

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**OPTIMIZATION OF CENTRIFUGAL FAN DESIGN USING
GENETIC ALGORITHM AND CST METHOD**



M.Sc. THESIS

Erkan BİÇER

Department of Mechatronics Engineering

Mechatronics Engineering Graduate Programme

MAY 2025

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**OPTIMIZATION OF CENTRIFUGAL FAN DESIGN USING
GENETIC ALGORITHM AND CST METHOD**

M.Sc. THESIS

**Erkan BİÇER
(518101053)**

Department of Mechatronics Engineering

Mechatronics Engineering Graduate Programme

Thesis Advisor: Prof. Dr. Tufan KUMBASAR

MAY 2025

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**GENETİK ALGORİTMA VE CST YÖNTEMİ İLE SANTRİFÜJ FAN
TASARIMININ OPTİMİZASYONU**

YÜKSEK LİSANS TEZİ

**Erkan BİÇER
(518101053)**

Mekatronik Mühendisliği Ana Bilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Tufan KUMBASAR

MAYIS 2025

Erkan BİÇER, a M.Sc. student of ITU Graduate School student ID 518101053 successfully defended the thesis entitled “OPTIMIZATION OF CENTRIFUGAL FAN DESIGN USING GENETIC ALGORITHM AND CST METHOD”, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Tufan KUMBASAR**
Istanbul Technical University

Jury Members : **Asst. Prof. Dr. Farzad HASHEMZADEH**
Istanbul Technical University

Asst. Prof. Dr. Mahdi BARADARANNIA
Tabriz University

Date of Submission : **23 March 2025**

Date of Defense : **05 May 2025**





To those exploring and forging their own paths,



FOREWORD

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Tufan Kumbasar, for his unwavering support, guidance, and encouragement throughout this journey. His inspiring mentorship, patience, and profound expertise were instrumental in the successful completion of this work. I am eternally grateful for his dedication and the intellectual freedom he granted me to explore and grow.

I extend my sincere thanks to my colleague and friend, Dr. Ali Rıza Yılmaz, for his invaluable advice, constructive feedback, and steadfast support during this project. His insights and encouragement motivated me to pursue this endeavor with confidence. I am equally indebted to my colleague and friend, Dr. Yusuf Kahraman, for his unwavering moral support throughout this journey.

My heartfelt appreciation goes to my family, whose unconditional love, patience, and encouragement have been my anchor. Their unwavering belief in me, even during the most demanding phases of this work, gave me the strength to persevere.

Over my 15-year career as a professional engineer—shaped by my journey through aviation, automotive, and electronic industries—I have immersed myself in challenges that demanded continuous growth and innovation. This journey taught me firsthand the struggle of finding reliable, actionable knowledge in fast-evolving fields. My aim was to serve curious colleagues exploring the world with wonder, as well as young peers forging their own paths, by bridging the gap between theory and practice. To achieve this, I sought to transform complexity into understanding, grounding the work in real-world applications. Throughout this project, I strove to present methodologies with precision, supported by extensive literature reviews synthesizing the most impactful studies.

It is my sincere hope that this work not only serves as a meaningful resource but also ignites the same spark of curiosity and intellectual fulfillment in its readers that drove its creation. May it be a catalyst for discovery, innovation, and meaningful dialogue.

May 2025

Erkan BİÇER
(Metallurgical and Materials Engineer)

TABLE OF CONTENTS

	<u>Page</u>
FOREWORD	ix
TABLE OF CONTENTS	xii
ABBREVIATIONS	xiii
SYMBOLS	xv
LIST OF TABLES	xvii
LIST OF FIGURES	xx
SUMMARY	xxi
ÖZET	xxiii
1. INTRODUCTION	1
1.1 Purpose of Thesis	3
1.2 Literature Review	4
1.2.1 Optimization of centrifugal fan design	6
1.2.2 Computational and experimental approaches in fan design	7
1.2.3 Optimization techniques for performance enhancement	9
1.2.4 Dissipation function and energy loss evaluation	9
1.2.5 Integrated performance evaluation for fan design	10
1.2.6 Conclusion	10
1.3 Hypothesis	11
2. EXPLORING PARAMETERIZATION AND THE CST METHOD FOR DESIGN OPTIMIZATION	13
2.1 Introduction	13
2.2 Background	14
2.3 Conventional Methods	16
2.3.1 Discrete approach	16
2.3.2 Analytical approach	17
2.3.3 Geometric approach	19
2.3.4 Polynomial and spline approach	20
2.4 CST Method	21
2.4.1 Two-dimensional CST	21
3. OPTIMIZATION SETUP AND GENETIC ALGORITHM CONFIGURATION	27
3.1 Introduction	27
3.2 Classical GAs	28
3.3 Genetic Algorithm Configuration	29
3.4 Genetic Operators	33
3.4.1 Encoding schemes	33
3.4.2 Selection methods	35
3.4.3 Crossover operators	38
3.4.4 Mutation operators	41
3.5 Challenges	44
3.5.1 Choosing initial population	44
3.5.2 Premature convergence	45
3.5.3 Choosing effective fitness functions	46
3.5.4 Crossover and mutation level	46
3.5.5 Selection of encoding schemes	46
4. TURBULENCE MODEL SELECTION AND APPLICATION IN COMPUTATIONAL FLUID DYNAMICS	49
4.1 Turbulence Models	49
4.1.1 $k-\varepsilon$ models	49
4.1.2 Spalart-Allmaras model	51
4.1.3 $k-\omega$ model	51
4.1.4 Transition sst model	52

4.1.5	Reynolds stress model (RSM)	52
4.1.6	Scale-adaptive simulation (SAS) model	52
4.1.7	Detached eddy simulation (DES).....	52
4.2	Near-Wall Treatment.....	53
4.2.1	Wall functions vs. near-wall model	54
4.2.2	Wall function strategies	56
4.2.3	Enhanced wall treatment (EWT)	56
4.2.4	Menter-Lechner ε -equation (ML- ε).....	57
4.3	Solver Theory	58
4.3.1	Discretization schemes.....	59
4.3.2	Pressure-based solver details	59
4.3.3	Hybrid initialization.....	60
5.	CENTRIFUGAL FAN OPTIMIZATION FOR ENHANCED PERFORMANCE.....	63
5.1	Software in the Loop for Optimization Process	63
5.2	Result	68
6.	CONCLUSION	79
	REFERENCES	81
	CURRICULUM VITAE	91



ABBREVIATIONS

CFD	: Computational Fluid Dynamics
GA	: Genetic Algorithm
CST	: Class-Shape-Transformation
NURBS	: Non-Uniform Rational B-Splines
AF	: Airfoil
BI	: Backward-Inclined
BC	: Backward-Curved
FC	: Forward-Curved
RT	: Radial Tip
RB	: Radial Blades
CAD	: Computer-Aided Design
FEA	: Finite Element Analysis
PDE	: Partial Differential Equation
PARSEC	: Parametric Airfoil Section
NACA	: National Advisory Committee for Aeronautics
Bezier	: Bezier Representation
B-spline	: Basis Spline Curve Representation
ACO	: Ant Colony Optimization
PSO	: Particle Swarm Optimization
RWS	: Roulette Wheel Selection
SUS	: Stochastic Universal Sampling
PMX	: Partially Matched Crossover
TSP	: Traveling Salesman Problem
PPX	: Precedence Preserving Crossover
RCX	: Reduced Surrogate Crossover
LX	: Laplace Crossover
MPTM	: Mutation (MPTM/NUM) Operator
LP	: Linear Programming
RANS	: Reynolds-Averaged Navier-Stokes
URANS	: Unsteady Reynolds-Averaged Navier-Stokes
LES	: Large Eddy Simulation
DES	: Detached Eddy Simulation
SAS	: Scale-Adaptive Simulation
RSM	: Reynolds Stress Model
SST	: Shear-Stress Transport
BSL	: Baseline



SYMBOLS

k	: Turbulent kinetic energy
ϵ	: Turbulence dissipation rate
ω	: Specific dissipation rate
J	: Objective function
$f_j(x)$	: Shape (or perturbation) function
t_1	: Adjustable parameter in the Hicks–Henne function
t_2	: Exponent parameter in the Hicks–Henne function
$y(x)$	: Airfoil shape function
y_{basis}	: Baseline shape
$\bar{\alpha}_j$	: Weight coefficient for shape perturbation
$\bar{c}_i$	: Coefficient vector in polynomial representation
p	: Polynomial (or Bezier) order
A_i	: Control points in Bezier/CST representation
$B_{i,p}(x)$	: Bernstein polynomial of degree p
$P(u)$	: B-spline curve
$\bar{A}_i$	: Control points (polygon) in B-spline representation
$N_{i,p}(u)$	: B-spline basis functions
u	: Parametric variable in B-spline representation
n	: Number of control points minus one
$\zeta(\psi)$	: Nondimensional vertical coordinate (z/c)
ψ	: Nondimensional horizontal coordinate (x/c)
ζ_T	: Nondimensional trailing edge thickness ($\Delta\zeta_{TE}/c$)
$C_{N_2}^{N_1}(\psi)$	: Class function in CST method
N_1, N_2	: Exponents defining the airfoil class
$S(\psi)$	: Shape function in CST method
R_{LE}	: Leading-edge radius
β	: Boat-tail angle
Δz_{TE}	: Trailing edge vertical displacement
$S_u(\psi)$	: Shape function for the airfoil upper surface
$S_l(\psi)$	: Shape function for the airfoil lower surface
Au_i	: Coefficients for the upper surface shape function
Al_i	: Coefficients for the lower surface shape function
$\zeta_u(\psi)$	: Nondimensional vertical coordinate for the upper surface
$\zeta_l(\psi)$	: Nondimensional vertical coordinate for the lower surface
ζ_{uT}	: Nondimensional trailing edge thickness for the upper surface
ζ_{lT}	: Nondimensional trailing edge thickness for the lower surface
$\Delta\zeta_{uTE}$	: Upper surface trailing edge thickness
$\Delta\zeta_{lTE}$	: Lower surface trailing edge thickness
ρ	: Air density (kg/m^3)



LIST OF TABLES

	<u>Page</u>
Table 2.1 : Definitions of typical shape functions [71].	19
Table 3.1 : An analysis of several encoding schemes.	34
Table 3.2 : Comparing various selection methods.	35
Table 3.3 : Comparison of crossover techniques.	41
Table 3.4 : Comparison of mutation techniques.	42
Table 3.5 : The optimal combination of different operators in the ideal setting. ..	43
Table 5.1 : Optimization variables.	67
Table 5.2 : Sensitivity analysis of CST variables.	70
Table 5.3 : Optimization result.	73
Table 5.4 : Performance values from randomly selected samples.	77



LIST OF FIGURES

	<u>Page</u>
Figure 1.1 : Airfoil as used in a backwardly curved centrifugal fan blade [5].	5
Figure 1.2 : Airfoil as used in a radial-tip centrifugal fan blade [5].	5
Figure 1.3 : Six blade shapes commonly used in centrifugal fans [5].	5
Figure 2.1 : The discrete approach [69].	16
Figure 2.2 : Hicks-Henne functions for $t_2=5$ [70].	18
Figure 2.3 : First (a) and second (b) kind Chebyshev polynomials; first (c) and second (d) kind Legendre functions [71].	18
Figure 2.4 : The geometric approach [72].	19
Figure 2.5 : An airfoil defined via a Bezier representation [69].	20
Figure 2.6 : Typical wing airfoil section [14].	21
Figure 2.7 : 2nd order [14].	23
Figure 2.8 : 3rd order [14].	23
Figure 2.9 : 7th order [14].	23
Figure 2.10 : 15th order [14].	23
Figure 2.11 : Terms for Bernstein polynomials for $p = 5$ [14].	24
Figure 2.12 : Airfoil for $C_{1,0}^{0.5}$ [14].	24
Figure 2.13 : $\bar{P}_i = \{1, 1, 1, 1, 1, 1\}$ (dashed line) and $\bar{P}_i = \{1, 2, 0.5, 2, 0.5, 1\}$ (solid line) were used to modify the airfoil [14].	24
Figure 2.14 : Airfoil with a rounded nose, $N_1 = 0.5$, $N_2 = 1.0$ [14].	25
Figure 2.15 : Airfoil with smooth curvature, $N_1 = 0.5$, $N_2 = 0.5$ [14].	25
Figure 2.16 : Biconvex airfoil featuring symmetric curvature, $N_1 = 1.0$, $N_2 = 1.0$ [14].	25
Figure 2.17 : Duct shape with minimal curvature effects, $N_1 = 0.001$, $N_2 = 0.001$ [14].	25
Figure 2.18 : Low-drag projectile exhibiting streamlined design, $N_1 = 0.75$, $N_2 = 0.25$ [14].	25
Figure 2.19 : Wedge-shaped airfoil with a sharp trailing edge, $N_1 = 1.0$, $N_2 = 0.001$ [14].	25
Figure 3.1 : Metaheuristic algorithms classification [76].	27
Figure 3.2 : The main steps of the GA.	28
Figure 3.3 : Operators used in the GA [76].	33
Figure 3.4 : Stochastic universal sampling approach.	36
Figure 3.5 : Single-point crossover [76].	38
Figure 3.6 : Two-point and k-point crossover [76].	39
Figure 3.7 : Uniform crossover [76].	39
Figure 3.8 : Partially matched crossover (PMX) [76].	39
Figure 3.9 : Shuffle crossover [76].	40
Figure 3.10 : Local and global optima [76].	45

Figure 4.1 : Resolved wake structures: SST-SAS vs. URANS [83].	53
Figure 4.2 : The ANSYS Fluent near-wall region subdivisions [83].	54
Figure 4.3 : The ANSYS Fluent near-wall treatments feature [83].	55
Figure 4.4 : Pressure-based solution methods [83].	57
Figure 4.5 : Density-based solution methods [83].	58
Figure 5.1 : Software in the loop.	63
Figure 5.2 : O-type mesh with five layers around the blade.	64
Figure 5.3 : H-type quadrilaterals at the inlet and outlet.	65
Figure 5.4 : Typical wing airfoil section.	66
Figure 5.5 : Best and mean values of cost function during optimization.	71
Figure 5.6 : Best, worst, and mean values of cost function.	71
Figure 5.7 : Fitness scaling of cost function during optimization.	72
Figure 5.8 : Score histogram of cost function after GA optimization.	72
Figure 5.9 : Contour image of velocity magnitude.	73
Figure 5.10 : Contour image of total pressure.	74
Figure 5.11 : Vector image of velocity magnitude.	74
Figure 5.12 : Vector image of velocity magnitude.	75
Figure 5.13 : Detailed contour image of velocity magnitude.	75
Figure 5.14 : Detailed vector image of velocity magnitude.	75
Figure 5.15 : Objval distribution across design parameters.	76
Figure 5.16 : Benchmarking selected airfoil shapes.	77

OPTIMIZATION OF CENTRIFUGAL FAN DESIGN USING GENETIC ALGORITHM AND CST METHOD

SUMMARY

Centrifugal fans, integral components in constrained spaces such as civil aviation ovens, are designed to meet stringent performance requirements. Their unique operational mechanism involves drawing air axially, redirecting it by 90 degrees, and passing it through the blades in a spiral trajectory. This redirection, combined with the application of centrifugal forces, significantly influences airflow and contributes to their ability to generate high total pressure. Consequently, centrifugal fans are ideal for high-pressure applications where reliable and efficient airflow is paramount.

Centrifugal fans possess different characteristics compared to axial fans. Initially, the airflow begins in an axial direction, but before reaching the blades, it undergoes a 90-degree change in direction. Within these fans, the blades guide the air in a spiral pattern, which combines with centrifugal force. This centrifugal force drives the airflow outward and determines its direction. For forward-curved blades, the change in airflow direction has a significant impact on performance. Centrifugal fans generate higher total pressure, and the lift force of the blade profile plays a limited role. Backward-curved blades do not produce traditional lift forces. Their convex side acts as the pressure side, which is an unconventional feature for blade profiles. Additionally, a blunt leading-edge design enhances efficiency and strengthens structural integrity.

Centrifugal fans are notable for their high efficiency and relatively low noise levels. Thanks to their robust structural strength, they can withstand pressures up to 7,500 Pa while maintaining stable performance. Commonly used in clean air, gas, and ventilation systems, they are also highly effective in systems with variable resistance. The transitions between blades are gradually widened. In most cases, blades are manufactured as hollow steel profiles welded to the backplate and shroud. In some cases, the fan wheel may also be fabricated using aluminum casting or CNC machining methods. It is well known that the performance of centrifugal fans is largely dependent on blade shape.

In this context, Bezier curves, B-spline curves, and NURBS curves are frequently employed for the parametric design of fan blades. However, with the increasing demand for high-speed or curved blade shapes, lower-order curves often fail to adequately define these designs. The Class-Shape Transformation (CST) method, proposed by Kulfan, is a highly accurate modeling approach capable of achieving precise shape designs with fewer variables. This method is widely applied across various aerospace fields, such as blade profile optimization, compressor blade design, and aircraft design, and is also beneficial for the parametric design of fan blades.

The research leverages the CST method to parameterize the fan geometry. The CST method is chosen for its capability to accurately describe complex geometrical shapes using a reduced number of variables, thereby simplifying the optimization process without compromising the fidelity of the design. By integrating CST with Genetic Algorithms (GA), this study seeks to explore a broader design space efficiently and uncover optimal configurations that maximize fan performance.

A critical component of the optimization process is the use of advanced numerical simulations. ANSYS Fluent, a robust computational fluid dynamics (CFD) tool, was employed to conduct fluid flow analyses and gather performance data for various design iterations. These simulations provided insights into the aerodynamic behavior of the fan, highlighting the impact of geometric changes on performance metrics such as total pressure, efficiency, and flow rate.

To facilitate an efficient optimization workflow, a communication loop was established between MATLAB's Genetic Algorithm Toolbox and ANSYS Fluent. This loop enabled seamless data exchange and automated the iterative process of design evaluation and optimization. Key parameters, such as blade angle, chord length, and hub-to-tip ratio, were defined and systematically varied within predefined limits to explore their influence on fan performance.

In this study, the blade design of a two-dimensional centrifugal fan was optimized through an integrated computational approach. Numerical simulations were performed using ANSYS software, with the necessary calculations to ensure robust analysis. Fan performance was evaluated based on Fluent CFD analysis, with the results systematically recorded and processed in Excel. The optimization process employed a GA implemented through MATLAB's built-in libraries. Parametric blade optimization was achieved using the CST method, the k-epsilon turbulence model, and the GA, forming an iterative computational loop between MATLAB and ANSYS. Following optimization, the blade designs were analyzed to identify the configuration with the most favorable parameters.

The study's findings underscore the significance of geometric parameterization and optimization in enhancing the performance of centrifugal fans. By systematically exploring the design space, the research achieved improvements in performance metrics, meeting the stringent requirements of civil aviation applications. The integration of the CST method and GA proved to be a powerful combination, offering a structured and efficient approach to fan design optimization.

In conclusion, this study provides a comprehensive framework for optimizing centrifugal fan designs, bridging the gap between traditional methodologies and modern computational tools. The results not only contribute to the field of fan design but also demonstrate the potential of advanced parameterization and optimization techniques in achieving superior performance in constrained engineering applications.

GENETİK ALGORİTMA VE CST YÖNTEMİ İLE SANTRİFÜJ FAN TASARIMININ OPTİMİZASYONU

ÖZET

Santrifüj fanlar, dar alanlarda, özellikle sivil havacılık fırınlarında kullanılan önemli bileşenlerdir ve yüksek performans gereksinimlerini karşılamak üzere tasarlanmıştır. Bu fanlar, havayı eksenel olarak çeker ve 90 derecelik bir yön değişikliğiyle kanatlardan geçirir. Bu süreçte hava spiral bir yol izler. Bu yönlendirme, santrifüj kuvvetleriyle birleşerek hava akışını etkiler ve yüksek toplam basınç üretimini sağlar. Dolayısıyla, santrifüj fanlar, güvenilir ve verimli hava akışı gereken yüksek basınçlı uygulamalar için idealdir.

Santrifüj fanların farklı özellikleri, eksenel fanlardan ayrılmasını sağlar. Başlangıçta hava akışı eksenel bir yönde ilerler, ancak kanatlara ulaşmadan önce 90 derecelik bir değişiklik geçirir. Kanatlar, havayı spiral bir desenle yönlendirir ve santrifüj kuvveti, hava akışını dışarı doğru sürükler. Öne eğik kanatlarda bu değişiklik performansı önemli ölçüde etkiler. Santrifüj fanlar, daha yüksek toplam basınç üretirken, kanat profilinin kaldırma kuvveti sınırlı bir rol oynar. Geriye eğik kanatlar ise geleneksel kaldırma kuvveti üretmez. Bu tür kanatların konveks tarafı basınç yüzeyi olarak işlev görür ve bu, kanat profilleri için alışılmadık bir özelliktir. Ayrıca, küt burun tasarımı verimliliği artırır ve yapısal bütünlüğü güçlendirir.

Bu fanlar, yüksek verimlilikleri ve nispeten düşük gürültü seviyeleri ile dikkat çeker. Sağlam yapısal dayanıklılıkları sayesinde, 7500 Pa'ya kadar basınçlara dayanırken kararlı performans gösterirler. Temiz hava, gaz ve havalandırma sistemlerinde yaygın olarak kullanılırlar ve değişken dirençli sistemlerde de oldukça etkilidirler. Kanatlar arasındaki geçişler kademeli olarak genişletilir. Çoğu durumda, kanatlar, arka plaka ile kapak levhasına kaynaklanmış içi boş çelik profiller olarak üretilir. Bazı durumlarda, fan ünitesi alüminyum döküm veya CNC işleme yöntemleriyle de imal edilebilir. Fan performansı büyük ölçüde kanat şekline bağlıdır.

Optimizasyon ve dönüşüm süreçleri, kanatçığın tasarımına yönelik modelleme yöntemlerinin ve optimizasyon algoritmalarının belirlenmesiyle başlar. CST (Class-Shape Transformation) modelleme yöntemi ve Genetik Algoritma (GA), kanatçık tasarımının ve parametrelerin optimizasyonu için kullanılmıştır. Bu süreç için MATLAB ve ANSYS Fluent kullanılarak gerekli hesaplamalar gerçekleştirilmiştir. CST kodu, MATLAB'da GA kodu ile entegre edilmiştir ve optimizasyon süreci bu entegre sistem üzerinden yürütülmüştür. GA optimizasyon sürecinde, Fluent tabanlı akışkanlar dinamiği yazılımı kullanılarak, üretilen iki boyutlu kanatçıklar için Toplam Basınç (Pa) [TP] ve Hava Akış Hızı (m^3/s) [Q] hesaplanmıştır.

GA, doğal seçim sürecine dayanan ve evrimsel algoritmaların bir parçası olarak kabul edilen bir optimizasyon yöntemidir. Doğal evrim teorisinden esinlenmiş bir

metasezgisel tekniktir ve karmaşık optimizasyon problemlerinde optimal çözümler elde etmek için kullanılır. Bu algoritma, problem alanını kodlanmış bir temsil alanına dönüştürür. Uygunluk fonksiyonunu değerlendirme temeli olarak kullanır ve kodlanmış popülasyonu evrimsel bir temel olarak kabul eder. Popülasyondaki bireysel bit dizilerinin genetik manipülasyonu, seçim ve diğer genetik mekanizmalar ile birleştirilerek yinelemeli bir süreç oluşturur. Bu süreçte, kodlanmış bit dizisinin bir veya birkaç biti rastgele seçilerek çaprazlama yapılır ve uygunluğa göre yeni bir bit dizisi kombinasyonu oluşturulur. Böylece, yeni nesil kod kombinasyonu eski nesil kod kombinasyonundan üstün hale gelir ve bu, son olarak problem alanına yansıtılarak optimal çözüme yaklaşımları sağlanır. Genetik algoritma, standart optimizasyon algoritmalarına uygun olmayan, kesintili, türevlenemeyen, stokastik veya yüksek derecede karmaşık olan hedef fonksiyonlara sahip çeşitli optimizasyon problemlerini çözmek için kullanılabilir. Ayrıca, bazı bileşenleri tamsayı olması gereken karmaşık programlama problemleriyle de başa çıkabilir.

CST yöntemi, analitik bir sınıf fonksiyonu $C(\psi)$ ile parametrik bir şekil fonksiyonu $S(\psi)$ birleştirir. Sınıf fonksiyonu, temel bir şekil sınıfını belirlerken, şekil fonksiyonu bu temel şekil etrafındaki varyasyonu tanımlar. Bu yaklaşım sayesinde, bir kanat kesitinin yuvarlak burnu ve keskin arka kenarı gibi belirli şekil sınıflarına özgü özellikler sınıf fonksiyonu içinde tanımlanabilir. Böylece, aynı şekil fonksiyonu farklı şekil sınıflarının optimizasyonunda kullanılabilir.

Araştırma, CST yöntemini kullanarak fan geometrisini parametrize etmektedir. Bu yöntem, karmaşık geometrik şekilleri daha az değişkenle doğru bir şekilde tanımlama yeteneği nedeniyle tercih edilmiştir. CST ile GA entegrasyonu, geniş bir tasarım alanını verimli bir şekilde keşfetmek ve fan performansını en üst düzeye çıkaracak optimal konfigürasyonları ortaya çıkarmak amacıyla kullanılmıştır.

ANSYS Fluent, çeşitli tasarım iterasyonları için akışkanlar dinamiği analizleri gerçekleştirmek ve performans verilerini toplamak üzere kullanılan güçlü bir sayısal simülasyon aracıdır. Bu simülasyonlar, fanın aerodinamik davranışına ilişkin önemli bilgiler sağlamış ve geometrik değişikliklerin performans ölçütleri üzerindeki etkisini vurgulamıştır. MATLAB GA Araç Kutusu ile ANSYS Fluent arasında bir iletişim döngüsü oluşturulmuş ve bu döngü, tasarım değerlendirmesi ve optimizasyon sürecini otomatikleştirmiştir. Kanat açısı, giriş uzunluğu ve göbek-ucu oranı gibi anahtar parametreler tanımlanmış ve performans üzerindeki etkilerini keşfetmek için sistematik olarak belirlenen sınırlar içinde değiştirilmiştir.

Yapılan optimizasyon çalışması, kanatçık tasarımının ve diğer parametrelerin maliyet fonksiyonu üzerindeki kritik etkisini vurgulamaktadır. Özellikle, tasarım parametrelerinin optimize edilmesi performans artışı sağlamak için hayati öneme sahiptir. Bu bulgular, fan tasarımında kullanılan modellerin ve yöntemlerin doğruluğunu değerlendirmek ve daha verimli sistemler geliştirmek için önemli bir referans noktası sunmaktadır. Ayrıca, farklı parametreler ve tasarım özellikleri arasındaki ilişkilerin incelenmesi, daha iyi performans sağlayan fan sistemlerinin geliştirilmesine katkıda bulunacaktır.

Sonuç olarak, bu çalışma, santrifüj fan tasarımlarının optimize edilmesi için kapsamlı bir çerçeve sunmakta ve geleneksel yöntemler ile modern hesaplama araçları arasındaki boşluğu doldurmaktadır. Elde edilen sonuçlar, sadece fan tasarımı alanına

katkıda bulunmakla kalmayıp, aynı zamanda kısıtlı mühendislik uygulamalarında üstün performans elde etmek için gelişmiş parametrelendirme ve optimizasyon tekniklerinin potansiyelini göstermektedir. Bu bulgular, gelecekte enerji verimliliği yüksek sistemlerin geliştirilmesine öncülük edebilir.





1. INTRODUCTION

Centrifugal fans are fluid machinery devices designed to move gas or air with a modest pressure increase, relying on impeller rotation that directs flow radially [1]. Compared with axial fans, centrifugal fans achieve higher pressure rises over a short axial distance, as the work imparted by the impeller is mainly due to centrifugal flow. Widely applied in power stations, ventilation systems, cooling devices, and various industrial processes, these fans can require substantial energy. Their internal flow is inherently complex due to centrifugal and Coriolis forces, as well as pressure gradients. These factors lead to unsteady flow characteristics, which can cause structural vibrations and aerodynamic noise [2]. Airflow begins in an axial direction, but before reaching the blades, it undergoes a 90° change in direction [3]. Inside these fans, the blades guide the air in a spiral pattern, which, combined with centrifugal force, drives the airflow outward and dictates its direction. For forward-curved blades (Figure 1.2) this change in airflow direction strongly influences performance [2]. Centrifugal fans produce higher total pressure, and the blade's lift force is relatively limited. Backward-curved blades (Figure 1.1) do not create typical lift forces because their convex side acts as the pressure side, an unconventional feature for blade profiles. Additionally, a blunt leading edge improves efficiency and enhances structural integrity [4,5].

