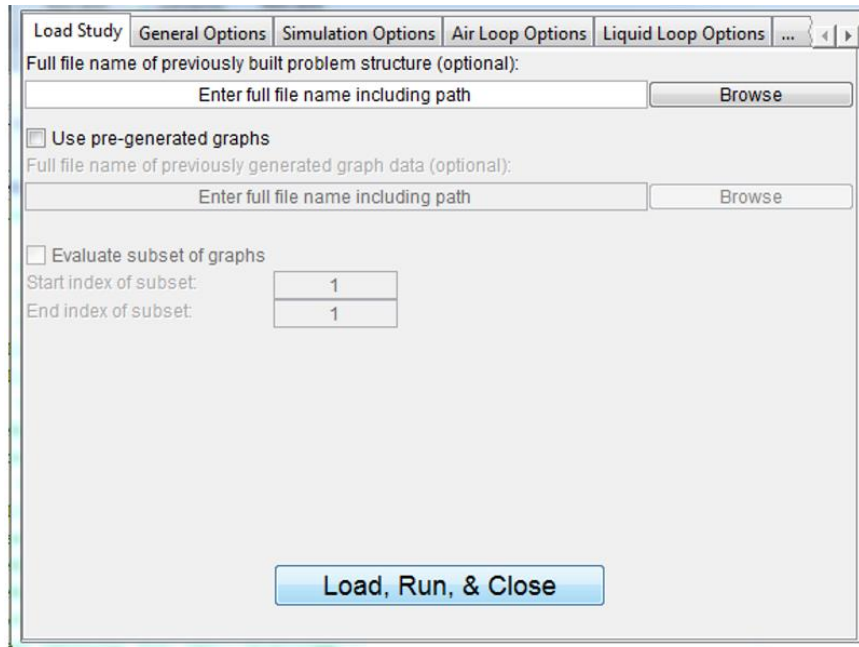
The third tab is shown in Fig. 5 and allows the user to set some basic simulation settings. These settings can affect the convergence of the study and the time required to obtain a solution.

The fourth tab is shown in Fig. 6 and allows the user to select which, if any, air circuits are to be present in the TMS. This tab also allows the user to select the number of components and provide sizing parameters using individual sizing GUIs.

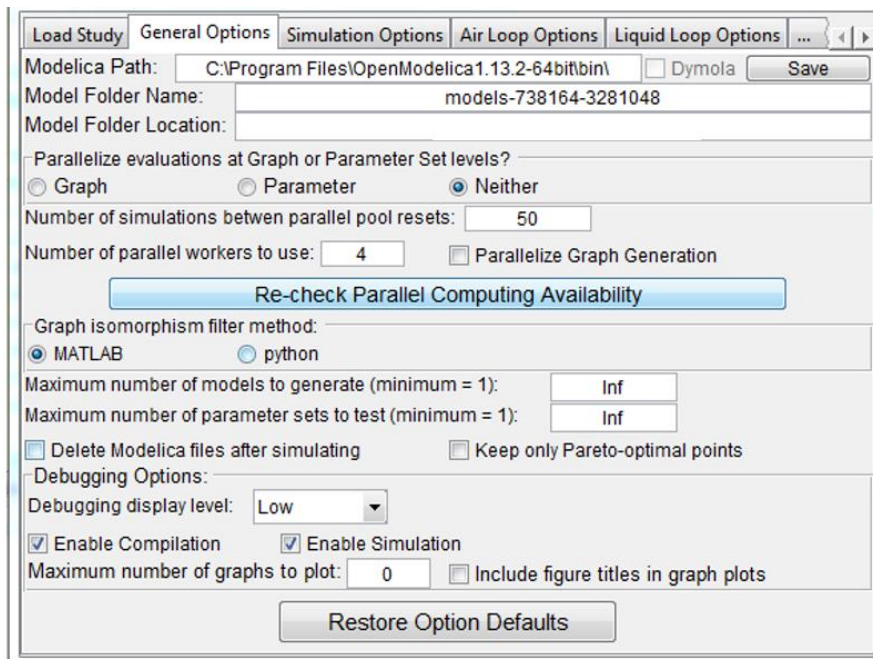
The fifth tab is similar to the fourth and is shown in Fig. 7. It allows for the configuration of liquid cooling circuits.

The sixth tab is shown in Fig. 8 and allows the user to configure the number and sizing parameters of heat loads, as well as the sizing of heat exchangers.

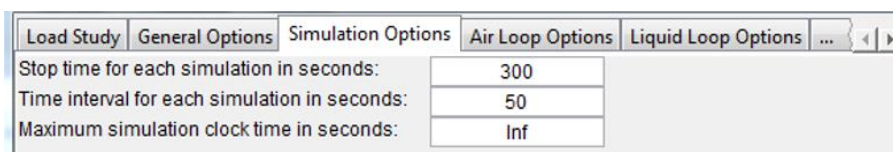
The seventh and final tab in the GUI wraps up the problem set up procedure and allows the user to enter boundary conditions and control options using pop-up GUIs. Additionally, this tab enables the user to decide if they want to run the study, or save it to run in another session. The finalization tab is shown in Fig. 9.



