Reports

Machine Learning and Inference Laboratory

An Optimized Design of Finned-Tube Evaporators Using the Learnable Evolution Model

Piotr A. Domanski
David Yashar
Kenneth A. Kaufman
Ryszard S. Michalski

MLI 04-1
P 04-1
February, 2004

School of Computational Sciences

George Mason University

AN OPTIMIZED DESIGN OF FINNED-TUBE EVAPORATORS USING THE LEARNABLE EVOLUTION MODEL

Piotr A. Domanski[1], David Yashar[1], Kenneth A. Kaufman[2] and Ryszard S. Michalski[2,3]

(1) Building and Fire Research Laboratory, National Institute of Standards and Technology,
Gaithersburg, MD 20899-8631

(2) Machine Learning and Inference Laboratory, George Mason University,
Fairfax, VA 22030-4444

(3) Also affiliated with Institute of Computer Science of the Polish Academy of
Sciences, Warsaw.

{piotr.domanski, david.yashar}@nist.gov;{kaufman, michalski}@gmu.edu

http://www.mli.gmu.edu

Abstract

Optimizing the refrigerant circuitry for a finned-tube evaporator is a daunting task for traditional exhaustive search techniques due to the extremely large number of circuitry possibilities. For this reason, more intelligent search techniques are needed. This paper presents and evaluates a novel optimization system, called ISHED1 (Intelligent System for Heat Exchanger Design). This system uses a recently developed non-Darwinian evolutionary computation method to seek evaporator circuit designs that maximize the capacity of the evaporator under given technical and environmental constraints. Circuitries were developed for an evaporator with three depth rows of 12 tubes each, based on optimizing the performance with uniform and non-uniform airflow profiles. ISHED1 demonstrated the capability to design an optimized circuitry for a non-uniform air distribution so the capacity showed no degradation over the traditional balanced circuitry design working with a uniform airflow.

Keywords: machine learning, evolutionary computation, engineering design, learnable evolution model, multistrategy learning

Acknowledgments

The basic research that led to the development of the Learnable Evolution Model and the machine learning methodology employed in it was supported in part by the National Science Foundation under grants IIS-0097476 and IIS9906858. The National Institute of Standards and Technology (NIST), the Air Conditioning and Refrigeration Technology Institute, and the U.S. Department of Energy provided the funding the development of the EVAP-COND simulation package. The development of ISHED1 was supported by the NIST. V. Payne, G. Kelly, and J. Michaloski of NIST provided comments on the draft manuscript.

.

1 INTRODUCTION

Performance of a finned-tube air-to-refrigerant heat exchanger is affected by a multitude of factors related to its design and operation. These factors include the overall heat exchanger dimensions, the type of refrigerant-side and air-side heat transfer surfaces, fin spacing, tube pitch, depth row pitch, refrigerant circuitry design, and air velocity distribution over the frontal heat exchanger surface. Typically during coil design, the outside dimensions are dictated by the available installation space, and most of the remaining parameters are imposed on the design engineer by established manufacturing practices, e.g., heat transfer surfaces or tube spacing. Hence, in many cases the heat exchanger optimization process focuses on identifying refrigerant circuitry that provides the maximum heat transfer rate for given environmental constraints. In fact, refrigerant circuitry may have a significant effect on the evaporator capacity (Chwalowski et al. (1989), Liang et al. (2001)).

Designing an optimized refrigerant circuitry is particularly difficult if the airflow is not uniformly distributed over the coil surface. Also, optimizing refrigerant circuitry for a new refrigerant may prove to be difficult since the design experience gained from work with conventional refrigerants may not extend itself to a new refrigerant with different thermophysical properties. If left only to laboratory experiments, heat exchanger optimization is very expensive due to compounding costs of engineering analysis, manufacturing of coils with different circuitry designs (architectures), and their testing.

One way to aid the design effort is to use a detailed heat exchanger simulation model that accounts for the refrigerant circuitry layout; such as the evaporator model EVAP, contained in the EVAP-COND simulation package (NIST, 2003). Figure 1 shows EVAP's representation of a heat exchanger's refrigerant circuitry. In this representation, the return bends are represented by the lines, solid lines are return bends on the near side of the evaporator and the broken lines represent the return bends on the far side. In addition, to aid in the visualization, each tube is given a tube number.

The model allows the user to specify refrigerant flow through the heat exchanger on a tube-by-tube basis. The user may perform a series of simulations for the best-guessed circuitry architectures. EVAP provides detailed simulation results for individual tubes, e.g. refrigerant temperature, pressure and quality. These results can guide the user to the optimal design, which can be validated later in a laboratory test.

The coil optimization process can be further upgraded if the optimization program replaces the design engineer in preparing candidate circuitry architectures. This paper describes a concept of such an automated scheme as implemented by an experimental program ISHED1 (Kaufman and Michalski, 2000), and presents examples of ISHED1's results.

