MULTI-OBJECTIVE OPTIMIZATION OF AN INTERIOR PERMANENT MAGNET MOTOR

Subhasis Ray, B.Eng.

Department of Electrical and Computer Engineering

McGill University, Montreal, Quebec, Canada

August, 2008

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF

MASTER OF ENGINEERING

Your file Votre référence
ISBN: 978-0-494-66957-0
Our file Notre référence
ISBN: 978-0-494-66957-0

Abstract

In recent years, due to growing environmental awareness regarding global warming, green cars, such as hybrid electric vehicles, have gained a lot of importance. With the decreasing cost of rare earth magnets, brushless permanent magnet motors, such as the Interior Permanent Magnet Motor, have found usage as part of the traction drive system in these types of vehicles. As a design issue, building a motor with a performance curve that suits both city and highway driving has been treated in this thesis as a multi-objective problem; matching specific points of the torque-speed curve to the desired performance output. Conventionally, this has been treated as separate problems or as a combination of several individual problems, but doing so gives little information about the trade-offs involved. As a means of identifying the compromising solutions, we have developed a stochastic optimizer for tackling electromagnetic device optimization and have also demonstrated a new innovative way of studying how different design parameters affect performance.

Résumé

Dans les années récentes, les véhicules verts comme les véhicules électriques hybrides sont devenus de plus en plus important en raison de la croissance de la conscience environnementale du réchauffement de la planète. Avec le coût décroissant des aimants permanents de terre rare, les moteurs à aimant permanent sans brosse, tels que le moteur à aimant permanent intérieur, sont utilisés dans les systèmes d'entraînement de traction dans ces types de véhicules. Comme question de conception, la construction d'un moteur avec une courbe d'exécution qui convient à la conduite à la ville et sur l'autoroute a été traitée dans cette thèse comme problème d'optimisation multi-objectif, dans lequel les points spécifiques de la courbe couple-vitesse sont jumelés à l'exécution désiré. Conventionnellement, ceci a été traité en tant que problèmes séparés ou comme combinaison de plusieurs problèmes différents, mais faire ainsi fournit peu d'informations sur les différences impliquées. En tant que des moyens d'identifier les solutions compromettantes, nous avons développé un optimiseur stochastique pour l'optimisation de dispositif électromagnétique. Nous avons également démontré une nouvelle manière innovatrice d'étudier comment les différents paramètres de conception affectent l'exécution.

Acknowledgements

First of all I would like to thank Prof. David A. Lowther for his invaluable suggestions, critical appreciation, supervision, and encouragement. It was my privilege working with him and sharing his expertise and knowledge in the field of Computational Electromagnetics. It was truly an enjoyable and a challenging experience.

I would also like to thank my parents and my sister for all their support and inspiration. Thanks also to my friends and colleagues from CADLab for their suggestions and ideas.