Biedermann [6] emphasized the need for robust yet efficient solutions in designing high-performance industrial fans, stressing that multiple disciplines such as aerodynamics, acoustics, vibroacoustics, and structural mechanics must be closely integrated to provide a customized multi-objective optimization tool for industrial centrifugal fans. Such an approach aims to minimize cost and material usage while ensuring safety and efficiency. Early centrifugal fan designs often follow simplified two-dimensional guidelines from Eck [7] or Bommers [8]. Casey [9] proposes an automated optimization technique using GA for impeller blades, focusing on aerodynamic performance, material stress, and vibrations.

It is well known that the performance of centrifugal fans is largely dependent on blade shape. In this context, Bezier curves [10], B-spline curves [11], and NURBS curves [12,13] are frequently employed for the parametric design of fan blades. However, with the increasing demand for high-speed rotation or curved blade shapes, lower-order curves often fall short of adequately defining these designs. The CST method, proposed by Kulfan [14], is a highly accurate modeling approach capable of achieving precise shape designs with fewer variables. This method is widely applied across various aerospace fields, such as blade profile optimization [15], compressor blade design [16], and aircraft design [17], and is also beneficial for the parametric design of fan blades [18]. Zhao [19] shows that blade height, blade outlet angle, blade inlet angle, blade count, and blade thickness strongly influence centrifugal fan aerodynamics. Such numerical simulations reduce development costs by allowing detailed flow analyses. Zhang [20] observes that increasing blade count raises turbulence intensity in the near wake, accelerating recovery and shifting tip and root vortices closer to the rotor. Ding [10] reports that enlarging the blade outlet angle can reduce impeller-channel vortices under particular conditions, although turbulence intensity may first decrease and then increase. Punnachet [21] finds that adjusting both blade inlet and outlet angles further refines fan performance. Kim [22] and Ottersten [23] conclude that altering the impeller's spatial position or geometry can also improve aerodynamic characteristics. Prezelj [24] demonstrates that a triangular impeller channel mitigates the “jet-wake” phenomenon, stabilizing the flow field and reducing vortices, especially when conical shapes are tested. Although many studies examine single parameters, Zhao [19] suggests that simultaneously optimizing multiple impeller parameters can yield greater efficiency gains.

Centrifugal fan performance relies heavily on stable internal flow fields. With rapid improvements in computational fluid dynamics (CFD), many studies have numerically explored unsteady flow phenomena in turbomachinery [25]. Researchers have employed the Realizable k - ε and SST k - ω turbulence models to investigate flow separation, secondary flows, and pressure fluctuations in fan impellers [26,27], sometimes adding oblique cuts to improve inlet conditions [28]. Analyses indicate that severe flow separation in blade passages significantly increases energy loss [29,30],

while rotor–stator interactions notably affect pressure pulsations [31]. Additionally, cutwater gap, vane arrangement, and geometry modifications can strongly influence performance [32]. High-velocity jet regions and low-energy wake areas often form a “jet-wake,” so minimizing the wake region and inhibiting boundary-layer separation are crucial for optimizing impellers [33]–[35]. Advanced approaches, including scale-adaptive simulation (SAS) and detached eddy simulation (DES), further clarify internal flow structures and guide performance improvements [36,37]. Adaptive blade or channel modifications can help reduce energy loss and noise while enhancing fan efficiency [21,38]–[40].

1.1 Purpose of Thesis

The blade design of a two-dimensional centrifugal fan was optimized through an iterative computational workflow. Numerical simulations were conducted via ANSYS software for robust analysis, and fan performance was assessed via Fluent CFD calculations, with results recorded and processed in Excel. A GA implemented through MATLAB’s built-in libraries provided the optimization framework. Parametric blade optimization employed the CST method, the $k-\varepsilon$ turbulence model, and a GA, generating a loop between MATLAB and ANSYS. After optimization, blade designs were examined to identify the most advantageous parameters, and the fundamentals and techniques of blade shape optimization were comprehensively discussed.

A key distinguishing factor is the aim to produce custom solutions for all boundary conditions under consideration, rather than limiting the analysis to a single design point or a single parameter change. By focusing on multiple boundary conditions, the approach ensures broader applicability and adaptability of the optimized blade geometry.

MATLAB’s built-in GA libraries managed the optimization, while ANSYS Fluent handled each CFD evaluation automatically. Unlike simpler parameterization methods (e.g., low-order polynomials or spline interpolations), the CST method employed here uses a ninth-order Bernstein polynomial, offering more direct geometric control of the airfoil with fewer parameters and more intuitive handling of the blade shape. This

expanded search space is especially useful when multiple parameters—such as blade angles, thickness, and the number of blades—must be simultaneously optimized.

Because each set of boundary conditions can yield a different “best” blade shape, the process leverages script-based automatic meshing and Fluent calculation for every new blade geometry generated by the CST method. As a result, the GA can evolve the design in fewer generations, converging on an optimal compromise between total pressure and airflow rate. Overall, this iterative, multi-parameter, and multi-boundary-condition approach seeks to surpass the limitations of traditional single-factor optimization and deliver a custom aerodynamic solution for centrifugal fans that can be extended to various industrial scenarios.

1.2 Literature Review

Centrifugal fans play a crucial role in various industrial applications, ranging from ventilation and material handling to thermal management in electronics. Their efficiency and performance depend on several design parameters, including impeller geometry, blade configurations, and volute casing optimization. This thesis explores the design and optimization methodologies for centrifugal fans, focusing on computational and experimental advancements that enhance their performance.

They consist of several primary components: fan housing, impellers, inlet and outlet ducts, drive shafts, and drive mechanisms [41]. The impeller plays a pivotal role in fan performance, as the gas enters from the side, turns 90 degrees, and accelerates due to centrifugal force as it exits through the fan housing [42]. The performance of centrifugal fans is typically analyzed using fan characteristic curves, which plot developed pressure and required power over a range of airflows [43].

Among the six types of centrifugal fan impellers—airfoil (AF), backward-inclined (BI), backward-curved (BC), forward-curved (FC), radial tip (RT), and radial blades (RB)—airfoil blades (Figure 1.3) are considered the most efficient [4]. The development of a numerical design procedure, coupled with CFD optimization, has significantly improved the design of centrifugal fans with airfoil blades. CFD optimization enables the reduction of volute clearances by 10-14% and increases the cut-off height by 5%,

which mitigates recirculation near the outlet and improves efficiency [4]. Numerical simulations are crucial for refining parameters such as flow rate, static and total pressure, and fan speed, which influence blade angles and volute geometry [5,44].

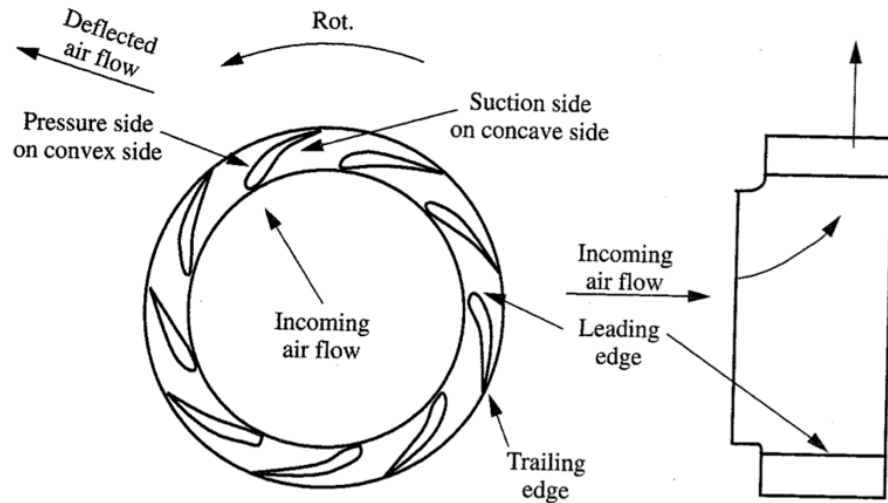


Figure 1.1 : Airfoil as used in a backwardly curved centrifugal fan blade [5].

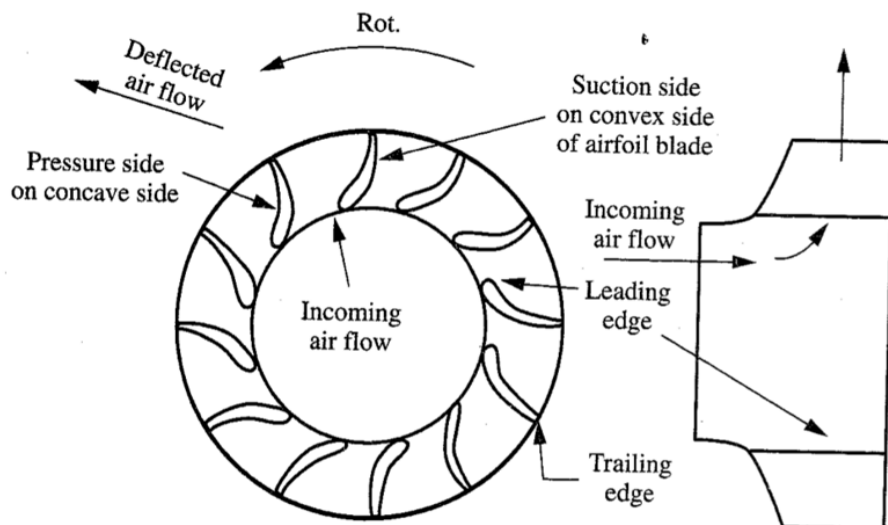


Figure 1.2 : Airfoil as used in a radial-tip centrifugal fan blade [5].



Figure 1.3 : Six blade shapes commonly used in centrifugal fans [5].

Experimental studies complement computational analyses by validating design improvements under real-world conditions. For instance, forward-curved blades (Figure 1.2) exhibit distinct characteristics in head rise and head loss, which can be optimized using GA [45]. Similarly, backward-inclined fans (Figure 1.1) commonly used for their quiet and reliable operation, undergo performance evaluation under different operating conditions using combined experimental and numerical approaches [44].

The radial gap between the impeller exit and diffuser entry also affects performance. Studies using particle image velocimetry (PIV) and CFD tools have demonstrated that an optimal radial gap improves dynamic and static heads, leading to higher efficiency [46]. Moreover, experimental investigations of tip clearance and impeller enlargement reveal significant impacts on fan efficiency. Enlarging the impeller diameter by 5-10% can improve airflow and pressure characteristics, though it requires careful alignment with design pressure and flow rate [47].

CFD simulations have further facilitated the integrated design of centrifugal fans for specialized applications. For example, small centrifugal fans designed for cooling laptop computers achieve high efficiency within constrained spaces [44]. By employing advanced turbulence models and optimizing the linkage profile between the inlet duct and impeller inlet, researchers have enhanced total pressure and efficiency compared to traditional designs [48].

In conclusion, the integration of numerical design methodologies, CFD simulations, and experimental validations has revolutionized centrifugal fan design. These advancements enable the optimization of impeller geometry, blade configurations, and volute casings, leading to improved efficiency and performance across various applications. Future research should continue to explore multi-objective optimization techniques and innovative materials to further enhance fan performance.

1.2.1 Optimization of centrifugal fan design

The optimization of centrifugal fan design plays a crucial role in various engineering domains, particularly in applications such as cooling systems, HVAC, and industrial

ventilation. The overall efficiency of centrifugal fans, including their aerodynamic performance, noise levels, and power consumption, depends heavily on critical design parameters such as blade number, inner diameter, and geometric configurations.

In recent years, the increasing demand for improved performance and reduced energy consumption has intensified research into optimizing centrifugal fans, particularly in applications where space and energy efficiency are significant concerns. A variety of computational and experimental techniques have been used to enhance fan performance by refining these parameters, leading to advancements in fan aerodynamics, noise reduction, and overall system efficiency.

1.2.2 Computational and experimental approaches in fan design

The growing significance of centrifugal fan optimization has been emphasized, especially in the context of cooling applications for electronics, where power consumption and space constraints impose strict limitations on system design. The impact of upstream components such as heat sinks and computer enclosures, which block airflow and reduce thermal dissipation efficiency, has been observed. These findings are consistent with previous studies that explore airflow reduction and heat dissipation challenges in confined cooling environments [49].

Studies conducted in the 1970s have explored the transition of airflow from axial to radial at the fan inlet, a process that creates an inactive separation zone. This zone is critical in understanding flow dynamics and its relationship to fan efficiency. More recent works have further exemplified how combining numerical simulations with experimental data can enhance the accuracy of fan design, reducing the reliance on costly physical testing while ensuring design optimization [44,50].

To address these challenges, fan laws provide quantitative relationships to guide system optimization [5]. When the fan geometry remains constant, the volumetric flow rate and static pressure scale with the rotational speed according to:

$$\frac{Q_2}{Q_1} = \frac{RPM_2}{RPM_1} \quad (1.1)$$

$$\frac{TP_2}{TP_1} = \left(\frac{RPM_2}{RPM_1} \right)^2 \quad (1.2)$$

If the blade angle (α), the number of blades (n), and the operating speed remain the same ($RPM_1 = RPM_2$), and all corresponding dimensions scale with the same ratio (d_2/d_1), then:

$$\frac{Q_2}{Q_1} = \left(\frac{d_2}{d_1} \right)^3 \quad (1.3)$$

$$\frac{TP_2}{TP_1} = \left(\frac{d_2}{d_1} \right)^2 \quad (1.4)$$

When the blade angle and number remain constant but the rotational speeds differ ($RPM_1 \neq RPM_2$) along with a consistent geometric scaling ratio (d_2/d_1), the relations become:

$$\frac{Q_2}{Q_1} = \left(\frac{d_2}{d_1} \right)^3 \cdot \frac{RPM_2}{RPM_1} \quad (1.5)$$

$$\frac{TP_2}{TP_1} = \left(\frac{d_2}{d_1} \right)^2 \cdot \left(\frac{RPM_2}{RPM_1} \right)^2 \quad (1.6)$$

Under the special condition when $RPM_2/RPM_1 = d_1/d_2$ the scaling relationships simplify to:

$$\frac{Q_2}{Q_1} = \left(\frac{d_2}{d_1} \right)^2 \quad (1.7)$$

$$\frac{TP_2}{TP_1} = 1 \quad (1.8)$$

$$\frac{TP_2}{TP_1} = \frac{\rho_2}{\rho_1} \quad (1.9)$$

These equations illustrate how changes in fan speed and geometric scaling affect the performance metrics such as flow rate and pressure. They serve as essential guidelines in the design and optimization of centrifugal fans for various applications.

1.2.3 Optimization techniques for performance enhancement

Recent studies have increasingly focused on optimizing fan blade shapes, which play a pivotal role in the overall efficiency of the centrifugal fan. Various approaches, such as neural networks and unsteady numerical simulations, have been employed to predict aerodynamic noise, revealing the limitations of traditional turbulence models in capturing low-frequency noise variations. Further research has analyzed the correlation between blade passing frequency and noise levels, with findings indicating that certain geometric configurations can reduce noise generation without compromising airflow efficiency [51,52].

One approach has proposed the development of advanced blade shape design techniques using parameterization methods such as Bezier curves, B-splines, and NURBS. Despite their applicability in simple cases, these methods are often insufficient for designing high-rotation or high-curvature blades. As an alternative, CST method has been utilized for high-precision shape design, which is advantageous for optimizing fan blades in systems requiring high efficiency [18].

The integration of surrogate models, such as Kriging, has further aided in the optimization process by reducing the computational cost typically associated with numerical simulations. By applying Kriging models in conjunction with the NSGA-II, significant improvements in fan efficiency, reduction in aerodynamic noise, and energy losses have been achieved. This approach, combined with the dissipation function, enables a comprehensive evaluation of flow energy losses, providing a clearer picture of the performance gains achieved through design optimization. Specifically, the optimization yielded a 9.03% increase in outlet static pressure and a 2.35% increase in fan efficiency [18].

1.2.4 Dissipation function and energy loss evaluation

One of the key contributions of recent studies is the use of the dissipation function to analyze energy losses in fans. Unlike traditional metrics like total efficiency or pressure loss, the dissipation function offers a more granular understanding of flow losses by

indicating both their location and magnitude. This method has shown great potential in centrifugal compressors and turbines [53,54], where viscous forces dominate the energy loss mechanisms. The dissipation function has been employed to identify areas where energy loss occurred and improve the flow fields, especially at the leading edge of the fan blade. However, the trailing edge still presented challenges, indicating areas for future research in further optimizing fan performance [18].

1.2.5 Integrated performance evaluation for fan design

To obtain a comprehensive understanding of centrifugal fan performance, an integrated approach that combines experimental testing with numerical simulations has been proposed. This method assesses various performance metrics, including efficiency, torque, and noise, providing a more holistic view of fan performance. Results based on an 80 mm diameter backward-inclined centrifugal fan showed a close agreement between experimental and numerical results, with deviations of only 3% in volumetric flow rate and 5% in static pressure. These findings demonstrate the effectiveness of numerical simulations in predicting fan performance and highlight the importance of integrating different testing methods for more accurate evaluations [49].

1.2.6 Conclusion

Optimizing centrifugal fan design is a multifaceted challenge that requires the integration of advanced computational techniques and experimental validation. As the demand for more energy-efficient and noise-reduced systems grows, advancements in methods such as CFD, surrogate modeling (including Kriging), and noise prediction models are crucial for achieving higher performance in centrifugal fans. Future research should continue to explore the refinement of turbulence models, the use of artificial intelligence in optimization, and the application of energy loss analysis techniques like the dissipation function to further enhance fan performance under varying operational conditions.

1.3 Hypothesis

Integrating GA, CST methods, and advanced CFD analysis can advance centrifugal fan design beyond the current state of the art. GA enables multi-objective optimization, allowing for the automatic selection of optimal design parameters to enhance efficiency and performance. The CST method provides a powerful approach for precisely shaping impellers and volutes, enabling smoother flow characteristics and reducing energy losses. When combined with high-fidelity CFD simulations, these methods offer unprecedented insights into aerodynamic behavior, leading to breakthroughs in fan efficiency, noise reduction, and overall system performance. This integrated approach has the potential to surpass existing methodologies, unlocking new frontiers in centrifugal fan optimization.

Therefore, this thesis focuses on developing a Software-in-the-Loop (SiL) system that integrates GA and the CST method to optimize centrifugal fan design within predefined boundary conditions. These boundary conditions include blade number, inner diameter, rotation speed, blade angle, upper and lower blade surface coordinates. The optimization process iterates through various design configurations, running simulations in ANSYS using CFD analysis to evaluate performance. The obtained results are recorded, and the system continuously refines the design to converge on the optimal solution. This approach establishes a highly efficient, automated methodology for centrifugal fan optimization, surpassing conventional design techniques.



2. EXPLORING PARAMETERIZATION AND THE CST METHOD FOR DESIGN OPTIMIZATION

2.1 Introduction

In any shape optimization problem, it is pivotal to represent the geometry with a minimal set of parameters to reduce the computational burden during the optimization process. This procedure is referred to as “shape parameterization” or “geometric representation”. Interestingly, the choice of parameterization techniques varies across different fields. For instance, in Computer-Aided Design (CAD), Non-Uniform Rational B-splines (NURBS) are widely used. These NURBS curves and surfaces provide a concise and intuitive geometric model that computer programs can manage with relative efficiency. Nevertheless, their complex formulation may hinder the exchange of identical geometries between software platforms. Within Finite Element Analysis (FEA), simpler polynomials are commonly employed to mitigate such compatibility issues, though they are generally less intuitive and less efficient than some of the alternative approaches.

In the context of CFD, Kulfan [55] outlined several desirable attributes for any geometric representation method, such as smoothness, mathematical effectiveness, ease of handling large design spaces, and the capacity to enforce constraints. Traditional approaches often fail to deliver all these features simultaneously. As a solution, Kulfan [55] introduced the CST method, which meets these design criteria comprehensively. CST represents a shape by combining a “class function,” capturing the essential form category, and a “shape function,” defining any deviation from that category. The latter relies on Bernstein polynomials, whose coefficients act as design variables. In an extensive review, Samareh [56] concluded that polynomial and spline-based representations offer distinct advantages, including:

- i. The capability to manage surface representations,
- ii. Cross-disciplinary consistency,

- iii. Accommodation of significant geometric modifications,
- iv. The availability of sensitivity derivatives.

However, for dealing with novel fan blade configurations, an additional criterion becomes crucial: the ability to efficiently handle local shape changes. Despite meeting the requirements identified by Samareh [56], the CST method faces a significant drawback in efficiently managing local shape changes. This suggests the need to augment Samareh's list of conditions with a fifth one:

- v. Ability to accommodate local shape modifications

Local control significantly broadens the range of options for dealing with geometric constraints. A number of constraint-handling methods have been investigated. For instance, Juhsz and Hoffmann [57] controlled B-spline knot values to generate a virtual envelope that inhibits shape crossing, a strategy also employed by Berglund [58]. Tuli [59] achieved geometric constraints by segmenting the B-spline curve into discrete sections. Pourazady and Xu [60] as well as Celniker and Welch [61] introduced physical properties (e.g., stretching and bending stiffness) into B-spline curves, where the imposition of constraints was accomplished by minimizing deformation energy. While effective, these methods do not enable the translation of geometric constraints into bounds on design variables. Such translation, however, is desirable for limiting computational time requirements.

2.2 Background

In the current landscape, a diverse array of optimization tools is readily available in the market. However, employing these methods effectively during the design of intricate products calls for accurate mathematical formulations and, preferably, streamlined availability of derivatives with respect to a set of specified design variables. Since design variables frequently dictate the geometry, and computational analysis tools often depend on both the geometry itself and its derivatives, the geometric details of product models have garnered extensive interest in the literature. The crux of the matter is illustrated in the following sample equation, which is applicable, for instance, to an

objective function related to fluid flows:

$$\frac{\partial J}{\partial x} = \frac{\partial J}{\partial R_f} \frac{\partial R_f}{\partial R_s} \frac{\partial R_s}{\partial R_g} \frac{\partial R_g}{\partial x} \quad (2.1)$$

The optimization process relies on computing the derivative of the objective function, (J), with respect to the design variables, (x). In particular, it requires obtaining the derivative of the objective function with respect to the field discretization (R_f), the derivative of the field discretization with respect to the product surface model (R_s), the derivative of the surface model with respect to the geometric model (R_g), and finally, the derivative of the geometric model with respect to the design variables, (x). It is crucial that the parametric model remain both flexible and consistently linked to the discrete models utilized by diverse analysis tools.

Surveys by Haftka-Grandhi [62] and Ding [63] covered shape optimization up to 1986. Samareh's [56] comprehensive survey of parametric models up to 2001, focused on their use in combined structural and aerodynamic optimization, emphasized six key criteria for a successful parameterization process: (1) automation, (2) uniform geometry changes across domains, (3) availability of sensitivity derivatives (preferably in analytical form), (4) suitability for prevailing product development timelines, (5) direct integration with CAD environments, and (6) a manageable collection of design factors for reasonable duration regarding solutions.

Parameterization methods can be broadly classified into eight main groups: basis vector, domain element, partial differential equation (PDE), discrete, polynomial and spline, CAD-based, analytical, and freeform deformation (FFD). The FFD approach, initially introduced by Barr [64] and based on the parallelepiped, is both straightforward to implement and computationally efficient, although its parallelepiped foundation restricts deformation topology. Sederberg and Parry [65] addressed this drawback by developing FFD techniques for non-parallelepiped geometries. Another popular FFD method involves using trivariate parametric volumes, such as Bezier volumes and NURBS volumes.

Samareh [56] also advanced parametric modeling based on CAD by introducing an approach that parameterizes shape perturbations instead of parameterizing the geometry directly. In parallel, Samareh proposed a modified freeform deformation

(FFD) technique as an alternative parameterization method, combining flexibility and freedom with reduced design variables and analytical sensitivity derivatives.

Nadarajah's [66] survey in 2007 applied three shape parameterization techniques: mesh points, B-spline surfaces, and the CST method [14]. Morris [67] introduced a generic approach to parameterizing domain elements that uses radial basis functions (RBFs) for interpolation, allowing for hierarchical parameterization of complex geometries. Sevant [68] applied a partial differential equation (PDE) method to surface generation for aerodynamic design, achieving parameterization of complex geometries with limited variables while preserving a wide range of feasible shapes.

2.3 Conventional Methods

2.3.1 Discrete approach

Describing a two-dimensional shape in one of the most straightforward ways involves recording the coordinates of boundary points and connecting them with straight lines. This approach is referred to as the discrete approach. This method is depicted in Figure 2.1.



Figure 2.1 : The discrete approach [69].

The discrete approach, known for its straightforward implementation due to its compatibility with various grid types, encounters a notable drawback: Its demand for a substantial amount of nodes to achieve an ideal form. Attaining consistency, essential for minimizing numerical errors in flow calculations and preventing flow separation, becomes challenging when describing curved shapes with straight line segments. Such continuity can only be approximated and is challenging to maintain, often proving impossible during the course of optimization [69]. The initial simple implementation is compromised by the need for a complex set of constraints.

Furthermore, the discrete approach is seldom employed in representing airfoil shapes. This is primarily attributed to the non-analytic nature of the round nose of an airfoil, characterized by infinite derivatives at the nose. As a result, accurately capturing this feature through a discrete function becomes unfeasible.

2.3.2 Analytical approach

Within the analytical framework, the baseline shape undergoes enhancement by incorporating a linear addition of perturbation or shape functions. These functions, whether defined numerically or analytically, contribute to the overall shape modification. Hicks and Henne [70] exemplify this analytical approach with shape functions defined analytically, introducing the Hicks-Henne bump functions, which are outlined as follows:

$$f_j(x) = \left[\sin\left(\pi x^{\frac{\log(0.5)}{\log(t_1)}}\right) \right]^{t_2}, \quad 0 \leq x \leq 1 \quad (2.2)$$

Two adjustable parameters, t_1 and t_2 , are introduced to modify the design solution. In Figure 2.2, five Hicks–Henne functions are shown, corresponding to $t_2 = 5$. The design variables consist of weights α_j , while the disturbance functions $f_j(x)$ can be added to the baseline form as follows:

$$y(x) = y_{\text{basis}} + \sum_{j=1}^M \overline{\alpha}_j f_j(x) \quad (2.3)$$

Hicks–Henne functions merely represent one instance among the numerous analytical shape functions available. Additional illustrations are presented in Figure 2.3, where first (a) and second (b) kind Chebyshev polynomials; first (c) and second (d) type Legendre polynomials are shown [71]. Table 2.1 provides a full explanation of each of their definitions.

An inherent limitation of the analytical approach lies in the non-intuitive nature of altering a shape by manipulating variables associated with perturbation functions. While this characteristic renders the method well-suited for automated optimization routines, it may prove less ideal for manual airfoil design where intuitive adjustments play a pivotal role.

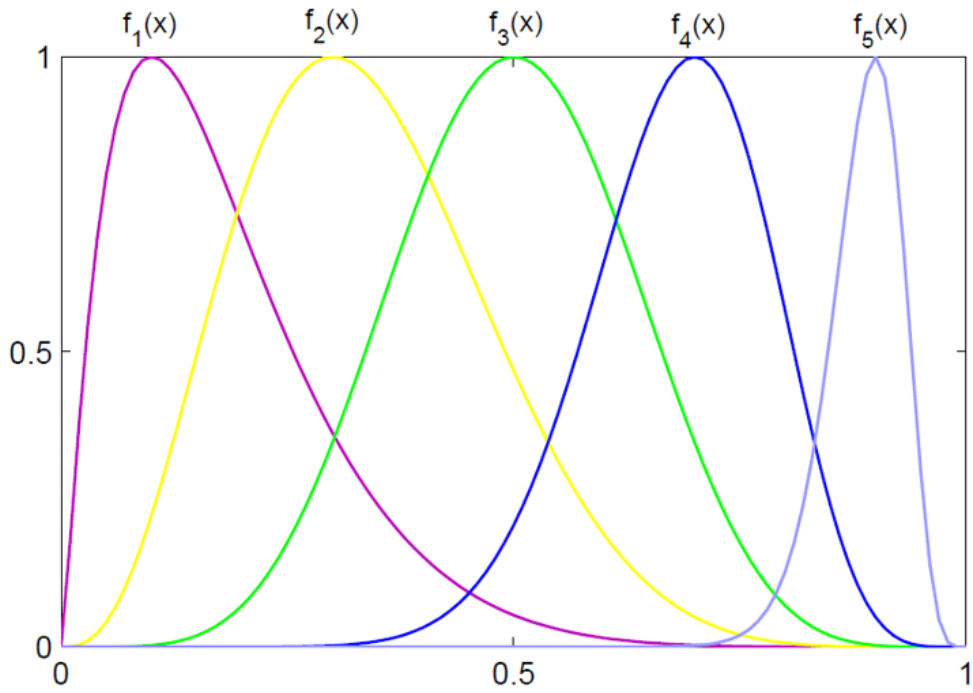


Figure 2.2 : Hicks-Henne functions for $t_2=5$ [70].