2 OVERVIEW OF ISHED1

Figure 2 presents a general diagram of the ISHED1 system. It includes the Control Module, the evaporator model EVAP, and two modules: the Knowledge-based Evolutionary Computation Module and the Symbolic Learning-based Evolutionary Computation Module.

These two modules guide the evolutionary process according to the concept referred to as the *Learnable Evolution Model*, or *LEM* (Michalski, 2000).

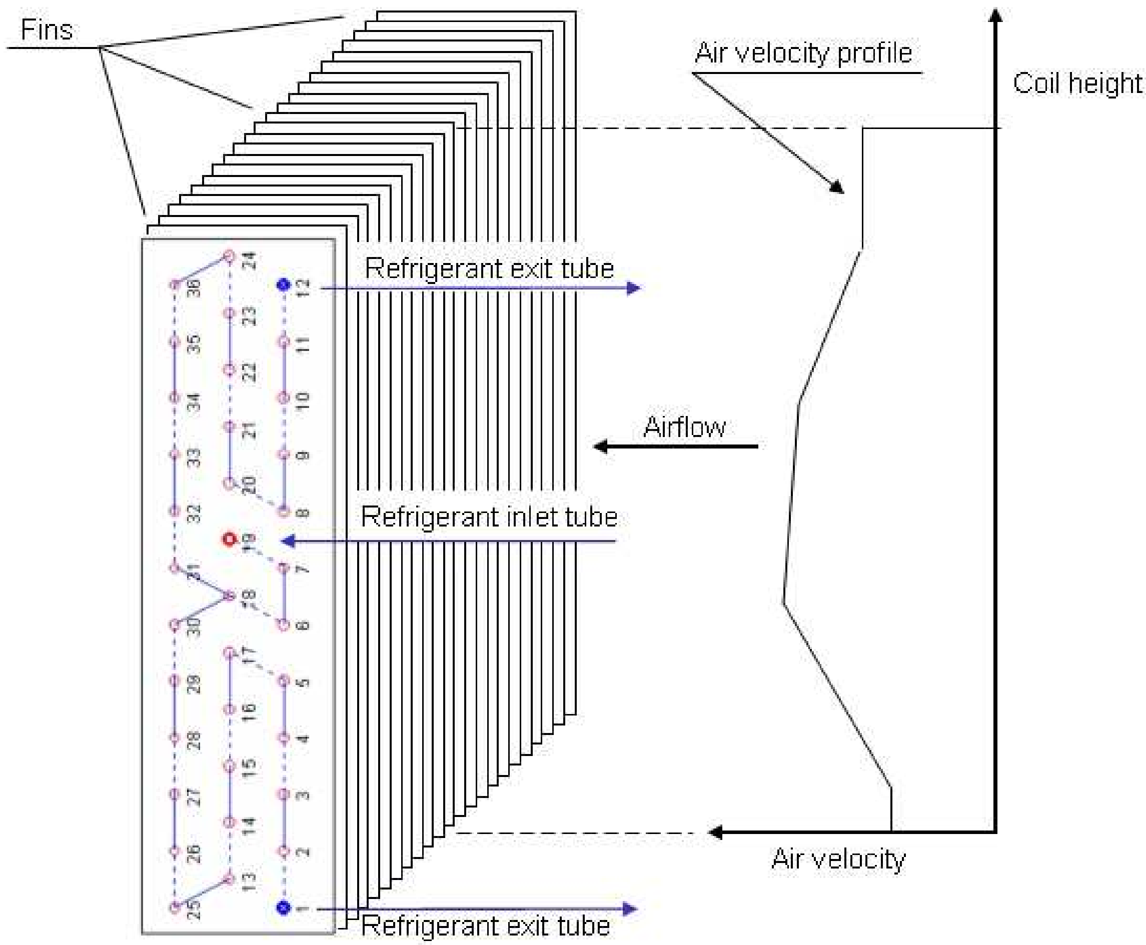


Figure 1. EVAP's representation of an evaporator

The novelty of LEM methodology is that it combines a conventional evolution program (Michalewicz, 1994) with a non-Darwinian evolutionary computation employing symbolic learning. An evolution program uses Darwinian-type operators, mutations and/or recombinations to generate new individuals (Goldberg 1989). In the ISHED1 knowledge-based module, these operators are not random, as in conventional genetic algorithms, but domain knowledge-based, i.e., they only perform changes that are deemed suitable according to the domain-knowledge. The symbolic learning method generates new individuals (designs) in an entirely different way, by hypothesis formation and instantiation (Michalski, 2000). These two distinct methods of generating new individuals are integrated in ISHED1.

Consistent with a conventional evolutionary computation approach, ISHED1 operates on one generation (population) of designs at a time. A population consists of a given number (determined by the user) of circuitry designs. Each member of the population is evaluated by EVAP, which simulates their performance and provides their cooling capacity as a single numerical fitness value. The designs and their fitness values are returned to the Control Module as an input for deriving the next generation of circuitry designs. Hence, the implemented process is a loop, and it is repeated for the number of generations specified by the user at the outset of the optimization run.