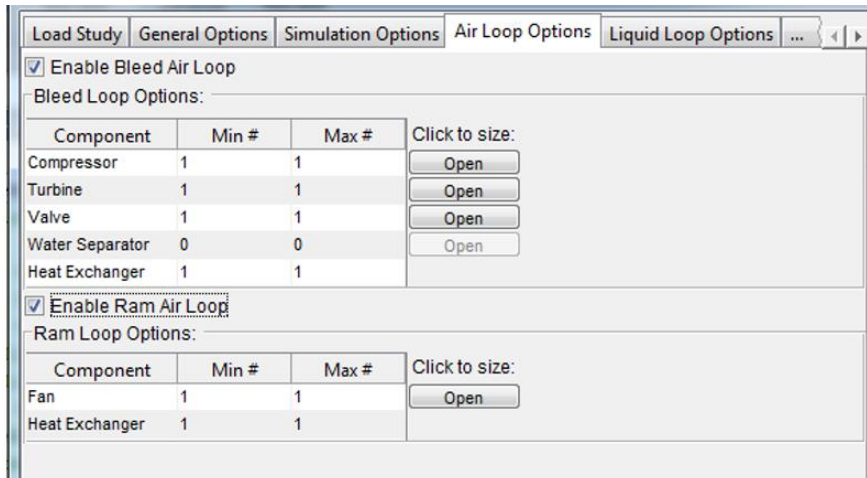
**Fig. 3 GUI tab #1: Load Study**



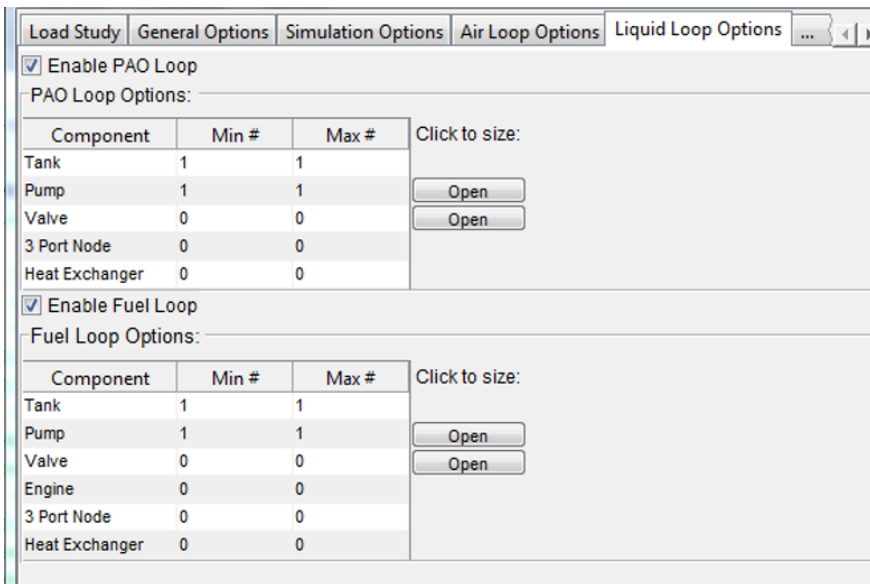
**Fig. 4 GUI tab #2: General Options**



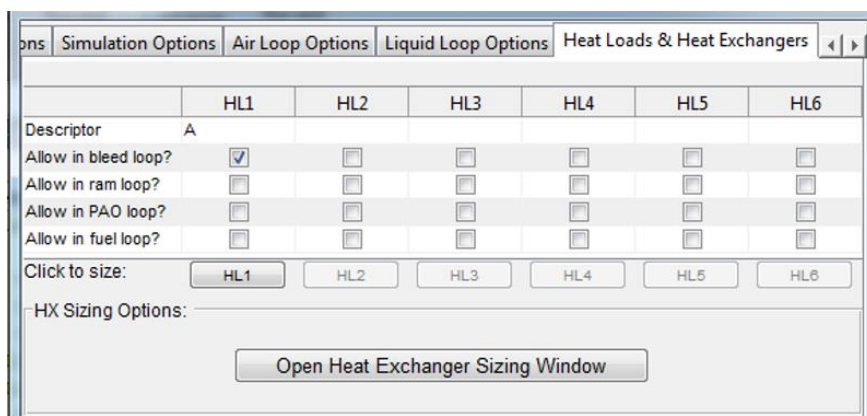
**Fig. 5 GUI tab #3: Simulation Options**



