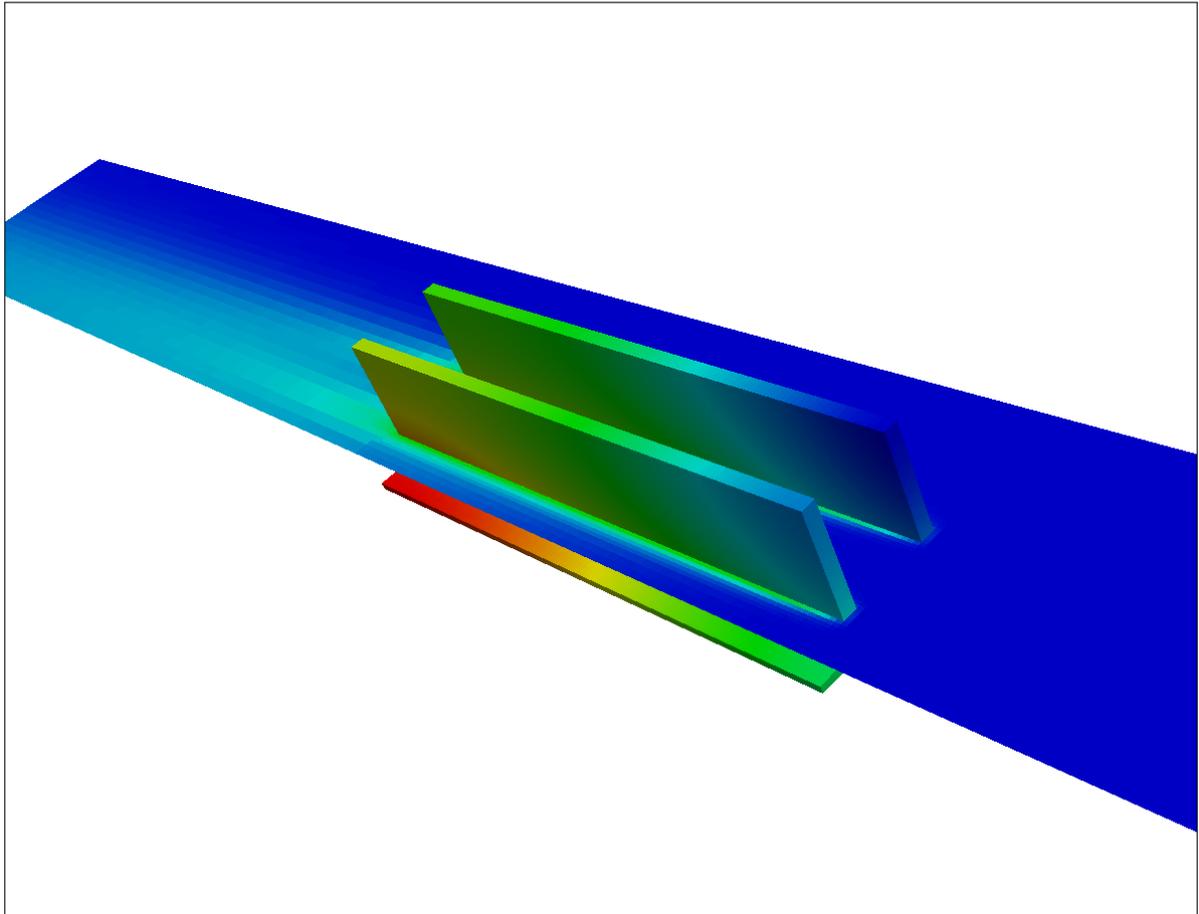




CHALMERS
UNIVERSITY OF TECHNOLOGY



Geometrical Optimization of Plate-Fin Heat-Sink

Master's thesis in Engineering Mathematics and Computational Science

DANI IRAWAN

MASTER'S THESIS 2016:NN

Geometrical Optimization of Plate-Fin Heat-Sink

DANI IRAWAN



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Applied Mathematics
Engineering Mathematics and Computational Science
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016

Geometrical Optimization of Plate-Fin Heat-Sink
DANI IRAWAN

© Dani Irawan, 2016.

Supervisor: Tommy Andersson, Fraunhofer-Chalmers Centre
Supervisor: Andreas Mark, Fraunhofer-Chalmers Centre
Examiner : Srdjan Sasic, Department of Fluid Dynamics

Master's Thesis 2016:NN
Engineering Mathematics and Computational Science
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Visualization of temperature distribution on a heat-sink and the air flowing through it.

Typeset in L^AT_EX
Printed by [Name of printing company]
Gothenburg, Sweden 2016

Geometrical Optimization of Plate-Fin Heat-Sink
Dani Irawan
Engineering Mathematics and Computational Science
Chalmers University of Technology

Abstract

In this thesis the geometrical configuration of a processor heat sink is optimized. Two objective functions, operational cost and maximum temperature, have been chosen and the design space has been limited to three degrees of freedom represented by number of fins, fin height and fin thickness.

The optimization is simulation-based and carried out using multi-objective particle swarm optimization (MO-PSO). Two variants of MO-PSO have been selected, adapted and implemented. The first is a variant introduced by Coello and Lechuga in "MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization" which is based on density of the pareto front. The second is a variant introduced by Fieldsend and Singh in "A Multi-Objective Algorithm Based Upon Particle Swarm Optimisation, an Efficient Data Structure and Turbulence" which is based on distance to the pareto front. Simulations have been performed using a state-of-the-art immersed boundary flow solver called IBOFlow developed by Fraunhofer-Chalmers research centre.

Comparisons presented in this thesis show that FS-method requires less evaluations to find the Pareto front, whereas the CL-method explores the Pareto front more evenly. The number of particles does not seem to give an apparent effect on exploration it is rather the distribution of initial points that determines the exploration. Many evaluated points have been observed to be clustered in objective space, especially in FS-method. This observation triggered the idea of a filter.

In the current work it has been found that the number of simulations can be significantly reduced by using filters. However, the threshold used in the filter must be chosen conservatively to prevent coarsening of the Pareto front. It has also been found that the Pareto front can be refined by projection of infeasible points onto the boundary of the feasibility region. The improvement from projection comes at the expense of additional evaluations.

Optimal designs of the heat sink are proposed based on lexicographic method, Analytical Hierarchy Process, and Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), respectively.

Keywords: CFD, multi-objective optimization, PSO, IBOFlow, conjugated heat transfer.

Acknowledgements

First of all, I would like to thank Allah S.W.T for His blessings and for giving me health and strength to finish this project. May His blessings and peace be upon the messenger of Allah, Muhammad S.A.W, his family and companions.

The work was done at the Fraunhofer-Chalmers Research Centre for Industrial Mathematics, FCC. I would like to express my gratitude and appreciation to my supervisor, Tommy Andersson and Andreas Mark, for the opportunity to work on this project and for their guidance, help, suggestions, and all support during the work of this project. I would also like to thank all employee of FCC, especially to Anders Ålund for all kindness and cooperation during my stay in this centre.

Finally, I wish to thank my family for their support and encouragement throughout my study.

Dani Irawan, Gothenburg, June 2016

Contents

List of Figures	xi
List of Tables	xiii
1 Introduction	1
1.1 Background	1
1.2 Objective	2
1.3 Scope	2
2 Fundamental Theories of Computational Fluid Dynamics and Heat Transfer	5
2.1 Navier-Stokes Equations	6
2.1.1 Incompressible Navier-Stokes Equations	6
2.1.2 Boussinesq Approximation for Buoyancy	6
2.2 Conjugate Heat Transfer	7
3 Multi-Objective Optimization	9
3.1 Pareto-front	10
3.2 <i>A Priori</i> Method	10
3.2.1 Lexicographic Method	10
3.2.2 Objective Value Scalarization	13
3.3 <i>A Posteriori</i> Method	14
4 Methods	15
4.1 Computational Fluid Dynamics	15
4.1.1 Boundary Conforming Mesh vs Immersed Boundary	16
4.1.2 Turbulence Modelling	17
4.2 Particle Swarm Optimization	17
4.2.1 Basic Particle Swarm Optimization Algorithm	19
4.2.2 Multi-Objective PSO (MO-PSO)	20
4.2.2.1 Coello and Lechuga’s Approach	20
4.2.2.2 Fieldsen and Singh’s Approach	21
5 Geometrical Optimization of Plate Fin Heat Sink	23
5.1 Optimization Problem	25
5.1.1 Objectives	25
5.1.2 Variables	26

5.1.3	Constraints	26
5.1.4	Complete Problem	27
5.1.5	Optimization Configurations	27
5.1.6	Design of Experiment	29
5.2	Results and Discussion	30
5.2.1	Pareto-front	31
5.2.2	Pareto-set	32
5.3	Post-processing	34
5.3.1	Design Space Filter	34
5.3.2	Exploring the Boundaries	39
5.3.3	Choosing the Best Design	44
5.3.3.1	Lexicographic Method	44
5.3.3.2	Analytical Hierarchy Process (AHP)	46
5.3.3.3	Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)	47
6	Conclusion	51

List of Figures

3.1	Objective function and objective space	11
3.2	Dominance	12
3.3	Pareto-set	12
4.1	Boundary conforming mesh	17
4.2	Immersed boundary	18
5.1	Illustration of a plate-fin heat-sink. Air will flow between the plates and cool down the plates. The plates are typically made of metals such as copper or aluminium.	23
5.2	Simulation setup	24
5.3	The objective space has two axes. Each axis corresponds to an objective function.	25
5.4	The design space is three-dimensional. Each axis corresponds to a design variable.	26
5.5	Feasibility cuts due to constraints	28
5.6	Feasibility contour for different N	28
5.7	Temperature distribution from a simulation	30
5.8	Velocity profile of a simulation	30
5.9	Explored objective space	33
5.10	Explored design space, FS-method 3 particles	35
5.11	Explored design space, FS-method 12 particles	36
5.12	Explored design space, CL-method 3 particles	37
5.13	Explored design space, CL-method 12 particles	38
5.14	Effect of filtering	40
5.15	Projection-evaluation on Pareto-front of configuration 3	43
5.16	Projection-evaluation on Pareto-front of configuration 4	44
5.17	Pareto-optimum solutions from all simulations conducted. The line represent the operating temperature limit 449K.	46
5.18	Ideal and negative-ideal solution without normalization and equal weights	47
5.19	Effect of weights on the Pareto-front	49
5.20	Best designs	50

List of Tables

5.1	Constraints due to bonding fabrication-method	26
5.2	Constraints due to design specification	27
5.3	Symbols and definitions for equation 5.1	27
5.4	MO-PSO configurations	29
5.5	Maximum speed for each axis in design space	29
5.6	Filter test configurations	41
5.7	Compounded Pareto-optimum solutions	45
5.8	Best design with respect to relative weight of T_{max} using AHP and TOPSIS	49

1

Introduction

Processors and electrical components generate heat by joule heating (heating due to passage of electric current), therefore their temperatures are ever increasing when they are working. A high temperature is feared to affect the hardware as an overheated component will wear out and the damages are usually permanent [31; 39]. Some components may even melt if the temperature is too high. Although some semiconductors can withstand 300° C, the packaging, solder materials, reliability considerations, and some other factors limit the operating temperature at approximately 175° C [22].

To confront the heating problem, a heat sink system is applied. For processors, the heat sink system employed usually in the form of placing solids with high heat conductance near or in contact with the processor to draw heat from the processor and then use flowing fluids to cool down the solids. Further in this thesis, heat exchange between solids and fluids is termed as conjugate heat transfer.

1.1 Background

Heat sink design have been evolving and there are many different types of heat sinks available currently: from a simple fin array heat sink to a complex staggered pin heat sink. These different designs affect the heat sinks performance due to different flow profile that passes through the heat sink elements. Different flow profile induces different heat transfer coefficient between the heated solids and the cooling fluid.

Heat sink designs are usually constrained by their fabrications method. A heat sink made by extrusion can have very thin fins but they must be linear. Heat sink made by stamping or die casting on the other hand cannot be as thin as extrusion-made heat sink but a complex shape can be easily made. The materials that can be used to fabricate is also constrained by the fabrication method. This material choice in turn will also limit the heat sink's thermal conductivity. An example of this limitation is the die casting method which requires that the material is a metal. Comparison of the different methods is available in [25] and [9].

Fluids flowing through the different designs of heat sinks will have different

flow profile. For example, the pin heat sink may induce larger pressure drop (i.e. pressure gradient) compared to a plate fin heat sink due to repeated expansion [21]. The different designs will also give different temperature profiles which in turn will affect buoyancy.

Another important factor regarding the flow is turbulence. Turbulence usually credited for increasing the heat transfer rate. In heat sinks, we need to consider two things regarding turbulence: the length of the heat sink along the direction of the flow; and the space between the fins. Air flowing at 5m/s above a plate would need a trailing length of approximately 1.68m to become turbulent, relatively large compared to the typical dimensions of processor heat sinks, however the spacing between fins only need to be at least 3.5mm to be turbulent [17].

In this thesis, plate fin heat sinks are simulated to evaluate their performance. The simulations are conducted using a finite volume method with an immersed boundary at the location of the solids.

1.2 Objective

The objective of this master thesis is to suggest an optimal heat-sink design in terms of thermal-performance and operating-cost. The thesis is also aimed to test and evaluate the usage of some optimization methods in finding the optimal heat-sink design.

The optimization is based on simulations which in this case implies that simulation results are treated as output from a black box function characterizing the performance of the heat sink system. The heat sink geometry is modified following a multi-objective particle swarm optimization (MO-PSO) algorithm.

1.3 Scope

In this thesis, constraints due to the fabrication methods, design limitations as well as constraints arising from simulation limitations are considered. The optimization is conducted in an *a posteriori* approach. Some possible optimization methods are reviewed thoroughly in paper by Reyes-Sierra and Coello[33] (MO-PSO) and master thesis by Rudholm and Wojciechowski [36] (surrogate-assisted optimization).

In simulation based problems, the function being solved does not contain information regarding its gradient/derivative thus gradient-based methods are not applicable.

One of the design variable in the case considered in this thesis is discrete, therefore surrogate-assisted methods which rely on the continuity of the problem[41]

are also not applicable.

The remaining option is to consider a stochastic algorithm e.g. MO-PSO or multi-objective genetic algorithm (MO-GA). The MO-PSO algorithm is chosen because it is superior compared to MO-GA [15; 37]. Another reason for choosing MO-PSO rather than MO-GA is that GA is normally applied to problems with large number variables but we are only considering 3 degrees of freedom[45].

Two variants of MO-PSO are used, the first method was developed by Coello and Lechuga[8] and the second was developed by Fieldsend and Singh [14]. The two methods are chosen due to their simplicity. Also, Fieldsend and Singh developed their method after Coello and Lechuga and claimed their new method is better, thus it would be interesting to compare them in this optimization problem. The performance of the methods are compared based on the density of Pareto front, the Pareto front shape and location, and the size of the area explored in the design space.

The optimization algorithms will give a Pareto front as the result. The best design will then be chosen from the Pareto front by doing post-Pareto analysis. There are many methods for post-Pareto analysis and three of them are implemented in this work.

The outline for this thesis is as follows:

- Chapter 2 describes the fundamental theory of fluid dynamics and conjugate heat transfer,
- Chapter 3 describes multi-objective optimization problems and general methods to solve them,
- Chapter 4 describes the simulations and optimization methods used in the thesis,
- Chapter 5 presents the main results of the work, discussion, and *a posteriori* analysis of the work,
- Chapter 6 summarize the main conclusions of the work.

2

Fundamental Theories of Computational Fluid Dynamics and Heat Transfer

The simulations conducted in this thesis involve fluid flow and heat transfer in both fluids and solids. The first section of this chapter describes the physics in the flow using the assumption that the fluid is Newtonian and incompressible. The second section of this chapter is about heat transfer. Means of heat transfer and the heat transfer between solids-fluids are discussed in this part.

Equations used in both sections are transport equations which describe how a quantity is transferred from one place to another. Generally, the equation consists of transient term, diffusion term, convection term, and external source/sink [44],

$$\underbrace{\frac{\partial \rho \phi}{\partial t}}_{\text{transient term}} + \underbrace{\nabla \cdot (\rho \mathbf{v} \phi)}_{\text{convection term}} = \underbrace{\nabla \cdot (\Gamma \nabla \phi)}_{\text{diffusion term}} - \underbrace{S_\phi}_{\text{ext. source/sink}}, \quad (2.1)$$

ρ is the mass density, ϕ is the quantity of interest (e.g. energy, mass, momentum, etc.), \mathbf{v} is velocity of the element¹, Γ is the diffusion rate, and S_ϕ is the source (positive value) or sink (negative value).

The transient term appear when the system is changing with respect to time. Convection term appear due to the velocity of the element \mathbf{v} , this term is apparent in fluid flow where the quantities are carried away with the fluid following the flow. Diffusion happens when there exist difference of the quantities at different points in the element, i.e. when there are gradients. Generally a quantity of an element will diffuse to neighboring element with lower value, e.g. an element with high concentration will transport (diffuse) its mass to another element with lower concentration thus increasing the latter's concentration.