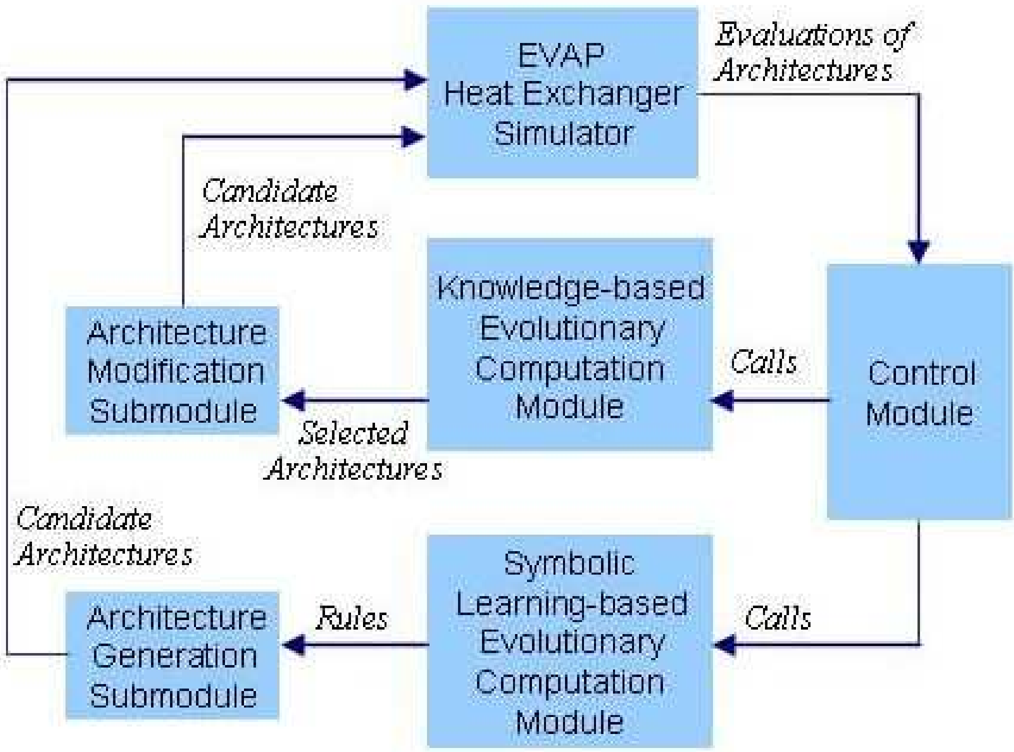


Figure 2. A functional architecture of ISHED1.

The Control Module determines which of the two modules, the Knowledge-based Evolutionary Computation Module or the Symbolic Learning Module, is utilized to produce the next population. At the outset of an optimization run, the Control Module applies Darwinian evolution until the population no longer improves (both in terms of the best individual and the population overall). It then switches to symbolic learning until the performance under that module ceases to improve. The Control Module alternates between the two models until a specified number of iterations have been completed.

2.1 Knowledge-based Evolutionary Computation Module

The optimization process using a Knowledge-based Evolutionary Learning Module follows the three-step pattern that is implemented by a conventional evolution program (Michalewicz, 1994): 1) a selection of individual designs for the next generation, with selection probability proportional to their evaluated fitness values (evaporator capacities in our case); 2) modification of the selected designs by structure modifying operators; and 3) evaluation of the new population, member by member, to acquire their fitness values (simulations using the evaporator model to obtain the capacity of the proposed designs).

Because of the constraints on feasible refrigerant circuitry, traditional genetic operators (replication, crossover, and mutation) would be, for the most part, unworkable for the problem at hand; therefore eight domain knowledge-based structure modifying operators were developed and implemented. They are: (1) SPLIT, creating a split point and two refrigerant paths starting at a given split point, (2) BREAK, creating, from one refrigerant path, two full paths from the input to the output, (3) COMBINE, taking two paths and creating one branch splitting into two, (4) INSERT, taking two paths and inserting one into the other at some break point, (5) MOVE-SPLIT, moving the existing split point an even number of tubes upstream or downstream, (6) SWAP, reversing the order of two adjacent tubes in a flow structure, (7) INTERCROSS, swapping two consecutive tubes between two individual circuits that are not upstream of one another; and (8) NEW-SOURCE, assign a new feeding tube for a randomly selected tube (to avoid loops, the new feeding tube can not be downstream of the breaking point). The system probabilistically selects an operator to apply based on the topology of the heat exchanger. If it seems that the operator will not lead to a feasible change in the circuitry, another operator is tried.