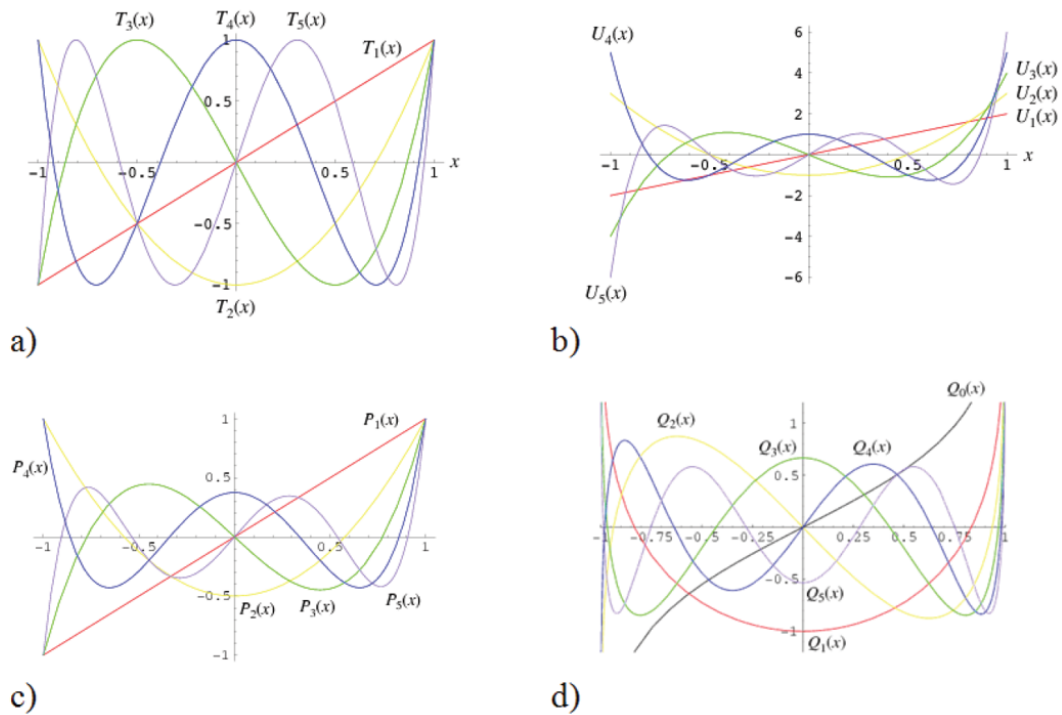


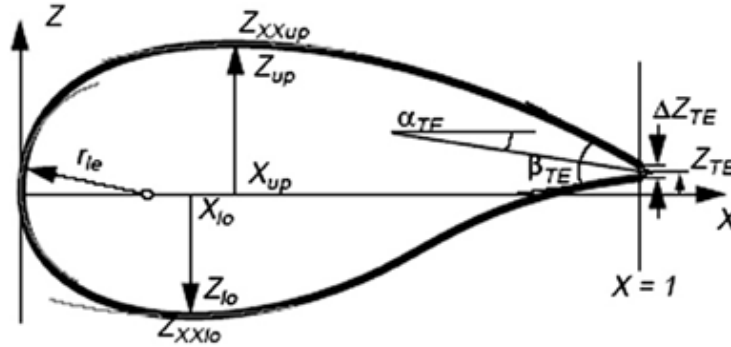
Figure 2.3 : First (a) and second (b) kind Chebyshev polynomials; first (c) and second (d) kind Legendre functions [71].

Table 2.1 : Definitions of typical shape functions [71].

	First kind	Second kind
Chebyshev	$T_0(x) = 1$	$U_0(x) = 1$
	$T_1(x) = x$	$U_1(x) = 2x$
	$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$	$U_{n+1}(x) = 2xU_n(x) - U_{n-1}(x)$
Legendre	$P_0(x) = 1$	$Q_0(x) = \frac{1}{2} \ln\left(\frac{1+x}{1-x}\right)$
	$P_1(x) = x$	$Q_1(x) = \frac{x}{2} \ln\left(\frac{1+x}{1-x}\right) - 1$
	$P_{n+1}(x) = \frac{(2n+1)xP_n(x) - nP_{n-1}(x)}{n+1}$	$Q_{n+1}(x) = \frac{(2n+1)xQ_n(x) - nQ_{n-1}(x)}{n+1}$

2.3.3 Geometric approach

A third method involves defining a shape based on physical parameters, including leading-edge radius, thickness-to-chord ratio, or trailing-edge angle. Figure 2.4 displays an airfoil created via the PARSEC geometric method [72]. In this case, the airfoil is characterized by 11 fundamental design parameters.

**Figure 2.4** : The geometric approach [72].

While this method is undeniably intuitive, it comes with a drawback. Each change in shape needs to be mapped onto a physical airfoil parameter, a process that is not always straightforward and may limit design flexibility. Furthermore, the design variables may lack a direct correlation with the flow properties, adding another layer of complexity to the matter.

2.3.4 Polynomial and spline approach

By employing a polynomial or spline representation, the necessary number of variables for creating a smooth shape can be significantly minimized. Although one can express a polynomial in its conventional power basis structure (see equation 2.4), determining how adjustments to the value of the coefficient vector $\bar{c}_i$ will alter the shape of the curve can prove challenging in such instances.

$$y(x) = \sum_{i=0}^p \bar{c}_i x^i \quad (2.4)$$

A more effective method for representing a curve involves utilizing the Bezier representation, defined below:

$$y(x) = \sum_{i=0}^p A_i B_{i,p}(x) \quad (2.5)$$

Here, a vector of control points is represented by A_i , whereas Bernstein polynomials of degree (p) are indicated by $B_{i,p}(x)$. This description is significantly more intuitive than the power base structure since the resulting curve tends to follow the control polygon formed by the points A_i . An example of an airfoil represented by a Bezier representation is shown in Figure 2.5.

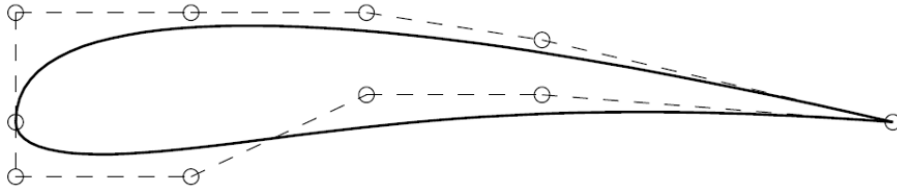


Figure 2.5 : An airfoil defined via a Bezier representation [69].

A drawback of employing Bezier curve surfaces when dealing with more intricate shapes is that the necessary degree of the Bernstein polynomials notably escalates, resulting in inefficiency during curve computation [69]. This stems from the global influence each Bernstein polynomial holds over the curve's shape. A remedy to this limitation is the utilization of a series of lower-order curves known as a B-spline. The description of a B-spline curve is as follows:

$$P(u) = \sum_{i=0}^n \bar{A}_i N_{i,p}(u) \quad (2.6)$$

Compared to equation 2.5, the Bernstein polynomials $B_{i,p}(x)$ are replaced by the basis functions of B-spline $N_{i,p}$, while the Bernstein coefficient vector A_i is replaced with the control polygon of B-spline $\bar{A}_i$. The parametric variable u replaces the Cartesian coordinate x . In particular, the order $p - 1$ of the different Bezier curves does not affect the control points number $(n+1)$.

2.4 CST Method

The preceding section outlined several parameterization techniques along with their advantages and disadvantages. This section delves into a method introduced by Kulfan [14], known as the CST method. The CST method integrates an analytical "class function" $C(\psi)$ with a parametric "shape function" $S(\psi)$. Here, the class function defines a fundamental category of shapes, while the shape function characterizes the deviations from this basic form. This formulation enables distinctive features inherent to a specific class, such as the rounded nose and sharp trailing edge of an airfoil, to be encapsulated within the class function $C(\psi)$. As a result, regardless of the class of forms being optimized, the same shape function can be used for any class.

2.4.1 Two-dimensional CST

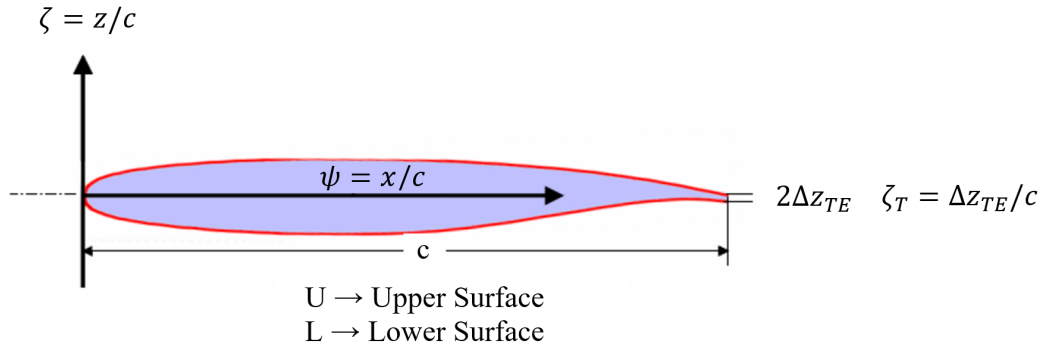


Figure 2.6 : Typical wing airfoil section [14].

Figure 2.6 displays a typical cross-section of a subsonic wing airfoil. Airfoils with round noses exhibit a zero-bound slope and a second component at the leading edge, as well as significant curvature variations across the airfoil surface. Describing such airfoils mathematically involves addressing a non-analytic complex function over the airfoil surface. As a result, a substantial number of "x,z" coordinates are

often necessary, coupled with a suitable interpolation approach, to achieve a valid mathematical or numerical model of the cambered airfoil surfaces [14].

The CST technique can be expressed symbolically as follows:

$$\zeta(\psi) = C_{N_2}^{N_1}(\psi) \cdot S(\psi) + \psi \cdot \zeta_T \quad (2.7)$$

In this context, $(\zeta = z/c)$, $(\psi = x/c)$ and $(\zeta_T = \Delta\zeta_{TE}/c)$. Here, “c” denotes the chord length in the x-direction, “x” represents the x-coordinate, and “z” signifies the z-direction. Notably, the dimensionless trailing edge thickness, ζ_T , is taken to be zero for simplicity. $C_{N_2}^{N_1}(\psi)$ and $S(\psi)$ correspond to the class and shape functions, respectively. Specifically, the following is how the class function is expressed:

$$C_{N_2}^{N_1}(\psi) = \psi^{N_1} (1 - \psi)^{N_2}, \quad 0 \leq \psi \leq 1 \quad (2.8)$$

For an airfoil featuring a NACA-type rounded leading edge and a sharp trailing end, the class function $C_{1,0}^{0.5}(\psi)$ is applicable. In other words, $C(\psi) = \sqrt{\psi}(1 - \psi)$. Regarding the shape function, Kulfan employs a representation based on Bernstein polynomials. According to the definition provided in Section 2.3.4, the shape function is formulated as:

$$S(\psi) = \sum_{i=0}^p A_i B_{i,p}(\psi) \quad (2.9)$$

The typical definition of the Bernstein polynomial terms is as follows:

$$B_{i,p} = \binom{p}{i} \psi^i (1 - \psi)^{p-i} \quad (2.10)$$

Here, ‘p’ represents the Bernstein polynomial order. A comparison between equation 2.9 and equation 2.10 highlights that the order of the polynomial consistently equals one less than the number of control points (or the vector length A_i). For example, the following phrases are identified using six control points ($p = 5$):

$$\begin{aligned}
B_{0,5} &= (1 - \psi)^5 \\
B_{1,5} &= 5(1 - \psi)^4 \psi \\
B_{2,5} &= 10(1 - \psi)^3 \psi^2 \\
B_{3,5} &= 10(1 - \psi)^2 \psi^3 \\
B_{4,5} &= 5(1 - \psi) \psi^4 \\
B_{5,5} &= \psi^5
\end{aligned} \tag{2.11}$$

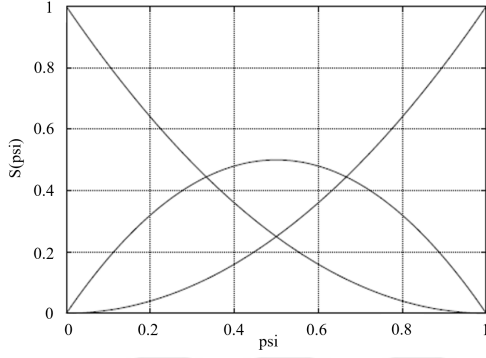


Figure 2.7 : 2nd order [14].

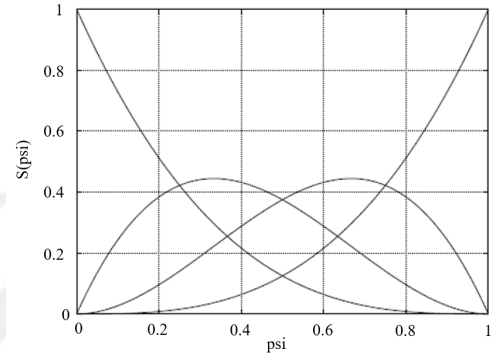


Figure 2.8 : 3rd order [14].

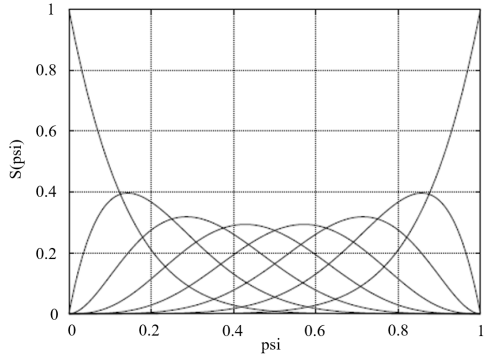


Figure 2.9 : 7th order [14].

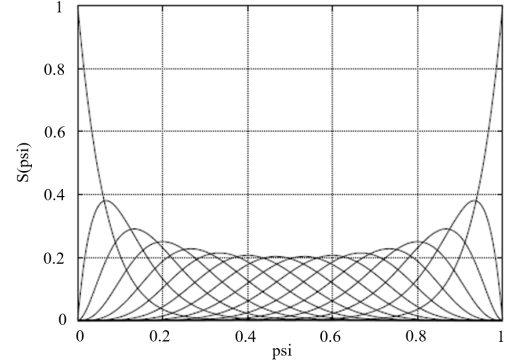


Figure 2.10 : 15th order [14].

When these polynomial terms are plotted, the results are depicted in Figures 2.6–2.11. An interesting property of these polynomials is that they guarantee a division of harmony, signifying that their sum always equals one, a property depicted by the thick black line in Figure 2.11. Essentially, if all the coefficients in the vector A_i are uniformly set at one, and if N_1 and N_2 are selected to be 0.5 and 1.0, then equation (2.9) will likewise equal one. Consequently, equation (2.7) simplifies to:

$$\zeta(\psi) = (\sqrt{\psi})(1 - \psi) \tag{2.12}$$

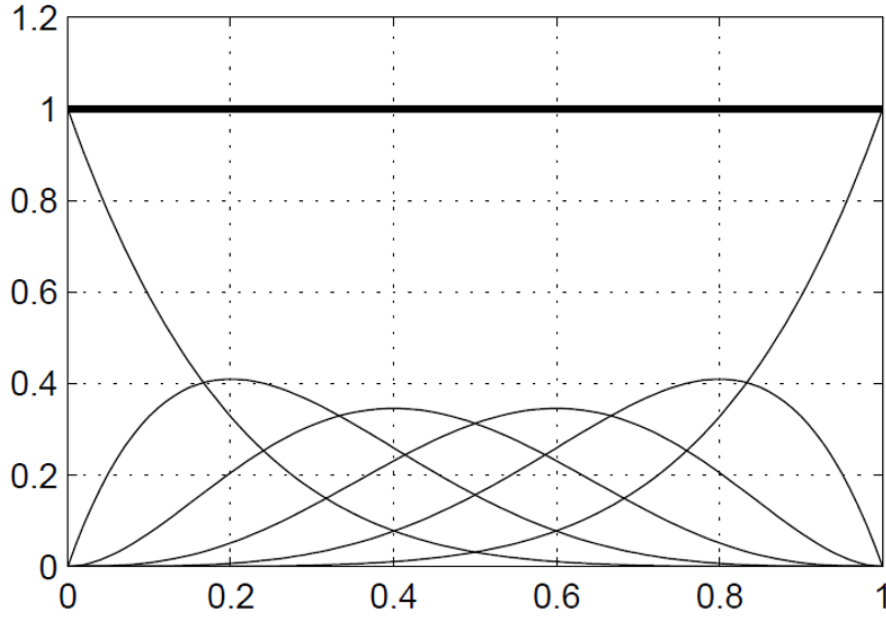


Figure 2.11 : Terms for Bernstein polynomials for $p = 5$ [14].

The unit $C_{1,0}^{0.5}$ airfoil, shown in Figure 2.12, is described by this equation. Alterations in the coefficient vector A_i induce variations in the form surrounding the unit airfoil. Figure 2.13 displays both the resulting and the deformed airfoil obtained by utilizing a coefficient vector of $\{1, 2, 0.5, 2, 0.5, 1\}$.

It is demonstrated that a variety of basic forms are mathematically defined by different combinations of N_1 and N_2 . Figures 2.14–2.19 showcase general classes that may be obtained by modifying the two exponents in equation (2.8).

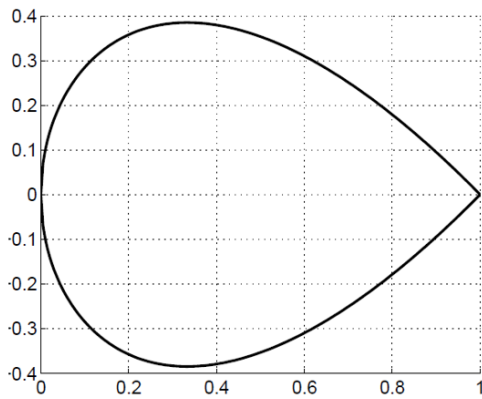


Figure 2.12 : Airfoil for $C_{1,0}^{0.5}$ [14].

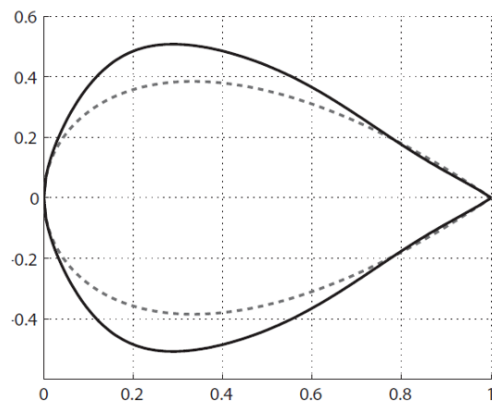


Figure 2.13 : $\bar{P}_i = \{1, 1, 1, 1, 1, 1\}$ (dashed line) and $\bar{P}_i = \{1, 2, 0.5, 2, 0.5, 1\}$ (solid line) were used to modify the airfoil [14].



Figure 2.14 : Airfoil with a rounded nose, $N_1 = 0.5$, $N_2 = 1.0$ [14].

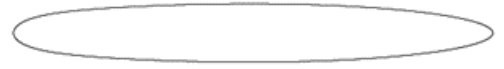


Figure 2.15 : Airfoil with smooth curvature, $N_1 = 0.5$, $N_2 = 0.5$ [14].

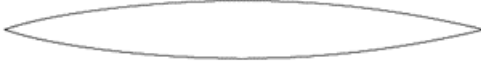


Figure 2.16 : Biconvex airfoil featuring symmetric curvature, $N_1 = 1.0$, $N_2 = 1.0$ [14].

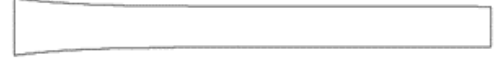


Figure 2.17 : Duct shape with minimal curvature effects, $N_1 = 0.001$, $N_2 = 0.001$ [14].

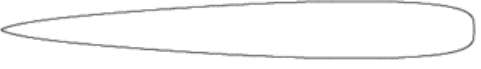


Figure 2.18 : Low-drag projectile exhibiting streamlined design, $N_1 = 0.75$, $N_2 = 0.25$ [14].



Figure 2.19 : Wedge-shaped airfoil with a sharp trailing edge, $N_1 = 1.0$, $N_2 = 0.001$ [14].

In this thesis, the parameters $N_1 = 0.5$ and $N_2 = 1.0$ are utilized. These values are suggested to fit the National Advisory Committee for Aeronautics (NACA) airfoils with round noses and pointed tails, making them commonly used for general airfoils with a round nose and an aft-end trailing edge [14]. These shapes are among the most common in aerodynamic design, underscoring the versatility and applicability of the CST method in optimizing airfoil performance. Round noses provide structural strength, while sharp trailing edges reduce turbulence, contributing to improved aerodynamic efficiency.

A noteworthy characteristic of the CST approach is that the shape function values at $\psi = 0$ and $\psi = 1$ are explicitly linked to the leading-edge nose radius and the boat-tail angle, respectively [14], as per equations (2.13) and (2.14). In the instance depicted in Figure 2.13, given that the initial and last values of the A_i vector are the same for both airfoils, the boat-tail angle and the leading-edge nose radius should be the same for both. A visual examination of the figure indeed supports this observation.

$$S(0) = \sqrt{\frac{2R_{LE}}{C}} \quad (2.13)$$

$$S(1) = \tan \beta + \frac{\Delta z_{TE}}{c} \quad (2.14)$$

The benefit of using the CST technique when coupled with Bernstein polynomials is that it produces a realistic and smooth form for any possible parameter vector A_i . However, it's important to note that any alteration in A_i will induce a global change in geometry, precluding localized modifications.

Once the overall shape function structure has been established, it becomes straightforward to use this framework to represent the cambered airfoil upper and lower surfaces. In particular, after determining the order (n) of the Bernstein polynomials, any desired shape modification can be achieved by suitably scaling the component shapes with coefficients that are chosen based on the study's objectives. In summary, the complete equations for the upper and lower surfaces of the airfoil, as defined by the overall shape function, are:

Airfoil upper and lower surfaces, respectively, are expressed as:

$$S_u(\psi) = \sum_{i=0}^p Au_i B_{i,p}(\psi) \quad (2.15)$$

$$S_l(\psi) = \sum_{i=0}^p Al_i B_{i,p}(\psi) \quad (2.16)$$

The values of Au_i and Al_i can be calculated using a least-squares fit to match a particular geometry, specified as an adjustable form variation, or predetermined in a design optimization algorithm. Ultimately, an airfoil form with a blunt nose with finite trailing edge thickness can be represented in Cartesian components (x, z) as [14]:

$$\zeta_u(\psi) = C_{N_2}^{N_1}(\psi) \cdot S_u(\psi) + \psi \cdot \zeta_{uT} \quad (2.17)$$

$$\zeta_l(\psi) = C_{N_2}^{N_1}(\psi) \cdot S_l(\psi) + \psi \cdot \zeta_{lT} \quad (2.18)$$

The ratios of trailing edge thickness for an airfoil's upper and lower surfaces can be defined as follows:

$$\zeta_{uT} = \Delta \zeta_{uTE} / c \quad (2.19)$$

$$\zeta_{lT} = \Delta \zeta_{lTE} / c \quad (2.20)$$

3. OPTIMIZATION SETUP AND GENETIC ALGORITHM CONFIGURATION

3.1 Introduction

Metaheuristic algorithms have attracted substantial interest in solving complex real-life problems in fields such as economics, engineering, politics, and management [73]. They are based on two principles, intensification and diversification, which must be balanced for effective problem-solving. Many are inspired by biological evolution, swarm behavior, and physical laws [74]. Metaheuristics are broadly classified as single-solution or population-based, as illustrated in Figure 3.1. Single-solution algorithms (for example, tabu search, guided local search, and simulated annealing) optimize a single nominee but risk being trapped in local optima [75]. In contrast, population-based algorithms such as ant colony optimization (ACO), particle swarm optimization (PSO), and GA employ multiple solutions simultaneously, preserving diversity and reducing the likelihood of local optima entrapment [76].

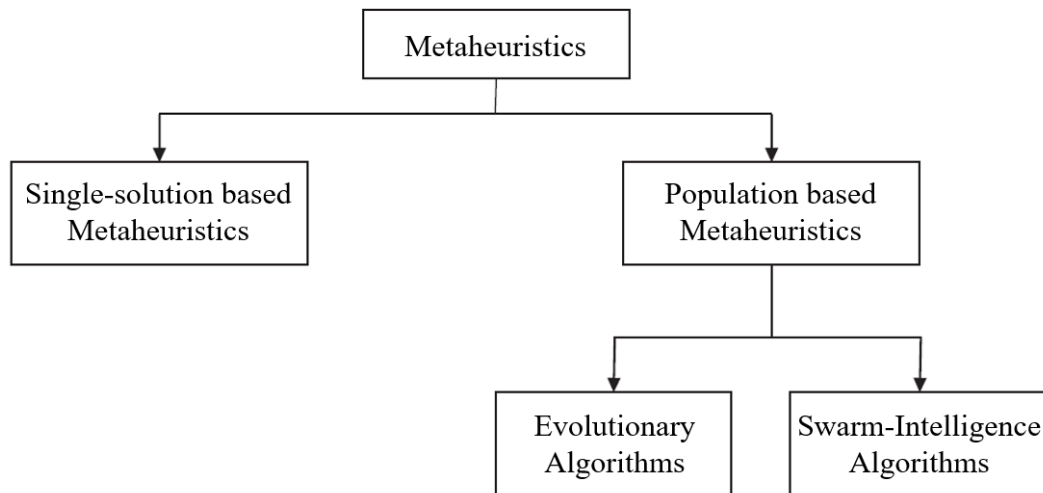


Figure 3.1 : Metaheuristic algorithms classification [76].

GA, in particular, is a prominent population-based metaheuristic, guided by Darwin's theory of natural selection. Proposed by J.H. Holland in 1975, it encodes solutions as chromosomes, employs fitness selection, and uses biologically inspired operators such

as crossover and mutation. Each chromosome's fitness score governs its selection for further processing. During crossover, offspring are generated by exchanging segments between chromosomes, while mutation alters random bits. GA has evolved alongside enhancements in its operators, representation, and fitness functions [77]–[79].

3.2 Classical GAs

The GA is an optimization approach that leverages a population of solutions and draws inspiration from natural selection and the "survival of the fittest" concept [78]. It iteratively forms new populations using genetic operators. The primary elements of GA include chromosome representation, selection, crossover, mutation, and fitness function evaluation.

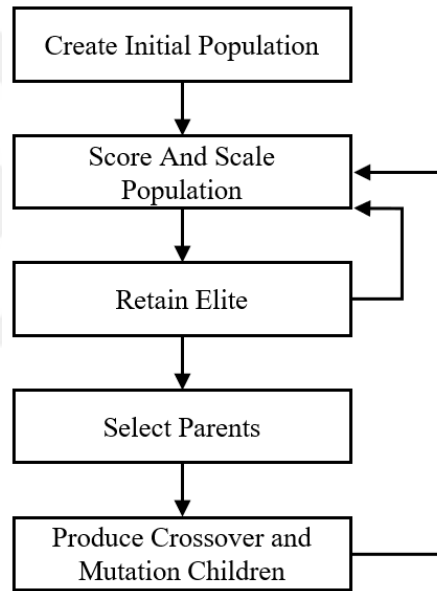


Figure 3.2 : The main steps of the GA.

GA starts by randomly initializing a population Y consisting of n chromosomes. The fitness of each chromosome is then evaluated, and based on these assessments, two chromosomes, $C1$ and $C2$, are selected. A single-point crossover is performed with a crossover probability C_p , resulting in the creation of an offspring O . Next, a uniform mutation operator with a mutation probability M_p is applied to produce a mutated offspring O' , which is then added to the new population. This cycle repeats until the new population is fully formed, as shown in Figure 3.2.