¹element refers to a very small part of the fluids/solids which carry the quantity of interest

2.1 Navier-Stokes Equations

Navier-Stokes equations can be derived from the conservation of mass ($\phi = 1$ in equation 2.1) and momentum ($\phi = \mathbf{v}$ in equation 2.1). The Navier-Stokes equations are

$$\rho \frac{Dv_i}{Dt} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial v_k}{\partial x_k} \delta_{ij} \right] + \rho f_i, \quad (2.2)$$

$$\frac{D\rho}{Dt} + \rho \frac{\partial v_i}{\partial x_i} = 0. \quad (2.3)$$

where ρ is the fluid density, v is the fluid velocity, P is the pressure, μ is the kinematic viscosity, f_i is the body force per unit mass, and t is time.[10]

2.1.1 Incompressible Navier-Stokes Equations

In this thesis, the fluid is considered to be incompressible. In incompressible fluid, the density ρ is constant with respect to pressure, but this would also imply that the density variation with respect to time is very small and thus negligible, i.e. $\frac{D\rho}{Dt} = 0$, therefore:

$$\rho \frac{\partial v_i}{\partial x_i} = 0, \quad (2.4)$$

ρ cannot be zero, therefore

$$\frac{\partial v_i}{\partial x_i} = 0. \quad (2.5)$$

Substituting equation 2.5 to 2.2 and using $\rho \frac{Dv_i}{Dt} = \rho \frac{\partial v_i}{\partial t} + \rho v_j \frac{\partial v_i}{\partial x_j}$ yields:

$$\rho \frac{\partial v_i}{\partial t} + \rho v_j \frac{\partial v_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] + \rho f_i. \quad (2.6)$$

2.1.2 Boussinesq Approximation for Buoyancy

For the cases considered in this thesis, the flow is subjected to a temperature difference which may cause natural convection. In equation 2.6, this effect is not stated explicitly. The temperature gradients affect the flow by means of buoyancy force

[43]. There is no clear guideline on when we should consider including buoyancy to the equation, but it is generally accepted that its effect on heat transfer decreases to negligible level when the air speed exceeds 1.5-2 m/s[25].

To be able to model the effect of temperature gradients, buoyancy force need to be added to equation 2.6. To model buoyancy, Boussinesq approximation for buoyancy is used(notice that there are other approximation named "Boussinesq approximation", e.g. in turbulence modelling and water waves). In Boussinesq approximation, the only property variation to consider in the fluid is its density variation, other properties variations are completely ignored [43]. Variations in density are also ignored except where the term is affected by gravity, so gravitational force is included as body force in equation 2.6, yielding

$$\rho \frac{\partial v_i}{\partial t} + \rho v_j \frac{\partial v_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} [\mu (\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i})] + \rho g. \quad (2.7)$$

Variation in density need to be considered only in the last term, ρ in the first term is replaced into a constant ρ_0 , while in the last term into $\rho = \rho_0 + \Delta\rho$. $\Delta\rho = -\alpha\rho_0\Delta T$ is added here to address the variation giving the equation

$$\rho_0 \frac{\partial v_i}{\partial t} + \rho_0 v_j \frac{\partial v_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} [\mu (\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i})] + (\rho_0 - \alpha\rho_0\Delta T)g. \quad (2.8)$$

Often, to avoid potential round-off errors from the buoyancy term, a "pressure shift" is employed[1],

$$-\frac{\partial p}{\partial x_i} + (\rho_0 - \alpha\rho_0\Delta T)g = -\frac{\partial P}{\partial x_i} - \alpha\rho_0\Delta Tg, \quad (2.9)$$

with $P = p + \rho_0gh$, and h is the elevation. The Navier-Stokes equation with Boussinesq approximation can then be written as

$$\rho_0 \frac{\partial v_i}{\partial t} + \rho_0 v_j \frac{\partial v_i}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} [\mu (\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i})] - \alpha\rho_0\Delta Tg. \quad (2.10)$$

2.2 Conjugate Heat Transfer

Heat transfer between fluids and solids is termed conjugate heat transfer. Heat transfer in fluids arises from convective and diffusive heat transfer, while in solids no convective term is used. This situation means that the boundary between solids and fluids need to be treated carefully to correctly model the heat transfer. A simple method to model the heat transfer would be by using an overall heat transfer

coefficient, but a more precise method would be to address local heat transfer at the boundary by using the transport equation for temperature.

The transport equation for temperature are derived from transport equation for internal energy (using $\phi = u$ in equation 2.1, u is the internal energy) and Fourier's law[10]. The transport equation for temperature is

$$\rho c_p \frac{\partial T}{\partial t} + \nabla \cdot (\rho c_p \mathbf{v} T) = \nabla \cdot (\Gamma \nabla T) - S_T. \quad (2.11)$$

For equation 2.11, Γ is the thermal conductivity $k = \frac{\alpha}{\rho c_p}$, with α is the thermal diffusivity. It should be noted that in solids, the convection term is zero.

To use equation 2.11 in a heat exchanger system, it is needed to solve the equation for both the solid and fluid element. The problem is how to define the boundary, i.e. where should the properties of the fluid be used and where should properties of the solid be used instead. The methods to tackle this problem are discussed in chapter 4.

3

Multi-Objective Optimization

Optimization is a process to bring somethings (referred as objective values) to its best state, i.e. maximum or minimum[5]. Often the system is limited by some conditions, known as constraints. The general form of an optimization problem is

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && f(\mathbf{x}), \\ & \text{subject to} && g_i(\mathbf{x}) \leq b_i, \quad i = 1, \dots, k. \end{aligned} \tag{3.1}$$

Here, x is the decision variable, $f(x)$ is the objective function, $g_i(x)$ are constraints, b_i are constants, and k is the number of constraints.

An optimization problem with several objective functions often occurs, for example, maximizing capacity of a facility while minimizing the cost to build it, therefore resulting 2 objective functions, one for capacity, the other one for cost. In reality, there can be more than 2 objective functions. Such problem will result in a vector of objective values, each representing its own objective function:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && F = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x})), \\ & \text{subject to} && g_i(\mathbf{x}) \leq b_i, \quad i = 1, \dots, k. \end{aligned} \tag{3.2}$$

By having multiple design parameters and multiple objective functions, we get 2 spaces of decisions. First, we have design space with dimension m , as the number of input/controllable variables. The other is objective space where each vector in design space is mapped to an objective vector. The objective space has dimension n , the number of objective functions.

Often, the objective functions are in conflict with each other, i.e. an optimum for one objective function, is not the optimum for other objective functions. To handle conflicting objective functions, designers' preference is needed to specify the notion of optimality, i.e. a trade-off (balance) between the objective functions. The results can vary depending on the designers' preference; thus we have several possible optimal values called the Pareto-optimum values or Pareto-front. The designers' preference can be imposed before (*a priori*) or after (*a posteriori*) the optimization process.

3.1 Pareto-front

Pareto-optimum vectors are defined as vectors in the design space where improvement of one of its corresponding objective values can only be achieved by the worsening of at least one other objective [3; 32].

The mapping of such vectors into objective space will create a frontier which is non-dominated [3]. Assuming a minimization problem, a vector F^1 dominates F^2 if and only if:

$$\begin{aligned} F_i^1 &\leq F_i^2 \quad \forall i \in 1, \dots, n, \\ F_i^1 &< F_i^2 \quad \text{for at least one } i. \end{aligned} \tag{3.3}$$

Pareto-optimum vectors form a set called the Pareto-optimal set or Pareto-set. Graphical representations of the Pareto-front are presented in figure 3.1, 3.2, and 3.3

3.2 *A Priori* Method

In a priori methods, the designers' preference are inputted before the process started. The result from the optimization is a single optimal vector in the objective space. This can be done, for example, by sorting the objective function based on their importance or applying weights to the objective functions.

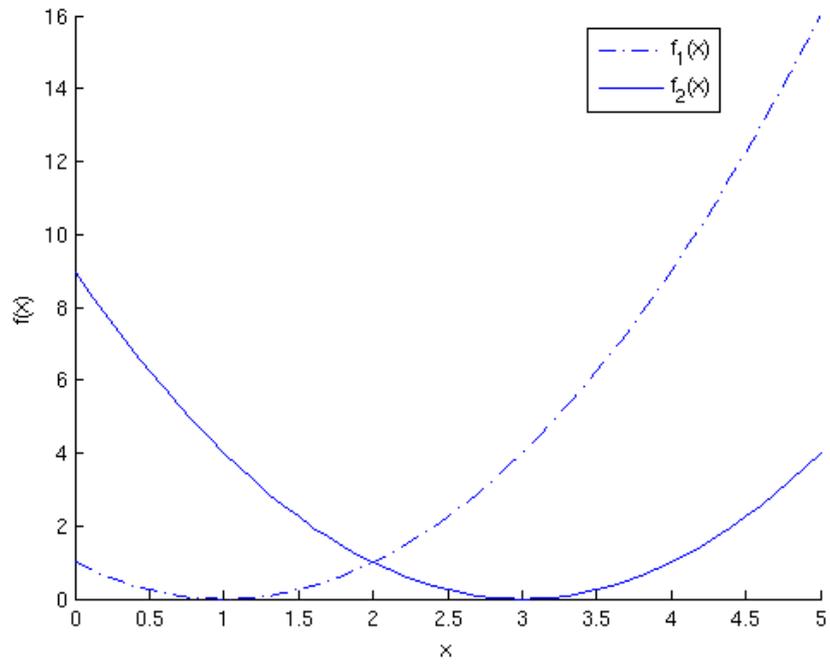
3.2.1 Lexicographic Method

The basis of lexicographic method is sorting the objective functions by its importance [13]. The method is similar with the process of sorting words in dictionaries:

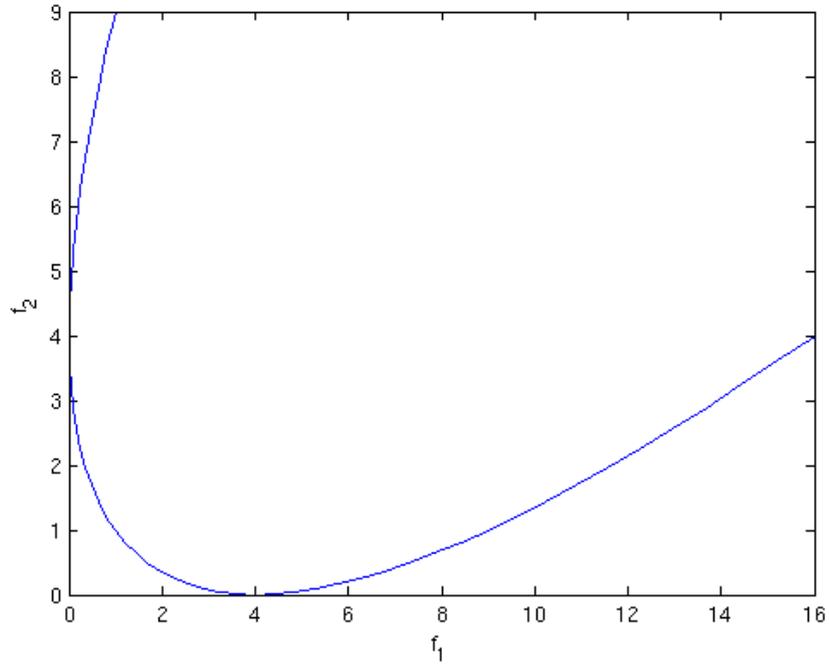
- sort by the first letter
- if the first letter is same, then sort by the second letter
- continue to the next letters until all items have different ranks or all letters in the word are used.

The lexicographic method in multi-objective optimization does exactly the same (analogously) procedure:

- do a single objective optimization on the most important objective function, if it is unique or unbounded then stop, we have an optimum or the problem is unbounded



(a)



(b)

Figure 3.1: (a) Two objective functions $f_1 = (x - 1)^2$ and $f_2 = (x - 3)^2$. (b) the objective functions drawn in the objective space.

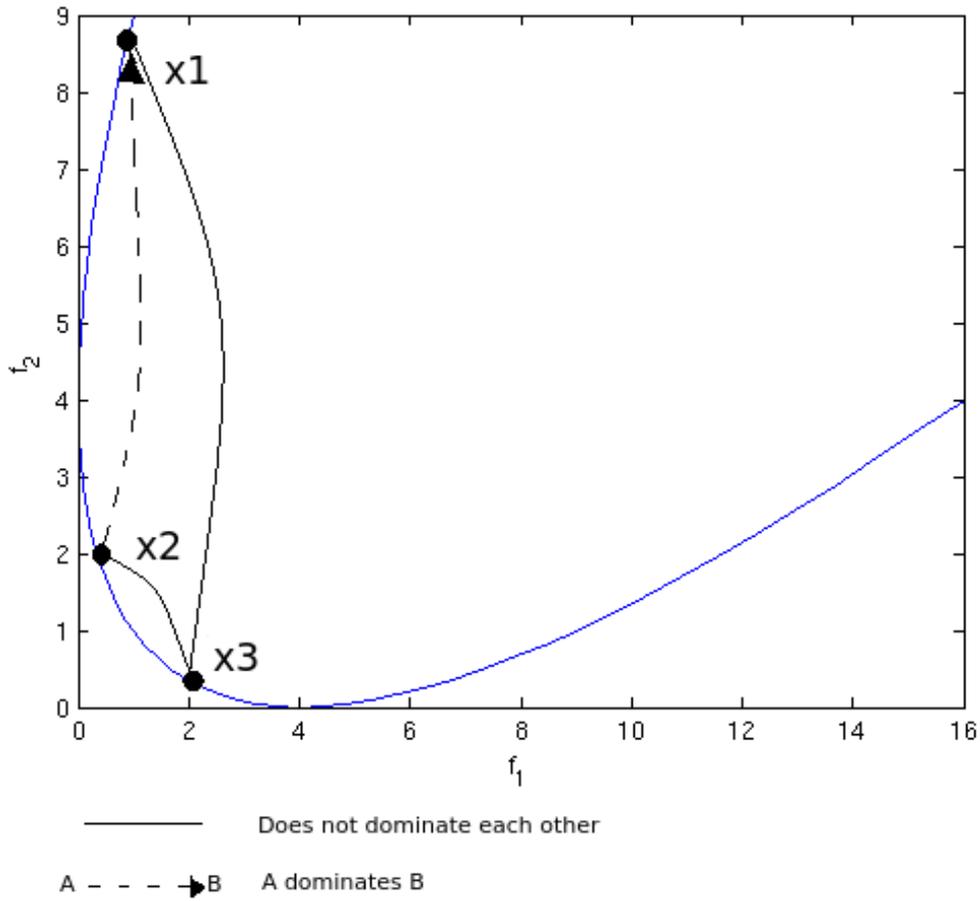


Figure 3.2: The concept of dominance. We see that $f_1(\mathbf{x}_1) < f_1(\mathbf{x}_3)$ but $f_2(\mathbf{x}_1) > f_2(\mathbf{x}_3)$ therefore \mathbf{x}_1 does not dominates \mathbf{x}_3 and vice versa. This is also the case for the pair \mathbf{x}_2 and \mathbf{x}_3 . However, $f_1(\mathbf{x}_1) < f_1(\mathbf{x}_2)$ and $f_2(\mathbf{x}_1) < f_2(\mathbf{x}_2)$ therefore \mathbf{x}_2 dominates \mathbf{x}_1 and \mathbf{x}_1 does not belong to the Pareto-front.

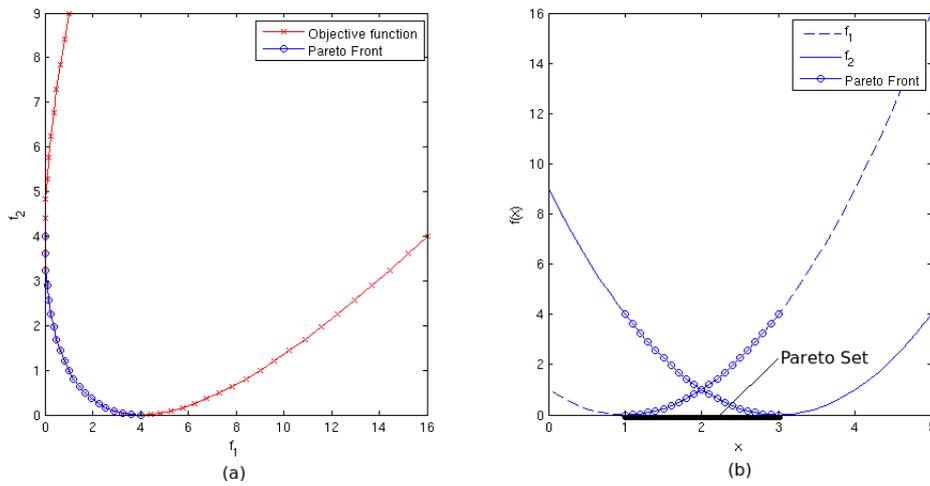


Figure 3.3: The Pareto-front formed from all non dominated points in the objective space is shown in (a). The corresponding Pareto-set is shown in (b).

- if the value of the most important objective function is not unique, i.e. several vectors in design space mapped to the same value, then from this list of vectors optimize the second most important objective value, if it is unique or unbounded then stop, we have an optimum or the problem is unbounded
- continue to the next objective functions until an optimum found or the problem is found unbounded or all objective functions have been evaluated.

This method is especially useful when the objective functions have very clear priority ordering. This method does not seek balance, it seeks instead the best for one objective function at a time.

3.2.2 Objective Value Scalarization

One way to impose preference is by element-wise multiplication (scalar product) between the objective vector with a weight vector (hence it is also called as the weighted sum method).

$$\begin{aligned}
 \bar{F}(\mathbf{x}) &= \mathbf{f}(\mathbf{x}) \cdot \mathbf{z} \\
 z_i &> 0 \quad i = 1, \dots, n \\
 \sum_{i=1}^n w_i &= 1
 \end{aligned} \tag{3.4}$$

Using this method, the problem is reduced to a single objective optimization and we can use methods for optimizing single objective optimization problems.

$$\begin{aligned}
 &\underset{\mathbf{x}}{\text{minimize}} \quad \bar{F}(\mathbf{x}) \\
 &\text{subject to} \quad g_i(\mathbf{x}) \leq b_i, \quad i = 1, \dots, k.
 \end{aligned} \tag{3.5}$$

The problem with scalarization is scaling [3; 11]. Often objective functions and design variables have very different scale, for example in this thesis we have several length scales:

- length of the whole simulation area can be in the order of 10^0m
- width and length of the heat sink in the order of 10^{-2}m
- thickness of the fins in the order of 10^{-4}m

Scaling becomes a critical problem because the algorithms to solve the problem usually are not scale-invariant [11].