2.2 Symbolic Learning-based Evolutionary Module

When applied, the Symbolic Learning Module divides the members of the current population into three classes based on their fitness values (cooling capacity); "good" class, "bad" class, and "indifferent" class. The "good" and "bad" classes contain members of the population whose fitnesses are in the top and bottom 25 % of the current generation's fitness range, respectively. Then, the module examines the characteristics of both well- and poorly performing designs, and creates hypotheses in the form of attributional rules that characterize the better-performing architectures. These rules are applied to generate a subsequent population of designs. During consecutive generations, rules are used in the context of their predecessors, so as to further focus the concept of design optimality.

2.3 Evaporator Simulation Model

The evaporator simulation model used in this study, EVAP, is a component of the simulation package EVAP-COND developed to facilitate preparing optimized designs of finned-tube evaporators and condensers (NIST, 2003). EVAP uses a tube-by-tube modeling scheme. That is, the program recognizes each tube as a separate entity for which it calculates heat transfer. These calculations are based on inlet refrigerant and air parameters, properties, and mass flow rates. The simulation begins with the inlet refrigerant tubes and proceeds to successive tubes along the refrigerant path. At the outset of the simulation, the air temperature is only known for the tubes in the first row and has to be estimated for the remaining tubes. A successful run requires several passes (iterations) through the refrigerant circuitry, each time updating inlet air and refrigerant parameters for each tube.

The tube-by-tube modeling approach used by EVAP makes it suitable for use within the ISHED1 scheme. This modeling approach is important for both heat transfer and refrigerant pressure drop calculations. Consequently, it is also essential for simulations of refrigerant distribution in different circuitry architectures because refrigerant distribution is affected by pressure drops in individual refrigerant tubes and circuitry branches. When calculating the total pressure drop, EVAP includes the pressure drop in return bends, whose lengths are determined based on the relative location of the connected tubes.

EVAP can account for one-dimensional air maldistribution, as it is conceptually shown in Figure 1. This feature allows optimizing refrigerant circuitries for installations with complicated air velocity profiles. Additional information on EVAP is available at the EVAP-COND website (NIST, 2003). Validation of EVAP is presented in Domanski and Payne (2002) and Payne and Domanski (2003).

2.4 ISHED1 System Operation

An optimization run starts by reading a file with the control parameters for the run. This file contains the basic geometric characteristics of the heat exchanger and the heat exchanger operating condition information. The read control parameters, which override defaults when read, are as follows:

- Parameters defining the characteristics of the initial population: its size and any user-defined first population individual design
- Parameters defining the length of the evolutionary process
- Parameters controlling the optimization run, including the persistence of the knowledge-based and symbolic modes, and the level of detail to be presented in the output file.
- Parameters defining the general dimensions of the heat exchanger
- Parameters defining the airflow distribution over the front face of the heat exchanger

ISHED1 allows the user to define individual architectures in the first population; or if the user does not define them, the system will generate the initial set randomly. It is also possible for the user to define only a portion of the initial population, in which case the system randomly generates the remaining designs.

During preparation of new architectures, ISHED1 applies experience-based knowledge to constrain the search to plausible architectures. These constraints are ranked from "suggested" to "essential". The program rejects structures that violate a required constraint, and only under special circumstances (namely when designing a more compliant architecture is very difficult) will accept structures that violate the most lenient constraints. The constraints include a user-defined parameter (or its default value) which imposes limitations on the length of return bend directing the refrigerant to the subsequent tube. Another constraint states that exit tubes should not have inlet tubes as their neighbors, but rather they should be located next to the tubes that feed them with refrigerant. The intent of this constraint is to limit internal tube-to-tube heat transfer which occurs via heat transfer through common fins between neighboring tubes if they are at different temperatures. Since the exit tubes typically have superheated refrigerant and are warmer than the tubes with two-phase refrigerant, it is preferable to have them surrounded in the coil assembly by tubes with somewhat superheated refrigerant as well. A similar constraint suggests that the exit tube should be in the first depth row. This constraint reflects the recognition that the overall heat transfer is most effective if semi-counterflow configuration is established between the temperature profiles of refrigerant and incoming air.

To summarize, for a given set of operating conditions, general evaporator geometry information, and user-specified run control parameters, ISHED1 performs an optimization process involving two distinct modes: the knowledge-based evolutionary mode and the symbolic learning mode. The Control Module decides which mode to apply at any decision time during the run. The optimization process involves evaluating sequential design populations whose size and number are predefined by the user. When the run is completed, ISHED1 produces a report with the best architectures and their capacities, as determined by the evaporator model. It should be stressed that except for the pre-coded experience-based design constraints and the evaporator model itself, no other components of ISHED1 have any recognition of the physical processes taking place in a heat exchanger. Simply, the system is concerned with a single numerical fitness value (cooling capacity) obtained by each architecture, and manipulates strings representing the refrigerant flow path through the evaporator with the goal of maximizing the coil capacity.