Mathematically, GA refines the search process by adjusting C_p and M_p , moving toward the best answer. It is capable of altering the encoded genes, evaluating a broad range of individuals, and generating several optimal solutions, offering strong global search capabilities. Crossover between parent chromosomes can alter favorable genetic schemas, and its mathematical expression is given as [80]:

$$R = \frac{G + 2\sqrt{g}}{3G} \quad (3.1)$$

Where g represents the current generation number and G denotes the total amount of evolutionary generations predetermined for the population. This equation clearly shows that R is dynamic, increasing as the evolutionary process progresses. In the initial phases of GA, when the similarity among individuals is low, a lower value of R is desirable to avoid disrupting valuable genetic schemas. Conversely, toward the end of the evolutionary process, when individual closeness is high, R should be high to preserve excellent genetic schemas. The Schema Theorem indicates that the original schema should be replaced with a modified version, maintaining population diversity during the initial stages of evolution and refining the schema toward optimal solutions by the end [81,82].

3.3 Genetic Algorithm Configuration

The population configuration and the fitness scale significantly influence the convergence and efficiency of the algorithm. The genetic algorithm defines various parameters related to the population structure to ensure optimal search performance. These settings include the population type, size, initialization methods, and reproduction.

- The population type determines how individuals are represented in the algorithm. Double Vector is the default representation, suitable for both continuous and mixed-integer problems. It allows for a more flexible solution space while maintaining integer feasibility. In this thesis, we have selected this method to ensure both the feasibility of integer constraints and the diversity of potential solutions. Bitstring representation consists of binary vectors (0 or 1). However, GA treats these vectors as numerical arrays rather than actual strings. Using bitstring

representation may cause GA to ignore constraints, including bounds and linear constraints. Users can define a custom representation, which requires specifying a compatible crossover and mutation function. Custom population types allow greater flexibility but necessitate careful constraint handling.

- The population size determines how many individuals exist in each generation. A larger population size enhances the algorithm's ability to explore the search space, reducing the likelihood of premature convergence to local optima. However, this comes at the cost of increased computational time. The default is "50 when numberOfVariables $\leq$ 5, else 200" [83]. If a vector is provided for the "PopulationSize" function, GA creates multiple subpopulations corresponding to the vector entries. In this thesis, we selected 50 as our population size and 20 generations, even though the number of variables is 13. To correctly define the population, we initially ran a simulation with a population size of 100 and a maximum generation of 50. This preliminary run helped analyze convergence behavior and computational efficiency, leading to the decision to use a population size of 50.
- The initial population can be generated using predefined functions. This option allows defining the characteristics of the population at each generation. It is an important detail since the algorithm's effectiveness is influenced by the population size and the genetic diversity among individuals. An incorrect configuration of these characteristics may lead to a lack of homogeneity among individuals or premature convergence of the results. Therefore, a trial-and-error approach is the only way to interpret results and determine the best solution. Finding an appropriate result often requires a compromise between solution quality and computational time. In many cases, faster but less effective algorithms are preferred. The genetic algorithm randomly creates an initial population using a creation function while respecting the constraints. In this case, the "gacreationuniformint" function was used. It generates individuals randomly within artificial limits set by the function itself while ensuring integer constraints.

- Fitness scaling is a crucial step in GA optimization, as it transforms raw fitness scores into a range suitable for the selection function. This process ensures that individuals with higher fitness values have a greater probability of being selected as parents for the next generation. However, an improperly defined scaling range can significantly impact the algorithm's performance.

A large disparity in scaled values can cause the fittest individuals to dominate the population too quickly, limiting the exploration of alternative solutions and increasing the risk of premature convergence. Conversely, if the variation in scaled values is too small, all individuals have nearly equal chances of selection, which slows down the search process and increases computational costs due to a larger number of potential parents being analyzed.

In this thesis, we employ rank-based fitness scaling (MATLAB: "fitscalingrank"), where the fitness score of an individual is determined by its rank within the sorted population rather than its absolute fitness value. This method helps regulate selection pressure by assigning scaled fitness values according to the following relationship:

$$\text{Scaled Fitness} \propto \frac{1}{\sqrt{n}} \quad (3.2)$$

where n represents the rank of the individual in the sorted population. The fittest individual is assigned a value of 1, the second-best receives $\left(\frac{1}{\sqrt{2}}\right)$, and so on. Additionally, the sum of all scaled values across the population equals the number of parents required for the next generation.

By using rank-based scaling, this study aims to maintain diversity within the population while avoiding premature convergence, ensuring a more effective search for optimal solutions in the defined problem space.

- Reproduction options determine how the next generation is created based on the individuals in the current generation. These settings are crucial for controlling the selection and crossover processes, ensuring an efficient search for optimal solutions.
- In the MATLAB "EliteCount" function specifies the number of individuals with the best fitness values from the current generation that are guaranteed to survive and be passed onto the next generation. These individuals are referred to as "elite children".

The elite children are not subjected to further selection processes, ensuring that the best solutions found thus far are preserved.

When "EliteCount" is at least 1, the best fitness value can only decrease from one generation to the next. This is a desired characteristic, as the genetic algorithm aims to minimize the fitness function. However, setting "EliteCount" to a high value may lead to the dominance of the fittest individuals in the population, which can reduce the diversity and effectiveness of the search. Therefore, it is important to balance the elite count to maintain both quality and diversity within the population.

For continuous problems, the default value of "EliteCount" is given by $\text{ceil}(0.05 \times \text{PopulationSize})$. For mixed-integer problems, the default value is typically $(0.05 \times \text{default PopulationSize})$. The value of "EliteCount" should be a positive integer and must be less than or equal to the population size.

The "CrossoverFraction" determines the fraction of individuals in the next generation, excluding the elite children, that are created through crossover. Crossover is a genetic operator used to combine the genetic information of two parent individuals to produce offspring. This parameter specifies how much of the next generation should be derived from the crossover process.

The number of crossover children in the next generation is controlled by the "CrossoverFraction" parameter in MATLAB. The value of "CrossoverFraction" should be a fraction between 0 and 1. The default value is set to 0.8, meaning that 80% of the next generation will be created by crossover, while the remaining individuals will be chosen from the elite group or created through mutation processes. For example, if the population size is 20, the elite count is 2, and the crossover fraction is 0.8, there will be 2 elite children, 14 crossover children, and the remaining 4 individuals will be mutation children. Adjusting the "CrossoverFraction" can have a significant impact on the performance of the genetic algorithm. A higher value generally promotes more exploration of the solution space, while a lower value can lead to more exploitation of the current solutions.

It is important to note that elite individuals, who have already been evaluated, are not reassessed during the reproduction process. This assumption holds true if the

fitness function is deterministic. However, if the fitness function is random or can change during the search, an output function can be used to reevaluate the elite individuals. This can be controlled through the "EvalElites" option in the state structure.

3.4 Genetic Operators

Genetic algorithms utilize a range of operators throughout the search process, including crossover, mutation, selection, and encoding methods, as illustrated in Figure 3.3. These operators are essential for the efficiency and functionality of GAs, and their selection depends on the problem domain [84,85].

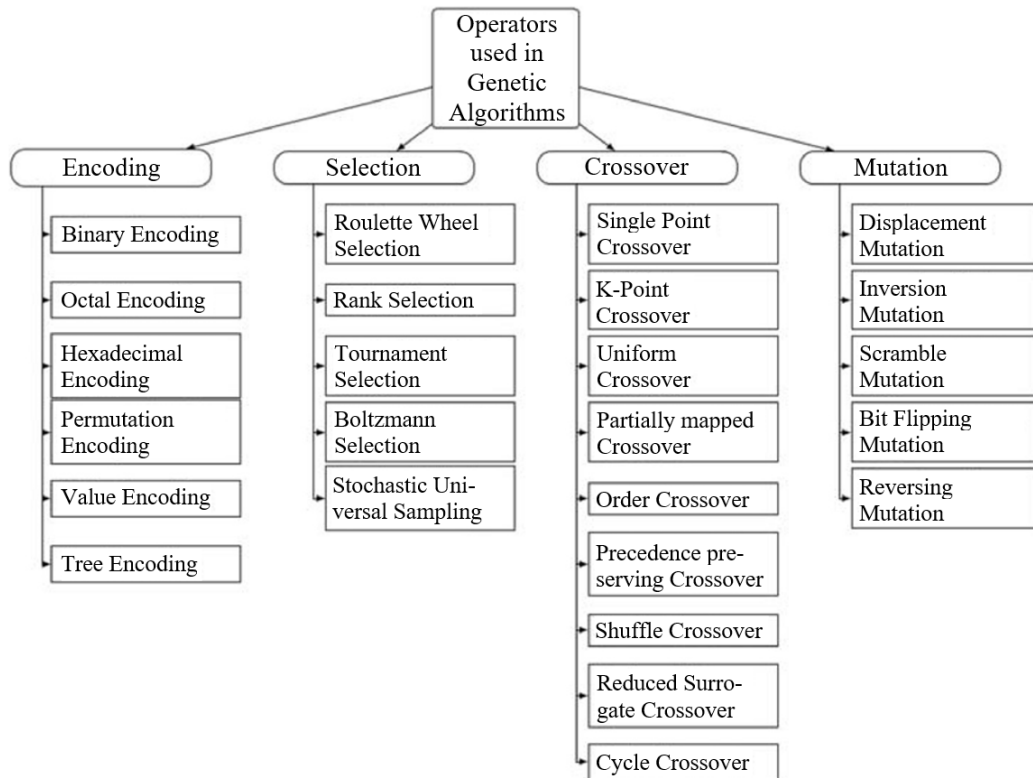


Figure 3.3 : Operators used in the GA [76].

3.4.1 Encoding schemes

The encoding scheme in GA transforms information into a specific format (typically a bit string) for processing [84,85]. Different encoding schemes are employed based on the problem being addressed. Binary, octal, hexadecimal, permutation, value-based, and tree encoding are examples of common encoding systems, see Table 3.1.

Table 3.1 : An analysis of several encoding schemes.

Encoding Scheme	Pros	Cons	Application
Binary	Simple to implement, fast execution	No support for inversion operator	Problems that support binary encoding
Octal	Simple to implement	No support for inversion operator	Limited use
Hexadecimal	Simple to implement	Lacks support for the inversion operator	Limited use
Permutation	The inversion operator is supported	Does not support binary operators	Task ordering problem
Value	No need for value conversion	Requires specific crossover and mutation	Neural network problems
Tree	Operator can easily be applied	Designing a tree might be challenging for certain issues	Evolving programs

- Binary Encoding: In this scheme, each chromosome is depicted as a sequence of 0s and 1s. This encoding is effective for crossover and mutation operations, but epistasis may cause complications in certain engineering design applications [86].
- Octal Encoding: Here, chromosomes are encoded using octal digits (0–7), providing a compact representation.
- Hexadecimal Encoding: In this method, chromosomes are expressed by hexadecimal digits (0-9, A-F), offering a more compact form than binary encoding [86]–[88].
- Permutation Encoding: Utilized in ordering problems, where chromosomes represent positions in a sequence.
- Value Encoding: This scheme uses real, integer, or character values and is suitable for problems requiring more complex representations, such as finding optimal weights in neural networks [89].

- Tree Encoding: Chromosomes are represented by trees of functions or commands, often utilized in dynamic programming or expressions [90].

3.4.2 Selection methods

GAs use selection as a crucial phase to decide which individuals participate in reproduction [89,90]. Various selection techniques affect the algorithm's convergence and performance, as summarized in Table 3.2.

- Roulette Wheel Selection: Chromosomes are assigned segments on a wheel proportional to their fitness. The wheel is spun to select individuals for the next generation. However, its stochastic nature can lead to errors [90].
- Rank Selection: Individuals are ranked by fitness, and selection is based on these ranks rather than absolute fitness values [90].

Table 3.2 : Comparing various selection methods.

Selection Technique	Pros	Cons
Roulette Wheel	Straightforward implementation; simple and unbiased	Prone to premature convergence; relies heavily on fitness variance
Rank	Maintains diversity; inherently unbiased	Convergence can be slow; requires sorting and is computationally intensive
Tournament	Retains diversity; amenable to parallelization; no sorting needed	May reduce diversity if the tournament size is too large
Boltzmann	Capable of reaching a global optimum	Computationally expensive
Stochastic Universal Sampling	Fast and unbiased	Can lead to premature convergence
Elitism	Ensures the best individual is preserved in the population	Crossover and mutation operators can cause the loss of the best individual

- **Tournament Selection:** This method pairs people and selects the fitter from each pair for the next generation [90].
- **Stochastic Universal Sampling (SUS):** Using a random beginning point and selecting individuals at uniformly spaced intervals, as shown in Figure 3.4. It is an extension of roulette wheel sampling [91]. SUS is particularly effective for the Traveling Salesman Problem, but its effectiveness decreases as the problem size increases [92].

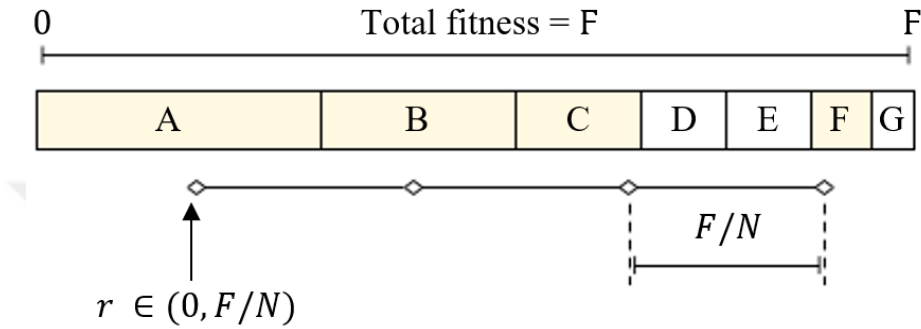


Figure 3.4 : Stochastic universal sampling approach.

- **Boltzmann Selection:** This methodology, which bases itself on entropy and Monte Carlo techniques, favors individuals with higher fitness to avoid premature convergence [93]. The danger of knowledge loss, however, can be reduced by elitism [94].
- **Elitism:** Elitism ensures that the best individual from each generation is always carried over to the next, preventing the loss of optimal solutions [90].

In this thesis, stochastic uniform sampling (MATLAB: "selectionstochunif") was preferred. This method arranges individuals along a line, where the length of each individual's section is proportional to its expectation value. The algorithm moves along the line in uniform steps, selecting the individual in the corresponding section as a parent. This method ensures diversity in selected individuals while maintaining a probabilistic bias toward fitter candidates.

For problems with integer constraints, MATLAB ensures that all integer, bound, and linear constraints remain feasible at each iteration. When using custom selection,

mutation, crossover, or initialization functions, MATLAB applies additional feasibility checks post-processing.

The algorithm (algorithm 1) operates as follows:

Algorithm 1 Stochastic Universal Sampling (SUS)

Input: Population, N (number of offspring to keep)
 $F \leftarrow$ total fitness of Population
 $P \leftarrow F/N$ distance between the pointers
Start $\leftarrow$ random number between 0 and P
Pointers $\leftarrow [\text{Start} + i \cdot P \mid i \in [0, N-1]]$
Return: RWS(Population, Pointers)

Algorithm 2 Roulette Wheel Selection (RWS)

Input: Population, Pointers
Keep $\leftarrow []$
for each P in Pointers **do**
 $I \leftarrow 0$
 while sum of fitness of Population[0.. I] $< P$ **do**
 $I \leftarrow I + 1$
 end while
 add Population[I] to Keep
end for
Return: Keep

- Compute Total Fitness: The total fitness of the population is calculated.
- Calculate Distance Between Pointers: The distance between the selection pointers is calculated by dividing the total fitness (F) by the number of offspring to retain (N): $P = F/N$
- Generate a Random Starting Point: A random number is generated between 0 and P for the starting point.
- Place Pointers: N equally spaced pointers are placed along the cumulative fitness distribution. These pointers are spaced by the distance P and are calculated as:
Pointers = $\{\text{Start} + i \cdot P \mid i \in [0, N-1]\}$
- Use Roulette Wheel Selection (RWS): The Roulette Wheel Selection (RWS) method (algorithm 2) is used to determine the individuals corresponding to these pointers.

- **Input:** The population and the list of pointers are given as input.
- **Initialize Keep List:** An empty list, `Keep`, is initialized to store the selected individuals.
- **Loop Through Pointers:** For each pointer, the algorithm loops through the population and adds the individual whose cumulative fitness sum exceeds the pointer value to the `Keep` list.
- **Return Selected Individuals:** Once all the pointers have been processed, the selected individuals are returned as the output.

3.4.3 Crossover operators

Crossover operators in GAs integrate genetic material from two or more parent individuals to create offspring. Common crossover operators, summarized in Table 3.3, include:

- **Single-Point Crossover:** A random crossover point is selected, and the genetic information past that juncture, is exchanged between two parents to produce offspring, Figure 3.5. This operator is simple and widely used [95].
- **Two-Point and K-Point Crossover:** Two or more randomly chosen crossover points are identified, and the segments of genetic material between these points are exchanged between parents [95]. These techniques allow more varied recombination compared to single-point crossover, as illustrated in Figure 3.6.

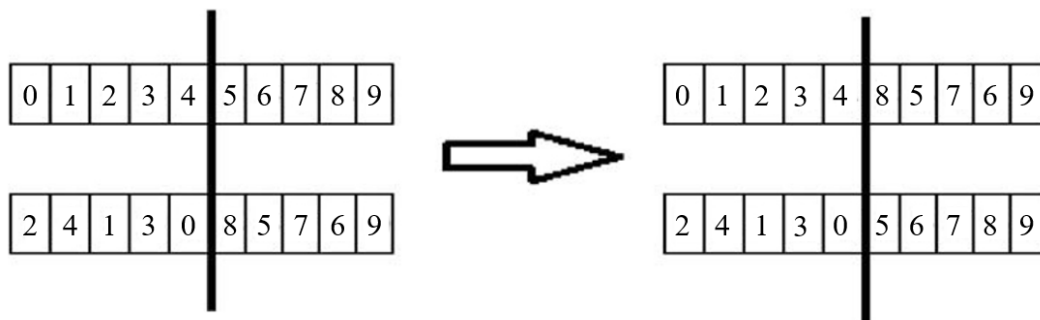


Figure 3.5 : Single-point crossover [76].



Figure 3.6 : Two-point and k-point crossover [76].



Figure 3.7 : Uniform crossover [76].

- Uniform Crossover: Each gene is treated individually; a random decision is made for each gene on whether to swap it with the corresponding gene from the other parent, Figure 3.7 [95].
- Partially Matched Crossover (PMX): Swaps corresponding segments between parents and fills the remaining positions with alleles from the other parent, making it effective for problems like the Traveling Salesman Problem (TSP), Figure 3.8 [96,97].
- Order Crossover (OX): Copies portions of parent chromosomes into the offspring, while the remaining positions are filled with elements not present in the copied segments [98]. However, it has limited efficiency in some problems such as TSP [99].

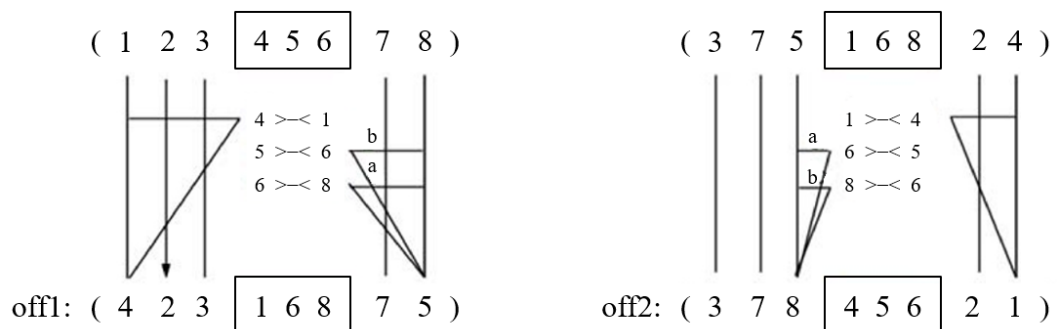


Figure 3.8 : Partially matched crossover (PMX) [76].

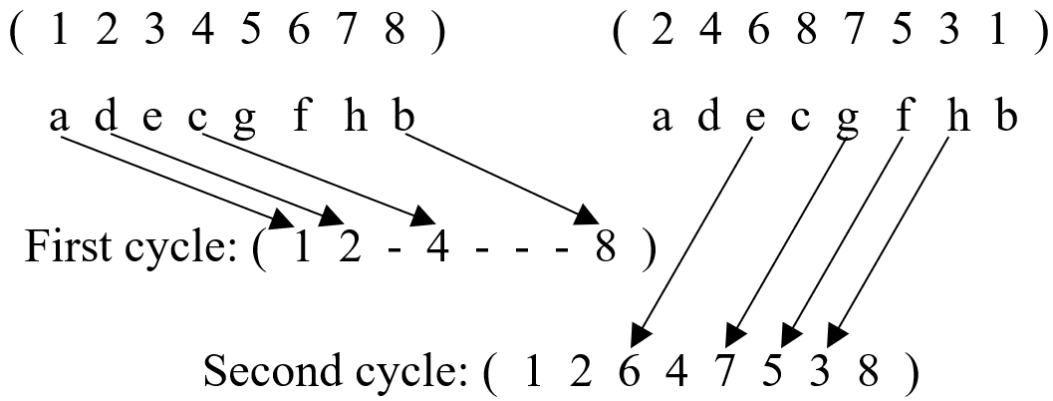


Figure 3.9 : Shuffle crossover [76].

- **Precedence Preserving Crossover (PPX):** It maintains the original sequence of genes as they appear in the parent chromosomes. The offspring is initialized randomly, with 1s and 0s determining which genes are included [100]. This operator is often applied in multi-objective scheduling problems.
- **Shuffle Crossover:** Shuffle crossover shuffles an individual's genetic material before and after crossover to reduce bias from the crossover point, Figure 3.9. Its usage has declined in the past few years [101].
- **Reduced Surrogate Crossover (RCX):** Minimizes unimportant crossovers by recognizing when two parents have identical genetic sequences. It assumes that genetic diversity is essential for improvement, but it is less effective when parents are identical [101,102].
- **Cycle Crossover:** Creates offspring by selecting genes from each parent based on their positions, following a cycle-based approach to preserve the genetic structure of the solutions [99].

In cases where the problem involves both integer and linear constraints, the most suitable crossover function to use in MATLAB is "crossoverlaplace". This crossover method is specifically designed for integer constraints [103]. It generates children by applying one of two formulas (equation 3.3 and 3.4) involving a Laplace distribution, which helps to maintain feasibility with respect to integer constraints.

Table 3.3 : Comparison of crossover techniques.

Technique	Advantages	Disadvantages
Single/Two and K-Point	Easy to implement	May not explore solution space thoroughly
Uniform	Suitable for large subsets	Can struggle with maintaining diversity
Order/Cycle	Better exploration and handling of ordering problems	May not be efficient for all problem types
Partially Mapped (PMX)	Provides better exploration than others	May not handle all problem types well
Reduced Surrogate	Reduces unnecessary crossovers when parents are similar	Not effective for parents with identical genes

The algorithm (algorithm 3) ensures that all individuals are feasible with respect to both integer and linear constraints. For problems with both integer and linear constraints, the algorithm adjusts generated individuals to maintain feasibility. The algorithm attempts to round linearly feasible points to integers when necessary, ensuring the points remain feasible with respect to the constraints.

The Laplace crossover generates children using either of the following formulas (chosen at random) [103]:

$$\beta = \begin{cases} a - b \log_e(u), & u \leq \frac{1}{2} \\ a + b \log_e(u), & u > \frac{1}{2} \end{cases} \quad (3.3)$$

Here, (a) and (b) are variables with default values of 0 and 1, respectively, and (u) is a random variable. The offspring are given by the equations:

$$y_i^1 = x_i^1 + \beta |x_i^1 - x_i^2| \quad (3.4)$$

$$y_i^2 = x_i^2 + \beta |x_i^1 - x_i^2| \quad (3.5)$$

3.4.4 Mutation operators

Mutation is a critical operator in GAs that maintains genetic diversity by introducing random alterations in offspring. Common mutation operators, summarized in Table 3.4, include:

Algorithm 3 Pseudo-code for Laplace Crossover (LX)

Begin
 $m \leftarrow 0$
Initialize $P(m)$
Evaluate fitness of $P(m)$
while termination criterion is not satisfied **do**
 $m \leftarrow m + 1$
 Select $P(m)$ from $P(m - 1)$
 Apply LX on $P(m)$
 Apply mutation (MPTM/NUM) Operator
end while
End

- Displacement Mutation (DM): A substring from a solution is moved to a different position randomly, ensuring the solution remains valid. Variants include exchange mutation (swapping parts) and insertion mutation (inserting a part into another position) [90].
- Simple Inversion Mutation (SIM): A substring's order is reversed between two specified positions and then reinserted into the solution at random [90].
- Scramble Mutation (SM): Elements within a specified range are randomly reordered, and the new solution's fitness is evaluated [90].

Table 3.4 : Comparison of mutation techniques.

Mutation Technique	Description	Advantages	Disadvantages
Displacement Mutation (DM)	Randomly displaces a substring within the solution	Straightforward and keeps the solution valid	May not be efficient for large problem spaces
Simple Inversion Mutation (SIM)	Reverses the order of a substring between two specified locations	Can introduce significant genetic diversity	May not always lead to better solutions for some problems
Scramble Mutation (SM)	Randomly rearranges the elements within a chosen segment	Good for local search improvements	Can be computationally expensive

Table 3.5 : The optimal combination of different operators in the ideal setting.

Encoding Scheme	Mutation	Crossover
Binary	Inversion	Uniform, Arithmetic, Single-Point, or N-Point
Permutation	Inversion	Partially Matched, Cycle, or Order
Value	Displacement	Uniform, Arithmetic, Single-Point, or N-Point
Tree	Scramble	Uniform or Single-Point

The ideal encoding, mutation, and crossover approach combinations are shown in Table 3.5. It indicates that both uniform and single-point crossovers are applicable to most encoding and mutation operators. In particular, partially matched crossover combined with inversion mutation and permutation encoding yields the optimal solution.

The proposed mutation operator is based on the power distribution and is referred to as the Power Mutation (PM) (MATLAB: "mutationpower") [104]. The distribution function of the power distribution is given by:

$$f(x) = px^{p-1}, \quad 0 \leq x \leq 1 \quad (3.6)$$

where p is the distribution index. The corresponding density function is:

$$F(x) = x^p, \quad 0 \leq x \leq 1. \quad (3.7)$$

The PM operator generates a new candidate solution y in the vicinity of a given parent solution $\bar{x}$. First, a uniform random number t in the range $[0, 1]$ is generated, followed by a random number s sampled from the power distribution. The mutated solution y is then computed as follows [104]:

$$y = \begin{cases} \bar{x} - s(\bar{x} - x^l), & \text{if } t < r, \\ \bar{x} + s(x^u - \bar{x}), & \text{if } t \geq r. \end{cases} \quad (3.8)$$

Here, t is the scaled distance from the lower bound x_l and x_u represent the lower and upper bounds of the decision variable, while r is a uniformly distributed random number in the range $[0, 1]$. The influence of mutation strength is controlled by the

parameter p . Lower values of p result in minor perturbations, preserving solution stability, whereas higher values introduce greater diversity. The probability of generating a mutated solution y on the left (or right) side of x is proportional to its distance from the respective boundary x_l (or x_u), ensuring that the generated solution remains feasible [104].

3.5 Challenges

While GAs offer many advantages, their implementation and application face several challenges that must be addressed for future advancements. Some key challenges include:

3.5.1 Choosing initial population

The choice of the initial population is critical in determining GA performance. Both its size and diversity influence the algorithm's efficiency and solution quality.

- Large populations: They explore a wider solution space but increase computation time, slowing the algorithm.
- Small populations: They restrict exploration, resulting in lower solution quality.

Determining the optimal population size requires balancing exploration and computation time.

Harik and Lobo [105] examined population size using a self-adaptation method, proposing two approaches:

- Self-adaptation before execution, where the population size remains constant.
- Self-adaptation during execution, where the population size adjusts dynamically based on the fitness function.

This dynamic adjustment can optimize both efficiency and solution quality.

3.5.2 Premature convergence

Premature convergence in GAs occurs when early loss of diversity traps the algorithm in local optima, hindering the discovery of the global optimum [105,106].