3.3 *A Posteriori* Method

A posteriori methods aim to produce the Pareto-front before setting any preference, then the designer pick a choice from the Pareto-front later.

The easiest way to obtain the Pareto-front is by repeating the objective value scalarization method with different weights. This method however has a drawback that the objective vectors found are not well distributed and objective vectors in non-convex region are missed entirely. The problems can be mitigated by applying a modified version of the method called the Adaptive Weighted Sum (AWS) method [4; 23]. Another drawback is that the each optimization run will only yield a single Pareto-optimum point.

Another possible method is using multi-objective evolutionary algorithms such multi-objective genetic algorithm (MOGA) or multi-objective particle swarm optimization (MOPSO)[6]. The main advantage of these methods is that they can find multiple Pareto-optimal vectors in a single run. The methods also do not require any knowledge of derivatives of the objective functions, thus they are especially useful when we are dealing with "black-boxes" where we do not know how the system will response to changes in the design, e.g. the results of simulations. However, as these methods usually require a lot of evaluation vectors, they become inefficient when each evaluation takes very large resource, i.e. when they are very long to evaluate. These methods are initially designed for unconstrained optimization, thus it is necessary to transform the optimization problem into an unconstrained problem by using penalty functions or skipping the non-feasible points.

A computationally more efficient method is to use surrogate models to predict the response of the black-box function to changes in design. However, the method is based on the assumption that the objective functions are smooth and continuous therefore surrogate-assisted methods simply cannot solve a discrete optimization problem.

4

Methods

In this chapter the simulation and optimization methods used in the thesis will be presented. Often, researchers use a boundary conforming mesh to do computational fluid dynamics (CFD) simulations. In this thesis however, the method for simulating the fluid flow and heat transfer is the hybrid immersed boundary method, developed by Mark et.al. [26; 29].

For the optimization method, a number of methods have been described in chapter 3. The chosen method to be performed in this thesis is the multi-objective particle swarm optimization method. The reasons to choose this method are:

- No information on the gradient of objective function (black box function), thus gradient-based methods are not feasible.
- There is a discrete variable in the system (see 5.1.4), thus surrogate-assisted optimization methods are not feasible.
- Evaluation time of each point in the design space is not too long (in average 1 hour per simulation on standard FCC workstation¹), thus evaluating a large number of points is affordable.
- The MO-PSO method can find multiple Pareto-optimum points in a single run.
- The MO-PSO method does not require encoding and decoding of variables (as in genetic algorithm) needed.

4.1 Computational Fluid Dynamics

The Navier-Stokes equations (2.6), are solved using IBOFLOW, a finite volume based incompressible flow solver used to simulate a number of industrial applications [20; 27; 28; 40]. The equations are discretized on a Cartesian octree grid that can be dynamically refined to get a higher resolution of the interface. A segregated

¹Intel i7-5930K 3.50GHz, 64GB RAM, NVIDIA GeForce GTX 650 Ti 1GB

solution technique, the SIMPLEC method [12], of the Navier-Stokes equations is employed to couple the velocity and pressure fields. First, the momentum equation is approximated with an estimated pressure field, and the pressure field is then corrected by the continuity equation. A new velocity field is then obtained by applying the corrected pressure field to the previously obtained velocity field. The method iterates until both the momentum and continuity equations are satisfied. All variables are stored in a co-located grid arrangement and the pressure weighted flux interpolation by Rhie and Chow [34] is used to prevent pressure oscillations. The temporal discretization, the unsteady part of Navier-Stokes equations, is done using implicit backward Euler time scheme. To handle interfaces between solids and fluids the hybrid immersed boundary method[26; 29] is employed. See section 4.1.1 for a comparison of boundary conforming and immersed boundary methods.

In this thesis, the fan speed is set at 1 m/s on all simulations. Using this speed, the flow is expected to be turbulent when the spacing between the fins is more than 17.5mm. Most of the models simulated have smaller spacing, thus a laminar flow is expected in every simulation if the velocity is constant and uniform in the whole simulation area. However, when entering the heat sink, the flow face a contraction therefore its speed may increase (alternatively, the flow can bypass through the side and above the heat sink). The increase may be large enough to make the flow turbulent. Lee[25] in his experiment found that there is a momentary lag of performance when the spacing decrease from 11.25 mm to 8.75 mm which he attributes to flow regime change from turbulent to laminar. For this reason, a turbulent flow model needs to be considered.

4.1.1 Boundary Conforming Mesh vs Immersed Boundary

In boundary conforming mesh, the mesh where we solve the equations is made such that it conforms (i.e. follows) the shape of the boundary. This means the method starts from having the geometry of the boundaries, then we set the meshes to follow them. A small change in the geometry may change the structure of the mesh greatly and gives disproportionate change in the output which may be regarded as noise [36]. Figure 4.1 shows how the boundary conforming mesh is made.

Contrary to what we do in the boundary conforming mesh, an immersed boundary method starts with building a base mesh, usually a structured one. After the base mesh is set, the boundary is then inserted into the mesh. After defining the boundary, the cell type is then determined whether it is inside or outside of the boundary. Each cell type has its own system of equation to solve. The advantage of this method is that complex boundaries can be easily inserted without changing the structure of the mesh. Figure 4.2 shows how the immersed boundary mesh is made.

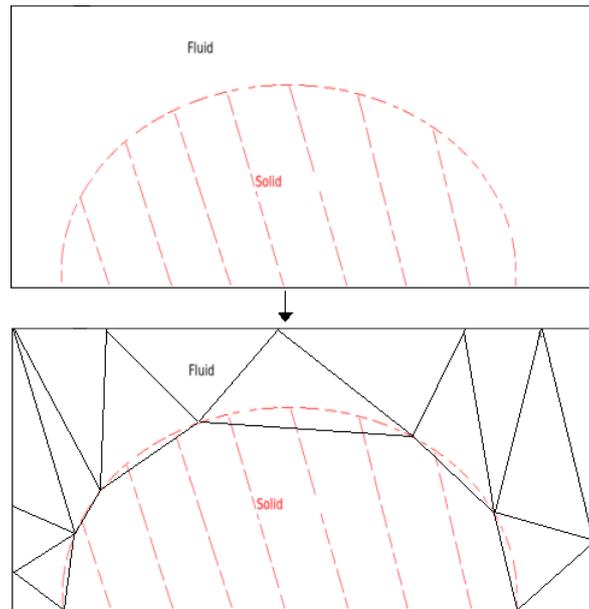


Figure 4.1: Boundary conforming mesh method: the meshes (here for solving the fluid mechanics) follow the curvature of the boundary ("solid" section, depicted by dashed area).

4.1.2 Turbulence Modelling

As briefly mentioned in chapter 1, the spacing between fins may be enough for the flow inside the heat sink to be turbulent. However, if the spacing is small enough, the flow will stay in the laminar regime. This means that we need a turbulence model that is able to model both laminar and turbulent regime and also the transition. Another requirement for the turbulence model is that we need high resolution near the solids because we are interested in the heat exchange between the solids and fluids. The latter requirement yields turbulence modelling with wall functions may not be sufficient because of the inaccuracies associated with the wall functions.

In this work the LVEL turbulence model, which fulfill the requirements, is used. The LVEL turbulence model only require knowledge of the wall distances (L) and the local velocities (VEL), hence its name. The LVEL model is based on Spalding's law of the wall (see [2] for more details).

4.2 Particle Swarm Optimization

Particle swarm optimization (PSO) is a biologically-inspired iterative method for solving optimization problems. It was inspired by swarming behaviour of many organisms.

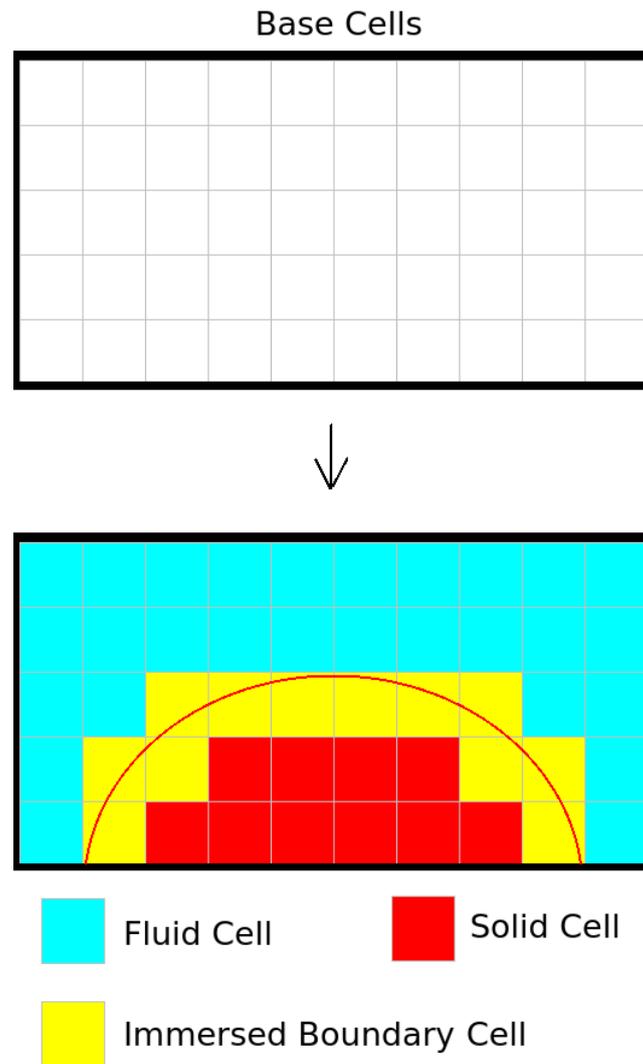


Figure 4.2: Immersed boundary method: The method starts with a base grid/cells. The red line represent the boundary is then immersed into the grid. The cells then assigned a value indicating whether it is inside the boundary (solid cells), outside (fluid cells), or at the boundary (IB cells).

4.2.1 Basic Particle Swarm Optimization Algorithm

The algorithm uses a concept of position and velocity. "Position" x is a vector of design variables, while "velocity" v is used to determine which "position" should be evaluated at the next iteration.

In PSO we have a number of candidate solutions named "particles". Each particle has its own position and velocity. In each iteration (called "generation"), the positions of all particles are evaluated (i.e. the value of design variables inserted into the objective function) to give its corresponding objective value. The best value (i.e. smallest objective value in minimization, or largest in maximization) is then saved and will be used to update the "velocity" (i.e. "acceleration"). We have 2 kinds of best values:

- Particle best: the best position each particle had so far,
- Swarm best (often called "leader"): the best position found by the algorithm so far (the best among particle best).

There are many versions of PSO algorithms, but all of them share a basic algorithm, as described by Wahde[45]:

1. Set initial position and velocity for each particle.
2. Evaluate the objective value of each particle.
3. Update the best positions (swarm-best and particle-best).
4. Update velocities and positions:

- (a) Calculate new velocities

$$v_{ij} = wv_{ij} + \frac{c_1q(x^{pb} - x_{ij})}{\Delta t} + \frac{c_2r(x^{sb} - x_{ij})}{\Delta t}. \quad (4.1)$$

- (b) Restrict velocities such that $|v_{ij}| \leq v_{max}$.

- (c) Update positions:

$$x_{ij} = x_{ij} + v_{ij}\Delta t. \quad (4.2)$$

5. Return to 2 until stopping criteria are reached.

$i = 1, 2, \dots, N$ and $j = 1, 2, \dots, M$, with N denotes the number of particles and M denotes the number of variables. w is called "inertia weight", a relaxation factor on the velocity. c_1 and c_2 are constants, typically set to 2. q and r are uniform random numbers in the interval $[0,1]$. x^{pb} and x^{sb} are positions of the best point in each particle history (particle best) and best point among all evaluated points (swarm best) respectively. Δt typically set as 1.

4.2.2 Multi-Objective PSO (MO-PSO)

For multi-objective optimization problem (MOOP), multi-objective variants of PSO are developed. These variants of PSO are using the basic PSO algorithm with modifications on determining swarm-best and particle-best values. In MO-PSO, several points are considered best (the "current" Pareto-front²) therefore there are several candidates for swarm-best and particle-best position. There are various ways of determining swarm-best and particle-best vectors. A summary of these methods can be found in [33]. Among the methods, the schemes described by Fieldsend-Singh[14] and Coello-Lechuga[8] are used in this thesis. For both methods, the velocities are restricted because this is a crucial step in preventing the swarm from expanding indefinitely[45].

4.2.2.1 Coello and Lechuga's Approach

Coello and Lechuga [8] used density of Pareto-front and roulette-wheel selection to choose the leaders for each particle. The leaders are not unique, different particles can use different points in the Pareto-front. The motivation is to create a well distributed Pareto-front by preferring search on areas with low Pareto-optimum-points density. The "particle best" point is the last non-dominated point in the particle's history. The algorithm (further will be referred as CL-method or Coello's method) can be described as follows:

1. Set initial position and velocity for each particle.
2. Evaluate the objective value of each particle.
3. Update the best positions (swarm-Pareto and particle-best).
4. Create several hypercubes of Pareto-front explored so far and calculate its fitness (the method of creating the hypercubes and calculating the fitness described below)
5. Update velocities and positions:
 - (a) For each particle, pick a hypercube of current Pareto-front using roulette-wheel selection based on its fitness
 - (b) For each particle, pick a non-dominated point from the chosen hypercube as leader
 - (c) Calculate new velocities

$$v_{ij} = wv_{ij} + \frac{c_1 q(x^{pp} - x_{ij})}{\Delta t} + \frac{c_2 r(x_i^{sp} - x_{ij})}{\Delta t}. \quad (4.3)$$

²It is termed "current" because it is Pareto-front found by the algorithm so far, i.e. non-dominated points found so far. The algorithms attempt to get as close as possible to the "true" Pareto-front. In this thesis Pareto-front and Pareto-optimum points refers to the non-dominated front and non-dominated points found by the algorithm, not the "true" Pareto-front.

(d) Update positions:

$$x_{ij} = x_{ij} + v_{ij}\Delta t. \quad (4.4)$$

(e) Restrict position such that they stay in feasible region.

6. Return to 2 until stopping criteria are reached.

The hypercubes mentioned above are created by dividing the objective space bounded by the minimum and maximum value of objective values found-so-far in each axis into a constant number of equally sized partitions. A hypercube fitness $f_{hypercube}$ is set as zero if the hypercube contains zero non-dominated point, otherwise the fitness is calculated as:

$$f_{hypercube} = \frac{A}{N_{pareto}}. \quad (4.5)$$

A is a constant larger than 1 (in this thesis 10 is used), and N_{pareto} is the number of non-dominated points inside the hypercube.

Some modifications are done on the implementation. The first modification is not to restrict the positions to stay in the feasible region because the restriction would limit exploration. Instead, to handle the points on non-feasible region, a penalty method is used. These non-feasible points are saved and after the PSO finishes, the points are projected back into feasible region for *a posteriori* analysis.

Another modification is on picking particle-best point: instead of comparing current point with a single last particle-best point, in this thesis all non-dominated points in the particle's history are used. Fieldsend and Singh in [14], referring to other researches, stated that the original method is prone to oscillation. Also, all of the points considered are non-dominated and potentially lead closer to the true Pareto-front so it is unfair to only consider one or two of them.

4.2.2.2 Fieldsend and Singh's Approach

Fieldsend and Singh [14] used a measurement of "closeness" to determine which point in the Pareto-front found-so-far to be used as the leader. Closeness is defined as the distance between points in the objective space. A particle will choose a currently non-dominated point with the most similar objective values as its leader. Similar with Coello and Lechuga's approach, each particle can have its own choice of leader. Fieldsend and Singh's approach does not require the particle positions to be restricted in the feasible region.

The downside is that the method relies on a relationship between "closeness" in objective space and "closeness" in design space, i.e. the algorithm may perform poorly if points "close" in the objective space are faraway in the design space or vice versa. The simulations in this thesis are expected to have close (i.e. similar) objective values if the designs are similar. However, points close to each other in the

objective space are not necessarily close in the objective space due to the existence of discrete variable in the design space.

To promote exploration and prevent early/premature convergence, a "turbulence" operator is introduced. The turbulence operator randomly includes a new term into the velocity equation with probability p_{turb} . In this thesis, the turbulence probability is 0.2 and the turbulence magnitude b is a random number $U(-0.05V_{max}, 0.05V_{max})^3$.

The algorithm (further will be referred as FS-method or Fieldsend's method) is as follows:

1. Set initial position and velocity for each particle.
2. Evaluate the objective value of each particle.
3. Update the best positions (swarm-Pareto and particle-Pareto).
4. Calculate distance between each particle's objective value with all points in the current Pareto-front.
5. Update velocities and positions:

- (a) For each particle, pick the closest non-dominated point
- (b) Calculate new velocities

$$v_{ij} = wv_{ij} + \frac{c_1q(x^{pp} - x_{ij})}{\Delta t} + \frac{c_2r(x_i^{sp} - x_{ij})}{\Delta t}. \quad (4.6)$$

- (c) For each particle check for turbulence:

- Get a random number r ,
- If r is larger than p_{turb} include turbulence term b_j

$$v_{ij} = v_{ij} + b_j. \quad (4.7)$$

- (d) Update positions:

$$x_{ij} = x_{ij} + v_{ij}\Delta t. \quad (4.8)$$

6. Return to 2 until stopping criteria are reached.

³In [14], the distribution for b is $N(0, 0.1R)$ with R the absolute range of the variables. It is changed so it complies with the velocity restriction.