3 COIL DESIGN EXPERIMENTS

We confronted ISHED1 with the task of designing refrigerant circuitry for a 36 tube R-22 evaporator consisting of three-depth rows with 12 tubes located in each depth row. Figure 3 shows the evaporator's detailed design information as it is displayed by the EVAP-COND graphical user interface. The design operating point was defined by the condenser subcooling and evaporator superheat of 5.0 °C, and the condenser bubble point and evaporator exit saturation temperature of 40 °C and 7.2 °C respectively. The inlet air was at 101.325 kPa pressure and 26.7 ºC dry-bulb temperature with 50 % relative humidity. The volumetric flow of air was 15.0 m^3 per minute.

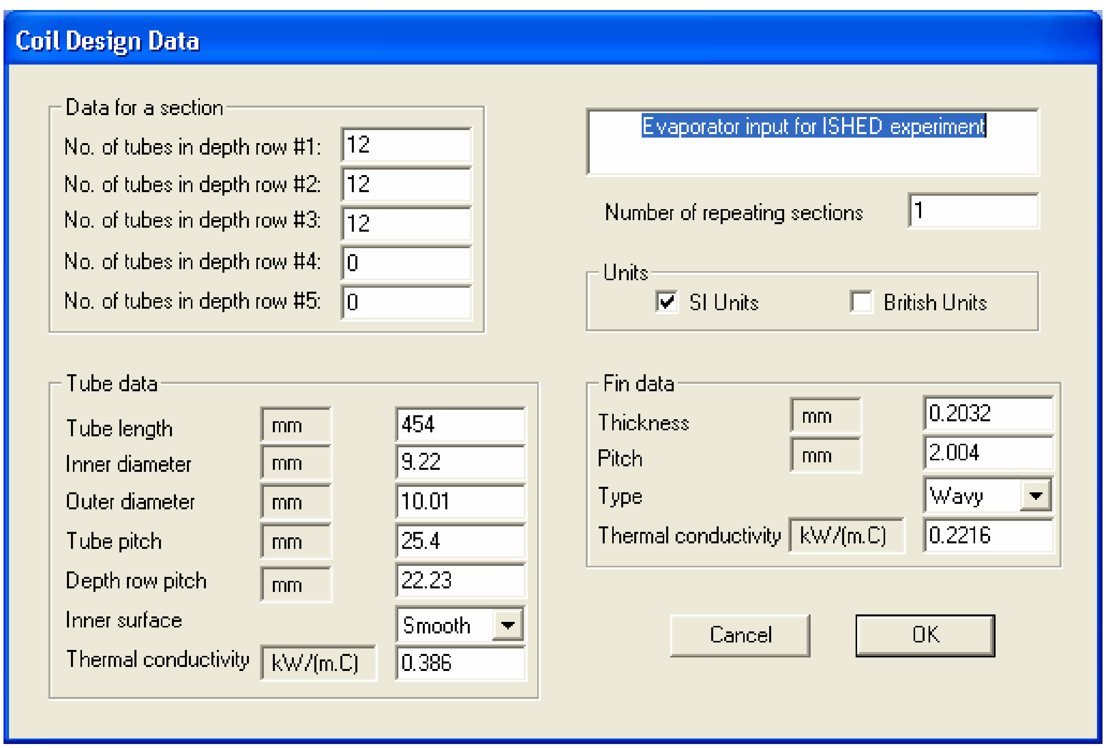


Figure 3. Evaporator Design Information.

In all simulations, the population size was set to 15 architectures (members) and the number of generations was set to 300. The ISHED1 defaults were used for all of the other parameters. When starting each optimization run, we did not specify any initial circuitry architectures; all of the architectures in the initial population were generated by ISHED1. We performed simulations for uniform and non-uniform air distributions. Since ISHED1 is not equipped with a windows-based interface, we used EVAP-COND user's interface to display ISHED1-generated circuitry designs.

3.1 Simulations with uniform air distribution

Figure 4 presents the circuitry developed for a uniform velocity profile of the incoming air. The inlet tubes are denoted by partially open circles (1 and 11), and the outlet tubes are denoted by closed circles (5 and 9). It takes only a quick look at the design to notice that the proposed circuitry is difficult and expensive to manufacture. Clearly, ISHED1 does not have the intelligence to recognize and avoid manufacturing difficulties. However, we have to realize that the proposed design gives us valuable information which we can use to produce a good, manufacturable design.