Contents

Abstract ii

Résumé iii

Acknowledgements iv

Contents v

List of Figures vii

List of Tables x

List of Symbols xi

List of Acronyms xiii

1 Introduction 1

1.1 Aim 3

1.2 Structure 3

2 Interior Permanent Magnet Motor 4

2.1 IPM: Theory 6

2.2 Parameterized Torque Equation 11

3 Evolutionary Algorithms 15

3.1 Single Objective Evolutionary Algorithm 16

3.2 Multi-objective Evolutionary Algorithm (MOEA): Definitions 18

3.2.1 MOEA: Review of some of the common Algorithms 20

4 Multi-Objective Optimization 27

4.1 Proposed Algorithm 31

4.1.1 The Algorithm: Detailed 33

4.1.2 The Algorithm: Salient Points 41

4.1.3 The Algorithm: Discussions 43

4.2 Optimization Framework 46

4.3 Analytical Test Functions 47

4.4 Benchmark Problem 58

4.5 Advantages of mERR-MOEA 62

5 Optimizing the Motor 64

5.1 Calculating Torque from Finite Element Analysis 64

5.2 Modeling the IPM 66

5.2.1 Modeling the IPM: Key Aspects 70

5.3 Optimizing the IPM 75

6 Conclusion 89

6.1 Future Research 90

Appendix A – Basic Definitions 91

Appendix B – Parameterization of the IPM for automation 96

References 98

List of Figures

Figure 1: Interior Permanent Magnet Motor [2] 4
Figure 2: Torque-Speed Plot 5
Figure 3: d-q axis of IPM 7
Figure 4: Equivalent q-axis circuit [14] 7
Figure 5: Equivalent d-axis circuit [14] 7
Figure 6: Phasor diagram of IPM during flux weakening operation 9
Figure 7: IPM parameters [13] 12
Figure 8: Evolution cycle in Evolutionary Algorithms 15
Figure 9: Family tree 16
Figure 10: For example, individual represented as 1.5x(2+a) 17
Figure 11: Real-valued vector 18
Figure 12: The Pareto front representing the boundary of all the dominated solutions . 20
Figure 13: Decision making as a posteriori process in a maximization problem [16] 21
Figure 14: Ranking 22
Figure 15: A well distributed Pareto front in a maximization problem [16] 28
Figure 16: Simplified Flow Chart 34
Figure 17: Step 3 is depicted here 35
Figure 18: Non-dominated individuals represented in function space as filled circles, remaining are dominated 36
Figure 19: Members of the external and main population sets after Step 6 37
Figure 20: Crossover process in the gene level 38
Figure 21: Crossover and mutation processes represented in parameter and function space 40
Figure 22: Members of the main population for next generation 40

Figure 23: Optimization framework diagram 47
Figure 24: Deceptive Pareto front at iteration 1 and true front at iteration 22 48
Figure 25: First discovery of a point along the true Pareto front at iteration 7 49
Figure 26: More points of the true Pareto front discovered at iteration 11 49
Figure 27: Shifting towards the true Pareto front discovered at iteration 12 50
Figure 28: True Pareto front at iteration 22 50
Figure 29: Pareto front 52
Figure 30: Pareto front of Test function 1 from [23] 53
Figure 31: Pareto front 54
Figure 32: Pareto front of Test function 2 in [23] 54
Figure 33: Pareto front of test function 4 55
Figure 34: Pareto front of same test function given in [23] 56
Figure 35: Pareto front of test function 5 57
Figure 36: Plot showing diversity in parameter space 58
Figure 37: SMES device configuration 59
Figure 38: Pareto front of SMES 60
Figure 39: Parameter space (SMES) 61
Figure 40: Progress measurement 62
Figure 41: Torque-Current plot 67
Figure 42: Torque plot at 200A at different positions of the rotor 68
Figure 43: Detailed parametric diagram of the IPM 68
Figure 44: 4 layers of air-gap (outer two grey ones are virtual air and the inner two are air) in IPM 71
Figure 45: Coil connections of the IPM 71
Figure 46: Detailed 3-phase distributed winding structure of the IPM 72
Figure 47: Symmetry axes of the IPM 72

Figure 48: 1/8th part of the IPM for FEA 73
Figure 49: d-axis Field distribution of the IPM at 200A 74
Figure 50: Solution mesh of the IPM 74
Figure 51: Effect of different parameters on torque 75
Figure 52: IPM optimization: Pareto plot 77
Figure 53: Actual Torque plot of the global non-dominated solutions 78
Figure 54: Half of the pole-arc angle from non-dominated solutions against the respective torques independent of 'Wm' and 'Lm/2' 80
Figure 55: 3D plot of the non-dominated solutions in parameter space-view 1 80
Figure 56: 3D plot of the non-dominated solutions in parameter space-view 2 81
Figure 57: Width of the magnets from non-dominated solutions against the respective torques independent of 'a' and 'Lm/2' 82
Figure 58: Half of the length of the magnet from non-dominated solutions against the respective torques independent of 'a' and 'Wm' 83
Figure 59: The circled region showing a constant 'a' and a varying 'area' 85
Figure 60: The circled region showing a constant 'area' and a varying 'a' 85
Figure 61: Effect of the area of the magnet on the torque at 200A 86
Figure 62: Effect of 'a' on the torque at 200A 86
Figure 63: Crossover in GA 92
Figure 64: Relationship between decision space and function space [16] 94
Figure 65: Mutation in GA 94
Figure 66: Parameterization of the permanent magnet for automation 96