5

Geometrical Optimization of Plate Fin Heat Sink

Heat sinks utilizing parallel plates are widely used to cool electronic modules (e.g. CPU, graphic processors)[38]. This type of heat-sink are known as plate-fin heat-sinks. The fins are stacked together on top of a base plate. An illustration of plate-fin heat-sink is presented in figure 5.1.

This thesis is considering simulation-based optimization, i.e. the objective functions f_i are output from a simulation rather than evaluations of mathematical functions. In other words, the design variables are inserted into a "black box", i.e. the simulation software with its internal workings, and the objective values are returned as outputs.

The simulation setup is shown in figure 5.2. The heat sink is placed in the middle of an air channel. Due to symmetry considerations, only half of the heat sink is used in the simulations. The base plate is heated uniformly by 20W.

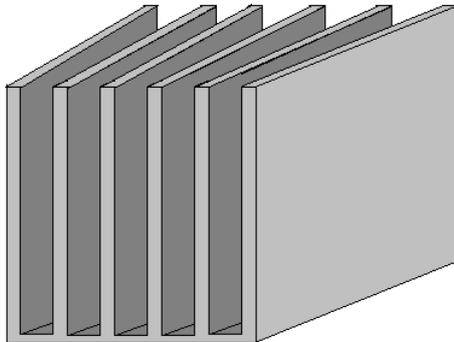


Figure 5.1: Illustration of a plate-fin heat-sink. Air will flow between the plates and cool down the plates. The plates are typically made of metals such as copper or aluminium.

5. Geometrical Optimization of Plate Fin Heat Sink

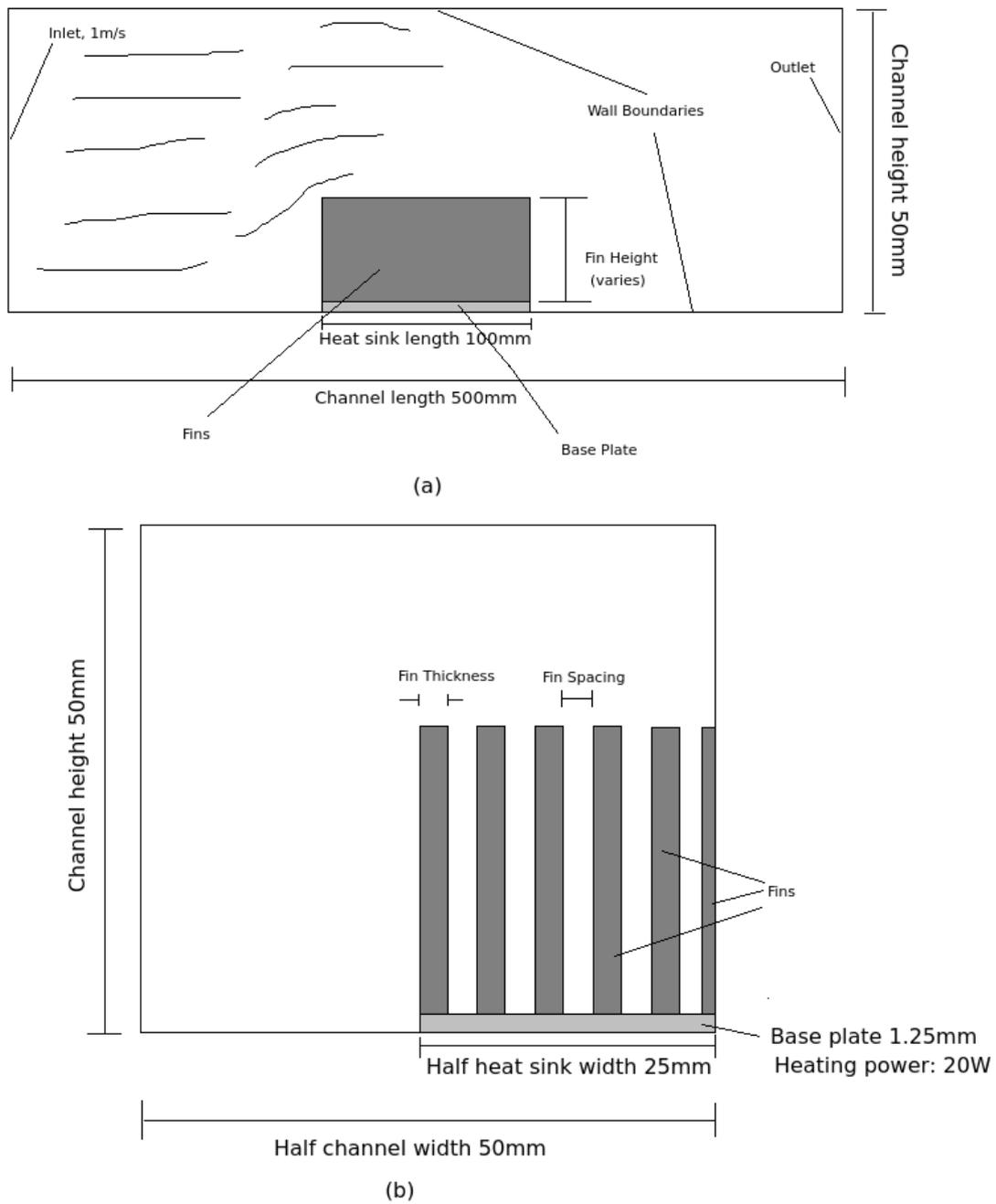


Figure 5.2: Simulation setup (a) side view, (b) streamwise view.

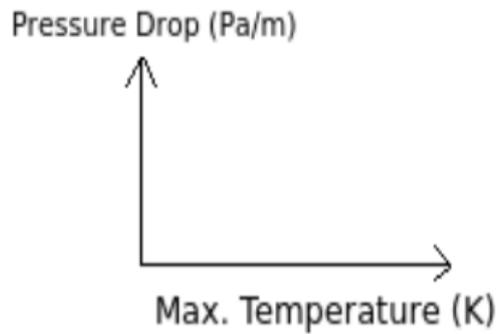


Figure 5.3: The objective space has two axes. Each axis corresponds to an objective function.

5.1 Optimization Problem

An optimization problem is formulated from objective functions and design variables (which creates the objective space and design space respectively) with respect to some constraints. The variations in the objective space and design space will be limited by some constraints. These constraints define the border between the feasible and non-feasible points in the spaces.

5.1.1 Objectives

We can use several objectives for the optimization problem. Some of the possible objectives are:

- Minimize the steady state temperature,
- Minimize the operating cost,
- Minimize the production cost,
- Minimize the dimension of the heat sink.

In this thesis, minimization of steady state temperature and minimization of operating costs are used as objective functions. The operating cost is assumed to be proportional to the energy needed to propel the air flow through a pressure difference. To measure performance with respect to this objective, we set a constant flux of air at the inlet. It is then sufficient to gauge the average pressure at the inlet and outlet to get the pressure drop. The pumping power, i.e. the energy required to propel the air flow, is proportional to the pressure drop. The objective space is illustrated in figure 5.3.

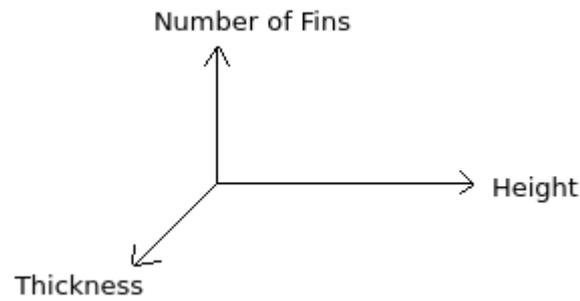


Figure 5.4: The design space is three-dimensional. Each axis corresponds to a design variable.

5.1.2 Variables

There are many variables that can be modified for the optimization problem such as fin spacing/density, fin thickness, fin height, dimension of the heat sink, air flow velocity, etc. However, we limit the problem to be a geometrical optimization and only vary the thickness, height, and number of fins. The design space is illustrated in figure 5.4.

5.1.3 Constraints

When optimizing heat-sinks by means of simulations, the constraints may appear due to different reasons. Some are due to design limitation e.g. maximum dimension of the heat sink, some may come from the fabrication process e.g. minimum fin thickness and spacing.

The first set of constraints comes from the fabrication and defined by Iyengar [19]. There are several fabrication methods and the bonding method is chosen for this thesis. The constraints are presented in table 5.1. Some design constraints also introduced and shown in table 5.2.

Table 5.1: Constraints due to bonding fabrication-method

Minimum fin thickness	0.75 mm
Maximum fin height:spacing ratio	60:1
Material	Al, Cu, or Mg
Minimum spacing between fins	0.8 mm

The total height is the sum of the fin height and base plate thickness. The base plate thickness is kept at 1.25 mm, therefore we can rewrite the constraint as: maximum fin height = 48.75 mm.

Table 5.2: Constraints due to design specification

Minimum fin height	0.625 mm
Maximum total height	50 mm
Minimum number of Fins	2
Heat sink width	50 mm
Heat sink length	100 mm

Table 5.3: Symbols and definitions for equation 5.1

Symbol	Definition
τ	Thickness
h	Fin height
N	Number of fins
T_{max}	Maximum temperature of solid
ΔP	Pressure drop
s	Fin spacing
w	Heat sink width

5.1.4 Complete Problem

As we have specified the objective, variables and constraints, we can then formally state the optimization problem,

$$\begin{aligned}
 & \underset{\tau, h, N}{\text{minimize}} && (T_{max}, \Delta P), \\
 & \text{subject to} && \tau \geq 7.5 \times 10^{-4} && \text{(Minimum thickness),} \\
 & && 6.25 \times 10^{-4} \leq h \leq 4.875 \times 10^{-2} && \text{(Minimum and maximum height),} \\
 & && N \geq 2 && \text{(Minimum number of fins),} \\
 & && h/s \leq 60 && \text{(Maximum height:spacing ratio),} \\
 & && s \geq 8 \times 10^{-4} && \text{(Minimum spacing),} \\
 & && N\tau + (N - 1)s = w && \text{(Spacing equality).}
 \end{aligned} \tag{5.1}$$

N is an integer variable, while the others are positive real numbers. The feasibility cut is shown in figure 5.5. The last constraint in 5.1.4 is due to the width of the heat-sink set at a constant value w . The boundary shape resembles stairs, with number of fins defining the steps. The boundary contour is shown in figure 5.6.

5.1.5 Optimization Configurations

The MO-PSO has been run 6 times with different configurations, see table 5.4. For all configurations, the maximum number of evaluations is 120. Each simulation takes 1 hour on average, so in total an optimization run would require approximately 5 days. However, non-feasible points are not simulated and given ∞ for both objective

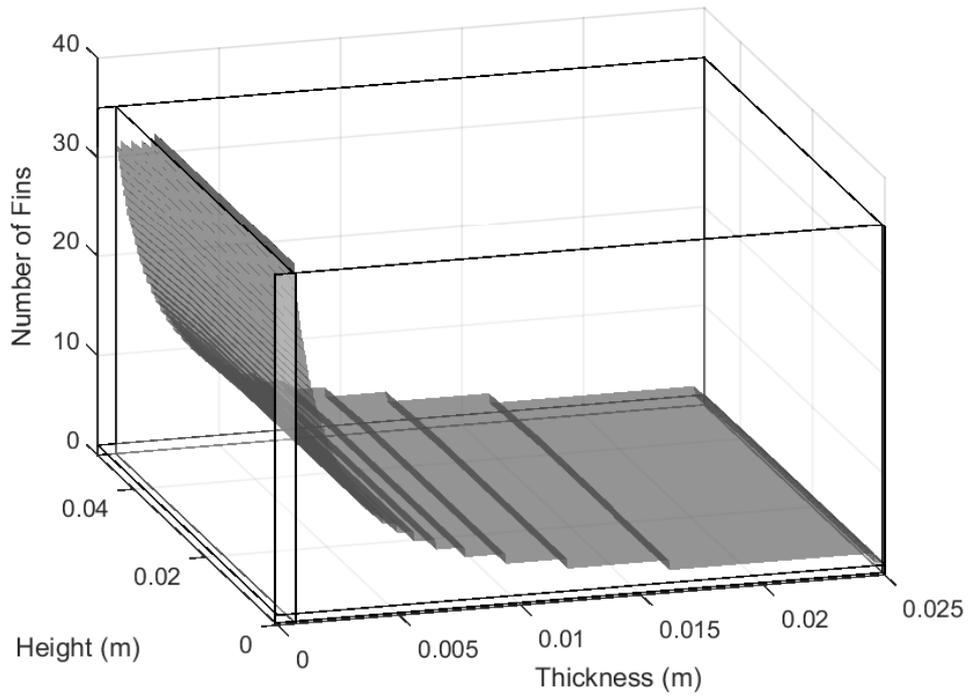


Figure 5.5: Feasibility cuts due to constraints

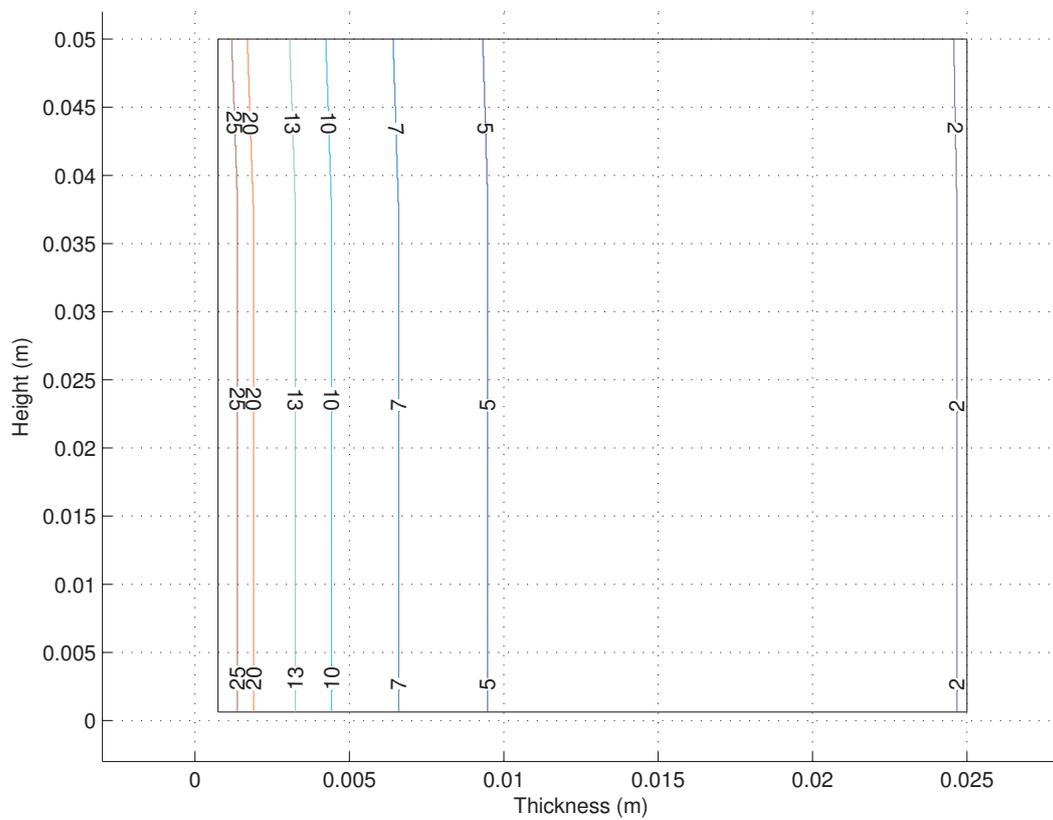


Figure 5.6: Feasibility contour for different N

values. The number of non-feasible points evaluated varies depending on how the particles moved, but it is in the range of 20-80 evaluations which corresponds to 20-80 hours reduction in run time.

As mentioned in section 4.2.2, velocity restrictions are used. The restrictions are based on the maximum and minimum values for thickness and height. The velocity for the number-of-fin is more restricted because adding or reducing even just one fin may give very different performance. The restrictions are presented in table 5.5.

Table 5.4: MO-PSO configurations

No.	Leader Choosing Method	Number of Particles	Number of Generations
1	FS	3	40
2	FS	6	20
3	FS	12	10
4	CL	3	40
5	CL	6	20
6	CL	12	10

Table 5.5: Maximum speed for each axis in design space

Axis	Max Speed
τ	49.375 mm
h	24.25 mm
N	1

5.1.6 Design of Experiment

The standard PSO algorithm chooses starting points for the first generation particles randomly with a uniform distribution in each axis of design space. The choice of starting points is then modified to suit our optimization problem and obtain more meaningful data at the start of the algorithm.

The algorithm is modified in such a way that the starting point is restricted to be inside the feasible region. This is achieved by randomly (uniform distribution) choosing fin height and number of fins, and then after the two are set, the choice of thickness is restricted to follow the constraints. This method is chosen due to two reasons. First, the complicating constraint for height ($h/s \leq 60$) is an inequality (as opposed to equality constraint on N and τ) so it is more flexible and actually can be chosen almost freely in its range (minimum to maximum height). Second, it is expected that the performance difference due to changing the number of fin to be larger than the effect of changing thickness therefore exploration on number of fins is more important than exploration on thickness.

5.2 Results and Discussion

The simulation is performed using IBOFlow and the PSO algorithm is implemented in MATLAB. To use IBOFlow, the fins and base-plate mesh need to be generated beforehand and immersed into the base grid constructed in IBOFlow. The generation of the base-plate only needs to be run once while the fins need to be generated multiple times for different dimensions of fins. The fin generation process has been scripted using Paraview's python interface *pvpython*. Simulation set-up and configuration in IBOFlow is written in Lua. The administration of the simulation results is managed using MATLAB scripts. The entire simulation and optimization process is automated by a master script in MATLAB.