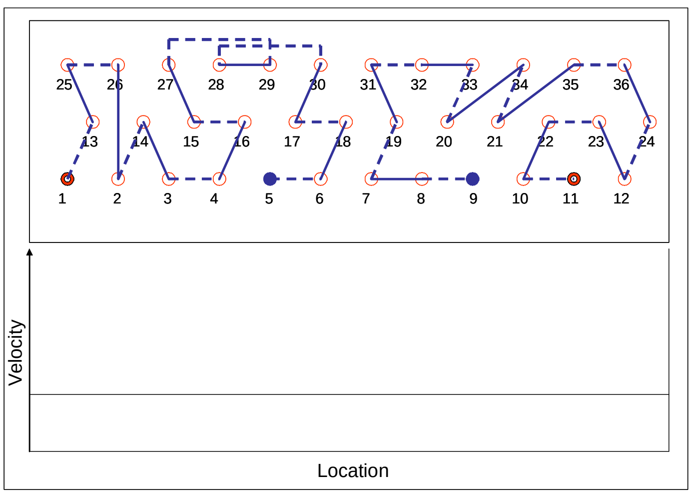
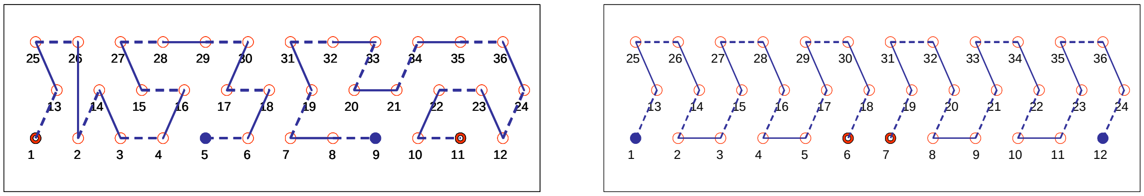


Figure 4. ISHED1-generated refrigerant circuitry for uniform air distribution:
Capacity Q = 5.25 kW

Based on the architecture proposed by ISHED1, we generated two different circuitry designs, shown in Figure 5. The first design was a "cleaned" version of the ISHED1 design; we eliminated the over-lapping long return bend and made a few minor alterations. The return bends were not modified, and the inlet and outlet tubes remained the same. In the second design, the two inlets and outlets were the only commonality with the ISHED1 design. These examples show that the more we depart from the ISHED1-recommended design, the lower capacity of the evaporator is, as determined by the evaporator model. However, we may debate in this case that the capacity degradation is not that significant and a manufacturer might select the simplest design for production.



Capacity Q = 5.18 kW

Capacity Q = 5.12 kW

Figure 5. Two modified circuitry designs for uniform air distribution

3.2 Simulations with non-uniform air distribution

For the purpose of this experimentation, we devised a simple non-uniform velocity profile for the inlet air. For this profile, the left half of the evaporator was subject to a uniform flow of air, while the right half was subject to a linear profile with the maximum velocity being twice that of the minimum. The total volumetric flow of air was 15.0 m^3 per minute (the same as for the previous examples with uniform air distribution). Figure 6 shows the non-uniform velocity profile and the refrigerant circuitry recommended by ISHED1. The obtained capacity is close, or actually slightly higher, than that obtained with the uniform velocity profile by the ISHED1-recommended architecture (5.35 kW vs. 5.25 kW). We should note that the design generated by ISHED1 for the uniform air velocity profile (shown in Figure 4) had a capacity of only 4.82 kW when simulated with this non-uniform velocity profile.

It is also interesting to note that ISHED1 has the ability to evolve to a design which is principally different from the one proposed for uniform air. The design based on a uniform airflow profile consists of two inlets and two outlets; with each individual circuit primarily being located in its own portion of the heat exchanger. This design would be a poor choice for a highly non-uniform air profile because the available airflow would be very different for each of the circuits. The design based on this non-uniform airflow (Figure 6, below), on the other hand, consists of one inlet, two outlets, with a single split occurring at tube #33. Also, each branch of the circuitry in this design tends to span across the heat exchanger, making it less sensitive to air side maldistributions.

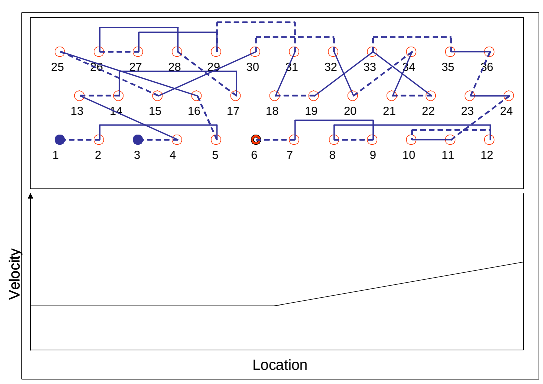
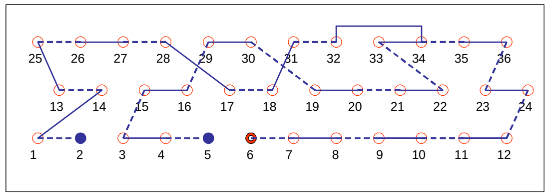


Figure 6. ISHED1-generated refrigerant circuitry for non-uniform air distribution: Capacity Q = 5.35 kW

Considering that the ISHED1-generated circuitry arrangement is not easy to manufacture, we modified it to arrive with a more practical design, by eliminating over-lapping and long return bends. The modified design of this architecture is shown in Figure 7. The capacity loss due to the modification was minimal (5.34 vs. 5.35 for the ISHED1-recommended architecture).