**Fig. 6 GUI tab #4: Air Loop Options**



**Fig. 7 GUI tab #5: Liquid Loop Options**



**Fig. 8 GUI tab #6: Heat Loads & Heat Exchangers**



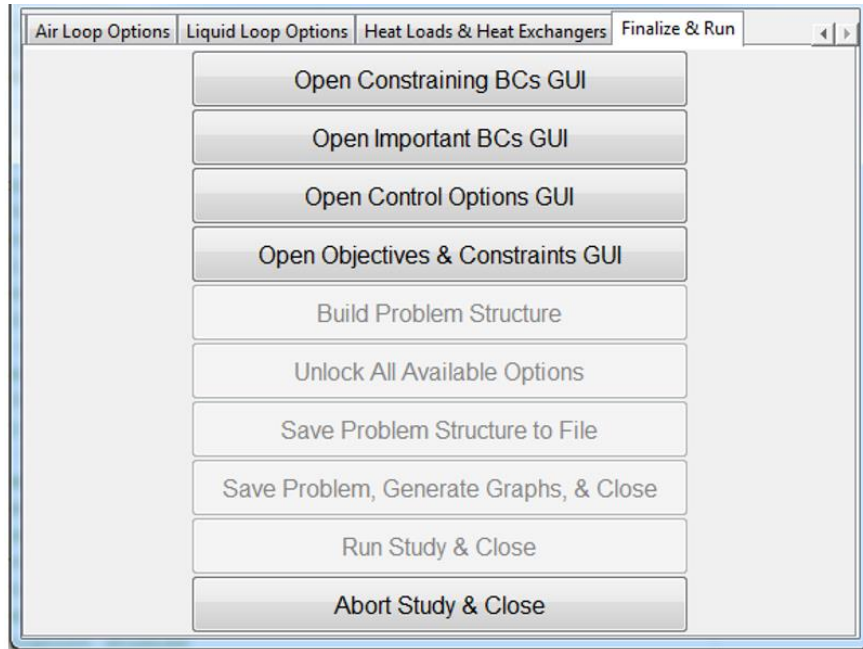


Fig. 9 GUI tab #7: Finalize & Run

### III. Hybrid Electric Propulsion TMS Study

#### A. General Information and Components

The aircraft studied here is a notional medium altitude long endurance vehicle with a series hybrid electric propulsion system. The primary power source for the system is a turboshaft engine connected to an electric generator. The propulsion is provided by an electric motor connected to a propeller. A battery serves as a secondary source of electrical power on the aircraft. The TMS features three cooling loops: A ram air duct (also referred to as the ram air loop), a polyalphaolefin (PAO) loop, and a fuel loop assuming JP-8 as the fuel. The number of each type of component and their sizing parameters, both required and optional, are presented in the remainder of this section. Note that the minimum and maximum numbers of components set here govern all architectures generated in this study.

##### 1. Heat Loads

The heat loads for this study are summarized in Table 1 and are considered a good representation of the loads for this class of vehicle.

Name	Description	Location
Auxiliary	Lumped radar, flight controls, and housekeeping thermal waste	PAO loop
Battery	Propulsion system electrical power storage thermal waste	Ram air loop
Generator	Propulsion system electrical power generation thermal waste	Fuel loop
Motor	Propulsion system electric motor thermal waste	Fuel loop

##### 2. Heat Load Sizing

The sizing parameters for the heat loads are given in Table 2. There is only one option for each sizing parameter to simplify the study.

Heat Load	Design mass flow rate (kg/s)	Design pressure drop (Pa)
Auxiliary	1	2,000
Battery	0.1	200
Generator	3	2,000
Motor	3	2,000



### 3. Sources, Sinks, and Reservoirs

Each cooling loop must have at least one sink and one source, or a reservoir that serves as both. For the ram air loop, this comes in the form of the ram air source and ram air sink. The PAO and fuel loops must have one tank each to serve as reservoirs. These are default components and cannot be changed without internal modifications to the tool. The fuel loop has an additional sink to account for fuel consumption by the engine.

### 4. Fans and Pumps

The ram air loop must contain a fan to maintain at least a minimum prescribed air flow rate at points where little air flow is provided by the motion of the aircraft. Both the PAO and fuel loops must contain one pump each to induce flow. Pumps are always located just after the fluid tank exit port. The existence and location of the pumps are automatically handled by the tool.

### 5. Fan and Pump Sizing

The fan in this study has a minimum flow rate of 0.5 kg/s with an isentropic efficiency of 0.65 and a 0.98 mechanical efficiency value. Pumps were assigned a 0.85 isentropic efficiency value.

### 6. Heat Exchangers

The ram air loop must contain two heat exchangers to interface thermally with the fuel and PAO loops. The fuel loop and PAO loop must contain one heat exchanger each to interface with the ram air loop.

### 7. Heat Exchanger Sizing

Both sides of every heat exchanger were given a design mass flow rate and pressure drop of 1 kg/s and 1,000 Pa, respectively. Effectiveness was allowed to be either 0.7 or 0.8 for any given heat exchanger.