Illustrations of simulation results are shown in figure 5.7 and 5.8. The figures show the temperature and velocity profile simulated in IBOFlow and visualized in Paraview.

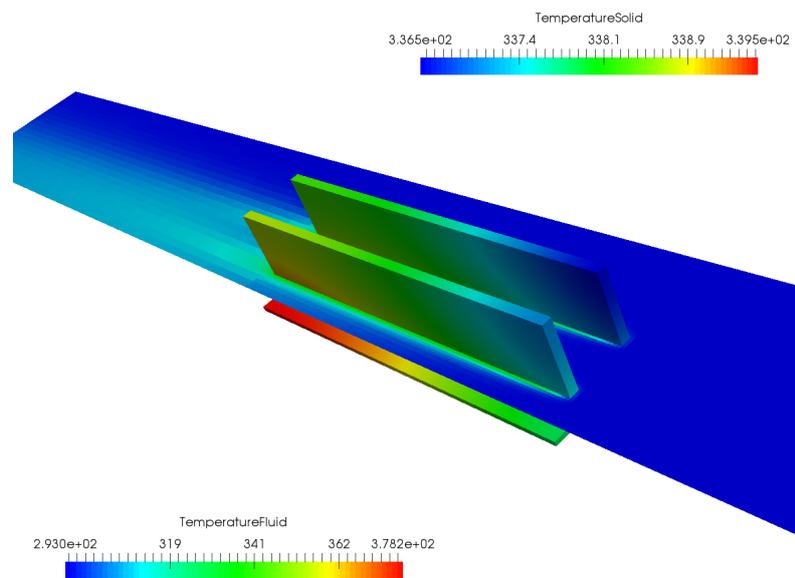


Figure 5.7: Temperature distribution from a simulation

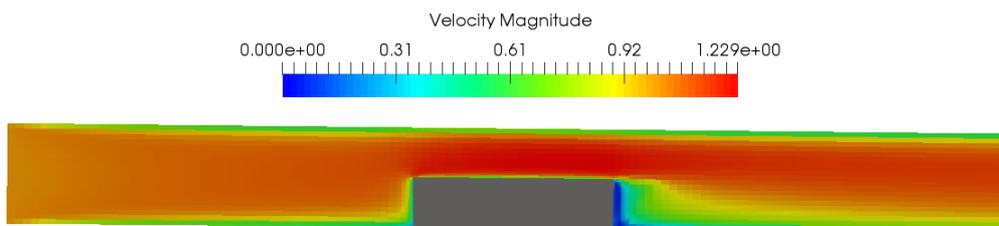


Figure 5.8: Velocity profile from a simulation. The heat sink is the gray area in the middle.

5.2.1 Pareto-front

Each configuration explores the design and objective space differently, not only due to the different number of particles but also due to randomization in the algorithm. Thus, each configuration yields a Pareto-front with characteristic qualities. Figure 5.9 shows the objective space explored by each configuration and the associated Pareto-front.

The figures show that by using 120 evaluations, FS-method could create a better Pareto-front. "Better" here defined by having smaller values, i.e. several dominating points, compared to the Pareto-front from CL. Notice that the figures use the same scale so it can be easily compared. At a glance it can be seen that the "knee" of the Pareto-front from FS-method (bottom row) is lower than CL-method Pareto-front.

The relatively bad performance of the CL-method may be attributed to the number of evaluations which are too low compared to Coello's suggestion. Coello suggested to use 20-80 particles with 80-120 generations so the minimum suggested number of evaluation is 1600 evaluations, more than 10 times the 120 evaluations conducted in this thesis. It can be concluded that the FS-method is more efficient in finding the Pareto-front.

The FS-method yields a dense Pareto-front in some areas and coarser in other areas. This behaviour is expected and consistent with the assumption that similar designs will give similar performances. By choosing leaders based on distance, the particles will move closer and closer to the Pareto-set. Looking at equation 4.6, when the distance is small, the acceleration will also be small making the search around a Pareto-optimum point more extensive (i.e. more points sampled). If similar designs indeed give similar performances the points would be located very close to each other both in objective space and design space. Thus, we observe crowding/clustering of points which is a characteristic trait of FS.

None of the configurations from CL-simulations resulted in dense Pareto-fronts. The spirit in CL is attraction towards less populated Pareto-front regions. Thus, the observations from CL are consistent with the expectations. It can then be concluded that CL will give a coarser Pareto-fronts compared with FS. However, coarser Pareto-fronts should not be considered worse because several data points giving similar objective values is most likely not efficient. Rather, it could be argued that the Pareto-front from CL has a more reasonable distribution.

Considering that FS finds the Pareto-front faster whereas CL yields less clustering, a combination of the methods may be considered. The suggested improvement would be to use distance-based method to quickly find the Pareto-front, but avoid clustering/crowding of points. A proposal is to change the craziness (turbulence) operator from giving a random velocity change into changing the leader to less populated Pareto-front. The method should quickly refine the Pareto-front by extensively searching around a Pareto-optimum point, but sometimes it will change its leader thus exploring a different parts of the design space. However, the evaluation of the

method is not in the scope of the thesis.

Generally, PSO performs better with few particles but more evaluations[45], but the optimum number of particles may actually differ depending on the case[7; 35]. In this thesis, the number of particles (3, 6, or 12) does not seem to show an apparent consistent-effect in the objective space.

5.2.2 Pareto-set

While getting a good Pareto-front is the optimization objective, in the end the goal is to decide which designs are optimum, i.e. the Pareto-set. The Pareto-set is a set consisting of all Pareto-optimal points. Pareto-sets from the simulations are shown in figure 5.10-5.13.

Intuitively, the smallest fin located at the boundary (shortest and thinnest) will also have the smallest pressure drop as it is the least obstructing configuration therefore it should be a Pareto-optimum point. A heat-sink with many fins also generally believed to have higher thermal performance compared to heat-sink with few fins[25]. This means that the Pareto-set would be located near or at the boundaries.

Looking at figure 5.10-5.13 it can be seen that many of the points are indeed located near the boundaries, particularly in the direction of "thickness" axis. It can be seen in the figures (especially by looking at the rightmost figure in each figure set) that several of the Pareto-optimum points are near the thinnest allowed fin.

Top-right figure in figure 5.12 (CL, 3 particles) shows a very small exploration along the "number of fin" (N)-axis, but remembering that the number of evaluations is constant then the exploitation (local search) is more intensive which fits Röhler and Chen's[35] theory regarding number of particles. This is due to the uneven distribution (crowding) of initial points (the initial points are 32, 23, and 20 fins) thus no information is acquired for other (lower) number of fins. The starting Pareto-optimum points used as leaders will also be among these initial points therefore the exploration is only in a small area around the initial points. The velocity restriction also prevents the particles from exploring too far. This behaviour is not observed in the other three figures (including the FS-3 particles) due to the initial points being more spread.

Even though the behaviour is not observed in the FS-3 particles case, the number of particles can affect the spread of initial points. More particles means more initial points will be seeded thus it is more likely that the initial points are spread out because they are seeded randomly.

The MO-PSO algorithms do not require the points to be evenly spread in the design space. It may even start from the infeasible region and find a good solution, such as the tests reported by Castro, et.al.[24, Chapter 9]. However, considering the small number of evaluations conducted, it is beneficial if the design of experiment is

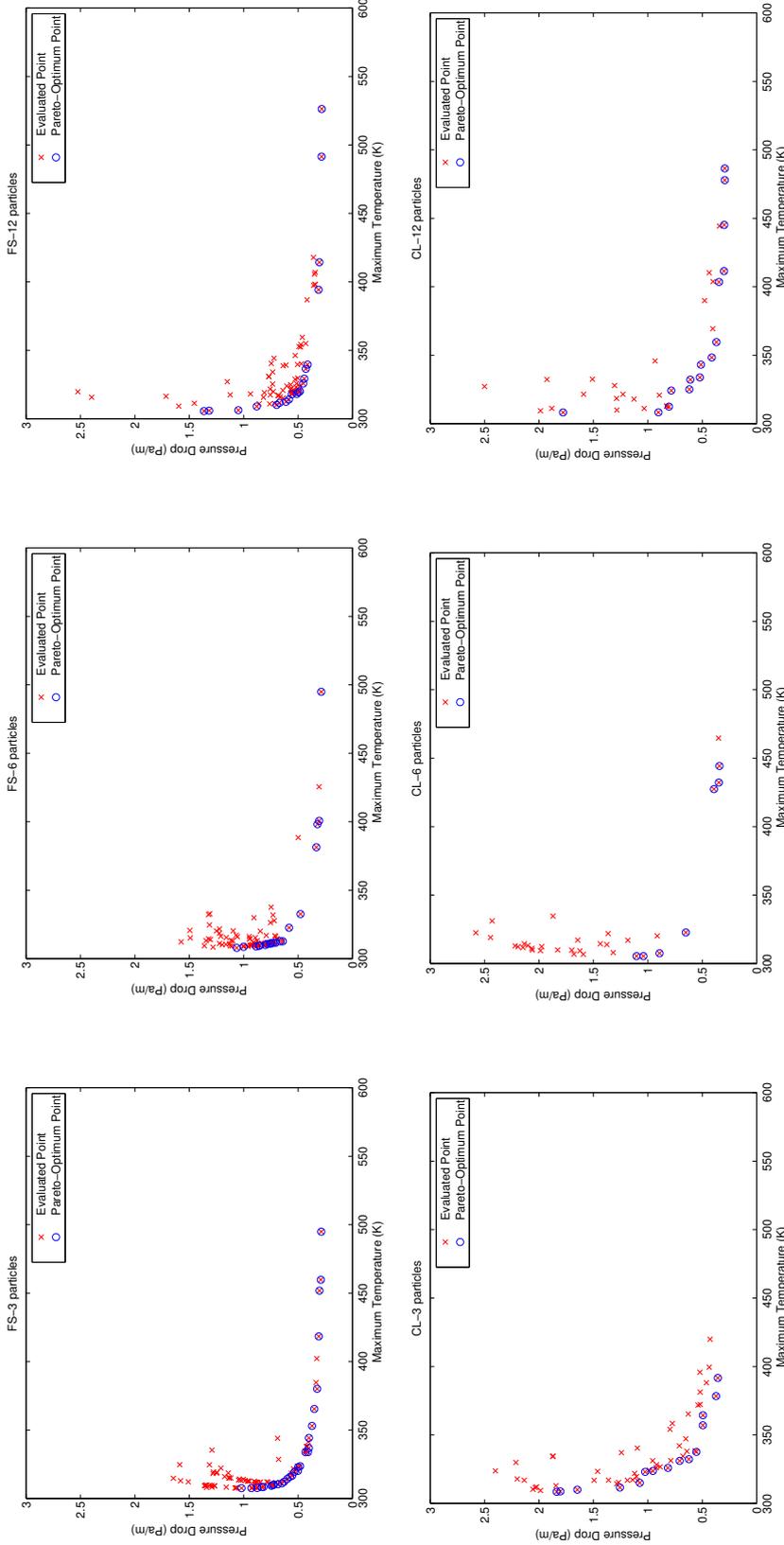


Figure 5.9: Explored objective space and its Pareto-front found using MO-PSO. The upper row, from left to right are configuration number 1-3. The bottom row, are configuration number 4-6. (see table 5.4 for the configurations).

modified to guarantee the initial points being spread in the design space. Examples of methods to promote spreading of initial points are the Latin Hypercube Sampling (LHS) (see [30]) and orthogonal sampling (see [42]) method.

If LHS or orthogonal sampling are used, all areas of the design space are guaranteed to be represented by at least one particle [30]. The number of particles will affect the sizes of the areas, more particles mean more areas can be sampled, thus the areas can be smaller.

5.3 Post-processing

5.3.1 Design Space Filter

It appears in the objective space that the evaluated points are clustered in some areas. The clustering suggests that introducing a design space filter can speed up the optimization process by reducing the number of evaluated designs. Since, as stated in section 4.2.2.2, points close to each other in the design space should be close in the objective space, but not necessarily vice versa. The filtering should then be based on this assumption that two designs with very similar dimensions will show similar performance (maximum temperature and pressure drop), thus evaluating only one of the two candidates is expected to be reasonable.

In order to filter the design space, a measure of similarity is required. There exist several measurements of similarity. In this thesis, three similarity measures are evaluated to test the filtering proposal:

- Volume difference (ΔV): $\tau_{min}\Delta h + h_{min}\Delta\tau + \Delta h\Delta\tau^1$,
- L₂ Norm: $N\sqrt{\Delta h^2 + \Delta\tau^2}$,
- L_∞ Norm: $\max(\Delta h, \Delta\tau)$.

In addition to similarity measures, thresholds must be introduced to distinguish similar designs from unique designs. Three different values of the similarity-threshold for each of the similarity measures mentioned above are evaluated, see table 5.6. An additional tracker is required to record which points are evaluated (evaluated-points) and which are filtered (filtered-points) because filtered-points should not be used as references in similarity comparisons. The filter is tested using the following algorithm:

1. Load a data set from previous MO-PSO run.

¹ τ_{min} is the smaller τ value between the two point, as is h_{min} is the smaller h value

²The L₂ norm is multiplied by the number of fins because the total change is subject to the number of fins.

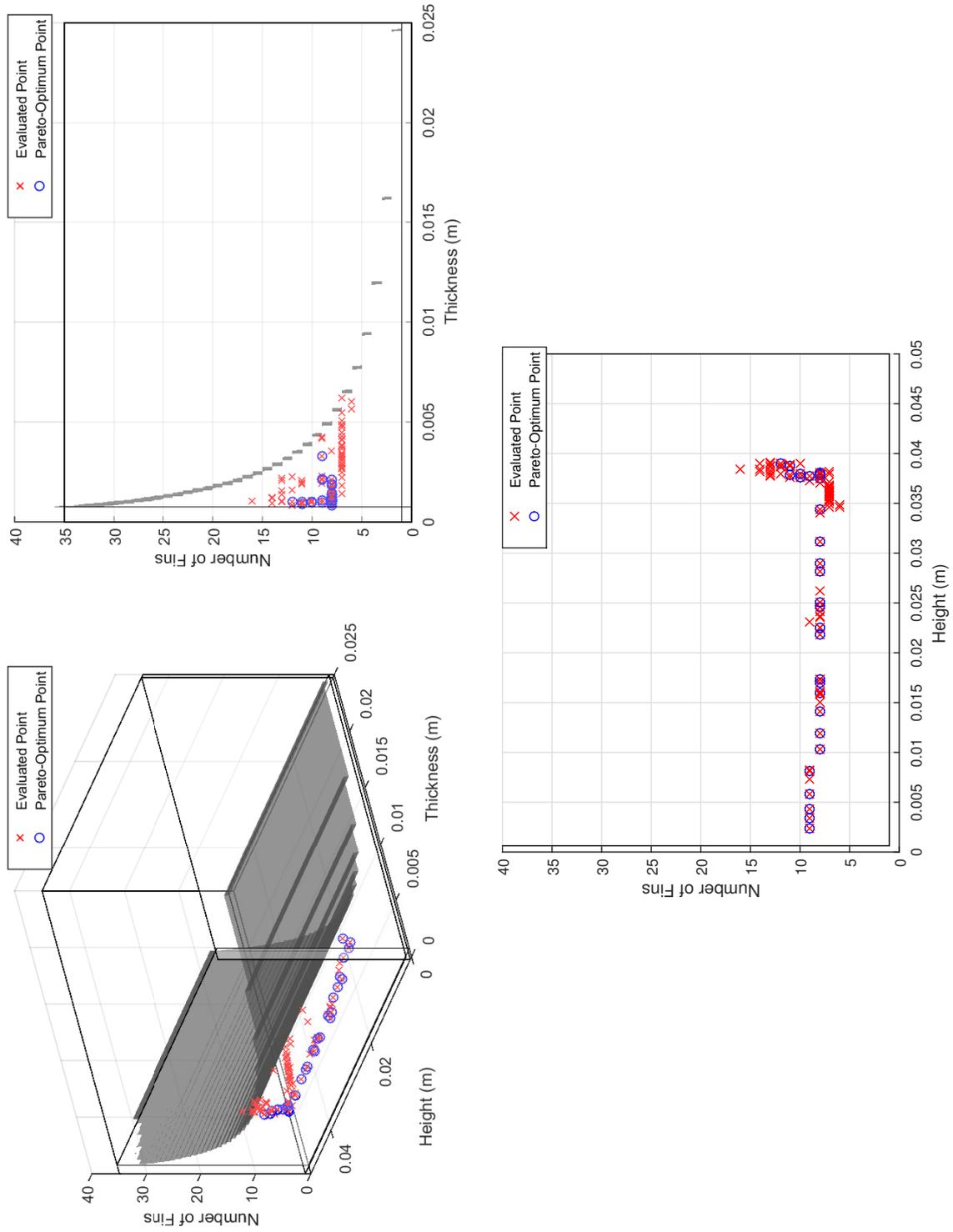


Figure 5.10: Explored design space and the Pareto-set found using MO-PSO using configuration 1 table 5.4).