- **Selection Pressure:** High selection pressure favors superior individuals early, reducing diversity and causing rapid convergence that may miss the global optimum.
- **Mitigation:** Adjusting selection pressure to balance intensification (exploiting good solutions) and diversification (exploring new ones) can help prevent premature convergence.
- **Global vs. Local Search:** Combining GAs' global search with local search methods (e.g., Tabu search) enhances the equilibrium among search and extraction, improving the likelihood of achieving the global optimum, as shown in Figure 3.10.

In conclusion, addressing both optimal population selection and premature convergence is crucial for enhancing GA performance and ensuring efficient convergence to global optimal solutions.

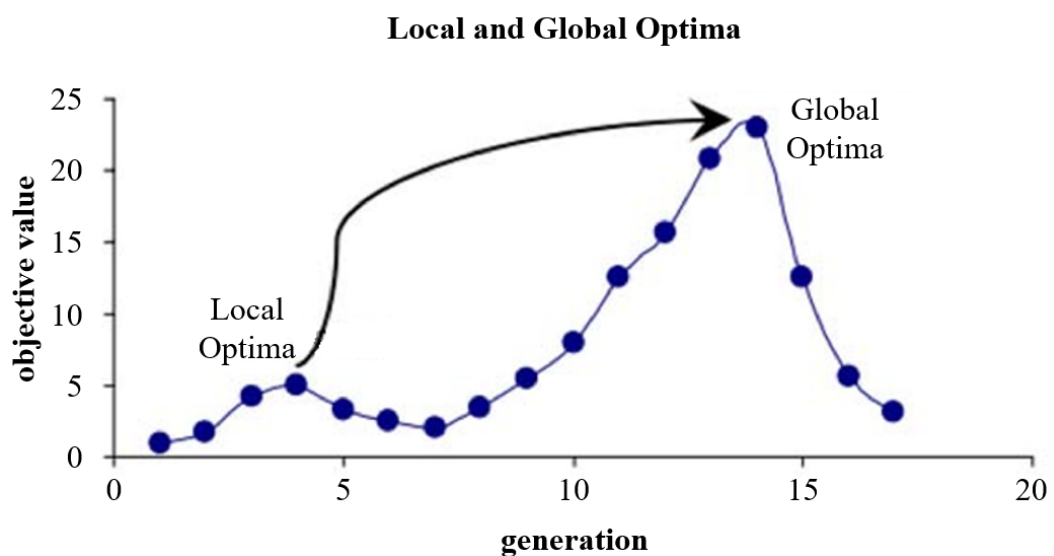


Figure 3.10 : Local and global optima [76].

3.5.3 Choosing effective fitness functions

The selection of an efficient fitness function plays a pivotal role in guiding GA, as it evaluates an individual's performance. However, choosing an appropriate fitness function presents several challenges:

- **Computational Cost:** A costly fitness function may be manageable with few iterations, but its computational burden grows significantly as iterations increase.
- **Suitability:** The function must accurately represent the optimization objective while remaining computationally efficient.
- **Example:** In [107], the Davies-Bouldin index was used for document classification, illustrating that fitness function selection depends on the application.

3.5.4 Crossover and mutation level

Mutation and crossover operators are essential for maintaining diversity and achieving convergence to an optimal solution. However, setting their appropriate levels poses challenges:

- **Mutation:** A mutation rate that is too low limits exploration and may lead to stagnation in local optima, while a rate that is too high can disrupt promising solutions and slow convergence.
- **Crossover:** Without crossover, genetic material is not exchanged, reducing the algorithm's ability to combine beneficial traits and explore new solutions.
- **Balancing Exploration and Exploitation:** Fine-tuning mutation and crossover rates is crucial for reaching global optima. Due to the probabilistic nature of these operators, there is no exact rule for determining their optimal values [108].

3.5.5 Selection of encoding schemes

GAs necessitate a suitable encoding scheme tailored to the specific problem type. However, selecting an appropriate scheme is challenging because:

- No Universal Encoding Methodology: A single encoding scheme does not work for all problems; different problems may require binary, integer, or real-valued encodings.
- Handling Redundancy: Some encoding schemes introduce redundant representations, leading to inefficiencies. Ronald [109] suggested designing encoding schemes that minimize redundancy and implementing genetic operators that avoid bias toward these redundancies.

Addressing these challenges requires a problem-specific approach where fitness function design, genetic operator tuning, and encoding scheme selection are carefully optimized to enhance GA performance.



4. TURBULENCE MODEL SELECTION AND APPLICATION IN COMPUTATIONAL FLUID DYNAMICS

The Boussinesq hypothesis [110] is commonly applied to correlate Reynolds stresses with mean velocity gradients in turbulence modeling. This approach underlies models like Spalart-Allmaras, k - ϵ , and k - ω , providing computational efficiency by requiring only one or two additional transport equations. However, it simplifies turbulent viscosity as an isotropic scalar. While this assumption holds effectively for shear-dominated flows, such as boundary layers, mixing layers, and jets, it may be less accurate for flows with significant anisotropy.

An alternative approach is the Reynolds Stress Model (RSM), which solves transport equations for each component of the Reynolds stress tensor (requiring five additional equations in 2D and seven in 3D). While Boussinesq-based models perform well in many cases, RSM is superior when turbulence anisotropy significantly influences the mean flow (e.g., in highly swirling or stress-driven secondary flows). However, due to its higher computational cost, RSM is used only when necessary.

4.1 Turbulence Models

4.1.1 k - ϵ models

Standard k - ϵ Model: A widely used two-equation model [111] for its balance of cost, robustness, and accuracy. Solves transport equations for turbulence kinetic energy (k) and dissipation rate (ϵ), where the k equation is derived from first principles and the ϵ equation relies on empirical scaling. Assumes fully turbulent flow, neglecting molecular viscosity effects. Enhanced variants in ANSYS Fluent include the realizable k - ϵ [112] and RNG k - ϵ models.

RNG k - ϵ Model: Derived via renormalization group (RNG) theory [113], this model refines the standard k - ϵ framework. Key improvements include: an additional term in the ϵ equation for rapid strain flows; explicit swirl modification for vortical

flows; analytical turbulent Prandtl numbers replacing user-defined constants; and a differential viscosity formula accommodating low-Re effects. These enhancements extend accuracy across broader flow regimes, particularly in near-wall regions with appropriate resolution. A full theoretical foundation is detailed in [113].

Realizable k - ε Model: Introduces significant enhancements compared to the standard k - ε model [112]:

- It adopts a modified turbulent viscosity formulation with a dynamic coefficient C_μ , which typically approximates 0.09 in the logarithmic region of equilibrium boundary layers and around 0.05 in strong homogeneous shear flows. This coefficient adjusts based on mean flow deformation and turbulence characteristics (k , ε).
- It incorporates a revised transport equation for ε , derived from the exact transport equation governing mean-square vorticity fluctuations.

The term "realizable" signifies that the model adheres to mathematical constraints on Reynolds stresses, effectively preventing anomalies such as negative normal Reynolds stresses under highly strained flow conditions. Both the Realizable and RNG k - ε models surpass the standard k - ε model, particularly in flows characterized by pronounced curvature, swirling motions, and rotational effects. Initial investigations further highlight the Realizable model's superior performance in scenarios involving flow separation and intricate secondary structures.

However, a recognized limitation of the Realizable k - ε model is its potential to generate non-physical turbulent viscosities when the computational domain includes both rotating and stationary fluid regions, such as those encountered in multiple reference frames or rotating sliding mesh simulations. This issue arises because the model accounts for mean rotation effects within its turbulent viscosity formulation—an advantage in single moving reference frames but a source of inconsistency in multiple reference frame (MRF) simulations. Consequently, careful consideration is advised when applying the model in such contexts.

4.1.2 Spalart-Allmaras model

- Single-equation turbulence model [114], developed for aerospace wall-bounded flows. Excels in boundary layer prediction under adverse pressure gradients; widely used in turbomachinery.
- Low-Re formulation requires fine boundary layer meshes. ANSYS Fluent's y^+ -insensitive treatment combines viscous sublayer and log-law, ensuring accuracy across $1 < y^+ < 30$.
- Limitations: Uncalibrated for general industrial flows; errors in free shear flows (e.g., jets); poor prediction of homogeneous/isotropic turbulence decay.

4.1.3 k - ω model

Standard Model: Solves transport equations for k (turbulence kinetic energy) and ω (specific dissipation rate, $\omega = \varepsilon/k$) based on Wilcox's framework [115]. Includes low-Re corrections, compressibility, and shear flow adjustments. Notable limitation is high sensitivity to freestream conditions [116], mitigated via empirical production terms in k and ω equations.

Baseline (BSL) Model: Addresses freestream sensitivity by blending near-wall k - ω accuracy with freestream k - ε independence [117]. Key enhancements include a damped cross-diffusion term in the ω equation and modified constants distinct from the standard model.

Shear-Stress Transport (SST) Model: Extends the BSL model by incorporating shear-stress transport into the turbulent viscosity definition, enhancing accuracy for adverse pressure gradients, airfoils, and transonic shock waves. Outperforms standard and BSL models in these scenarios. k - $k\ell$ - ω Transition Model: Predicts laminar-to-turbulent transition [118], capturing boundary-layer development and integrating transition processes into aerodynamic simulations.

4.1.4 Transition sst model

Extends the SST k - ω framework with two additional equations: intermittency and transition onset (via momentum-thickness Reynolds number). Leverages the Langtry-Menter correlation to simulate bypass transition and low-freestream turbulence flows. Features include customizable transition onset correlations and applicability to wall-bounded flows (free shear flows treated as fully turbulent). Limitations include non-Galilean invariance, exclusion of zero-freestream scenarios (e.g., fully developed pipe flows), and lack of calibration for buoyancy or multiphase turbulence.

4.1.5 Reynolds stress model (RSM)

Solves transport equations for Reynolds stresses and dissipation (5 equations in 2D, 7 in 3D) [119]–[121], avoiding isotropic eddy-viscosity assumptions. Excels in resolving anisotropic turbulence effects in flows with curvature, swirl, rotation, or rapid strain changes (e.g., cyclones, swirling combustors). Limitations include reliance on closure assumptions (pressure-strain and dissipation terms) and higher computational cost. ANSYS Fluent offers ε -based (quadratic/linear) and ω -based (Stress-BSL/Stress- ω) variants.

4.1.6 Scale-adaptive simulation (SAS) model

Advanced URANS methodology using the von Kármán length scale to dynamically resolve turbulence scales in unsteady flows. Maintains RANS behavior in stable regions while activating LES-like resolution in detached regions (Figure 4.1). Demonstrates superior scale resolution compared to standard URANS.

4.1.7 Detached eddy simulation (DES)

Hybrid approach combining URANS (boundary layers) with LES (separated regions). Compatible with turbulence closures including Transition SST, BSL/SST k - ω , realizable k - ε , and Spalart-Allmaras. Captures unsteady features in high-Reynolds-number

wall-bounded flows at lower computational cost than full LES. Limitations include higher CPU demand compared to RANS.

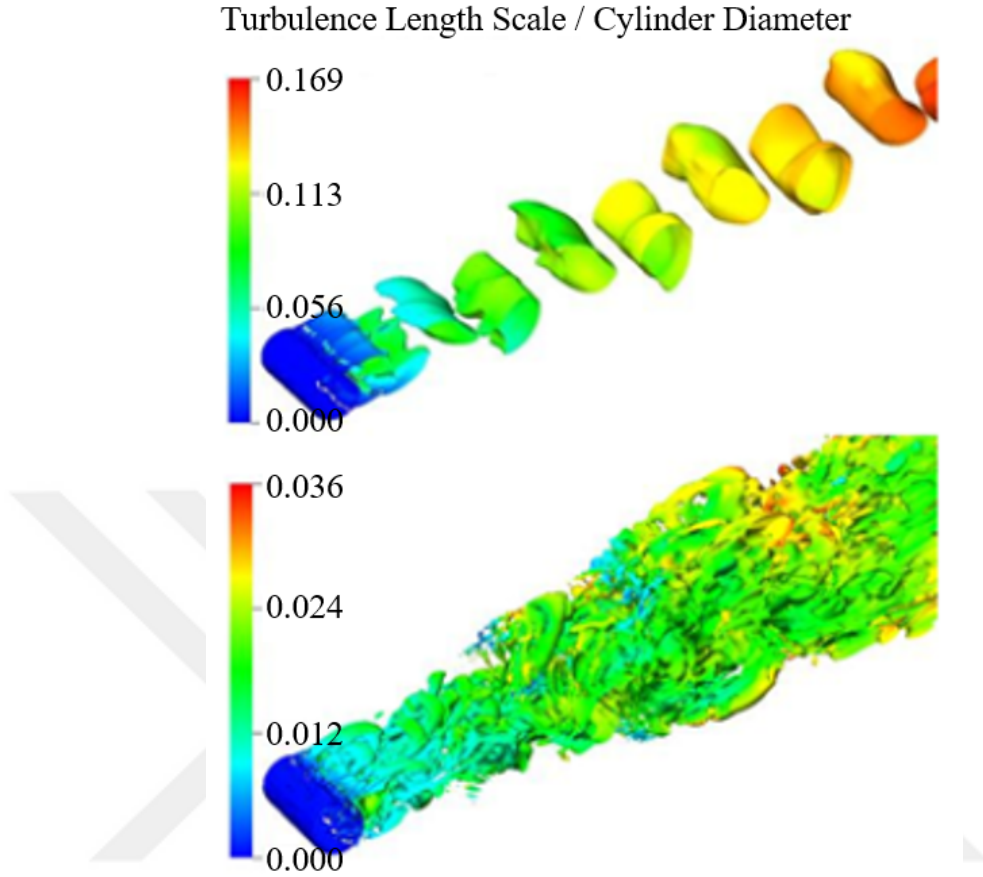


Figure 4.1 : Resolved wake structures: SST-SAS vs. URANS [83].

4.2 Near-Wall Treatment

Turbulent flows are strongly affected by the presence of walls, which alter both mean velocity distributions and turbulence properties. The no-slip boundary condition imposes a modified velocity profile at the wall, while viscous and kinematic effects further reshape the turbulence field. In close proximity to the wall, viscous damping reduces tangential fluctuations and kinematic blocking restricts normal fluctuations. Moving away from the wall, steep velocity gradients rapidly increase the turbulence kinetic energy.

Accurate near-wall modeling is essential because walls are major sources of vorticity and turbulence. The near-wall region features sharp gradients in solution variables and

active momentum, heat, and mass transfer, making its precise representation critical for reliable predictions in wall-bounded flows.

Experimental studies reveal that the near-wall region comprises three distinct layers, as shown in Figure 4.2 [83]:

- Viscous Sublayer: Behaves almost like laminar flow, where molecular viscosity dominates.
- Buffer Layer: An intermediate region influenced by both molecular viscosity and turbulence.
- Fully Turbulent Layer: Governed mainly by turbulence.

Understanding and accurately modeling these layers are key to improving the fidelity of turbulence simulations.

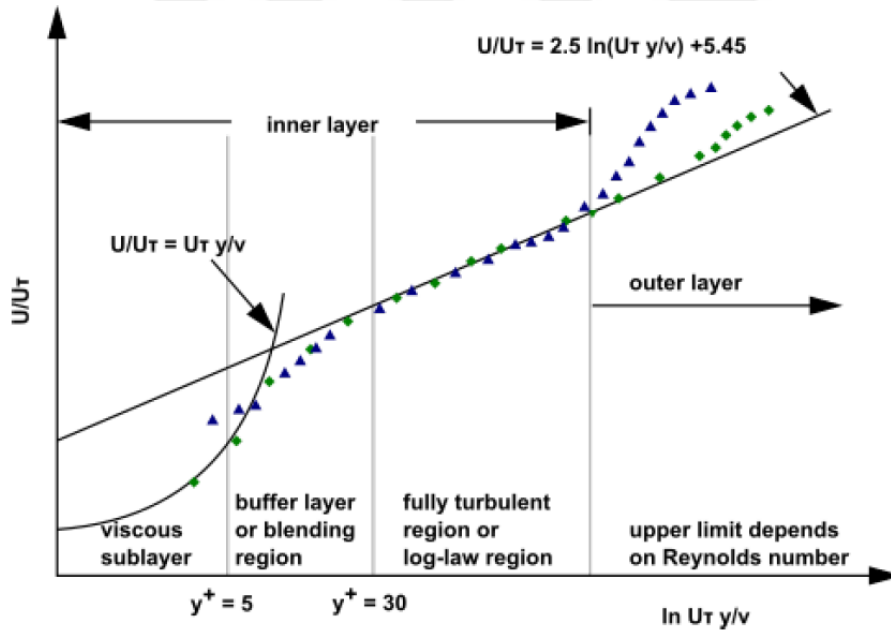


Figure 4.2 : The ANSYS Fluent near-wall region subdivisions [83].

4.2.1 Wall functions vs. near-wall model

Conventional wall-function techniques (with the exception of scalable wall functions) diminish in accuracy if the wall-normal grid is refined to the extent that y^+ drops below 15. In such scenarios, unbounded errors in wall shear stress and heat transfer

can arise. To address this, ANSYS Fluent provides more advanced, y^+ -independent wall treatments, which are the default for all ω -equation-based turbulence models. For ε -equation approaches, the Menter-Lechner and EWT methods perform a similar function, while the Spalart-Allmaras model also incorporates an approach that is insensitive to y^+ .

Achieving sufficient resolution in the boundary layer is more critical than targeting particular y^+ levels. A minimum of 10, and ideally 20, cells should be distributed across the boundary layer. In unstructured meshes, 10–20 prism layers near the wall are advised, with the prism-layer thickness exceeding that of the boundary layer to avoid constraining its development, as illustrated in Figure 4.3 [83].

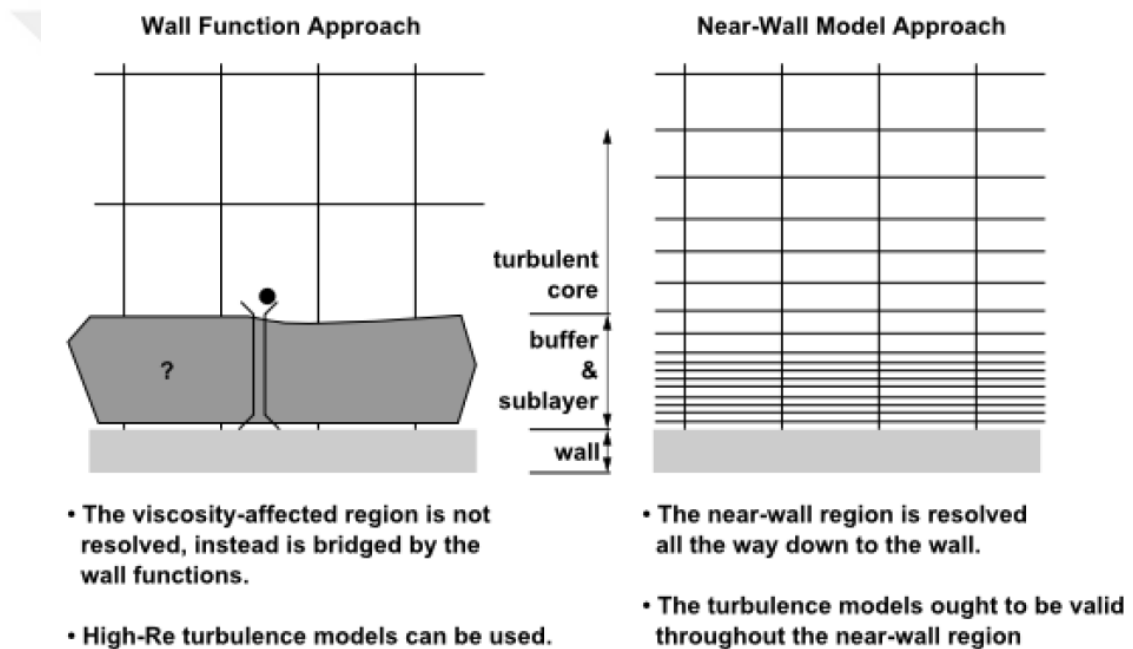


Figure 4.3 : The ANSYS Fluent near-wall treatments feature [83].

Recommendations:

- For ε -based models, opt for Menter-Lechner (ML- ε) or Enhanced Wall Treatment (EWT- ε). Scalable wall functions ought to be chosen if wall functions are desired.
- For ω -equation-based models and the Spalart-Allmaras model, keep the default y^+ -independent wall treatment.

4.2.2 Wall function strategies

Standard Wall Functions: Derived from Launder-Spalding [122], this method automatically imposes wall boundary conditions for velocity, temperature, and species. Suitable for industrial flows with constant shear and equilibrium conditions, its accuracy diminishes under strong pressure gradients or non-equilibrium effects. Compatible with k - ϵ and Reynolds Stress models.

Scalable Wall Functions: Designed to eliminate accuracy degradation on fine meshes ($y^+ < 11$), this approach matches standard wall function behavior for $y^+ > 11$. It ensures consistent predictions across grid resolutions by enforcing the log-law framework.

Non-Equilibrium Wall Functions: A two-layer method [123] modifies the velocity log-law to account for pressure gradients while retaining standard temperature/species laws. Recommended for flows with separation, impingement, or rapid pressure changes. Compatible with k - ϵ and Reynolds Stress Transport models.

Limitations: Wall functions lose reliability in low-Reynolds-number flows, near-wall mass injection/extraction, strong pressure gradients, body-force-dominated flows, or highly 3D boundary layers. For such cases, ANSYS Fluent recommends EWT or Menter-Lechner near-wall modeling.

4.2.3 Enhanced wall treatment (EWT)

This near-wall strategy combines a two-layer model with specialized wall functions. When the mesh resolves the viscous sublayer ($y^+ \approx 1$), EWT behaves like a standard two-layer zonal approach. To balance computational cost and accuracy, EWT integrates the two-layer model with enhanced wall functions, ensuring high fidelity on fine grids and minimal accuracy loss on coarse wall-function meshes.

Two-Layer Model Implementation. The domain is split into:

- A viscosity-affected near-wall region (resolved down to the viscous sublayer).

- A fully turbulent outer region (governed by k - ϵ or Reynolds Stress equations).

The transition between regions depends on the turbulence Reynolds number Re_y (normal wall distance at cell centers), with $Re_y^* = 200$ as the threshold. Compatible with all k - ϵ models (excluding Quadratic RSM), EWT is supplemented by the Menter-Lechner formulation for y^+ -insensitive predictions. ω -based models and Spalart-Allmaras automatically use y^+ -insensitive near-wall methods.

4.2.4 Menter-Lechner ϵ -equation (ML- ϵ)

This approach [124] replaces problematic two-layer methods with a novel low-Re formulation, ensuring y^+ -independent accuracy. It smoothly transitions between wall functions and low-Re models during mesh refinement, resolving issues like unphysical laminar zones and numerical instability.

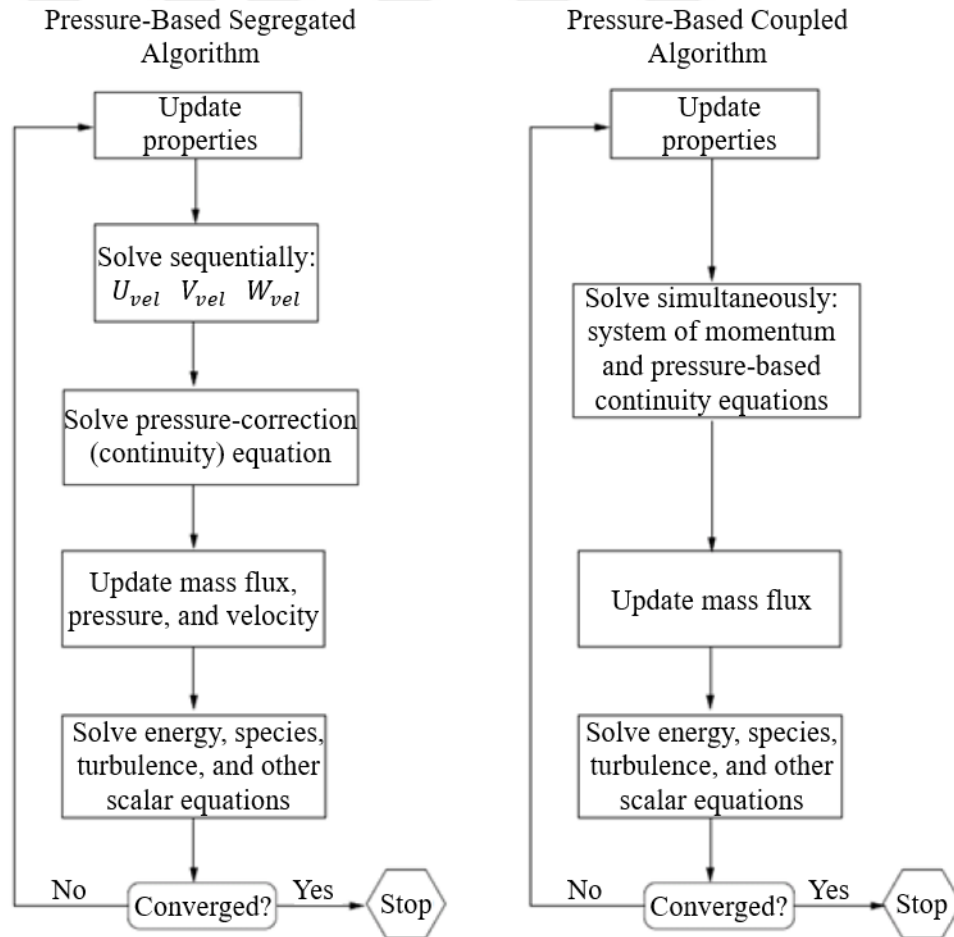


Figure 4.4 : Pressure-based solution methods [83].

4.3 Solver Theory

Pressure-based Solver: Originally designed for incompressible flows, now extended to various regimes. Enforces mass conservation through pressure-velocity coupling, as illustrated in Figure 4.4.

- **Segregated Algorithm:** Sequential equation solving (memory-efficient, slower convergence).
- **Coupled Algorithm:** Simultaneous momentum/continuity solution (faster convergence, higher memory).

Density-based Solver: Optimized for compressible flows. Solves continuity, momentum, energy, and species equations concurrently, as illustrated in Figure 4.5.

Approaches:

- **Implicit:** Faster convergence with higher memory.
- **Explicit:** Lower memory but slower convergence.

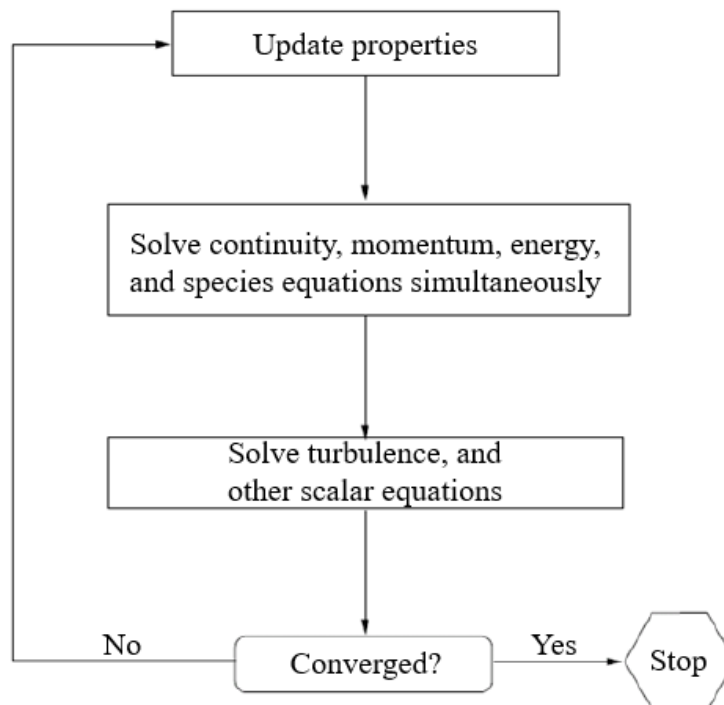


Figure 4.5 : Density-based solution methods [83].

4.3.1 Discretization schemes

Spatial Discretization: Uses finite-volume method with upwind interpolation for face values:

- First-Order Upwind: Robust but low accuracy.
- Second-Order Upwind: Linear reconstruction for improved precision.
- QUICK Scheme: Hybrid approach effective on structured meshes.
- Third-Order MUSCL: High accuracy on arbitrary meshes (caution near discontinuities).

Gradient Evaluation: Methods for computing governing equations and ensuring numerical accuracy.