### 8. Three-port Nodes

The fuel loop must have at least one three-port node from which flow to the engine will be split off from the main flow. The flow was allowed to be split at any location downstream of the pump fluid exit. The fuel loop was also allowed to have two additional three-port nodes that must both be downstream of the pump fluid exit and upstream of the engine or tank inlet. No nodes were allowed in the PAO loop to simplify the study. The location and function of three-port nodes (i.e., whether they are used to split or join the flow) are handled automatically by the tool and simulation environment. The use of three-port nodes in the ram air loop is in development and not yet available.

### 9. Valves

The fuel loop may have, at most, one valve. An additional restriction to the study is that there may only be a valve present when the fuel loop has all three possible three-port nodes. This is because adding a flow restriction to a pump-driven loop with only a single flow path (neglecting the engine branch) would do nothing more than increase the pumping power required, which is undesirable.

### 10. Valve Sizing

The valve was sized with a nominal hydraulic conductance of 0.0003 (kg/s)/Pa.

## B. Optimization Points and Boundary Conditions

Optimization points were selected based on the notional mission profile in [10]. Takeoff and top of the initial climb were chosen as constraining points. End of loiter and the end of the final cruise segment were chosen as important points. Engine flow rates were determined by cross-referencing the thrust and specific fuel consumption graphs in [10]. Motor and Generator heat loads were calculated based on data from [10] and assuming 97.77% efficiency for both components. The battery loads were determined using a similar method assuming 95% efficiency. The auxiliary heat load was taken from the profile in [10]. Atmospheric conditions were determined assuming 1962 standard atmospheric conditions. Stagnation conditions were used for the ram air inlet.

### 1. Constraining Points

A summary of the boundary conditions for the constraining points is given in Table 3.

### 2. Important Points

A summary of the boundary conditions for the important points is given in Table 4 along with the point weighting factors used for the overall weighted objective function value calculations.

### 3. Objectives and Constraints

The objectives and constraints were determined based on experience with other aircraft. Table 5 summarizes the objectives and constraints. Note that the ram air flow rate is reported as a negative value due to Modelica conventions and internal variable selection. The overall objective function at a point is the weighted sum of the individual objective functions. The overall objective function value for an architecture is obtained from the weighted sum of the objective function at each important point.

Variable	Value at top of climb	Value at takeoff
Ram source pressure	25,489.6 Pa	103,170.1 Pa
Ram source temperature	223.01 K	289.63 K
Ram sink pressure	23,846.5 Pa	101,342.4 Pa
Ram sink temperature	218.81 K	288.15 K
Fuel tank pressure	23,846.5 Pa	101,342.4 Pa
Fuel tank temperature	303 K	303 K
Fuel burn rate	0.0522 kg/s	0.0454 kg/s
PAO tank pressure	23,846.5 Pa	101,342.4 Pa
PAO tank temperature	303 K	303 K
Auxiliary heat load	3,200 W	3,200 W
Battery heat load	5,000 W	12.5 kW
Generator heat load	12.23 kW	12.23 kW
Motor heat load	14.45 kW	17.78 kW

Variable	Value at end of loiter	Value at end of final cruise
Ram source pressure	34,781.6 Pa	25,489.6 Pa
Ram source temperature	236.32 K	223.01 K
Ram sink pressure	32,938.2 Pa	23,846.5 Pa
Ram sink temperature	232.68 K	218.81 K
Fuel tank pressure	32,938.2 Pa	23,846.5 Pa
Fuel tank temperature	303 K	303 K
Fuel burn rate	0.0246 kg/s	0.0164 kg/s
PAO tank pressure	32,938.2 Pa	23,846.5 Pa
PAO tank temperature	303 K	303 K
Auxiliary heat load	7,600 W	3,200 W
Battery heat load	0 W	2,500 W
Generator heat load	8,892 W	7,780 W
Motor heat load	7,780 W	7,780 W
Point Weighting	0.7	0.3

Variable (x)	Objective function f(x)	Minimum	Maximum	Weight
Auxiliary load temperature	$\begin{cases} \frac{x - 285.9 K}{1 K}, x \geq 285.9 K \\ 0, x < 285.9 K \end{cases}$	223 K*	320 K	10
Battery temperature	$\begin{cases} \frac{x - 313.7 K}{1 K}, x \geq 313.7 K \\ 0, x < 313.7 K \end{cases}$	200 K*	323.7 K	100
Generator temperature	Null	200 K*	338.7 K	0
Motor temperature	Null	200 K*	338.7 K	0
Ram air flow rate	$\begin{cases} -\frac{x}{1 kg}, x \leq 0 kg/s \\ 0, x > 0 kg/s \end{cases}$	$-\infty$	0	10
Fuel pump power	$\begin{cases} \frac{x}{1 W}, x \geq 0 W \\ 0, x < 0 W \end{cases}$	0	$\infty$	0.01
PAO pump power	$\begin{cases} \frac{x}{1 W}, x \geq 0 W \\ 0, x < 0 W \end{cases}$	0	$\infty$	0.01