List of Tables

Table 1: Symbol reference table 13

Table 2: Final optimal solution of SMES device optimization 61

Table 3: SMES solution when using MOSCA 61

Table 4: SMES solution from [28] 62

Table 5: IPM dimensions 69

Table 6: Other information about the IPM to be optimized 70

Table 7: Optimal design specifications 79

Table 8: (Summary-1) Analysis of parameters v/s torque 87

Table 9: (Summary-2) Analysis of parameters v/s torque 88

List of Symbols

ω	angular speed
f	supply frequency
P	number of poles
E_f	back emf
V	terminal voltage
I	armature current
R	armature resistance
k_e	back emf constant
L_q	q-axis inductance
L_d	d-axis inductance
i_d	d-axis current of IPM
i_q	q-axis current
V_q	q-axis terminal voltage
V_d	d-axis terminal voltage
γ	angle between I and the d-axis
θ	angle between V and the d-axis
α	angle by which V leads I
P_{out}	output power
ψ_m	peak flux linkage
P_{in}	input power
T	output torque
B_g	air gap flux density
a	half of the magnet pole-arc angle or half spread angle
C_ϕ	flux concentration factor
B_r	remanence of the magnet
μ_{rec}	recoil permeability
K_c	Carter coefficient
g	width of the air gap
w_m	width of the magnet
d	width of the flux barrier

h_1	inner flux barrier height
h_2	outer flux barrier height
l_m	length of the magnet
A_m	cross-sectional area per pole of the magnet
A_{mm}	cross-sectional area of the thin iron bridge above the flux barriers
B_s	limit of the leakage flux density in the bridge due to saturation
t	thickness of the iron bridge
A_g	cross-sectional area per pole of the air gap
r_r	outer radius of the rotor
D	inner diameter of the stator
l	lamination stack length of the motor
K_{w1}	winding factor
N_{ph}	number of turns per phase
B	magnetic flux density
i	current through the conductor
L	length of the conductor
W_1	stored energy of the body at point 1
W_2	stored energy of the body at point 2
x_{12}	displacement between points 1 & 2
$B_x B_y, B_z$	flux densities along x, y & z axes

List of Acronyms

DM - Decision Maker

FEA - Finite Element Analysis

HEV - Hybrid Electric Vehicle

IMOEA - Incrementing Multi-objective Evolutionary Algorithm

IPM - Interior Permanent Magnet Motor

MOA - Multi-objective Algorithm

MOCSA - Multi-objective Clone Selection Algorithm

MOEA - Multi-objective Evolutionary Algorithm

MOGA - Multi-objective Genetic Algorithm

MOO - Multi-objective Optimization

MOP - Multi-objective Problem

NPGA - Niched Pareto Genetic Algorithm

NSGA - Non-Dominated Sorting Genetic Algorithm

PAES - Pareto Archived Evolutionary Strategy

SMES - Superconducting Magnetic Energy Storage

SOP - Single Objective Problem

SPEA - Strength Pareto Evolutionary Algorithm

TEAM - Testing Electromagnetic Analysis Methods

VB - Visual Basic

VEGA - Vector Evaluated Genetic Algorithm

1 Introduction

In recent years, growing environmental awareness regarding global warming, high emission standards, high fuel price, and other cost factors, have lead to the increase in the demand for green cars such as Hybrid Electric Vehicles (HEVs). The first mass produced HEV was built by Toyota in 1997 [1] and since then various models of HEVs from different automobile manufacturers have flooded the market. Even governmental organizations, such as the US Department of Energy have been part of this important research area for quite some time now [2] with their Freedom Car Project. In general, most of these HEVs, use brushless permanent magnet motors, such as the Interior Permanent Magnet motor (IPM), due to various advantages such as high torque-current ratio, small size, high efficiency, etc for a variable speed traction drive system.