- Green–Gauss Cell-Based: Efficient but inaccurate on skewed meshes.
- Green–Gauss Node-Based: Accurate for irregular meshes (higher cost).
- Least Squares Cell-Based: Balanced accuracy/efficiency.

4.3.2 Pressure-based solver details

Pressure Interpolation Schemes:

- Linear Scheme: Arithmetic mean pressure. For simple meshes.
- Standard Scheme: Momentum coefficient interpolation [125]. Requires refined meshes near walls.
- Second Order Scheme: Central differencing. Recommended for most flows.
- Body Force Weighted: Accounts for $\rho g x$ gradients. Ideal for buoyancy/swirl.
- Modified Body Force Weighted: Enhanced stability for VOF/Mixture models.

- PRESTO! Scheme: Staggered control volume method [126]. Default for multiphase flows.

Pressure-Velocity Coupling:

- SIMPLE/SIMPLEC: SIMPLEC [127] accelerates convergence. Skewness correction for distorted meshes.
- PISO [128]: Adds neighbor/skewness corrections. Stabilizes skewed meshes.
- Coupled Algorithm: Solves momentum/continuity simultaneously (incompatible with NITA).
- Fractional Step (FSM): For unstable flows with Non-Iterative Time Advancement.

4.3.3 Hybrid initialization

Generates initial conditions using boundary interpolation and predefined recipes. Constructs a velocity field via Laplace equation to conform to domain geometry and a smooth pressure field. Variables like temperature, turbulence, species, and volume fractions are initialized with domain-averaged or user-specified values.

- Velocity Field: Solved via Laplace equation with boundary conditions: zero normal velocity at walls, user-specified inlets, free-stream matching at far-field, and zero potential at outlets. The "Maintain Constant Velocity Magnitude" option enforces uniform velocity magnitude while preserving flow direction.
- Pressure Field: Derived from a Laplace equation requiring at least one inlet and outlet with explicit pressure data. Boundary treatments include: supersonic/subsonic classification at inlets (based on gauge pressure differences), prescribed gauge pressure at outlets, Supersonic/Initial Gauge Pressure at velocity/mass-flow inlets (if enabled), and zero normal gradient at walls.
- Temperature Field: Initialized to a uniform domain-averaged value by default, with options for alternative distributions.

- **Turbulent Parameters:** Turbulence quantities (e.g., k , ε , ω) are initialized using domain-averaged constants or specialized recipes, enabling uniform or refined distributions.
- **Species Fractions:** Secondary species in multispecies simulations default to zero mass/mole fraction. Users can specify non-zero or non-uniform initial concentrations.
- **Volume Fractions:** For VOF multiphase flows, the secondary phase is initialized to the minimum volume fraction across inlets. Additional phases (beyond two) are set to zero volume fraction to stabilize convergence.





5. CENTRIFUGAL FAN OPTIMIZATION FOR ENHANCED PERFORMANCE

5.1 Software in the Loop for Optimization Process

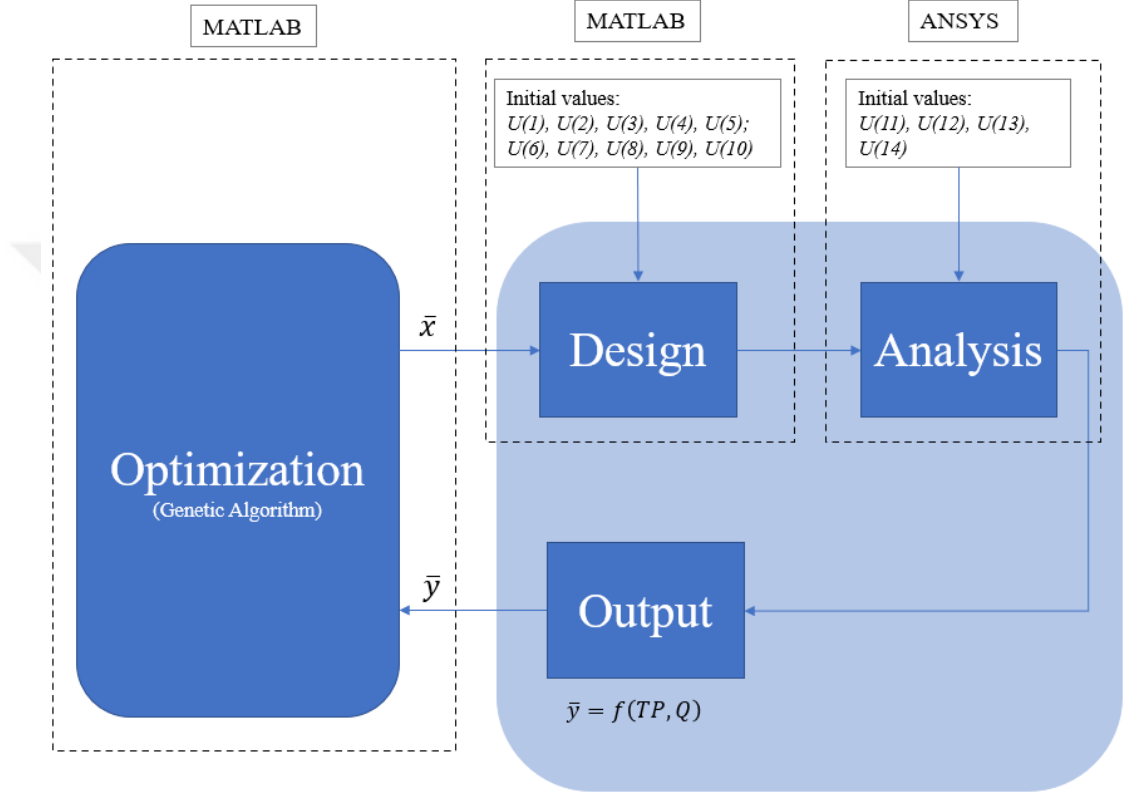


Figure 5.1 : Software in the loop.

The design details of the blade for a two-dimensional centrifugal fan, using the integrated computational approach, are depicted in Figure 5.1. The cost function of GA was defined as a separate function within MATLAB. The settings in the "Solver-Based Optimization Editor" from the MATLAB library were configured for the GA, and the cost function limits (Table 5.1) were established.

The blade optimization uses a CST-based approach and a GA in MATLAB, coupled with ANSYS Fluent for two-dimensional steady-state CFD. A realizable $k-\epsilon$ turbulence model with enhanced wall treatment is employed, using a pressure-based

solver with a coupled pressure-velocity scheme. Gradients are computed via the least squares cell-based method, and second-order schemes are applied to the pressure, momentum, turbulent kinetic energy, and dissipation rate equations. Air is treated as an incompressible fluid, with hybrid initialization, a flow courant number of 200, and 200 iterations per simulation. The nonlinear optimization loop proceeds for 20 generations with a population size of 50, evaluating each candidate blade geometry in Fluent for Total Pressure (Pa) [TP] and Airflow Rate (m^3/s) [Q] before returning results to MATLAB.

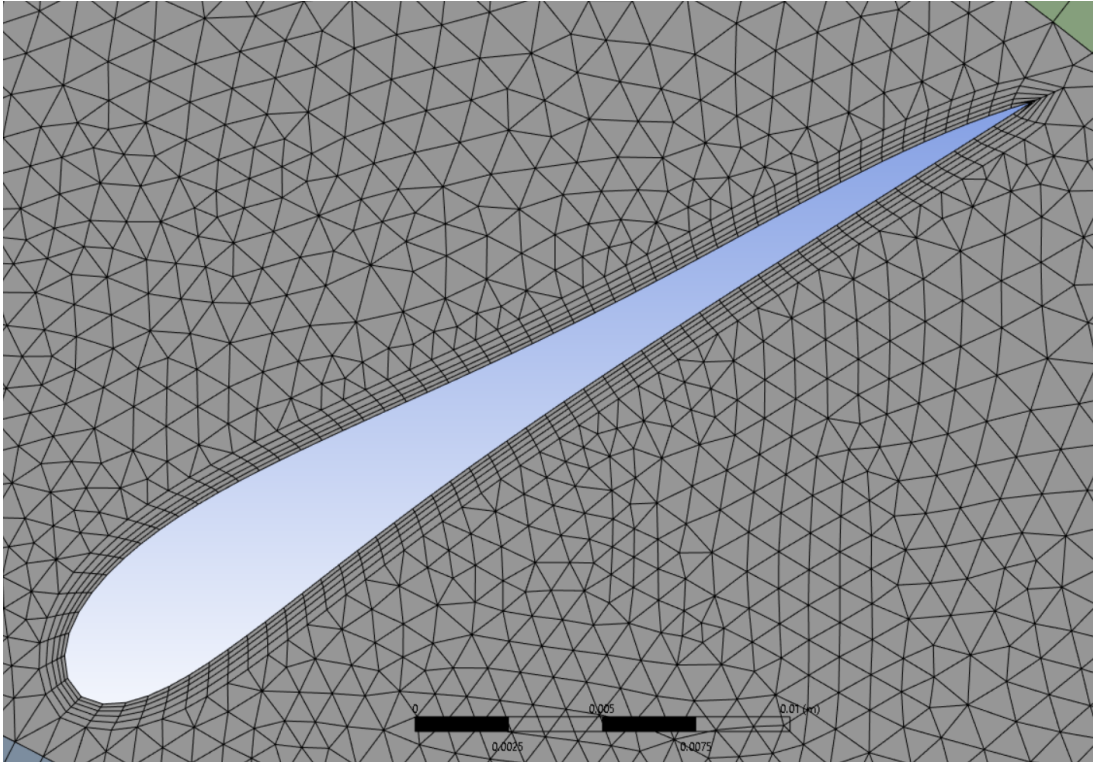


Figure 5.2 : O-type mesh with five layers around the blade.

The computational domain adopts triangular elements near the inlet, an O-type mesh with five layers around the blade (Figure 5.2), and H-type quadrilaterals (Figure 5.3) at the inlet and outlet, producing a final mesh of 56,038 nodes and 67,528 elements, a minimum quality of 7.742×10^{-2} , and an average quality near 0.89. Residuals are driven below 1×10^{-5} . Each solution requires about three minutes on an Intel Core i7 CPU (3.7 GHz), so the entire GA-driven optimization runs for roughly 48 hours. This duration is primarily due to the ANSYS Fluent analysis time, and it is independent of the optimization algorithm itself.

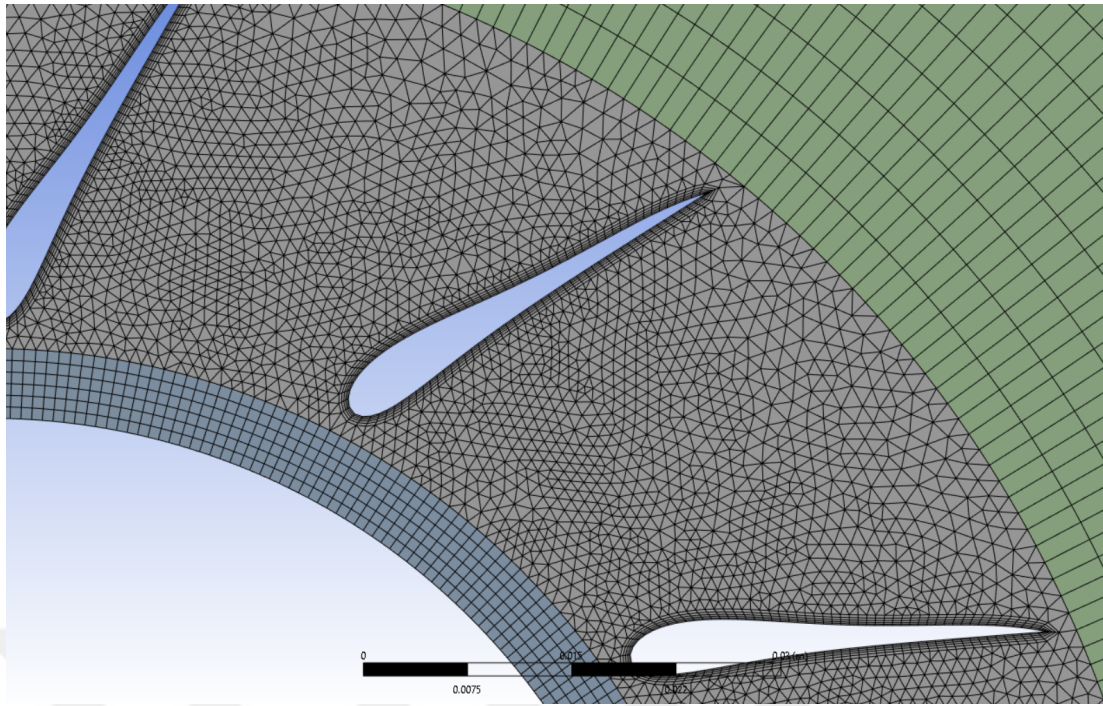


Figure 5.3 : H-type quadrilaterals at the inlet and outlet.

To create each blade profile, the GA produces a complete 14-parameter chromosome. The first ten genes—five lower-surface coefficients U(1)-U(5) and five upper-surface coefficients U(6)-U(10) define the CST curve, while the remaining four genes U(11)-U(14) specify the blade angle, blade count, inner diameter, and rotational speed, respectively. Allowed ranges for all genes are listed in Table 5.1. After MATLAB evaluates the CST equations, the GA-generated coefficients yield 100 coordinate points that form a complete blade cross-section; this geometry-creation step is executed automatically for every individual in every generation.

The initial GA population is random. Figure 5.4 shows a sample cross-section from that first generation, with the region above the $y = 0$ axis corresponding to the upper surface and the region below to the lower surface (defined by equations 2.17 and 2.18). Each GA-defined geometry is exported into a pre-parameterised ANSYS Workbench model, where meshing is performed and the case is solved in Fluent. The CFD solver computes a steady-state flow field, but only two outlet quantities are extracted: total pressure and airflow rate. These two scalar outputs are returned directly to the GA as the objective values driving selection and reproduction. The

entire “GA → geometry-build → CFD → objective-return” loop is fully automated and repeats until the optimization converges.

In this study, a two-dimensional design approach is employed for optimization and analysis instead of a three-dimensional one, for several reasons. First, three-dimensional modeling would require significantly more time to complete. Second, under the given boundary conditions and scenarios, there is no substantial difference between the results obtained from two-dimensional and three-dimensional analyses. Lastly, although three-dimensional modeling may provide more accurate results in cases involving noise, material strength, or complex obstructions, the two-dimensional approach is sufficient and more efficient for the current study’s conditions.

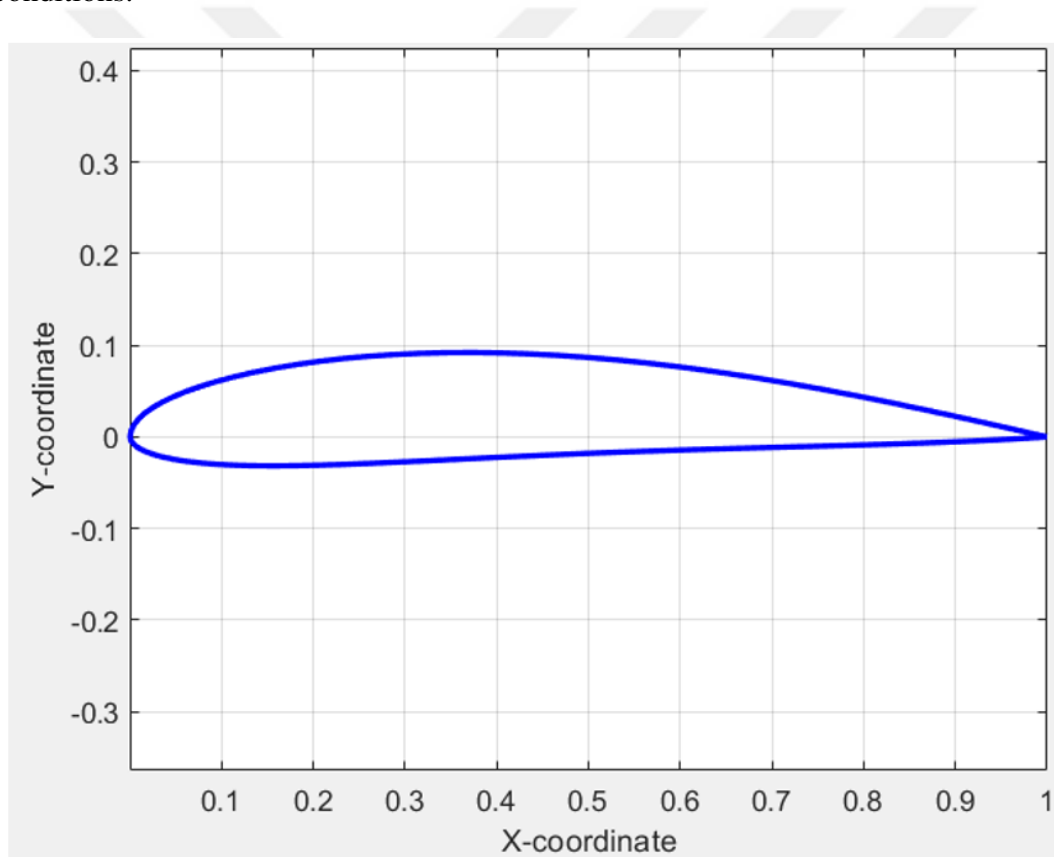


Figure 5.4 : Typical wing airfoil section.

The selection of parameter values in the table is based on several critical factors. For the lower surface vector values U(1) to U(5), the blade thickness is defined between a minimum value of -0.3 and a maximum value of 0. For the upper surface vector values U(6) to U(10), the thickness ranges from 0 to 0.3. To enhance machinability in

CNC processes and avoid excessive blade weight, the maximum thickness is limited to approximately 3 mm.

The blade angle $U(11)$ is set between 15° and 60° . However, when the blade angles at the leading and trailing edges become too large, the passage between adjacent blades may become overly wide and short. This can reduce flow guidance and lead to increased circulatory losses [5]. In such cases, increasing the number of blades can help by narrowing the blade passages. The trailing edge blade angle has a strong influence on both airflow and static pressure. A higher angle at the trailing edge increases air volume but reduces efficiency. Therefore, rather than increasing the trailing edge angle, it is often preferable to limit it for better efficiency and instead enlarge the wheel's outer diameter. A blade angle between 20° and 30° is typically recommended [5].

Table 5.1 : Optimization variables.

Parameter	Value	Min.	Max.	Step
Lower surface vector value	$U(1)$	-0.3	0	0.0001
Lower surface vector value	$U(2)$	-0.3	0	0.0001
Lower surface vector value	$U(3)$	-0.3	0	0.0001
Lower surface vector value	$U(4)$	-0.3	0	0.0001
Lower surface vector value	$U(5)$	-0.3	0	0.0001
Upper surface vector value	$U(6)$	0	0.3	0.0001
Upper surface vector value	$U(7)$	0	0.3	0.0001
Upper surface vector value	$U(8)$	0	0.3	0.0001
Upper surface vector value	$U(9)$	0	0.3	0.0001
Upper surface vector value	$U(10)$	0	0.3	0.0001
Blade angle (degree)	$U(11)$	15	60	1
Blade number	$U(12)$	6	12	1
Inner diameter (mm)	$U(13)$	60	108	1
Rotation speed (RPM)	$U(14)$	2000	6600	1

The number of blades $U(12)$ is set between 6 and 12. This range represents a compromise between two opposing requirements: the channels between blades must be narrow enough to guide the airflow effectively, yet wide enough to minimize resistance. A count of 8 to 12 blades is suitable for most designs, although up to 16

may be required for configurations with large diameter ratios or high inlet angles [5]. However, for the current fan size and application, increasing the blade count beyond twelve offers no aerodynamic advantage and is not practical for CNC manufacturing.

In addition to aerodynamic geometry, dimensional and operational parameters are also considered. The inner diameter $U(13)$ is defined between 60 mm and 108 mm, based on the physical constraints of the fan's installation environment. The rotational speed $U(14)$ is limited to a range of 2000 to 6600 RPM, determined according to the motor's performance characteristics. These values are chosen to optimize the fan's performance while adhering to physical limitations and operational requirements.

5.2 Result

In this study, the primary objective was to derive a blade design that satisfies the specified boundary conditions in the most effective way, achieving a geometry that balances minimized inner diameter and blade count against maximized power output.

$$J = W_1 U(12) + W_2 U(13) - W_3 (TPQ) \quad (5.1)$$

A single-objective optimization framework was employed, using a cost function with penalty terms to address deviations from desired design constraints. Initially, the algorithm was run with a population size of 100 and a maximum generation of 50 without penalties to observe potential outcomes. This preliminary run analyzed convergence behavior and computational efficiency, leading to the selection of a population size of 50 and 20 generations. Initial 100×50 setup shows that the best penalty value was achieved within the first 23 generations and then remains essentially flat for the next 27 generations, even though another 2700 CFD solutions are evaluated. In other words, more than 85% of the improvement occurs in the first 40% of the CPU budget. By contrast, 50×20 compresses the same steep descent into a run that is one-fifth as long: after only 1000 evaluations, the algorithm has already lowered the objective to -599. Measured on a “gain per CFD call” basis, the 50-individual/20-generation configuration is therefore almost four times more efficient

than the 100×50 configuration. Consequently, the thesis adopts the 50×20 setting as the standard, reserving the computational savings for verification cases.

The penalty weights ($W_1 = 100$, $W_2 = 30$, $W_3 = 1$) were assigned to ensure the design remains within the specified limits defined in equation 5.1. The penalty weights were selected by dimensional analysis of the design variables to keep every component of the cost function within a comparable numerical range while still reflecting their relative importance. The combined geometry penalty $W_1U(12) + W_2U(13)$ spans 2400–4440 when $W_1 = 100$ and $W_2 = 30$, a scale that exceeds the maximum achievable benefit from the performance term $W_3(TPQ)$, which lies between 0 and 5039 with $W_3 = 1$. This ordering guarantees that any violation of geometric limits—represented by blade count $U(12)$ and inner diameter $U(13)$ —cannot be fully offset by gains in total-pressure flow rate TPQ , thereby enforcing feasibility as the primary optimization objective. The larger weight on $U(12)$ compensates for its smaller numerical range (6–12) relative to $U(13)$ (60–108 mm), ensuring that both geometry variables exert roughly equal influence on the cost function. Conversely, assigning $W_3 = 1$ maintains sensitivity to aerodynamic performance without allowing extreme TPQ values to dominate the search. Together, these weights produce a balanced optimization landscape in which admissible designs with high performance are favored, yet solutions that exceed geometric constraints are consistently penalized.

The GA iteratively explored and refined the parameter space. As shown in Figure 5.5 and Figure 5.6, a significant improvement in solution quality occurred at the 14th generation. This improvement signaled that the algorithm had begun to converge on promising designs, and by Figure 5.7, the results stabilized, indicating that the best compromise between size, blade number, and power had been reached. The final geometry reflects a lighter, more efficient fan configuration, which was the aim of the penalty-enforced cost function. Figure 5.8 further underscores how crossover, mutation, and elite selection guided the population toward high-performance solutions, showing the transition from a dispersed distribution of individuals to one concentrated around the optimal region of the design space. This convergence highlights that the GA was both thorough in its exploration and effective in honing in on the final, penalty-governed objective.

Table 5.2 summarises the sensitivity of the CST coefficients. With blade angle $U(11)$, blade count $U(12)$, inner diameter $U(13)$ and rotational speed $U(14)$ held constant, the sensitivity study isolates the influence of the CST coefficients. The cost function decreases by 2.07%, confirming a modest overall improvement. On the lower surface, $U(1)$ rises by 0.52%, $U(2)$ and $U(3)$ decrease by 10.71% and 1.03%, $U(4)$ is unchanged, and $U(5)$ increases by 28.19%. On the upper surface, $U(6)$ and $U(7)$ vary by less than 0.05%, $U(8)$ increases by a factor of 14, $U(9)$ falls slightly (−0.08%), and $U(10)$ grows by 18.08%. Because the global blade parameters remain fixed, the performance gain can be attributed to the CST vector, with $U(5)$ and especially $U(8)$ emerging as the most influential variables.

Figure 5.9 shows the velocity magnitude contour plot that reveals how the airflow accelerates along the blade surface. Meanwhile, Figure 5.10 displays the total pressure distribution, showcasing elevated pressure regions near the leading edge and blade tip.

Table 5.2 : Sensitivity analysis of CST variables.

Parameter	Symbol	Case 1	Case 2	Delta (%)
Cost function	J	−1492,03	−1461,17	−2.07
Lower surface vector value	$U(1)$	−0,2685	−0,2699	0.52
Lower surface vector value	$U(2)$	−0,0028	−0,0025	−10.71
Lower surface vector value	$U(3)$	−0,0390	−0,0386	−1.03
Lower surface vector value	$U(4)$	−0,0106	−0,0106	0
Lower surface vector value	$U(5)$	−0,0149	−0,0191	28.19
Upper surface vector value	$U(6)$	0,2603	0,2604	0.04
Upper surface vector value	$U(7)$	0,2124	0,2125	0.05
Upper surface vector value	$U(8)$	0,0109	0,1614	1380.73
Upper surface vector value	$U(9)$	0,1233	0,1232	−0.08
Upper surface vector value	$U(10)$	0,2528	0,2985	18.08
Blade angle (degree)	$U(11)$	58	58	0
Blade number	$U(12)$	12	12	0
Inner diameter (mm)	$U(13)$	108	108	0
Rotation speed (RPM)	$U(14)$	6600	6600	0

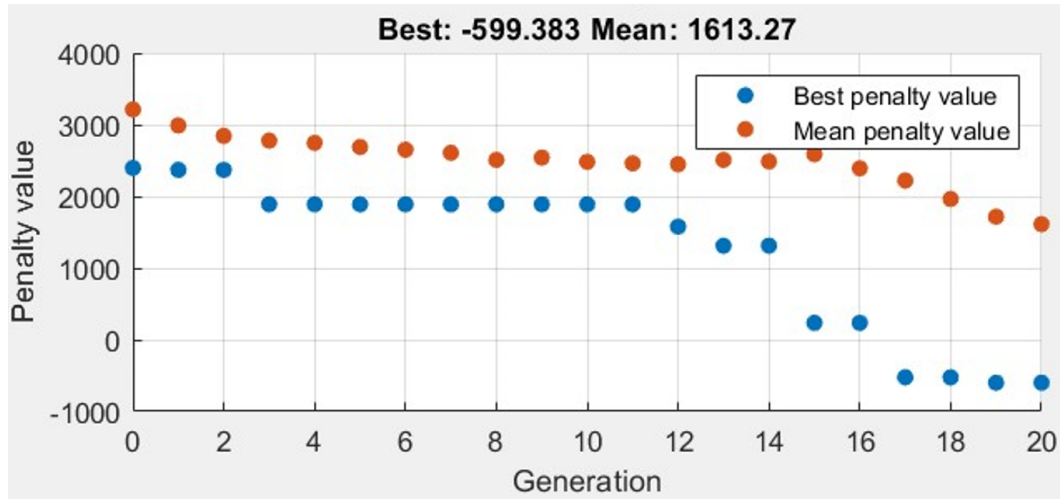


Figure 5.5 : Best and mean values of cost function during optimization.

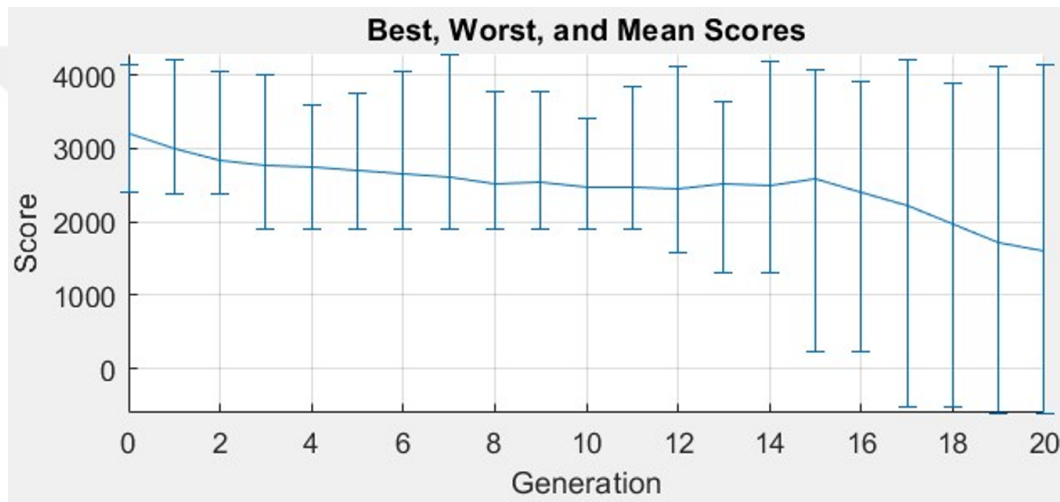


Figure 5.6 : Best, worst, and mean values of cost function.