\* Constraint imposed by fluid model limitations

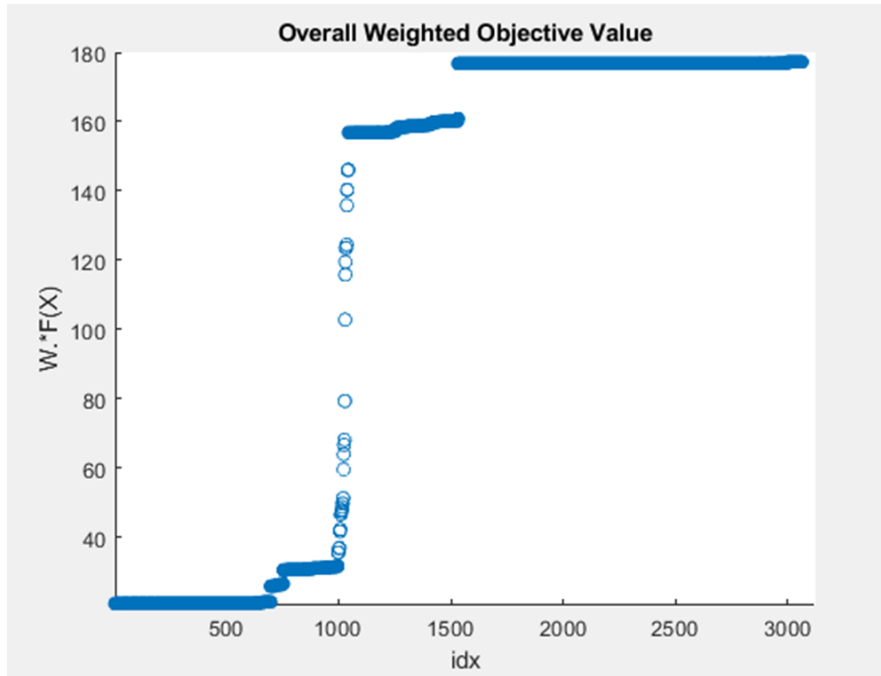
#### 4. Control Options

There are three controls available, which are the fuel and PAO pump flow rates and the fuel valve position. A summary of their possible values is given in Table 6. The fuel valve position is specified as an opening fraction and is only used when the valve is present in an architecture.

Control Variable (units)	List of Values
Fuel pump flow rate (kg/s)	1, 3, 6, 9, 12, 15
PAO pump flow rate (kg/s)	0.5, 1, 3, 6
Fuel valve position (n.d.)	0.2, 0.4, 0.6, 0.8, 1.0