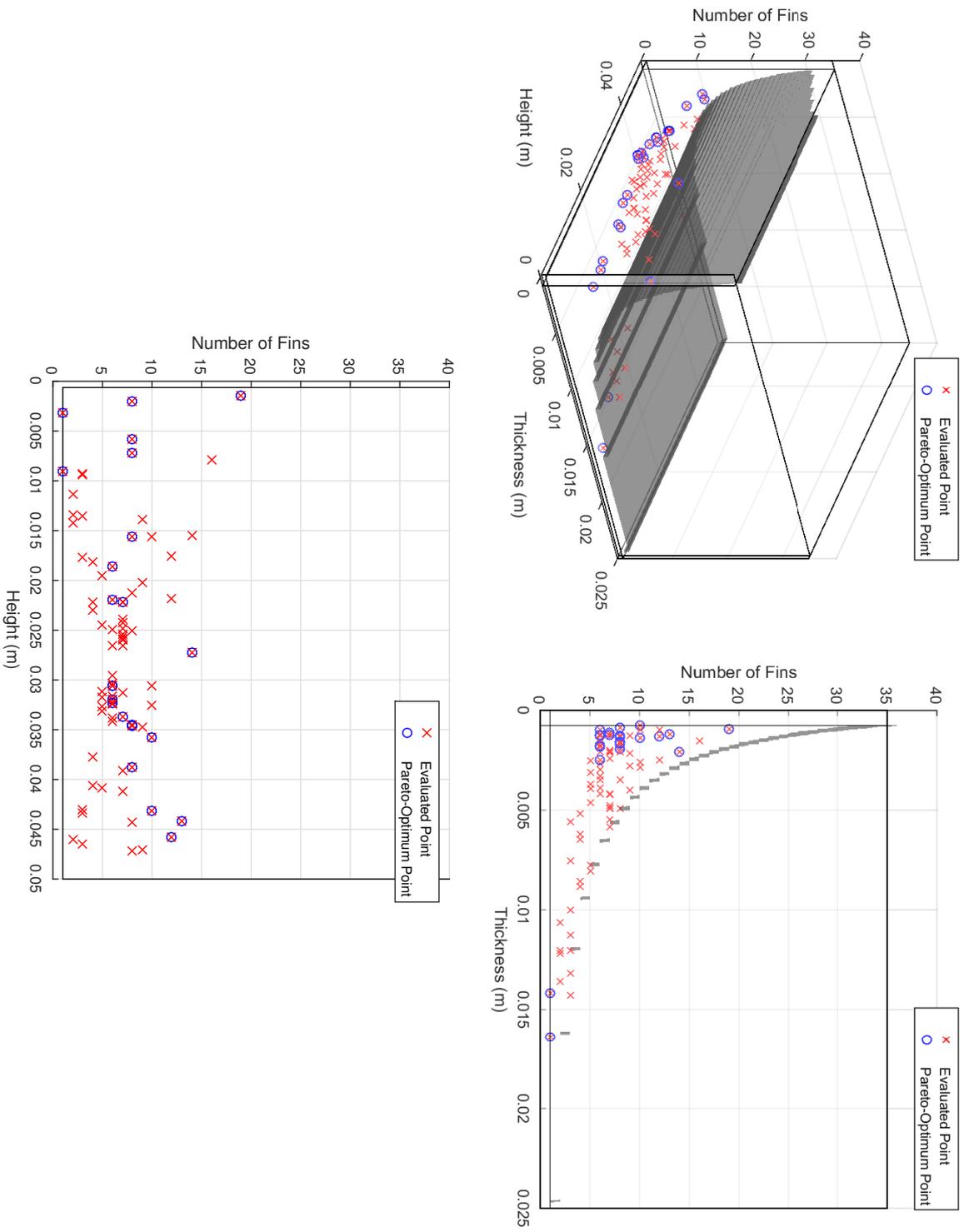


Figure 5.11: Explored design space and the Pareto-set found using MO-PSO using configuration 3 table 5.4).

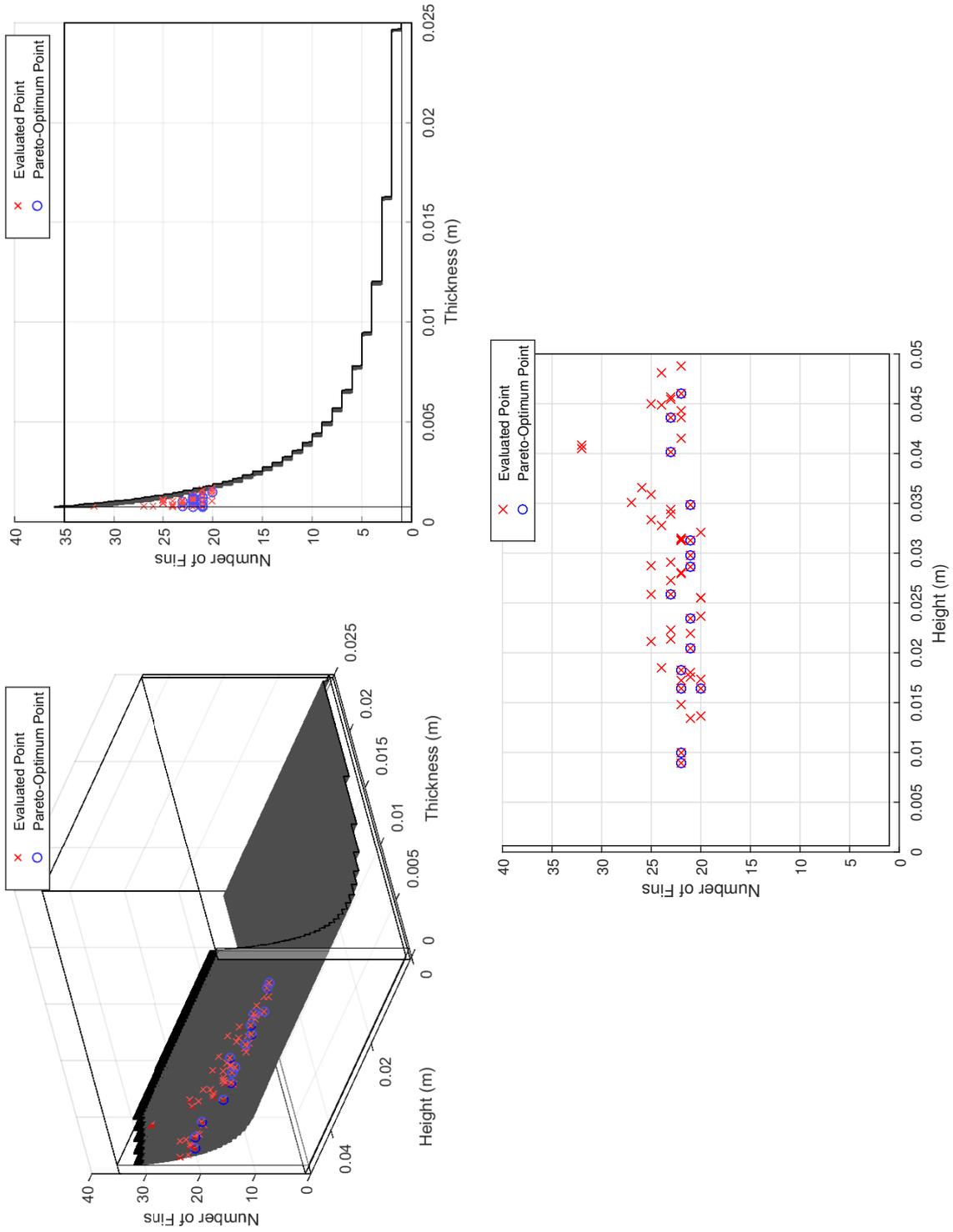


Figure 5.12: Explored design space and the Pareto-set found using MO-PSO using configuration 4 table 5.4).

5. Geometrical Optimization of Plate Fin Heat Sink

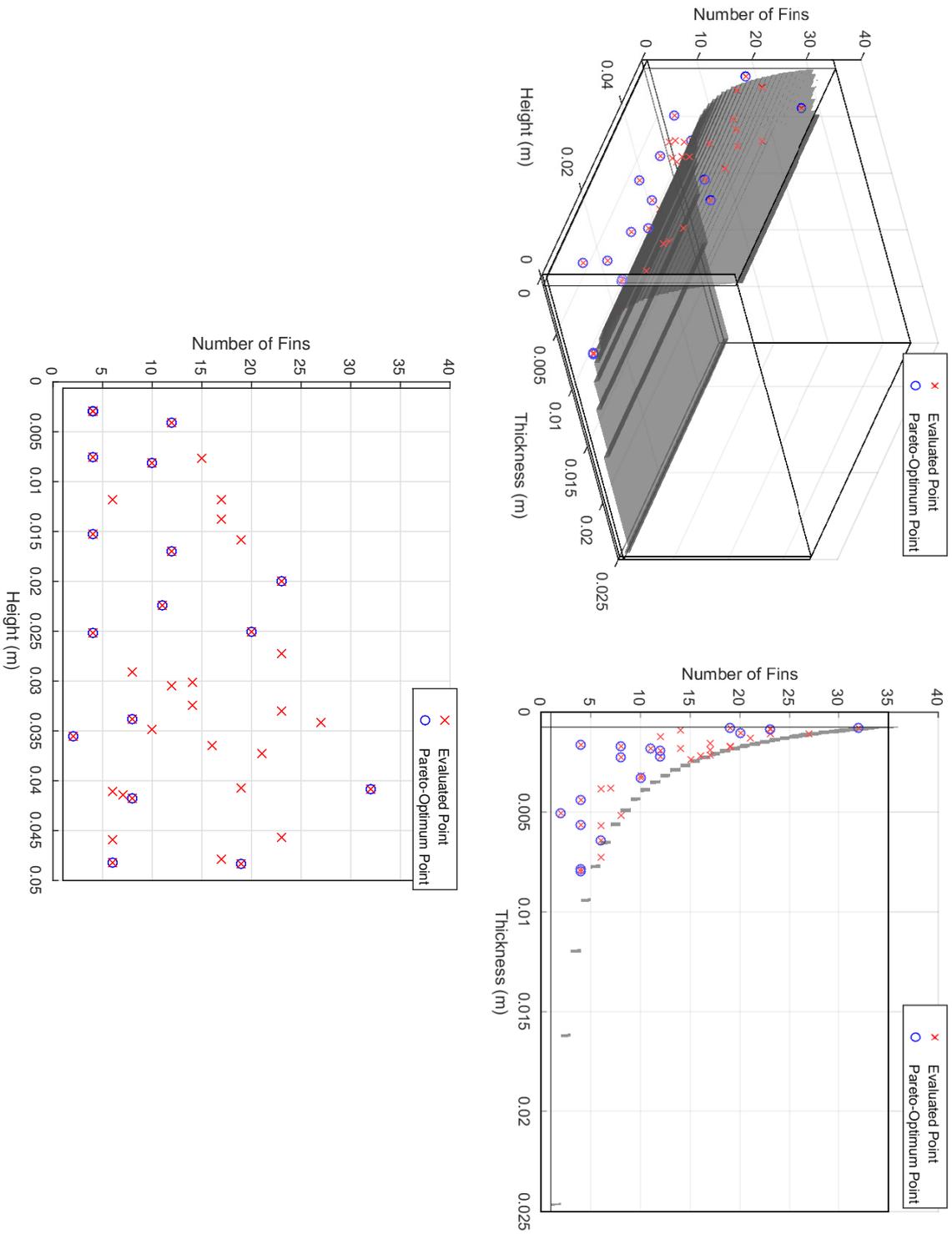


Figure 5.13: Explored design space and the Pareto-set found using MO-PSO using configuration 6 table 5.4).

2. Re-evaluate the points starting from oldest generation, first particle and do filtering:
 - If there are any similar evaluated-point, mark the new point as "filtered". Assign identical objective values with the most similar evaluated-point,
 - else, evaluate the new point, mark it as "evaluated".
3. Continue until all particles in the generation are re-evaluated.
4. Continue until all generations are re-evaluated.

The filtering test cases and their results are summarized in table 5.6. An example of re-evaluated MO-PSO data (FS, 3 particle) and the change to the Pareto-front are shown in figure 5.14. In the figure, some "x"-marks are not inside the upward triangles indicating the design corresponding to the objective vector are filtered.

The number of simulations can be decreased significantly using filtering. However, the Pareto-front also changes. Some Pareto-optimum points may be filtered (circles without triangle) and some sub-optimum points may be identified as Pareto-optimum (triangles without circle). A higher threshold increases the number of filtered points and also increase the risk of missing the real Pareto-front. By using filters and data from previous optimization runs, the MO-PSO can be re-run to explore the design space further. Doing this potentially improve the Pareto-front with a very short run-time because only unexplored designs will be simulated.

It should be mentioned that specifying a similarity-threshold before performing any simulations is not a trivial task. Too high a threshold will filter too many points and much information is lost, while too low a threshold means that almost no points are filtered. The remark is thus that there is not an obvious way to quantify "high" and "low" values of similarity threshold before performing simulations. A proposed guideline is to use inherent limits and uncertainties in the fabrication process e.g. if the uncertainty in thickness of the fin is estimated to 0.05mm it is reasonable to use this value or slightly larger as the threshold for the L_∞ -norm filter.

5.3.2 Exploring the Boundaries

The MO-PSO run had many points not evaluated due to the fact that it crossed the feasible region boundary. These points are ignored in the optimization, but recalling that the Pareto-optimum points should be close to or at the boundaries, exploration on the boundaries is considered particularly relevant.

The boundaries themselves have infinitely many points. To search the whole boundaries is practically impossible, therefore a guidance on which point in the boundary to evaluate is needed. One alternative is to use the non-feasible points in the PSO run. The points in the non-feasible area can be used as a guide because in the optimization run, these points are supposed to be results of pulling the particles

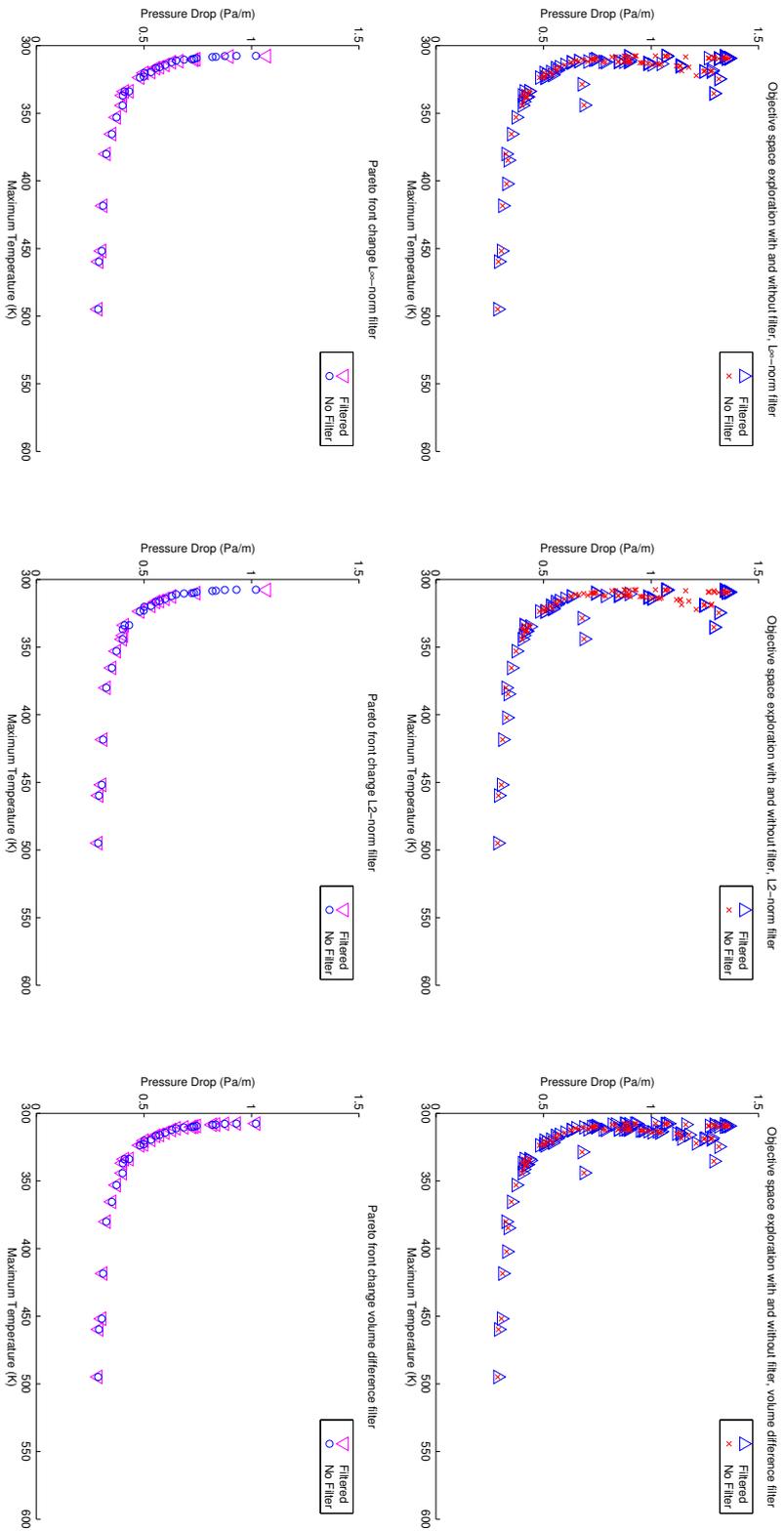


Figure 5.14: Filtering effect on the objective space. The whole objective space are shown on top row, while in the bottom row only the Pareto-fronts are shown. From left to right: filter test configuration 8,10, and 11 (see table 5.6).