Capacity Q = 5.34 kW

Figure 7. Modified circuitry designs for non-uniform air distribution

We also simulated performance of the circuitry design shown in Figure 6 with a uniform velocity profile. At this operating condition, the obtained capacity was 5.34 kW. Hence, the circuitry design that was optimized for the non-uniform air distribution showed robust performance with both uniform and non-uniform air distributions. In fact, the architecture developed for this case performed slightly better for the uniform air distribution than the one shown in Figure 4. This is interesting and should be noted although no additional experiments were performed with other non-uniform velocity profiles to give this observation more generality.

4 GENERAL OBSERVATIONS

To understand the scope of performed optimization runs, let us reiterate that each of the runs involved simulations for 15 circuitry designs (architectures) of each of the 300 populations. Hence, a completed optimization run encompassed simulations of 4500 evaporators. This may seem like a large number of test cases, however, it is a very small portion of possible circuitry designs. According to simple calculations based on the number of tubes and possible path attributes (splits, multiple paths, etc.), this 36-tube test case has approximately $2 \cdot 10^{45}$ possible architectures.

A valid question arises as to what combination of the population size and number of populations would make ISHED1 most effective. There is no simple response to this question because the answer will depend on the evaporator size, i.e., number of tubes in the evaporator. Simply, the larger the heat exchanger, the more different circuit arrangements can be devised, and more simulations should be performed. In any case, a practical consideration requires that an optimization run is completed within the interest span of the design engineer. This may imply completion of the run in 15 hours; that is to be able to start the run before leaving the office for home on one day and having the optimization results ready the next morning with a PC computer running overnight. In our case of an evaporator with 36 tubes, an optimization run was typically completed within four hours using a computer with a 1.7 GHz microprocessor, which yields an average simulation time for a single architecture of 3.2 seconds. The amount needed to optimize a design for larger architectures will be longer and may require an overnight run.

The number of populations used will also depend on the progress ISHED1 makes during the optimization process. The progress for our example optimizations can be reviewed in Figure 8, which presents capacity progression of the best architectures in each population for both cases studied. For the uniform air distribution case, the first population already included a design with a reasonably high cooling capacity. ISHED1 improved over this original design somehow, but the improvement shown is not dramatic. For the case with non-uniform air distribution, the initial capacity was low, and ISHED1 made gradual improvements. The figure shows that most capacity improvements were obtained in steps, which, in most cases indicate the instances when the Control Module switched between the two available optimization modes (the evolutionary and symbolic learning modes). It appears that switching to the other mode "shook up" the population and allowed for improved architectures with capacities exceeding those developed for the uniform air velocity profile.

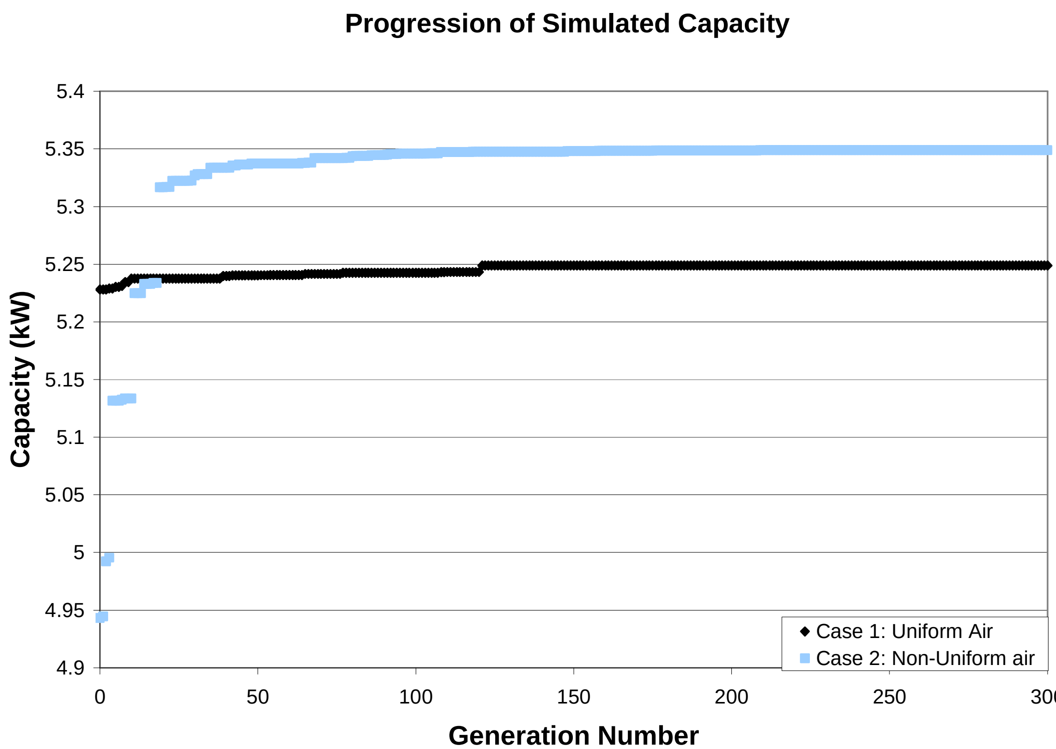


Figure 8. Capacity progression for optimizations run for uniform and non-uniform air distributions