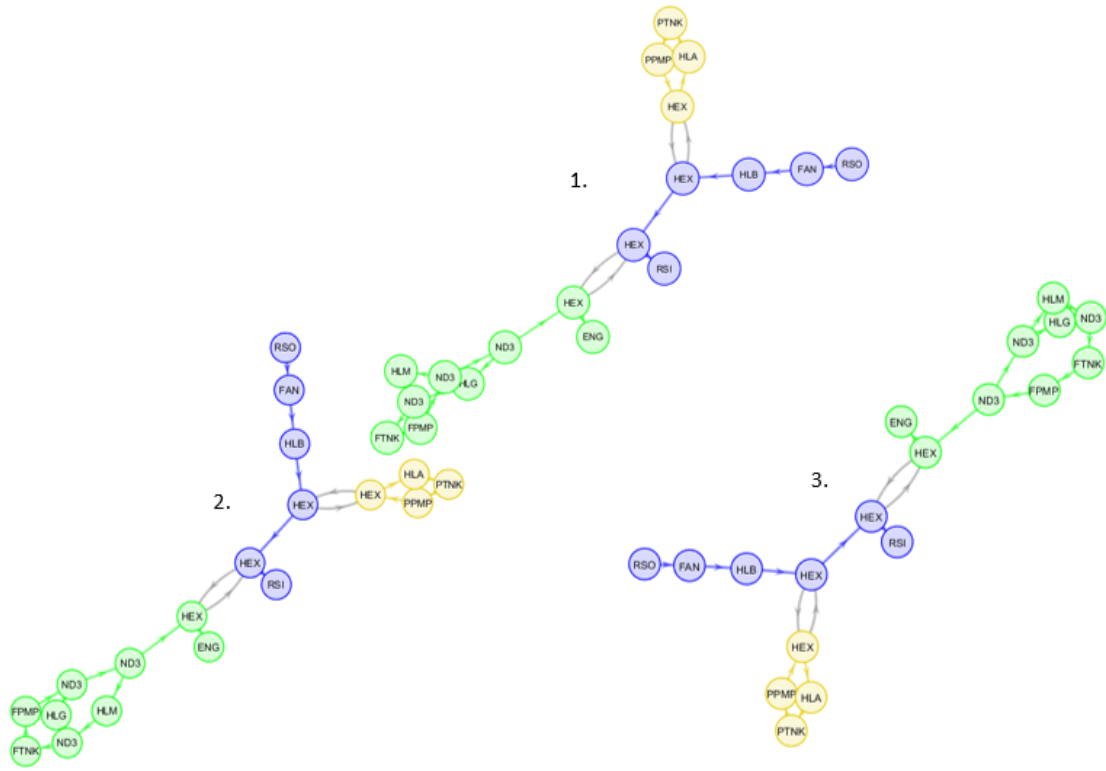
### C. Results and Discussion

The tool generated a total of 21,264 architectures within the design space specified by the study. Only 3,064 of these were capable of operating within the specified constraints for every specified set of boundary conditions. Fig. 10 shows the overall weighted objective function values (denoted  $W.*F(X)$  by the data visualization utility provided with the tool) for all successful architectures plotted in ascending order against an arbitrary index (denoted  $idx$  in the figure). The majority of the results are clustered near overall weighted values of 20, 30, 160, and 175 with a large number of architectures in the higher-valued clusters. Architectures with high objective function values would likely be poor choices for the system. An investigation as to the source of the clustering is beyond the scope of this demonstration, but it is likely due to architectural features that are common to each cluster. In a real-world use case, the user may wish to sample some of the architectures in each cluster to compare and contrast the various clusters and to determine if the objective functions are weighted correctly.

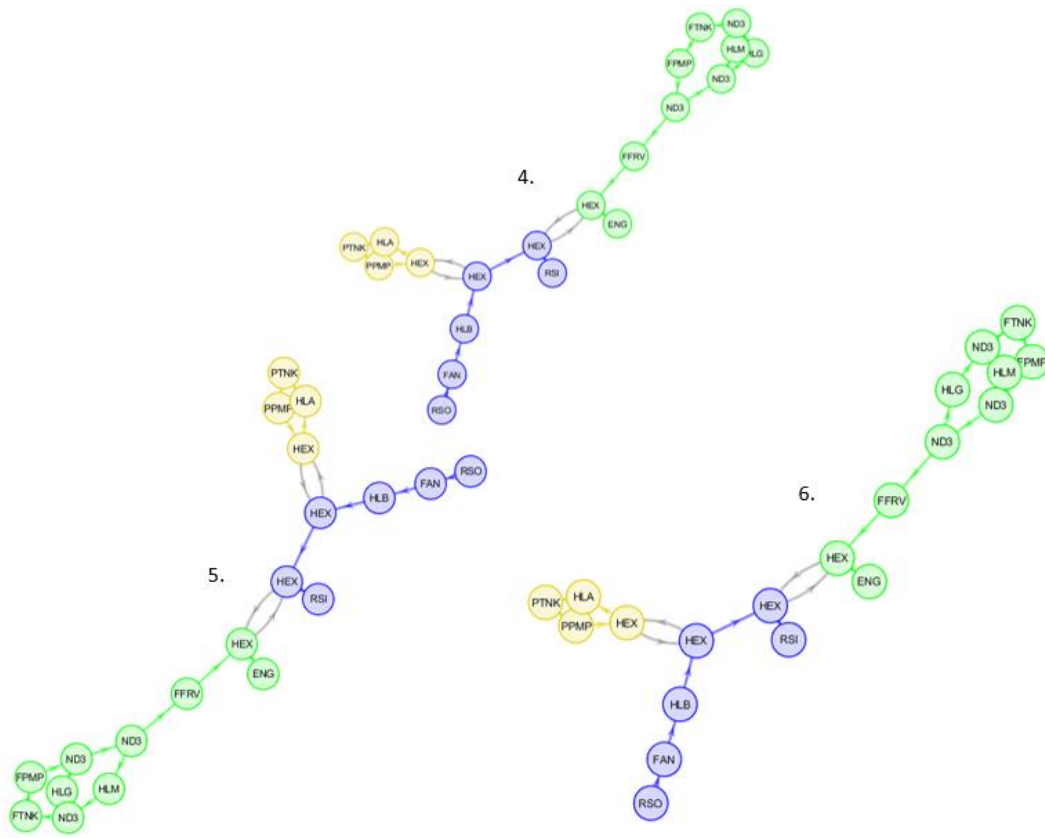


**Fig. 10 Overall weighted objective values for all successful architectures**

Typically, the user will only be interested in the architectures giving the lowest overall weighted objective values for more in-depth investigations. In this case, there are three architectures giving the lowest value and three architectures giving the second lowest value. These sets of architectures are shown in Fig. 11 Fig. 12, respectively.