Further detailing the flow pattern, Figure 5.11 and Figure 5.12 present velocity vector plots, clearly depicting how fluid vectors are oriented around the blade's tip and leading edge. These vector fields confirm that the flow is being guided smoothly according to the imposed rotational speed, with no major separation zones within the analyzed sector. Figure 5.13 and Figure 5.14 offer a more focused view of these contours and vectors, allowing closer inspection of local flow phenomena, such as slight velocity gradients near boundary layers.

These results collectively indicate that the CFD approach—combining mesh refinements and the selected turbulence model—effectively captures the evolving flow field, particularly the interplay of velocity and pressure around critical blade regions.

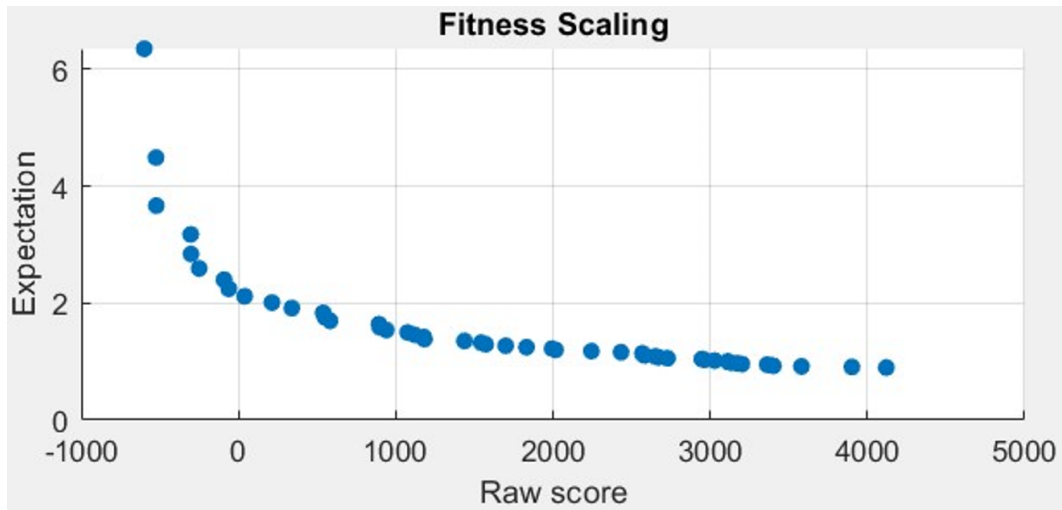


Figure 5.7 : Fitness scaling of cost function during optimization.

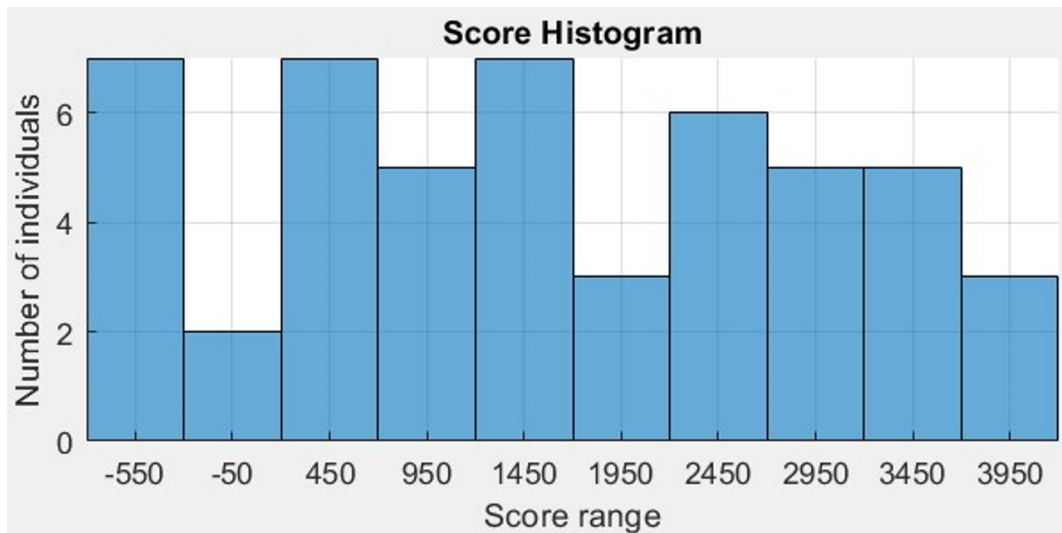


Figure 5.8 : Score histogram of cost function after GA optimization.

In practice, such visualizations assist not only in confirming that the flow is accurately computed given the rotational speed and blade geometry but also in identifying possible improvements, such as refining the blade tip clearance or adjusting local geometry near the leading edge, to achieve greater overall aerodynamic efficiency.

The best nine GA-based cost results are summarized in Table 5.3. Although the optimization initially aimed to reduce the number of blades and the fan’s inner diameter, the highest power outputs converged toward upper-limit values for these parameters. This trade-off reflects the inherent complexity of fan design, where seemingly beneficial constraints (e.g., fewer blades for weight reduction) can diminish performance if pushed too far. The optimization data indicate that total pressure and

airflow rate are closely linked to parameters such as blade count, blade angle, and inner diameter, underscoring the importance of tuning these geometric factors for maximum efficiency.

Table 5.3 : Optimization result.

Cost Function Score	Blade Angle (°)	Blade Number	Inner Dia. (mm)	Rot. Speed (RPM)	Total Pressure (Pa)	Air Flow Rate (m ³ /s)
-599	58	12	108	6571	356	14.14
-551	59	12	108	6543	351	14.20
-524	58	12	108	6544	350	14.18
-523	58	12	107	6538	356	13.84
-308	58	12	108	6532	345	13.72
-303	59	12	108	6557	340	13.92
-301	60	12	108	6593	341	13.89
-250	60	12	108	6515	332	14.11
-156	59	12	107	6551	331	13.76

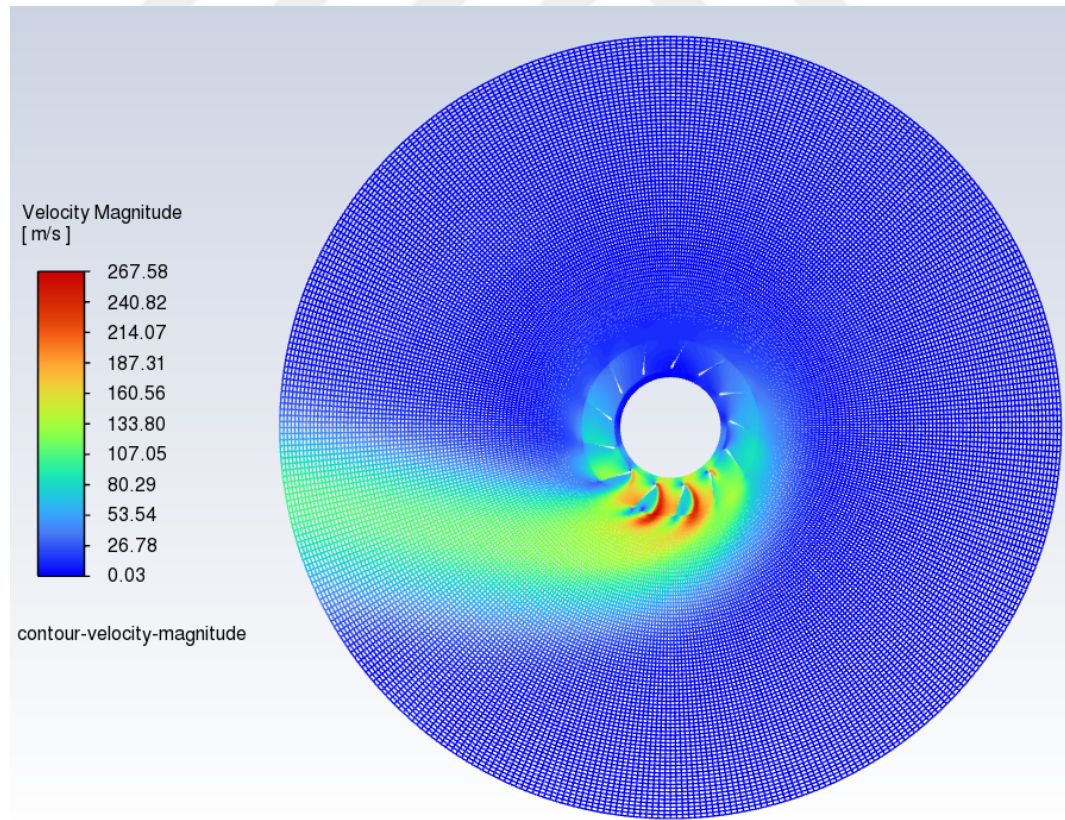


Figure 5.9 : Contour image of velocity magnitude.

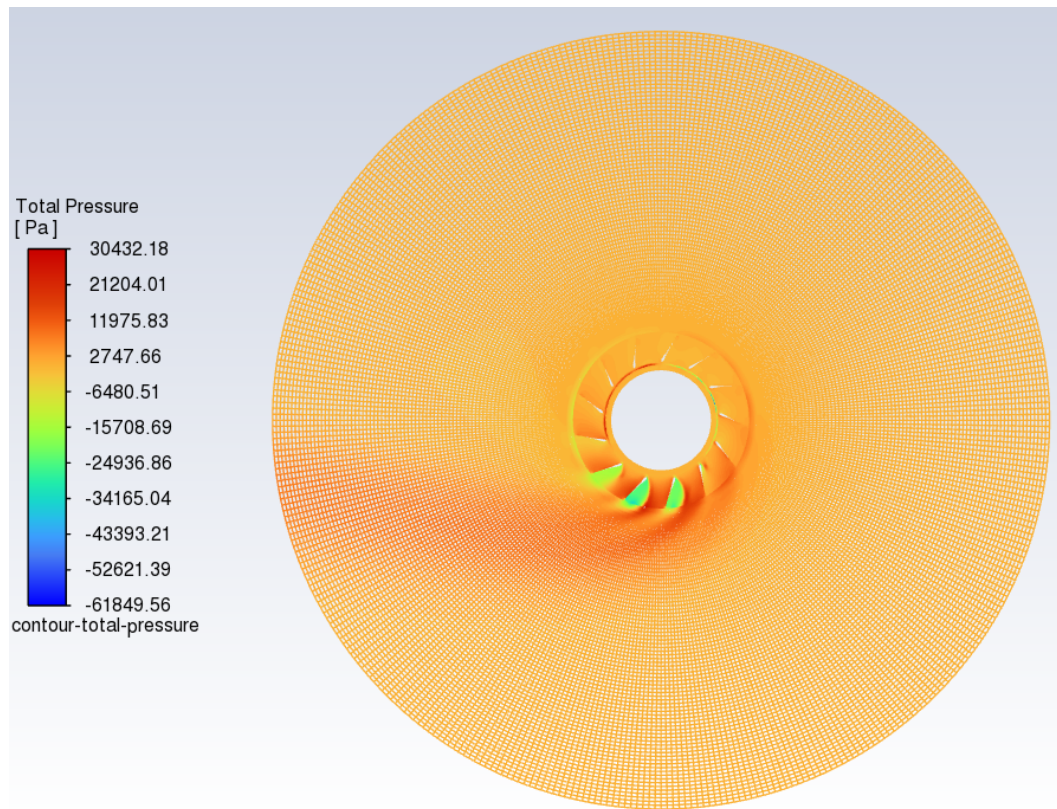


Figure 5.10 : Contour image of total pressure.

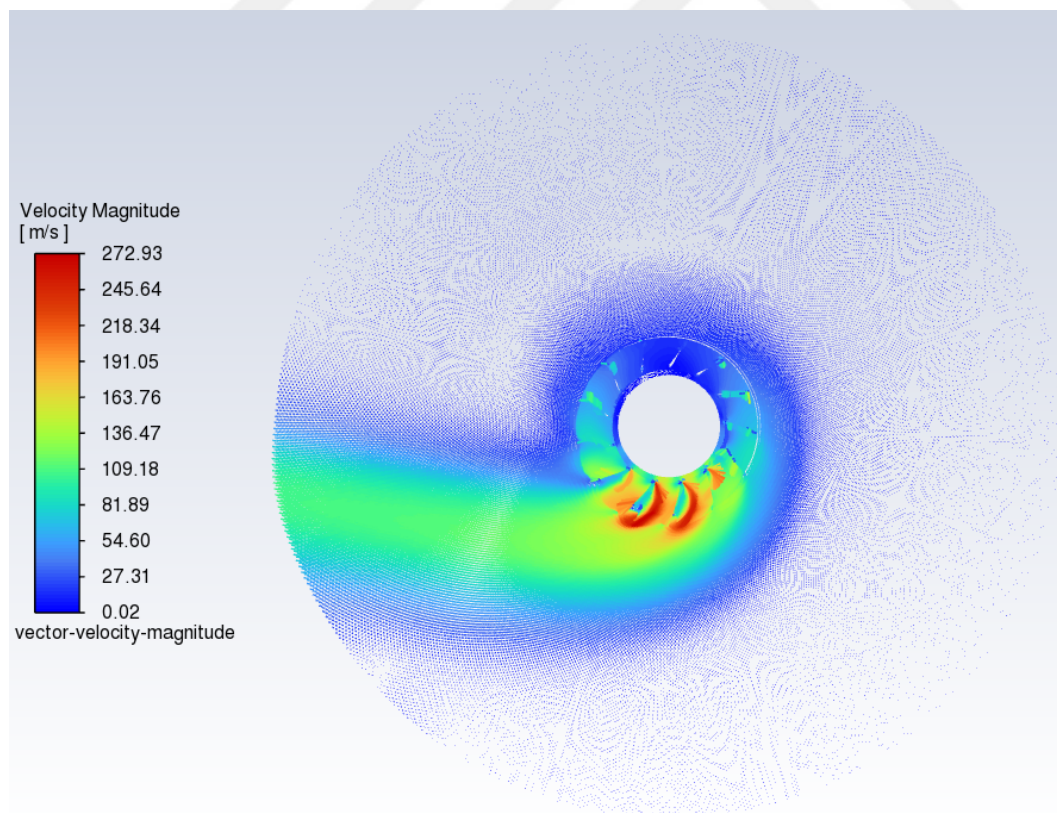


Figure 5.11 : Vector image of velocity magnitude.

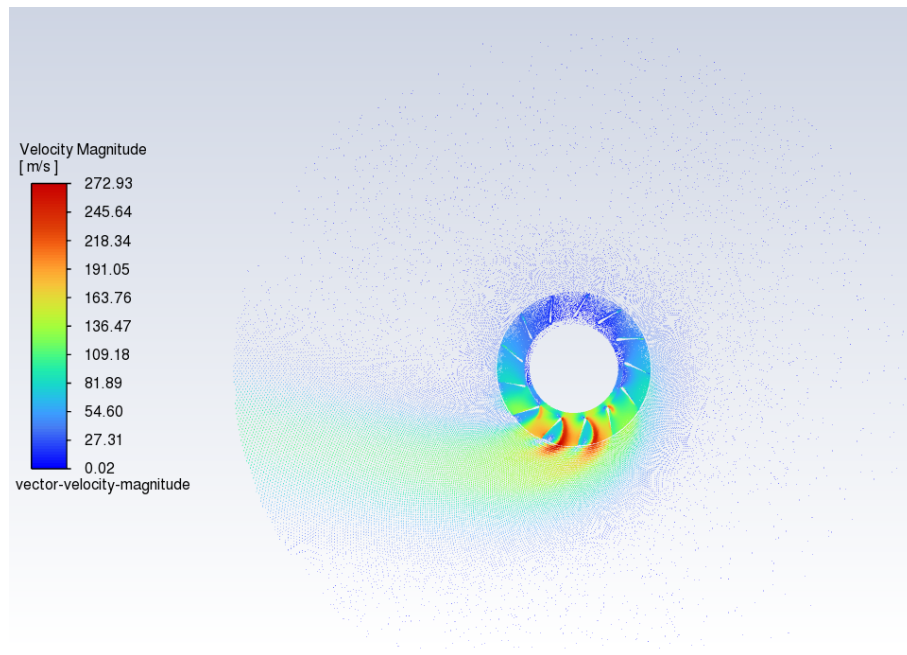


Figure 5.12 : Vector image of velocity magnitude.

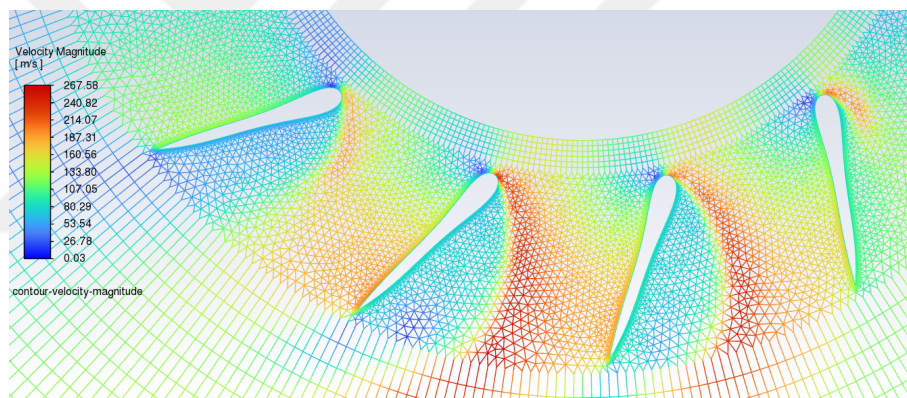


Figure 5.13 : Detailed contour image of velocity magnitude.

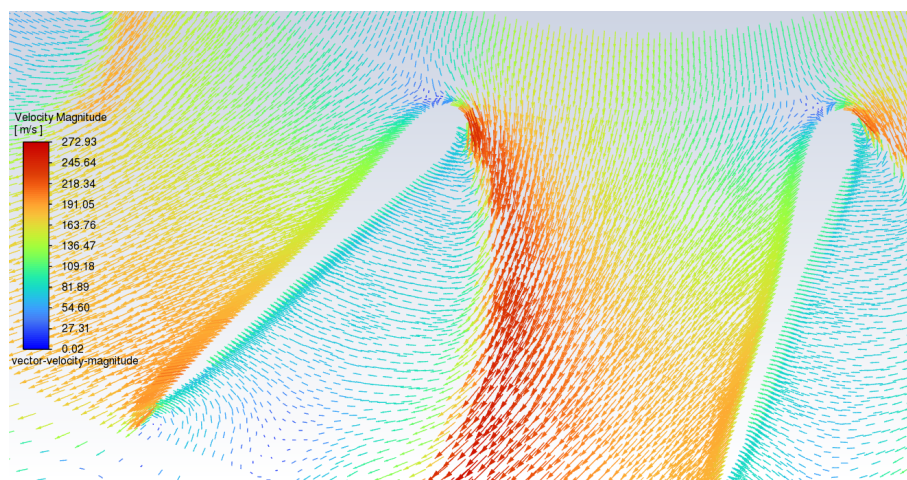


Figure 5.14 : Detailed vector image of velocity magnitude.

To further illustrate these relationships, Figure 5.15 shows a three-dimensional distribution of the objective function values (ObjVal) across three key design parameters: angle, blade number, and inner diameter. The color scale ranges from blue (lower ObjVal) to red (higher ObjVal), and a marker identifies the minimum ObjVal point. As seen in this plot, the denser clustering of high-performance solutions skews toward higher blade numbers and larger inner diameters, confirming the numerical trend that emerged in Table 5.3. However, the behavior of the angle parameter is less uniform, suggesting that it can be fine-tuned to exploit synergies with the other dimensions.

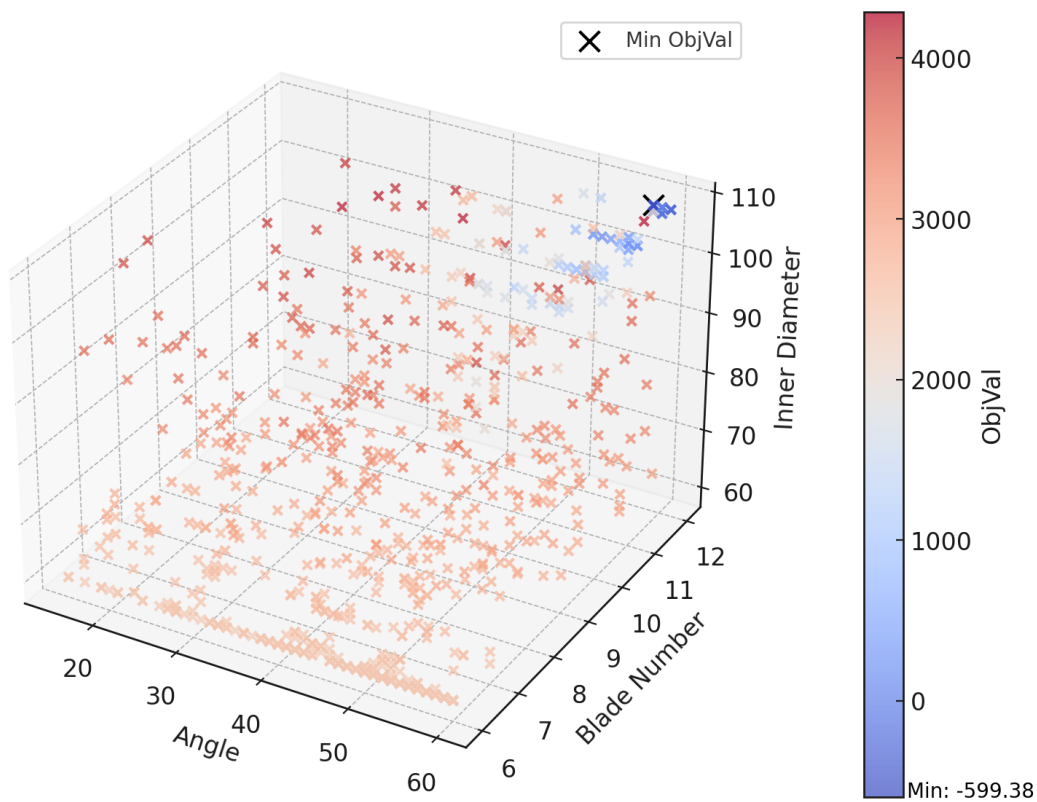


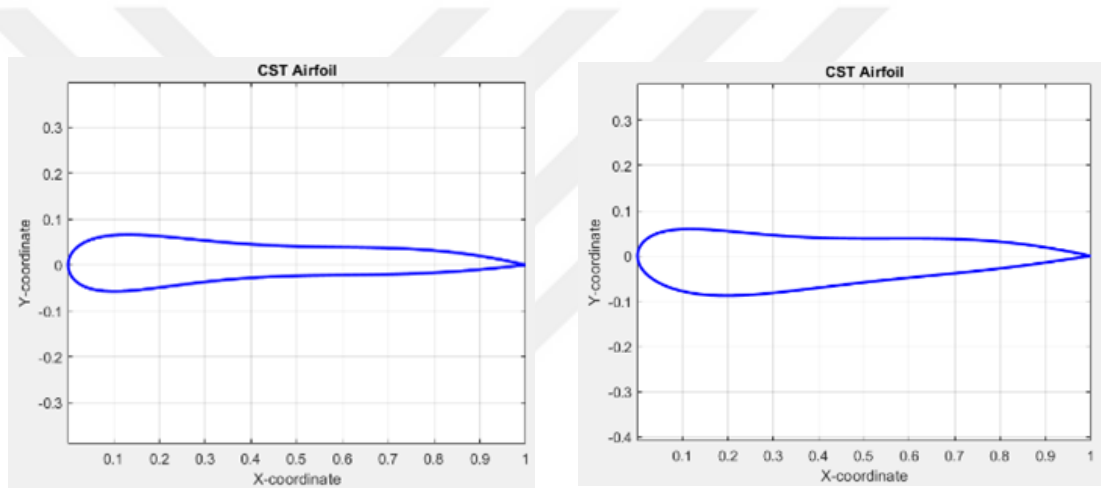
Figure 5.15 : Objval distribution across design parameters.

Since individual parameters can obscure the influence of blade geometry, the study further compares blade cross-sections under identical parameter conditions (Figure 5.16) to isolate the specific role of shape in enhancing or constraining fan performance. Table 5.4 correlates these geometric details with computed aerodynamic values, verifying that maximizing total pressure and airflow depends as much on

local blade contours as on the global parameters. These findings provide a solid reference point for testing theoretical models and validate the practical importance of multi-parameter adjustments.

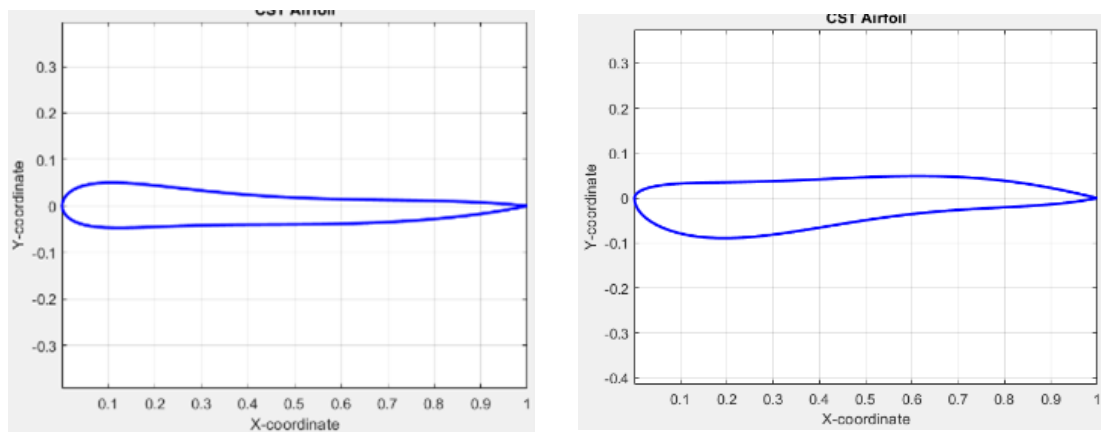
Table 5.4 : Performance values from randomly selected samples.

Figure 5.16	Cost Function Score	Blade Angle (°)	Blade Number	Inner Dia. (mm)	Rot. Speed (RPM)	Total Pressure (Pa)	Air Flow Rate (m ³ /s)
A	-599	58	12	108	6571	356	14.14
B	472	58	12	108	6537	310	12.76
C	2044	58	12	107	5865	219	10.79
D	3614	56	12	91	6597	72	4.33



(a) Sample - A.

(b) Sample - B.



(c) Sample - C.

(d) Sample - D.

Figure 5.16 : Benchmarking selected airfoil shapes.

Analyzing total pressure (Pa) and airflow rate (m^3/s) across designs A through D reveals notable performance shifts. From A to B, total pressure decreases by 12.92% and airflow by 9.76%, indicating a significant efficiency loss. This decline is attributed primarily to geometric modifications, as the accompanying rotational speed reduction (from 6571 rpm to 6537 rpm, a 0.52% drop) is too minor to explain such pronounced effects. The transition from B to C sees more pronounced declines of 29.35% in total pressure and 15.44% in airflow. The steep decline (29.35%) aligns with the quadratic relationship between pressure and rotational speed, proportional to RPM^2 . The 10.28% RPM reduction alone would theoretically reduce pressure by $\sim 19.7\%$ (equation 1.2) [5]. The remaining $\sim 9.65\%$ loss is attributed to geometric inefficiencies (e.g., blade shape or curvature). The 15.44% airflow reduction reflects both the linear dependence on RPM (10.28% drop) from equation 1.1 [5] and geometric changes (e.g., increased flow separation or reduced blade pitch), which account for the additional $\sim 5.16\%$ loss. The most significant reductions occur between C and D, with total pressure decreasing by 67.12% and airflow by 59.87%. The inner diameter decreases by 14.95% (from 107 mm to 91 mm), while the rotational speed increases by 12.48% (from 5865 RPM to 6597 RPM). Although the theoretical airflow (Q) reduction is 30.9% based on equation 1.5, the actual decrease is 59.87%, indicating an extra loss of 28.97% due to blade geometric design effects. Similarly, the expected pressure drop of 8.6% based on equation 1.6 is exceeded by an actual drop of 67.12%, meaning that an additional 58.52% loss is attributable to the design geometry.