Table 5.6: Filter test configurations

No.	Data set	Method	Threshold	Original ^a	Filtered ^b	Missing ^c	New Pareto ^d
1	CL,3 par ^e	ΔV	0.00007m ³	59	3	1	0
2			0.00014m ³		7	2	0
3		L ₂	0.01m		40	10	7
4			0.005m		10	3	7
5		L _∞	0.25mm		18	4	8
6			0.1mm		1	0	0
7	FS,3 par ^f	ΔV	0.00007m ³	97	32	1	1
8			0.00014m ³		64	13	2
9		L ₂	0.01m		83	17	3
10			0.005m		65	15	3
11		L _∞	0.25mm		43	11	5
12			0.1mm		12	4	2

^a Number of simulated points without filtering

^b Number of filtered points

^c Number of Pareto-optimum points missed due to being filtered

^d Number of new Pareto-optimum points. "New" means without filtering these points were suboptimum

^e CL, 3 particles data set

^f FS, 3 particles data set

to potentially better positions, therefore it is likely that these points can refine the Pareto-front when projecting back onto the feasible region.

Projection to the nearest point in the boundary, however, is quite difficult in this case due to the constraints being discontinuous. An alternative way is to address the constraints one by one and project the point in the direction of one axis every time it violates the constraint. The direction can be along the height, number of fins or thickness axis.

The first three constraints in equation 5.1 are straight-forward to deal with by assigning the minimum or maximum value for the axes. The remaining constraints, however, are complicated since they are coupling several design variables, and one of the design variables is discrete.

For the complicating constraints, it would be simpler to decompose the constraints. The spacing equality constraint in 5.1 can be rewritten as

$$s = \frac{w - N\tau}{N - 1}. \quad (5.2)$$

Substituting it into the minimum spacing constraint yields

$$\frac{w - N\tau}{N - 1} \geq 8 \times 10^{-4}. \quad (5.3)$$

Rearranging the equation gives

$$\tau \leq \frac{w - 8 \times 10^{-4}(N - 1)}{N}, \quad (5.4)$$

$$N \leq \frac{w + 8 \times 10^{-4}}{\tau + 8 \times 10^{-4}}, \quad (5.5)$$

Substituting equation 5.2 to the maximum height:spacing ratio also yields

$$\frac{h(N-1)}{w - N\tau} \leq 60. \quad (5.6)$$

Rearranging the equation, gives:

$$N \leq \frac{60(w - N\tau)}{h} + 1. \quad (5.7)$$

$$\tau \leq \frac{60w - h(N-1)}{N}. \quad (5.8)$$

$$h \leq \frac{60(w - N\tau)}{N-1}. \quad (5.9)$$

The maximum allowed value for each design variable then can be rewritten as 3 new constraints, replacing the last 3 constraints in 5.1:

$$\tau_{max} = \min \left[\frac{w - 8 \times 10^{-4}(N-1)}{N}, \frac{60w - h(N-1)}{N} \right], \quad (5.10)$$

$$h_{max} = \frac{60(w - N\tau)}{N-1}, \quad (5.11)$$

$$N_{max} = \min \left[\frac{60(w - N\tau)}{h} + 1, \frac{w + 8 \times 10^{-4}}{\tau + 8 \times 10^{-4}} \right]. \quad (5.12)$$

After knowing the maximum allowed value for each design variable, the violations Δ are calculated, i.e. for particle i :

$$\Delta\tau_i = \tau_i - \tau_{max}, \quad (5.13)$$

$$\Delta h_i = h_i - h_{max}, \quad (5.14)$$

$$\Delta N_i = N_i - N_{max}. \quad (5.15)$$

The violations are then normalized by the maximum allowed values.

$$\Delta\tau'_i = \frac{\Delta\tau_i}{\tau_{max}}, \quad (5.16)$$

$$\Delta h'_i = \frac{\Delta h_i}{h_{max}}, \quad (5.17)$$

$$\Delta N'_i = \frac{\Delta N_i}{N_{max}}. \quad (5.18)$$

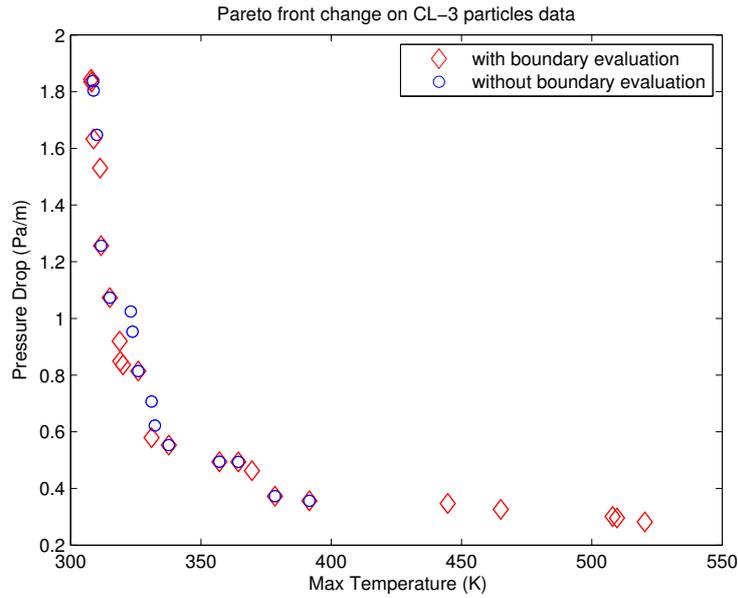


Figure 5.16: Projection-evaluation on Pareto-front of configuration 4

5.3.3 Choosing the Best Design

Once the Pareto-front is found, the final step is to pick the best design from the available choices in the Pareto-front. The process is known as post-Pareto analysis. All points in the Pareto-front are optimum, but a product can only have one design. At this step, designers' preferences are needed.

Several methods can be used to determine the best design once the designers' preferences are specified. Three such methods are considered, the corresponding results and best designs are presented in this section³. The data used in this section is a compounded data from all simulations conducted. The Pareto-optimum solutions are presented in table 5.7 and 5.17. It should be noted that some of the solutions presented in the table is not practical because the maximum temperature from the simulation exceed the operating temperature limit (around 175° C or 449 K) therefore the data should be filtered so these impractical designs are excluded in decision making.

5.3.3.1 Lexicographic Method

The easiest method to choose a design is the lexicographic method. The method has been described in section 3.2.1. Using this method, the solution is either solution number 1 if temperature is preferred or 40 if pressure drop is preferred.

³see Hwang and Yoon[18] for more methods

Table 5.7: Compounded Pareto-optimum solutions sorted by objective values. Greyed rows indicates the temperature exceed operating temperature limit and not considered in the decision making.

No.	τ (mm)	Height (mm)	Num. of fins	Spacing (mm)	T_{max} (K)	$-\Delta P$ (Pa/m)
1	0.8703	45.0496	12	3.6	305.4232	1.1037
2	0.7638	44.4611	12	3.7	305.4505	1.0396
3	1.0480	40.5090	10	4.4	307.4992	0.8918
4	0.8995	37.8903	11	4.0	307.7265	0.8757
5	1.0122	38.0074	10	4.43	308.3137	0.8342
6	0.9695	37.6391	10	4.48	308.495	0.8182
7	0.9882	37.6822	9	5.14	309.3	0.7443
8	1.2785	38.0434	8	5.68	309.8515	0.7328
9	1.2524	37.5514	8	5.71	310.0961	0.7226
10	0.7589	35.8034	10	4.71	310.133	0.6973
11	0.9909	37.9724	8	6.01	310.4181	0.6863
12	0.8092	37.8003	8	6.22	311.0514	0.6496
13	0.9413	34.4159	8	6.07	312.3095	0.6288
14	0.8613	34.4763	8	6.16	312.3955	0.6112
15	1.1356	33.6721	7	7.01	314.1755	0.5840
16	1.1485	28.9944	8	5.83	315.9209	0.5729
17	1.0997	28.1952	8	5.89	316.6367	0.5547
18	1.2645	31.9186	6	8.48	318.0909	0.5118
19	1.2310	30.5528	6	8.52	319.3324	0.4971
20	0.9654	30.6307	6	8.84	320.0806	0.4830
21	1.2714	21.8	8	5.69	323.676	0.4811
22	1.2299	22.2282	7	6.90	325.7606	0.4524
23	1.8328	21.9485	6	7.80	329.3331	0.4428
24	2.9677	34.9490	3	20.55	333.1692	0.4315
25	1.4385	17.3498	8	5.50	333.7974	0.4308
26	1.075	17.0107	8	5.91	333.8781	0.4098
27	1.2348	15.9273	8	5.73	336.8343	0.4011
28	2.4457	23.5315	4	13.4	339.5859	0.3998
29	2.0783	14.3779	7	5.91	344.8056	0.3885
30	0.9439	12.9312	9	5.19	346.8166	0.3749
31	1.8884	11.9501	8	4.98	353.022	0.3723
32	4.3833	15.2765	4	10.82	359.6116	0.3695
33	0.7996	9.9980	9	5.35	361.4313	0.3373
34	4.0003	19.9172	2	42.00	378.6095	0.3333
35	1.0395	15.2783	3	23.44	378.6349	0.3166
36	1.3551	7.2009	8	5.60	394.2132	0.3118
37	0.9339	6.4205	9	5.20	400.7254	0.3065
38	1.6475	7.4988	4	14.47	411.4077	0.2997
39	1.9443	4.0727	12	2.42	445.1884	0.2978
40	1.0622	4.1526	10	4.37	448.2688	0.2944
41	2.7060	4.7463	3	20.94	456.368	0.2901
42	2.6186	2.6398	10	2.65	473.2611	0.2897
43	1.6261	2.6219	12	2.77	479.5023	0.2886
44	7.0654	2.3908	4	7.25	485.3148	0.2881
45	1.6587	2.0672	8	5.25	491.4896	0.2839
46	0.9433	1.3897	19	1.78	526.3024	0.2825
47	1.7244	0.9566	12	2.66	572.993	0.2800

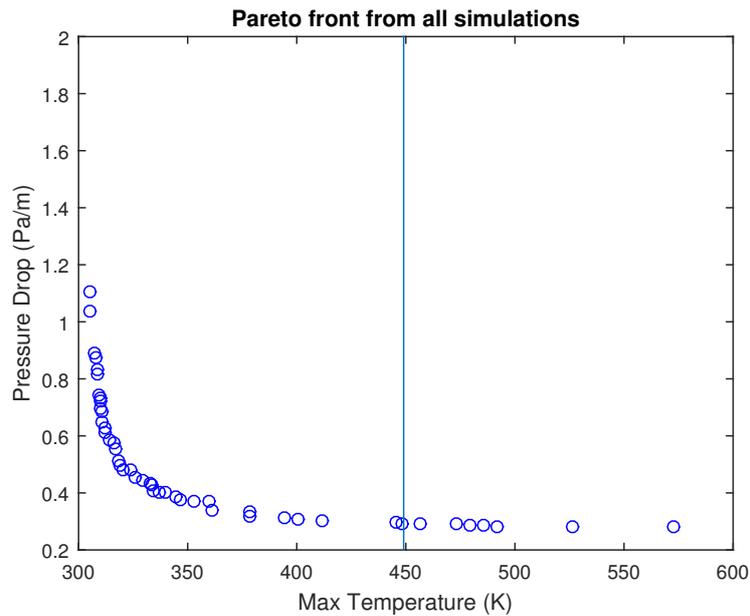


Figure 5.17: Pareto-optimum solutions from all simulations conducted. The line represent the operating temperature limit 449K.

5.3.3.2 Analytical Hierarchy Process (AHP)

In choosing a design, qualitative judgement frequently comes into the system. Preference of the objectives is one of them. The most prominent feature of AHP is its ability to assign weights to qualitative judgements. The requirement for AHP is that the qualitative judgements are given a score (number). An example of valid preference is "objective X is *twice* more important than objective Y" or "objective X score is *ten*, while objective Y score is *three*".

In reality, it may happen that more than 2 objectives are considered. To compare more than 2 objectives, a pairwise comparison should be done resulting an $n \times n$ matrix (n is the number of objectives). The importance (weight) of each objectives is then taken from the eigenvector of the matrix.

AHP is not scale invariant, therefore the objective values need to be normalized. After normalizing, it is then only a matter of evaluating each point in the Pareto-front multiplied by the weights making it a weighted sum approach similar with the method in 3.2.2, but done *a posteriori*. For a more detailed description and example, see [16] and the section "Hierarchical Additive Weighting Method" in [18].

There can be several possible results using this method depending on the designers' preferences. Some possible results are presented in table 5.8.

5.3.3.3 Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)

TOPSIS is a method to rank choices developed by Yoon and Hwang[18] in 1980. Similar with AHP, TOPSIS starts with normalizing and weighting the objectives. The weighting and normalizing method introduced in AHP can be used in TOPSIS, but other methods can also be used. The main difference between TOPSIS and AHP is instead of using a weighted sum like in AHP, TOPSIS consider 2 special points to choose the best design: the ideal solution and the negative-ideal solution.

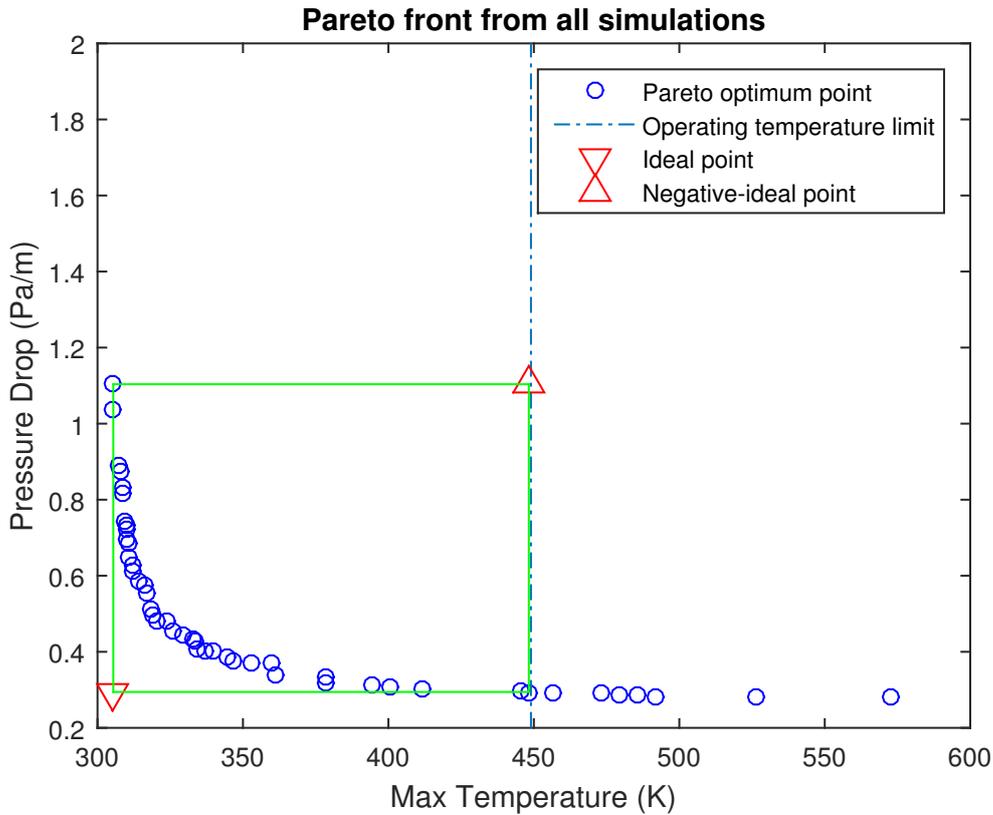


Figure 5.18: Ideal and negative-ideal solution without normalization and equal weights

The ideal solution is the point where all weighted normalized objective values are at their best possible at the same time, for example assuming equal weights and no normalization, in table 5.7 the T_{max}^{best} is 305.4232 K and $-\Delta P^{best}$ is 0.2944 Pa/m therefore the ideal solution is (305.4232, 0.2944). This point usually is not attainable by the system (otherwise the problem becomes trivial).

The negative-ideal solution is the opposite of the ideal solution, i.e. the point where all weighted normalized objective values are at their worst possible at the same time. Using table 5.7 assuming equal weights and no normalization, it is the point (448.268, 1.1037).

The idea is that the best design should be as close as possible to the ideal solution and as far away as possible to the negative-ideal solution. However, this may also become another multi-objective optimization problem. To prevent this, Yoon and Hwang[18] propose a method to consider the distance to both points simultaneously by taking the relative closeness to the ideal solution. Relative closeness of point i to the special points is calculated using equation 5.19.

$$c_i = \frac{s_{i-}}{s_{i+} + s_{i-}} \quad (5.19)$$

c_i is the relative closeness, s_{i+} is the Euclidean L^2 -norm between point i and the ideal solution, and s_{i-} is the Euclidean L^2 -norm between point i and the negative-ideal solution⁴. Notice that $s_{i+} = 0$ at the ideal solution so that $c_i = 1$ and $s_{i-} = 0$ at the negative-ideal solution so that $c_i = 0$.

It should be noted that different weights will give different values of relative closeness therefore changing the decision. The effect of changing weights is shown in figure 5.19. In the figure it can be seen that when maximum temperature is given a larger weight than pressure drop, the Pareto-front shape becomes flattened in the vertical direction, i.e. the distance in the pressure-drop axis is scaled to be smaller while the distance in maximum-temperature axis is scaled larger. On the opposite case, the Pareto-front is flattened in the horizontal direction. The weights also shifts the position of the ideal and negative-ideal point.

The best designs with respect to different preferences both using AHP and TOPSIS are presented in table 5.8 and in figure 5.20. In the figure, the appearance of the best designs can also be seen. In general, when the pressure drop (operating cost) is preferred, the fins will be short and/or has large spacing. On the other hand, when temperature is preferred the operating cost would be higher and the fins will be tall and dense. The best design when temperature is preferred have almost 4 times higher operating cost.