**Fig. 11 Architectures with the lowest overall weighted objective value**



**Fig. 12 Architectures with second lowest overall weighted objective value**

The architecture representations in Fig 12 & 13 are graph representations created using a utility built into the tool. Color-coded connections with single arrows represent fluid connections with the arrow indicating flow direction. Gray double connections indicate purely thermal connections with energy potentially flowing either direction. Components and connections in green indicate that they are in the fuel loop, blue indicates the ram loop, and yellow indicates the PAO loop. The component identification symbols used all architecture representations in this paper are summarized in **Error! Reference source not found.**

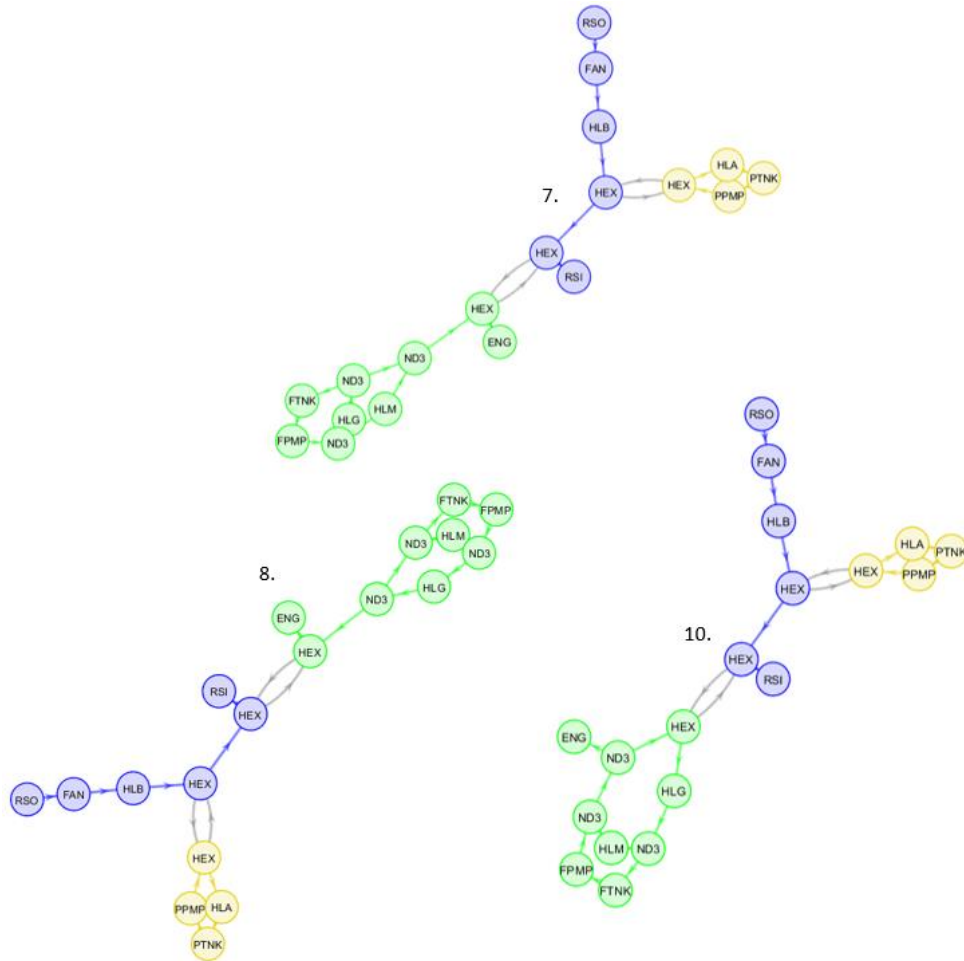
Closer inspection of Fig. 12 reveals that architectures 4, 5, and 6 have valves placed in the engine branch. The engine acts as an ideal flow rate sink, thus no valve is needed to regulate the flow of fuel and the valves in these cases are superfluous. For this reason, such architectures should be disregarded and future improvements to NSCs should prevent their generation. Discarding architectures with the bad valve placement feature, the remaining useful architectures in the group with the ten lowest objective values are shown in Fig. 13.

In Fig. 13, architecture 9 was deliberately left out since it contained the bad valve placement feature. Of the remaining architectures, 7 and 8 both had the third lowest objective value while the final architecture had the fourth lowest value.

Further investigation of all of the validity of the down selected architectures requires the use of higher-fidelity models and full-mission simulations. Such an in-depth study is beyond the scope of this study, however, some qualitative observations can be made. First, the PAO loops for all architectures are identical. This is not entirely surprising since only the order of the heat exchanger and auxiliary heat load can change in this loop, recalling that the pump and tank must be directly connected. Slightly more interesting is the fact that the ram air loops in most architectures are identical as well, including the order in which the PAO and fuel loops interface with the ram air loops through heat exchangers. It is also interesting to note that the heat exchanger in the fuel loop appears in the flow path going to the engine in all cases except for architecture 10. Additionally, all of the architectures make use of all three possible nodes in the fuel loop, indicating that a split between the generator and motor heat loads is optimal.