While discussing the obtained results we should emphasize that the ISHED1 scheme, as well as any other evolutionary optimization method, will not repeat optimization results from one optimization run to another for the same environmental constraints. This is due to the randomness that is inherently embedded in these schemes. This is in contrast to traditional calculus-based methods, which provide the same result, time after time, if the initial conditions have not changed. When running ISHED1 several times, we would obtain several different architectures, which, in most cases, would provide similar capacity. As a real-life analogy, we may think about different people who can perform a given task equally well although they may use somewhat different methods. Also, as in real life situations, we have no guarantees as to the optimization outcome when using ISHED1 or other genetic algorithms-based methods due to the randomized operators employed. However, we have clear evidence that ISHED1 is able to generate optimized designs, which in some cases would be difficult to formulate for a design engineer.

5 CONCLUSIONS

We described an experimental system, ISHED1, developed to assist a design engineer in optimizing finned-tube evaporators. Specifically, given input parameters and technical constraints, the system optimizes the refrigerant circuitry in the evaporator. The novelty of this approach is in applying the recently developed Learnable Evolution Model (LEM), which integrates Knowledge-based Evolutionary Computation with Symbolic Learning that guides the process of generating new designs. Generated designs are evaluated using the EVAP evaporator model, which simulates the designs.

ISHED1 can be applied to evaporator design for different air-conditioning and refrigeration applications as it is not constrained by the refrigerant used or heat transfer surfaces other than the simulation limitations of EVAP.

ISHED1 has only a few experience-based design principles incorporated in its code. It carries out the optimization process using randomized operators whose implementation may in effect resemble “out of the box” thinking. Experimentation with ISHED1 has demonstrated that it is capable of generating designs equal or superior to the best human designs, particularly in cases of non-uniform airflow. Circuitry architectures generated by ISHED1 require some manual adaptation to assure their manufacturability.

The system is not oriented toward displacing a design engineer, but aims at offering to him/her useful guidance, particularly for designing heat exchangers with new overall geometries, heat transfer surfaces, and refrigerants. The methodology underlying ISHED1 is general and could potentially be used for other engineering design problems.

REFERENCES

Domanski, P.A., Payne W. V., 2002. "Properties and Cycle Performance of Refrigerant Blends Operating Near and Above the Refrigerant Critical Point", report to ARTI, project ARTI-21CR/605-50010-02, http://www.arti-21cr.org/research/completed/index.html.

Goldberg, D.E., 1989. "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison Wesley Longman, Inc.

Kaufman K. and Michalski R.S., 2000. "Applying Learnable Evolution Model to Heat Exchanger Design," *Proceedings of the Seventeenth National Conference on Artificial Intelligence* (AAAI-2000) *and the Twelfth Annual Conference on Innovative Applications of Artificial Intelligence* (IAAI-2000), Austin, TX, pp. 1014-1019.

Michalewicz, Z.,1994, *Genetic algorithms + Data structures = Evolution Programs*, 2nd extended edition, Springer-Verlag

Michalski, R., 2003. "Learnable Evolutionary Model", US. Patent No. 6,518,988, February 11.

Michalski, R., 2000. "Learnable Evolution Model: Evolutionary Process Guided by Machine Learning. *Machine Learning* 38, pp. 9-40.

NIST, 2003. "EVAP-COND, Simulation models for finned-tube heat exchangers". National Institute of Standards and Technology, Gaithersburg, MD, http://www2.bfrl.nist.gov/software/evap-cond/

Payne W. V., Domanski, P.A., 2003. "Potential Benefits of Smart Refrigerant Distributors", report to ARTI, project ARTI-21CR/605-200-50-01, http://www.arti-21cr.org/research/completed/index.html.

A publication of the *Machine Learning and Inference Laboratory*
School of Computational Sciences
George Mason University
Fairfax, VA 22030-4444 U.S.A.
http://www.mli.gmu.edu

Editor: R. S. Michalski
Assistant Editor: K. A. Kaufman

The *Machine Learning and Inference (MLI) Laboratory Reports* are an official publication of the Machine Learning and Inference Laboratory, which has been published continuously since 1971 by R.S. Michalski's research group (until 1987, while the group was at the University of Illinois, they were called ISG (Intelligent Systems Group) Reports, or were part of the Department of Computer Science Reports).