The thickness seems to vary greatly between designs in the Pareto-set, ranging from 0.75mm to 4.4mm. Many designs have spacing less than 8.75mm which means they operate at laminar flow regime according to [25] thus it can be said that effect of turbulence on temperature performance is not significant.

⁴the L^2 -norm is the norm used in the main description of TOPSIS, other measure of distance can also be used

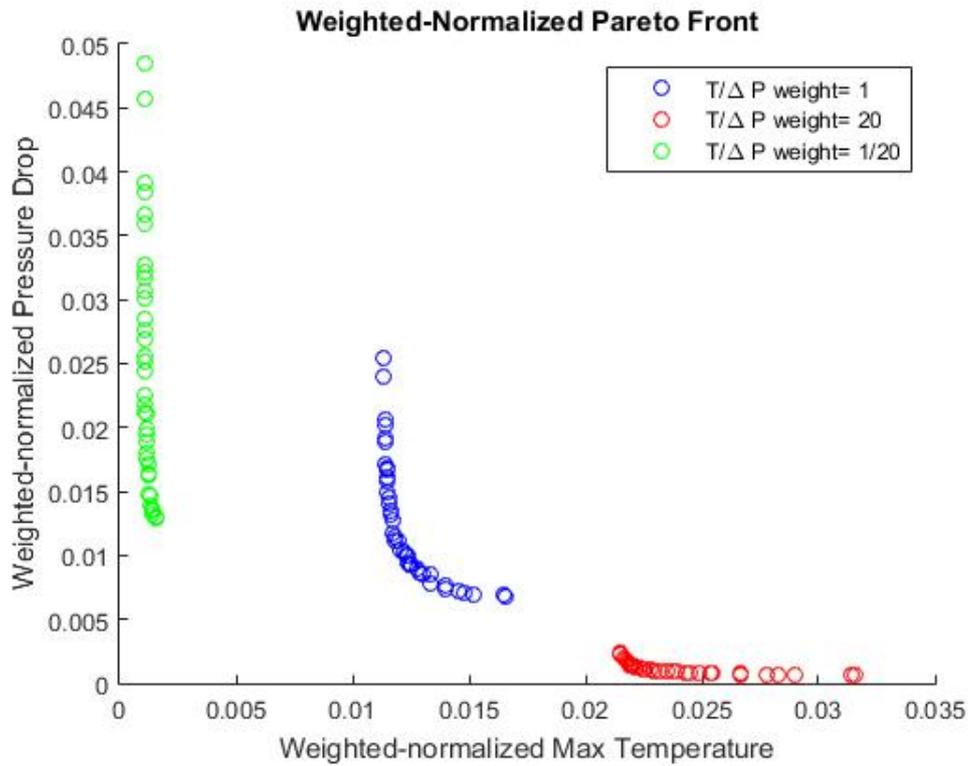


Figure 5.19: Effect of weights on the Pareto-front

Table 5.8: Best design with respect to relative weight of T_{max} using AHP and TOPSIS

$T_{max}/-\Delta P$ weight	Optimum Configuration ID	
	AHP	TOPSIS ^a
1/10000	40	40
1/20	40	38
1/5	38	35
1 ^b	33	33
5	20	20
20	12	14
10000	1	1

^a The weighting and normalizing method used is using same method as in AHP

^b Usually taken when no decision-maker/designer available and referred as "no-preference methods".

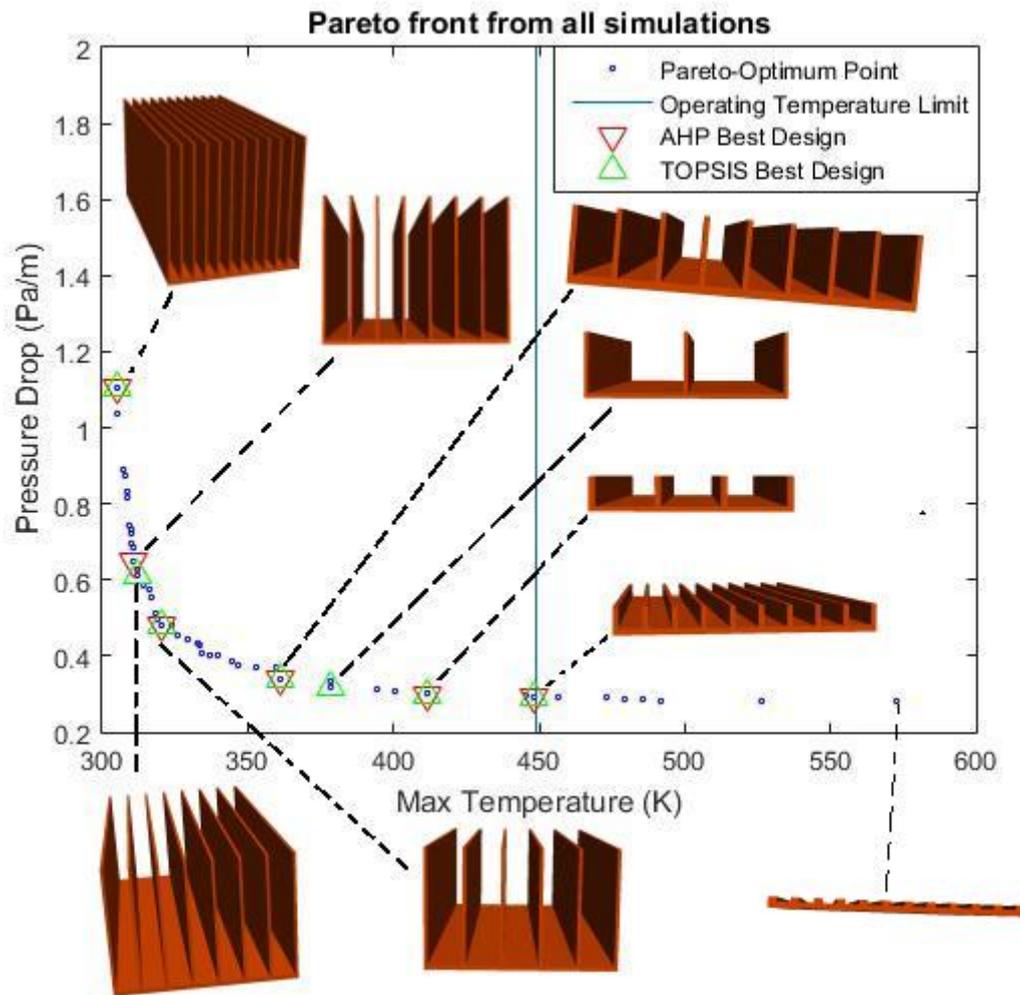


Figure 5.20: Location of the best designs on the Pareto-front and its image

6

Conclusion

The simulation based Multi-Objective Particle Swarm Optimization of a processor heat sink has been presented in this thesis. Two objective functions are chosen and the design space is three dimensional. Two MO-PSO variants from literature are selected, implemented and evaluated for the geometric optimization considered. The multi-physics software IBOFlow are used to simulate fluid flows and conjugate heat transfer.

The results show that Fieldsend and Singh's approach creates a more refined Pareto front than Coello and Lechuga's without any post-processing. However, the CL-method has a better-distributed Pareto front. A method is proposed to combine the benefit of both MO-PSO method by modifying the turbulence operator in FS-method, however the thesis does not cover the implementation and test of the proposed method.

The number of particles may affect the spread of initial points which in turn may increase the probability of crowding. Latin hypercube sampling and orthogonal sampling methods are proposed to mitigate the effect but not tested in this thesis.

Two post-processing methods for the multi-objective optimization are proposed and tested. The two methods have opposing effect. The first method, filtering, is proven to be able to reduce computation time significantly with reduced Pareto front quality as the price. The second method, projection of infeasible points, is capable of improving the Pareto front but it would increase the computational cost significantly, potentially doubling the cost.

Post-Pareto analysis to choose the best design is conducted. The analysis requires one or several decision makers to make ordinal or cardinal preference of the objective functions. Three methods to account for these preferences are tested. For generality, several scenarios of decision maker preference are evaluated. The obtained best point for each scenario under different methods have been shown. When operating cost is preferred, the fins are short and/or has large spacing. When temperature performance is preferred, the fins are tall and dense. Turbulence which was believed to improve thermal performance is observed to have insignificant effect.

Bibliography

- [1] The Boussinesq Approximation. URL <https://www.comsol.com/multiphysics/boussinesq-approximation>.
- [2] Dereje Agonafer, Liao Gan-Li, and Brian Spalding. The LVEL turbulence model for conjugate heat transfer at low reynolds numbers. *Proceedings of the 1996 ASME International Mechanical Engineering Congress and Exposition*, 18, 1996.
- [3] Anas Alfaris. Multidisciplinary system design optimization (msdo), multiobjective optimization (i), lecture 14, . URL http://ocw.mit.edu/courses/engineering-systems-division/esd-77-multidisciplinary-system-design-optimization-spring-2010/lecture-notes/MITESD_77S10_lec14.pdf.
- [4] Anas Alfaris. Multidisciplinary system design optimization (msdo), multiobjective optimization (ii), lecture 15, . URL http://ocw.mit.edu/courses/engineering-systems-division/esd-77-multidisciplinary-system-design-optimization-spring-2010/lecture-notes/MITESD_77S10_lec15.pdf.
- [5] N. Andreasson, A. Evgrafov, M. Patriksson, E. Gustavsson, and M. Onnheim. *Introduction to Continuous Optimization*. Edition 2 : 1. Studentlitteratur AB, 2013. ISBN 9789144060774. URL <https://books.google.se/books?id=AHqYngEACAAJ>.
- [6] M. C. Bhuvaneshwari, editor. *Application of Evolutionary Algorithms for Multi-objective Optimization in VLSI and Embedded Systems*. Springer, New Delhi, India, 2015.
- [7] Daniel Bratton and James Kennedy. Defining a standard for particle swarm optimization. *Swarm Intelligence Symposium*, pages 120–127, 2007. doi: 10.1109/SIS.2007.368035.
- [8] Carlos A. Coello Coello and Maximo Salazar Lechuga. Mopso: A proposal for multiple objective particle swarm optimization. *Congress on Evolutionary Computation (CEC'2002)*, 2, 2002.
- [9] *SMT*. Cooler Master Corporation. URL <http://odm.coolermaster.com/manufacture.php>.

- [10] Lars Davidson. Fluid mechanics, turbulent flow and turbulence modeling, 2015.
- [11] Olivier de Weck and Karen Willcox. Multidisciplinary System Design Optimization (MSDO), Post-Optimality Analysis, Lecture 16. URL http://ocw.mit.edu/courses/engineering-systems-division/esd-77-multidisciplinary-system-design-optimization-spring-2010/lecture-notes/MITESD_77S10_lec16.pdf.
- [12] J. P. Van Doormaal and G. D. Raithby. Enhancements of the simple method for predicting incompressible fluid flows. *Numerical Heat Transfer*, 7(2):147–163, 1984. doi: 10.1080/01495728408961817. URL <http://dx.doi.org/10.1080/01495728408961817>.
- [13] Matthias Ehrgott, editor. *Multicriteria Optimization*. Springer, Berlin, Germany, 2000.
- [14] Jonathan E. Fieldsend and Sameer Singh. A multi-objective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence. *Proceedings of the 2002 U.K. Workshop on Computational Intelligence*, 2002.
- [15] Adriana Cortes Godinez, Luis Ernesto Mancilla Espinosa, and Efrén Mezura Montes. An experimental comparison of multiobjective algorithms: Nsga-ii and omopso. In *Electronics, Robotics and Automotive Mechanics Conference (CERMA), 2010*, pages 28–33. IEEE, 2010.
- [16] Rainer Haas and Oliver Meixner. An Illustrated Guide to the ANALYTIC HIERARCHY PROCESS. URL <https://mi.boku.ac.at/ahp/ahptutorial.pdf>.
- [17] William R. Hamburgden. Optimal finned heat sinks, 1986.
- [18] Ching-Lai Hwang and Kwangsun Yoon. *Multiple Attribute Decision Making: Methods and Applications*. Springer-Verlag, New York, USA, 1981.
- [19] Madhusudan Iyengar. Design for manufacturability of forced convection air cooled fully ducted heat sinks. *Electronics Cooling*, August, 2007. <http://www.electronics-cooling.com/2007/08/design-for-manufacturability-of-forced-convection-air-cooled-fully-ducted-heat-sinks/> .
- [20] T. Johnson, P. Røyttä, A. Mark, and F. Edelvik. Simulation of the spherical orientation probability distribution of paper fibers in an entire suspension using immersed boundary methods. *Journal of Non-Newtonian Fluid Mechanics*, 229(1):1–7, 2016.
- [21] Hans Jonsson and Bahram Mosfegh. Modeling of the thermal and hydraulic performance of plate fin, strip fin, and pin fin heat sinks - influence of flow bypass. 2001.
- [22] Sukhvinder S. Kang. Advanced cooling for power electronics. *International Conference on Integrated Power Electronics Systems*, 2012.

-
- [23] Il Yong Kim and Olivier de Weck. Adaptive weighted sum method for multiobjective optimization. *Structural and Multidisciplinary Optimization*, 31, 2006. doi: <http://dx.doi.org/10.1007/s00158-005-0557-6>.
- [24] Natalio Krasnogor, María Belén Melián-Batista, José Andrés Moreno Pérez, J. Marcos Moreno-Vega, and David Alejandro Pelta, editors. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2008)*. Springer-Verlag GmbH, Berlin, Germany, 2008.
- [25] Seri Lee. Optimum design and selection of heat sinks. *Eleventh IEEE SEMI-THERMTM Symposium*, 2002.
- [26] A. Mark, R. Rundqvist, and F. Edelvik. Comparison between different immersed boundary conditions for simulation of complex fluid flows. *Journal of Fluid Dynamics & Materials Processing*, 7(3):241–258, 2011.
- [27] A. Mark, B. Andersson, S. Tafuri, K Engström, H. Söröd, F. Edelvik, and J.S. Carlson. Simulation of electrostatic rotary bell spray painting in automotive paint shops. *Atomization and Sprays*, 23(1):25–45, 2013.
- [28] A. Mark, R. Bohlin, D. Segerdahl, F. Edelvik, and J.S. Carlson. Optimisation of robotised sealing stations in paint shops by process simulation and automatic path planning. *International Journal of Manufacturing Research*, 9(1):4–26, 2014.
- [29] Andreas Mark and Berend G.M. van Wachem. Derivation and validation of a novel implicit second-order accurate immersed boundary method. *Journal of Computational Physics*, 227(13):6660 – 6680, 2008. ISSN 0021-9991. doi: <http://dx.doi.org/10.1016/j.jcp.2008.03.031>. URL <http://www.sciencedirect.com/science/article/pii/S0021999108001770>.
- [30] M. D. McKay, R. J. Beckman, and W. J. Conover. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code . *Technometrics*, 21(2):239–245, 1979. doi: <https://dx.doi.org/10.2307/1268522>.
- [31] Francisco J. Mesa-Martínez, Ehsan K. Ardestani, and Jose Renau. Optimum design and selection of heat sinks. 2010.
- [32] Achille Messac and Christopher A. Mattson. Normal constraint method with guarantee of even representation of complete pareto frontier. *AIAA Journal*, 42(10), 2004.
- [33] Margarita Reyes-Sierra and Carlos A. Coello Coello. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International Journal of Computational Intelligence Research*, 2(3):287–308, 2006.
- [34] C. M. Rhie and W. L. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, 21(11):1525–1532, 1983. doi: 10.2514/3.8284. URL <http://dx.doi.org/10.2514/3.8284>.
- [35] Antonio Bolufé Röhler and Stephen Chen. *AI 2011: Advances in Artificial*

- Intelligence: 24th Australasian Joint Conference, Perth, Australia, December 5-8, 2011. Proceedings.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-25832-9. doi: 10.1007/978-3-642-25832-9_28. URL http://dx.doi.org/10.1007/978-3-642-25832-9_28.
- [36] Johan Rudholm and Adam Wojciechowski. A method for simulation based optimization using radial basis functions. 2007.
- [37] Vladimír ŠEDĚNKA and Zbyněk RAIDA. Critical comparison of multi-objective optimization methods: Genetic algorithms versus swarm intelligence. *RADIOENGINEERING*, 19(3):369, 2010.
- [38] Robert E Simons. Estimating parallel plate-fin heat sink thermal resistance.
- [39] Jayanth Srinivasan, Sarita V. Adve, Pradip Bose, and Jude A. Rivers. The case for lifetime reliability-aware microprocessors. *ACM SIGARCH Computer Architecture News*, 32(2), 2004.
- [40] E. Svenning, Andreas Mark, F. Edelvik, E. Glatt, S. Rief, A. Wiegmann, L. Martinsson, R. Lai, M. Fredlund, and U. Nyman. Multiphase simulation of fiber suspension flows using immersed boundary methods. *Nordic Pulp and Paper Research Journal*, 27(2):184–191, 2012.
- [41] Laura P Swiler, Patricia D Hough, Peter Qian, Xu Xu, Curtis Storlie, and Herbert Lee. Surrogate models for mixed discrete-continuous variables. In *Constraint Programming and Decision Making*, pages 181–202. Springer, 2014.
- [42] Boxin Tang. Orthogonal Array-Based Latin Hypercubes . *Journal of the American Statistical Association*, 88(424):1392–1397, 1993. doi: <http://dx.doi.org/10.1080/01621459.1993.10476423>.
- [43] D. J. Tritton. *Physical Fluid Dynamics*. Van Nostrand Reinhold Company, Berkshire, England, 1977.
- [44] H. K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics*. Pearson Education, England, 2007.
- [45] Mattias Wahde. *Biologically Inspired Optimization Algorithm*. WITpress, Southampton, UK, 2008.