Note also that none of the valid architectures presented make use of the valve. This may be explained by the fact that the valve is a flow restriction which will cause the fuel pump power requirements to increase. This tradeoff does come with the benefit of better control of the system and it may be worthwhile to expand the number of architectures studied to include the best performing architectures that do use valves in valid placements. This may be especially important in the case that the architectures examined here do not allow for adequate control of the system over the flight envelope when examined with higher-fidelity tools.

Symbol	Component Name
ENG	Engine
FAN	Ram air fan
FPMP	Fuel pump
FTNK	Fuel tank
FFRV	Fuel valve
HEX	One side of a heat exchanger
HLA	Auxiliary heat load
HLB	Battery heat load
HLG	Generator heat load
HLM	Motor heat load
ND3	Three-port node
PPMP	PAO pump
PTNK	PAO tank
RSI	Ram air pressure source
RSO	Ram air pressure sink



**Fig. 13 Remaining valid architectures within the best ten**

A final point of interest are the parameter sets chosen for each of the architectures. All but architecture 10 use the same parameter set, using 0.7 for the effectiveness of one heat exchanger and 0.8 for the other. The user would need to investigate the Modelica models themselves to differentiate the two, but this process is omitted from the present paper for brevity. Architecture 10 uses 0.8 for the effectiveness of both heat exchangers.

#### IV. Conclusion

This paper introduces an automated graph theory-based aircraft TMS architecture design tool. Some details of the overall design procedure used were presented with emphasis on the multipoint optimization capability. A case study for an aircraft utilizing hybrid-electric propulsion with ram air, PAO, and fuel for cooling was presented to demonstrate the capabilities of the tool. The current results show that the tool can be effectively used to narrow down a large architecture design space to a few promising designs and identified some current limitations in the NSC implementation. To expand the design space to include additional TMS architectures, ongoing efforts include adding bleed loop regeneration and branching in both types of air loops. The introduction of closed-loop ACMs is planned for the near future followed by vapor cycle systems. Additionally, further improvements to NSCs are under investigation to narrow down the design spaces, thereby reducing the time and computational resources required to obtain a solution.

#### References

- [1] McCarthy, K., Walters, E., Heltzel, A., Elangovan, R. et al., "Dynamic Thermal Management System Modeling of a More Electric Aircraft," *SAE Technical Paper Series*, SAE International, 2008.

- doi: 10.4271/2008-01-2886
- [2] Walters, E., Amrhein, M., O'Connell, T. et al., "INVENT Modeling, Simulation, Analysis and Optimization," *AIAA 2020-287. 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. January 2010.  
doi: 10.2514/6.2010-287
- [3] Herber, D. R., Allison, J. T., Buettner, R., Abolmoali, P. and Patnaik, S. S., "Architecture Generation and Performance Evaluation of Aircraft Thermal Management Systems Through Graph-based Techniques," *AIAA Scitech 2020 Forum*. AIAA 2020-0159. January 2020.  
doi: 10.2514/6.2020-0159
- [4] McCarthy, P. T., McCarthy, K., Hasan, M., Boyd, M., Chang, M., Walters, E. and Niedbalski, N., "A Multi-Domain Component Based Modeling Toolset for Dynamic Integrated Power and Thermal System Modeling," *SAE Technical Paper 2019-01-1385*, 2019.  
doi:10.4271/2019-01-1385.
- [5] D. R. Herber, T. Guo, J. T. Allison. 'Enumeration of architectures with perfect matchings.' *ASME Journal of Mechanical Design*, 139(5), p. 051403, May 2017.  
doi: 10.1115/1.4036132
- [6] Herber, D. R., "Advances in Combined Architecture, Plant, and Control Design," Ph.D. thesis, University of Illinois at Urbana-Champaign, Urbana, IL, USA, Dec. 2017.
- [7] Rheaume, J. M. and Lents, C. E., "Design and Simulation of a Commercial Hybrid Electric Aircraft Thermal Management System," *2018 AIAA/IEEE Electric Aircraft Technologies Symposium*. AIAA 2018-4994. July 2018.  
doi: 10.2514/6.2018-4994
- [8] Kim, J. H., Kwon, K. S., Roy, S., Garcia, E. and Mavris, D. "Megawatt-class Turboelectric Distributed Propulsion, Power, and Thermal Systems for Aircraft," *AIAA Scitech 2018 Forum*. AIAA 2018-2024. January 2018.  
doi: 10.2514/6.2018-2024
- [9] Fritzon, P., and Bunus, P., "Modelica - A General Object-oriented Language for Continuous and Discrete-event System Modeling and Simulation," *Annual Simulation Symposium*, San Deigo, CA, USA, 2002.  
doi: 10.1109/simsym.2002.1000174
- [10] Abolmoali, P., Donovan, A., Patnaik, S. S., McCarthy, P., Dierker, D., Jones, N., Buettner, R., "Integrated Propulsive and Thermal Management System Design for Optimal Hybrid Electric Aircraft Performance," *AIAA Propulsion and Energy 2020 Forum*. AIAA 2020-3557. August 2020.  
doi: 10.2514/6.2020-3557