These modifications substantially degrade aerodynamic performance across Designs A to D, with the most pronounced losses observed at the final stage. Geometric deviations amplify losses beyond theoretical predictions, highlighting the critical role of design geometry in system efficiency. Among the four configurations, Design A demonstrates the highest total pressure and airflow rate, establishing it as the optimal choice for maximizing efficiency.

6. CONCLUSION

This study successfully demonstrates the robust optimization of a two-dimensional centrifugal fan blade design by integrating CST-based geometry parameterization, a MATLAB-driven genetic algorithm, and ANSYS-based CFD simulations. Key findings reveal that carefully calibrated design parameters—such as blade angles, rotational speed, and inner diameter—can significantly enhance total pressure and airflow rate while adhering to practical constraints. Although focused on a civil aircraft galley oven fan, the methodology is not confined to this application: its parametric flexibility and iterative optimization framework make it broadly adaptable to any scenario requiring high-performance fan design.

In future work, we plan to validate the predicted improvements through physical prototyping of three designs. Subsequent steps will include a comprehensive three-dimensional Fluent analysis to assess recirculation zones and watertight constraints, better aligning simulations with real-world operational dynamics. Further refinements may explore multi-section airfoil designs or expand the optimization framework from single-objective penalty-based methods to multiobjective framework competing priorities like noise reduction, structural integrity, and performance across multiple operating points. By extending beyond two-dimensional analysis, this methodology will enhance adaptability for high-fidelity applications in turbomachinery and aerodynamic systems, underscoring its versatility beyond civil aircraft galley ovens. This approach's modular architecture allows seamless integration of additional constraints or objectives, paving the way for scalable solutions in diverse engineering contexts. Such advancements could redefine efficiency benchmarks in industrial fan design, from HVAC systems to aerospace propulsion.



REFERENCES

- [1] **Pathak, M., Sharma, P. and Yadav, A.** International journal of engineering sciences research technology review of numerical analysis of aerofoil section of blade of axial flow fan.
- [2] **Cai, J.C., Chen, H.J., Brazhenko, V. and Gu, Y.H.** (2021). Study of the hydrodynamic unsteady flow inside a centrifugal fan and its downstream pipe using detached eddy simulation, *Sustainability*, 13(9), 5113.
- [3] **EP, D., IT, N. and OFA, S.S.** US Department of Energy Energy Efficiency and Renewable Energy.
- [4] **Pranav, C.A. and Raj, R.T.K.** (2012). Numerical design and parametric optimization of centrifugal fans with airfoil blade impellers, *Research Journal of Recent Sciences*, 1(10), 7–11.
- [5] **Bleier, F.P.** (2018). *Fan handbook: selection, application, and design*, McGraw-Hill.
- [6] **Biedermann, T.M., Moutamassik, Y. and Kameier, F.** (2021). Feasibility Study on the Effect of Blade Inclination for Heavy Duty Centrifugal Fans–Aerodynamic Aspects, *Turbo Expo: Power for Land, Sea, and Air*, volume84898, American Society of Mechanical Engineers, p.V001T10A001.
- [7] **Eck, B.** (1972). *Entwurf und Betrieb der Radial-, Axial- und Querstromventilatoren*.
- [8] **Bommes, L.** (2003). *Ventilatoren*, Vulkan-Verlag GmbH.
- [9] **Casey, M., Gersbach, F. and Robinson, C.** (2008). An optimization technique for radial compressor impellers, *Turbo Expo: Power for Land, Sea, and Air*, volume43161, pp.2401–2411.
- [10] **Ding, Y., Wang, J., Jiang, B., Li, Z., Xiao, Q., Wu, L. and Xie, B.** (2022). Multi-objective optimization for the radial bending and twisting law of axial fan blades, *Processes*, 10(4), 753.
- [11] **Heo, M.W., Kim, J.H., Seo, T.W. and Kim, K.Y.** (2016). Aerodynamic and aeroacoustic optimization for design of a forward-curved blades centrifugal fan, *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, 230(2), 154–174.

- [12] **Xiao, Q., Wang, J., Jiang, B., Yang, W. and Yang, X.** (2022). Multi-objective optimization of squirrel cage fan for range hood based on Kriging model, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 236(1), 219–232.
- [13] **Gan, X., Pei, J., Wang, W., Yuan, S. and Lin, B.** (2023). Application of a modified MOPSO algorithm and multi-layer artificial neural network in centrifugal pump optimization, *Engineering Optimization*, 55(4), 580–598.
- [14] **Kulfan, B.M.** (2008). Universal parametric geometry representation method, *Journal of aircraft*, 45(1), 142–158.
- [15] **Akram, M.T. and Kim, M.H.** (2021). CFD analysis and shape optimization of airfoils using class shape transformation and genetic algorithm—Part I, *Applied Sciences*, 11(9), 3791.
- [16] **Huang, S., Lu, X., Han, G., Zhao, S., Zhou, C. and Yang, C.** (2022). Research on aerodynamic optimization design method and flow mechanism of a high-subsonic compressor cascade, *Engineering Applications of Computational Fluid Mechanics*, 16(1), 316–334.
- [17] **Liu, C., Duan, Y., Cai, J. and Wang, J.** (2016). Application of the 3D multi-block CST method to hypersonic aircraft optimization, *Aerospace Science and Technology*, 50, 295–303.
- [18] **Zhu, M., Li, Z., Li, G., Ye, X., Liu, Y., Chen, Z. and Li, N.** (2023). An Investigation on Optimized Performance of Voluteless Centrifugal Fans by a Class and Shape Transformation Function, *Processes*, 11(6), 1751.
- [19] **Zhao, X., Guan, J., Wang, T., Liu, X., Xu, Q. and Zhou, J.** (2024). Optimization of Impeller Structure Parameters of a Centrifugal Fan in a Powered Air-Purifying Respirator Power System, *Processes*, 12(2), 353.
- [20] **Zhang, H., Wen, J., Zhan, J. and Xin, D.** (2022). Effects of blade number on the aerodynamic performance and wake characteristics of a small horizontal-axis wind turbine, *Energy Conversion and Management*, 273, 116410.
- [21] **Tantakitti, P., Pattana, S. and Wiratkasem, K.** (2018). The performance of backward curved centrifugal fan with various impeller inlet and outlet angles, *2018 Third International Conference on Engineering Science and Innovative Technology (ESIT)*, IEEE, pp.1–3.
- [22] **Kim, K.Y. and Seo, S.J.** (2006). Application of numerical optimization technique to design of forward-curved blades centrifugal fan, *JSME International Journal Series B Fluids and Thermal Engineering*, 49(1), 152–158.
- [23] **Ottersten, M., Yao, H. and Davidson, L.** (2022). Inlet gap effect on aerodynamics and tonal noise generation of a voluteless centrifugal fan, *Journal of Sound and Vibration*, 540, 117304.

- [24] **Prezelj, J. and Novaković, T.** (2018). Centrifugal fan with inclined blades for vacuum cleaner motor, *Applied acoustics*, 140, 13–23.
- [25] **Ballesteros-Tajadura, R., Velarde-Suárez, S., Hurtado-Cruz, J.P. and Santolaria-Morros, C.** (2006). Numerical calculation of pressure fluctuations in the volute of a centrifugal fan.
- [26] **Corsini, A., Delibra, G., Rispoli, F., Sheard, A.G. and Venturini, P.** (2013). Aerodynamic simulation of a high-pressure centrifugal fan for process industries, *Turbo Expo: Power for Land, Sea, and Air*, volume 55188, American Society of Mechanical Engineers, p.V004T10A009.
- [27] **Younsi, M., Djerrada, A., Belamri, T. and Menter, F.** (2008). Application of the SAS turbulence model to predict the unsteady flow field behaviour in a forward centrifugal fan, *International journal of computational fluid dynamics*, 22(9), 639–648.
- [28] **Wang, K., Ju, Y. and Zhang, C.** (2020). Experimental and numerical investigations on effect of blade trimming on aerodynamic performance of squirrel cage fan, *International Journal of Mechanical Sciences*, 177, 105579.
- [29] **Huang, Q., Qin, D. and Pan, G.** (2022). Numerical simulation of the wake dynamics of the pumpjet propulsor in oblique inflow, *Physics of Fluids*, 34(6).
- [30] **Ottersten, M., Yao, H.D. and Davidson, L.** (2021). Tonal noise of voluteless centrifugal fan generated by turbulence stemming from upstream inlet gap, *Physics of Fluids*, 33(7).
- [31] **Song, H., Zhang, J., Huang, P., Cai, H., Cao, P. and Hu, B.** (2020). Analysis of rotor-stator interaction of a pump-turbine with splitter blades in a pump mode, *Mathematics*, 8(9), 1465.
- [32] **Spence, R. and Amaral-Teixeira, J.** (2009). A CFD parametric study of geometrical variations on the pressure pulsations and performance characteristics of a centrifugal pump, *Computers & Fluids*, 38(6), 1243–1257.
- [33] **Yang, J., Meng, L., Zhou, L., Luo, Y. and Wang, Z.** (2013). Unsteady internal flow field simulations in a double suction centrifugal fan, *Engineering Computations*, 30(3), 345–356.
- [34] **Gao, Z., Zhu, W., Lu, L., Deng, J., Zhang, J. and Wuang, F.** (2014). Numerical and experimental study of unsteady flow in a large centrifugal pump with stay vanes, *Journal of Fluids Engineering*, 136(7), 071101.
- [35] **Yao, Z., Yang, M., Xiao, R. and Wang, F.** (2018). Influence of wall roughness on the static performance and pressure fluctuation characteristics of a double-suction centrifugal pump, *Proceedings of the Institution of*

Mechanical Engineers, Part A: Journal of Power and Energy, 232(7), 826–840.

- [36] **Menter, F.R., Kuntz, M., Langtry, R. et al.** (2003). Ten years of industrial experience with the SST turbulence model, *Turbulence, heat and mass transfer*, 4(1), 625–632.
- [37] **Bozorgasareh, H., Khalesi, J., Jafari, M. and Gazori, H.O.** (2021). Performance improvement of mixed-flow centrifugal pumps with new impeller shrouds: Numerical and experimental investigations, *Renewable Energy*, 163, 635–648.
- [38] **Li, D., Gong, R., Wang, H., Zhang, J., Wei, X. and Shu, L.** (2015). Numerical investigation in the vaned distributor under different guide vanes openings of a pump turbine in pump mode, *Journal of Applied Fluid Mechanics*, 9(1), 253–266.
- [39] **Li, K., Chen, X., Dou, H., Zhu, Z., Zheng, L. and Luo, X.** (2019). Study of flow instability in a miniature centrifugal pump based on energy gradient method, *Journal of Applied Fluid Mechanics*, 12(3), 701–713.
- [40] **Yang, H., Yu, P., Xu, J., Ying, C., Cao, W., Wang, Y., Zhu, Z. and Wei, Y.** (2019). Experimental investigations on the performance and noise characteristics of a forward-curved fan with the stepped tongue, *Measurement and Control*, 52(9-10), 1480–1488.
- [41] **Patel, J. and Makadia, R.** A Review on Performance of Industrial Induced Draft Centrifugal Fan Using Ansys Cfx.
- [42] **Engin, T., Gur, M. and Scholz, R.** (2006). Effects of tip clearance and impeller geometry on the performance of semi-open ceramic centrifugal fan impellers at elevated temperatures, *Experimental Thermal and Fluid Science*, 30(6), 565–577.
- [43] **Singh, R.R. and Nataraj, M.** (2012). Optimizing impeller geometry for performance enhancement of a centrifugal blower using the taguchi quality concept, *International Journal of Engineering Science and Technology*, 4(10).
- [44] **Lin, S.C. and Huang, C.L.** (2002). An integrated experimental and numerical study of forward-curved centrifugal fan, *Experimental thermal and fluid science*, 26(5), 421–434.
- [45] **Khalkhali, A., Farajpoor, M. and Safikhani, H.** (2011). Modeling and multi-objective optimization of forward-curved blade centrifugal fans using CFD and neural networks, *Transactions of the Canadian Society for Mechanical Engineering*, 35(1), 63–79.
- [46] **Karanth, K.V. and Sharma, N.Y.** (2009). CFD Analysis on the Effect of Radial Gap on Impeller-Diffuser Flow Interaction as well as on the Flow

Characteristics of a Centrifugal Fan, *International Journal of Rotating Machinery*, 2009(1), 293508.

- [47] **Chunxi, L., Ling, W.S. and Yakui, J.** (2011). The performance of a centrifugal fan with enlarged impeller, *Energy Conversion and Management*, 52(8-9), 2902–2910.
- [48] **Liu, X., Dang, Q. and Xi, G.** (2008). Performance improvement of centrifugal fan by using CFD, *Engineering Applications of Computational Fluid Mechanics*, 2(2), 130–140.
- [49] **Lin, S.C. and Tsai, M.L.** (2012). An integrated performance analysis for a backward-inclined centrifugal fan, *Computers & Fluids*, 56, 24–38.
- [50] **Raj, D. and Swim, W.** (1981). Measurements of the mean flow velocity and velocity fluctuations at the exit of an FC centrifugal fan rotor.
- [51] **Han, S.Y. and Maeng, J.S.** (2003). Shape optimization of cut-off in a multi-blade fan/scroll system using neural network, *International journal of heat and mass transfer*, 46(15), 2833–2839.
- [52] **Liu, Q., Qi, D. and Mao, Y.** (2006). Numerical calculation of centrifugal fan noise, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 220(8), 1167–1177.
- [53] **Wang, C., Zhang, Y., Yuan, Z. and Ji, K.** (2020). Development and application of the entropy production diagnostic model to the cavitation flow of a pump-turbine in pump mode, *Renewable Energy*, 154, 774–785.
- [54] **Wang, Z., Xie, B., Xia, X., Yang, H., Zuo, Q. and Liu, Z.** (2022). Energy loss of radial inflow turbine for organic Rankine cycle using mixture based on entropy production method, *Energy*, 245, 123312.
- [55] **Kulfan, B. and Bussoletti, J.** (2006). " Fundamental" parameteric geometry representations for aircraft component shapes, *11th AIAA/ISSMO multidisciplinary analysis and optimization conference*, p.6948.
- [56] **Samareh, J.A.** (2001). Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization, *AIAA journal*, 39(5), 877–884.
- [57] **Juhász, I. and Hoffmann, M.** (2004). Constrained shape modification of cubic B-spline curves by means of knots, *Computer-Aided Design*, 36(5), 437–445.
- [58] **Berglund, T., Jonsson, H. and Soderkvist, I.** (2003). An obstacle-avoiding minimum variation b-spline problem, *2003 International Conference on Geometric Modeling and Graphics, 2003. Proceedings, IEEE*, pp.156–161.

- [59] **Tuli, M., Reddy, N.V. and Saxena, A.** (2006). Constrained shape modification of b-spline curves, *Computer-Aided Design and Applications*, 3(1-4), 437–446.
- [60] **Pourazady, M. and Xu, X.** (2000). Direct manipulations of B-spline and NURBS curves, *Advances in Engineering Software*, 31(2), 107–118.
- [61] **Celniker, G. and Welch, W.** (1992). Linear constraints for deformable non-uniform b-spline surfaces, *Proceedings of the 1992 Symposium on Interactive 3D graphics*, pp.165–170.
- [62] **Haftka, R.T. and Grandhi, R.V.** (1986). Structural shape optimization—a survey, *Computer methods in applied mechanics and engineering*, 57(1), 91–106.
- [63] **Ding, Y.** (1986). Shape optimization of structures: a literature survey, *Computers & Structures*, 24(6), 985–1004.
- [64] **Barr, A.H.** (1984). Global and local deformations of solid primitives, *ACM Siggraph Computer Graphics*, 18(3), 21–30.
- [65] **Sederberg, T.W. and Parry, S.R.** (1986). Free-form deformation of solid geometric models, *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pp.151–160.
- [66] **Nadarajah, S., Castonguay, P. and Mousavi, A.** (2007). Survey of shape parameterization techniques and its effect on three-dimensional aerodynamic shape optimization, *18th AIAA computational fluid dynamics conference*, p.3837.
- [67] **Morris, A., Allen, C. and Rendall, T.** (2008). A Generic Parallel Framework for Shape Parameterisation and CFD-Based Optimisation, *26th AIAA Applied Aerodynamics Conference*, p.6220.
- [68] **Sevant, N., Bloor, M. and Wilson, M.** (1998). A hierarchical approach to optimal aerodynamic design, *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, p.4850.
- [69] **Straathof, M.H.** (2012). Shape parameterization in aircraft design: a novel method, based on B-splines.
- [70] **Hicks, R.M. and Henne, P.A.** (1978). Wing design by numerical optimization, *Journal of Aircraft*, 15(7), 407–412.
- [71] **Abramowitz, M.** (1964). Handbook of mathematical functions with formulas, graphs, and mathematical tables, nbs, *Applied Math. Series*, 55, 232.
- [72] **Sobieczky, H.**, (1999). Parametric airfoils and wings, Recent development of aerodynamic design methodologies: inverse design and optimization, Springer, pp.71–87.

- [73] **Kumar, V., Chhabra, J.K. and Kumar, D.** (2014). Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems, *Journal of Computational Science*, 5(2), 144–155.
- [74] **Bonabeau, E., Dorigo, M. and Theraulaz, G.** (1999). *Swarm intelligence: from natural to artificial systems*, 1, Oxford university press.
- [75] **Kumar, V. and Kumar, D.** (2017). An astrophysics-inspired grey wolf algorithm for numerical optimization and its application to engineering design problems, *Advances in Engineering Software*, 112, 231–254.
- [76] **Katoch, S., Chauhan, S.S. and Kumar, V.** (2021). A review on genetic algorithm: past, present, and future, *Multimedia tools and applications*, 80, 8091–8126.
- [77] **Holland, J.H.** (1975). Adaptation in natural and artificial systems, *University of Michigan Press google schola*, 2, 29–41.
- [78] **Michalewicz, Z.** (2013). *Genetic algorithms+ data structures= evolution programs*, Springer Science & Business Media.
- [79] **Michalewicz, Z. and Schoenauer, M.** (1996). Evolutionary algorithms for constrained parameter optimization problems, *Evolutionary computation*, 4(1), 1–32.
- [80] **Liu, D.** (2019). Mathematical modeling analysis of genetic algorithms under schema theorem, *Journal of Computational Methods in Sciences and Engineering*, 19(S1), 131–137.
- [81] **Goldberg, D.E.** (1989). Genetic Algorithm in Search, Optimization and Machine Learning, Addison, *Wesley Publishing Company, Reading, MA*, 1(98), 9.
- [82] **Helal, M.H., Fan, C.T., Liu, D.Y. and Yuan, S.M.** (2017). Peer-to-peer based parallel genetic algorithm, *2017 International Conference on Information, Communication and Engineering (ICICE)*, IEEE, pp.535–538.
- [83] **Fluent, A. et al.** (2011). Ansys fluent theory guide, *Ansys Inc., USA*, 15317, 724–746.
- [84] **Lee, J.Y., Kim, M.S., Kim, C.T. and Lee, J.J.** (2007). Study on encoding schemes in compact genetic algorithm for the continuous numerical problems, *SICE Annual Conference 2007*, IEEE, pp.2694–2699.
- [85] **Shi, G., Iima, H. and Sannomiya, N.** (1996). A new encoding scheme for solving job shop problems by genetic algorithm, *Proceedings of 35th IEEE Conference on Decision and Control*, volume 4, IEEE, pp.4395–4400.
- [86] **Sivanandam, S., Deepa, S., Sivanandam, S. and Deepa, S.** (2008). *Genetic algorithms*, Springer.
- [87] **Kumar, A.** (2013). Encoding schemes in genetic algorithm, *International Journal of Advanced Research in IT and Engineering*, 2(3), 1–7.

- [88] **Lima, S.J.d.A. and de Araújo, S.A.** (2018). A new binary encoding scheme in genetic algorithm for solving the capacitated vehicle routing problem, *Bioinspired Optimization Methods and Their Applications: 8th International Conference, BIOMA 2018, Paris, France, May 16-18, 2018, Proceedings 8*, Springer, pp.174–184.
- [89] **Fox, B. and McMahon, M.**, (1991). Genetic operators for sequencing problems, *Foundations of genetic algorithms*, volume 1, Elsevier, pp.284–300.
- [90] **Jebari, K., Madiafi, M. et al.** (2013). Selection methods for genetic algorithms, *International Journal of Emerging Sciences*, 3(4), 333–344.
- [91] **Abdulal, W. and Ramachandram, S.** (2011). Reliability-aware genetic scheduling algorithm in grid environment, *2011 International Conference on Communication Systems and Network Technologies*, IEEE, pp.673–677.
- [92] **Sharma, S. and Gupta, K.** (2011). Solving the traveling salesmen problem through genetic algorithm with new variation order crossover, *2011 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, IEEE, pp.274–276.
- [93] **Lee, C.Y.** (2003). Entropy-Boltzmann selection in the genetic algorithms, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(1), 138–149.
- [94] **Saini, N.** (2017). Review of selection methods in genetic algorithms, *International Journal of Engineering and Computer Science*, 6(12), 22261–22263.
- [95] **Soon, G.K., Guan, T.T., On, C.K., Alfred, R. and Anthony, P.** (2013). A comparison on the performance of crossover techniques in video game, *2013 IEEE international conference on control system, computing and engineering*, IEEE, pp.493–498.
- [96] **Goldberg, D.E. and Lingle, R.** (2014). Alleles, loci, and the traveling salesman problem, *Proceedings of the first international conference on genetic algorithms and their applications*, Psychology Press, pp.154–159.
- [97] **Hussain, A., Muhammad, Y.S., Nauman Sajid, M., Hussain, I., Mo-hamd Shoukry, A. and Gani, S.** (2017). Genetic algorithm for traveling salesman problem with modified cycle crossover operator, *Computational intelligence and neuroscience*, 2017(1), 7430125.
- [98] **Ray, S.S., Bandyopadhyay, S. and Pal, S.K.** (2004). New operators of genetic algorithms for traveling salesman problem, *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 2, IEEE, pp.497–500.
- [99] **Mudaliar, D.N. and Modi, N.K.** (2013). Unraveling travelling salesman problem by genetic algorithm using m-crossover operator, *2013 International Conference on Signal Processing, Image Processing & Pattern Recognition*, IEEE, pp.127–130.

- [100] **Ripon, K.S.N., Siddique, N.H. and Torresen, J.** (2011). Improved precedence preservation crossover for multi-objective job shop scheduling problem, *Evolving Systems*, 2, 119–129.
- [101] **Burkowski, F.J.** (1999). Shuffle crossover and mutual information, *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 2, IEEE, pp.1574–1580.
- [102] **Lim, S.M., Sultan, A.B.M., Sulaiman, M.N., Mustapha, A. and Leong, K.Y.** (2017). Crossover and mutation operators of genetic algorithms, *International journal of machine learning and computing*, 7(1), 9–12.
- [103] **Deep, K. and Thakur, M.** (2007). A new crossover operator for real coded genetic algorithms, *Applied mathematics and computation*, 188(1), 895–911.
- [104] **Deep, K. and Thakur, M.** (2007). A new mutation operator for real coded genetic algorithms, *Applied mathematics and Computation*, 193(1), 211–230.
- [105] **Harik, G.R., Lobo, F.G. et al.** (1999). A parameter-less genetic algorithm., *GECCO*, volume 99, pp.258–267.
- [106] **Grefenstette, J.J.** (2014). *Proceedings of the First International Conference on Genetic Algorithms and their Applications*, Psychology Press.
- [107] **Díaz-Manríquez, A., Ríos-Alvarado, A.B., Barrón-Zambrano, J.H., Guerrero-Melendez, T.Y. and Elizondo-Leal, J.C.** (2018). An automatic document classifier system based on genetic algorithm and taxonomy, *IEEE Access*, 6, 21552–21559.
- [108] **Hassanat, A., Almohammadi, K., Alkafaween, E., Abunawas, E., Hammouri, A. and Prasath, V.S.** (2019). Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach, *Information*, 10(12), 390.
- [109] **Ronald, S.** (1997). Robust encodings in genetic algorithms: A survey of encoding issues, *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, IEEE, pp.43–48.
- [110] **Hinze, J.** (1975). *Turbulence. 2nd Edition.*, McGraw-Hill Publishing Co., New York.
- [111] **Launder, B.E. and Spalding, D.B.** (1972). Lectures in mathematical models of turbulence, (*No Title*).
- [112] **Shih, T.H., Liou, W.W., Shabbir, A., Yang, Z. and Zhu, J.** (1995). A new k- eddy viscosity model for high reynolds number turbulent flows, *Computers & fluids*, 24(3), 227–238.
- [113] **Orszag, S.A.** (1993). Renormalisation group modelling and turbulence simulations, *Near-wall turbulent flows*.

- [114] **Spalart, P. and Allmaras, S.** (1992). A one-equation turbulence model for aerodynamic flows, *30th aerospace sciences meeting and exhibit*, p.439.
- [115] **Wilcox, D.** (1998). Turbulence Modeling for CFD, *DCW industries, La Canada*.
- [116] **Menter, F.R.** (2009). Review of the shear-stress transport turbulence model experience from an industrial perspective, *International journal of computational fluid dynamics*, 23(4), 305–316.
- [117] **Fr, M.** (1994). Two-equation eddy-viscosity turbulence models for engineering applications, *AIAA J*, 32, 269–289.
- [118] **Walters, D.K. and Cokljat, D.** (2008). A three-equation eddy-viscosity model for Reynolds-averaged Navier–Stokes simulations of transitional flow.
- [119] **Gibson, M. and Launder, B.** (1978). Ground effects on pressure fluctuations in the atmospheric boundary layer, *Journal of Fluid Mechanics*, 86(3), 491–511.
- [120] **Launder, B.E.** (1989). Second-moment closure: present... and future?, *International Journal of Heat and fluid flow*, 10(4), 282–300.
- [121] **Launder, B.E., Reece, G.J. and Rodi, W.** (1975). Progress in the development of a Reynolds-stress turbulence closure, *Journal of fluid mechanics*, 68(3), 537–566.
- [122] **Launder, B.E. and Spalding, D.B.,** (1983). The numerical computation of turbulent flows, Numerical prediction of flow, heat transfer, turbulence and combustion, Elsevier, pp.96–116.
- [123] **Kim, S.E.** (1995). A near-wall treatment using wall function sensitized to pressure gradient, *ASME FED*, 217.
- [124] **Menter, F.R.** (1992). Influence of freestream values on k-omega turbulence model predictions, *AIAA journal*, 30(6), 1657–1659.
- [125] **Rhie, C.M. and Chow, W.L.** (1983). Numerical study of the turbulent flow past an airfoil with trailing edge separation, *AIAA journal*, 21(11), 1525–1532.
- [126] **Patankar, S.** (2018). *Numerical heat transfer and fluid flow*, CRC press.
- [127] **Van Doormaal, J.P. and Raithby, G.D.** (1984). Enhancements of the SIMPLE method for predicting incompressible fluid flows, *Numerical heat transfer*, 7(2), 147–163.
- [128] **Issa, R.I.** (1986). Solution of the implicitly discretised fluid flow equations by operator-splitting, *Journal of computational physics*, 62(1), 40–65.

CURRICULUM VITAE

Erkan BİÇER:

EDUCATION:

- **B.Sc.:** 2009, Sakarya University, Engineering Faculty, Metallurgical and Materials Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2019 – Present, Senior Structural & Cabin Interior Design Engineer at Turkish Airlines Technic.
- 2014 - 2019, Senior R&D Engineer at Panasonic Co.
- 2014, Senior Engineer at Öztreyler.
- 2012 - 2014, R&D Engineer at Tırsan Treyler.
- 2011 - 2012, R&D Engineer at Deka Elektronik.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Biçer E., Kumbasar T., (2024).** Optimization of Centrifugal Fan Design Using Genetic Algorithm and CST Method. *25.Otomatik Kontrol Ulusal Konferansı*, September 12-14, 2024 Konya, Turkey. (Presentation Instance)
- **Biçer E., Kumbasar T., (2025).** Optimization of Centrifugal Fan Design Using Genetic Algorithm and CST Method. *ASME Turbo Expo, Turbomachinery Technical Conference & Exposition*, June 16–20, 2025, Memphis, USA. (Presentation Instance)