As a design issue, building a motor that matches certain performance criteria to suit both city and highway driving, i.e. high torque at low speed and over a substantial speed range, requires investigation of the underlying trade-offs, such as, a compromise between the size of the motor and the performance, the size of the permanent magnets and performance, the location of the permanent magnets and performance, etc. This physical requirement when represented mathematically means, finding the best suitable design, and can be achieved by searching for the maxima or the minima of the objective functions subject to certain specifications – a process also known as optimization.

Optimization problems can be sub-divided into Single objective problems (SOPs) [3] and Multi-objective problems (MOPs) [4]. Most real world applications have multiple objectives, frequently conflicting in nature, for example, maximizing profit while

maximizing quality or minimizing weight while maximizing strength or minimizing fuel consumption while maximizing output power. The solution to MOPs having conflicting objectives is a set of trade-off points or compromising solutions also known as non-inferior solutions [5] or Pareto Optimal solutions [6] (a set of Pareto Optimal points is known as the Pareto Optimal set). However, it is to be noted that, if the objectives don't conflict then MOPs lead to a single optimum instead of a set of solutions.

One of the ways to solve a MOP is by using weighting functions [4] thereby converting the MOP into a single-objective optimization (SOP) problem. For example, suppose if f_1 & f_2 are the two objectives of a MOP, then they can be represented as a SOP,

$$f = w_1 f_1 + w_2 f_2 \tag{1.1}$$

where w_1 and w_2 are the desired weights and

$$w_1 + w_2 = 1 \tag{1.2}$$

However, there are times when the human decision maker [7] would not know the weights at the start, i.e. *a priori,* due to the complexity of the device and would like to weigh different optimal solutions at the end, i.e. *a posteriori,* according to his requirements and eventually pick the best one.

In the context of the IPM, since the torque performance [8] of these high efficiency motors [9] & [10] has complex relationships with numerous parameters as has been researched in [11], [12] & [10], while assuming infinite permeability of iron, zero winding resistance, etc. [12], determining the exact weights can be tough or may not be possible at all. Therefore, achieving objectives such as maximizing efficiency [10], minimizing volume [13], maximizing torque, minimizing ripple torque [11], etc. by treating the process as a SOP may not be desirable to some human decision makers.

1.1 Aim

1. To optimize the torque performance of a IPM model over a substantial speed range 520-5400 rpm [2] by solving it as a MOP and
2. To study the effect of the design variables such as the permanent magnet dimensions on the torque performance from the information obtained by optimizing.

1.2 Structure

The thesis has been structured in such a way that it highlights all the key elements of a design process such as knowledge acquisition, designing, implementation and testing. Chapter 2 and Chapter 3 highlight the knowledge acquisition phase, wherein Chapter 2 deals with the theoretical aspects of IPM so that the reader can get an initial idea of the parameters to be optimized, and Chapter 3 reviews some of the commonly used Evolutionary algorithms. Chapter 4 describes the proposed algorithm, the mixed Elitist Restricted Resolution Multi-objective Evolutionary Algorithm (mERR-MOEA), and the results obtained from tests conducted on various benchmark analytical functions and a real device, the Testing Electromagnetic Analysis Methods (TEAM) Superconducting Magnetic Energy Storage (SMES) Problem 22. Chapter 5 deals with the modeling of the IPM and the final implementation. The Conclusions along with discussions about future work make up Chapter 6.

2 Interior Permanent Magnet Motor

In this chapter, we will present the theoretical aspects of the IPM, for example, we shall derive the torque equation from the phasor diagram, and provide equations that indentify the critical parameters affecting torque performance.

Permanent magnet motors are increasingly being used in applications, such as, variable speed traction drives, compressors, pumps, etc. Among the family of permanent magnet motors, the IPM has more advantages in terms of high efficiency, wide speed operation, and flux weakening ability. Further, it has a high torque to inertia ratio due to small size, increased mechanical robustness at high speeds as the magnets are embedded inside the rotor and a reduced air gap, which are essential requirements of HEVs since fuel efficiency comes at the cost of performance.

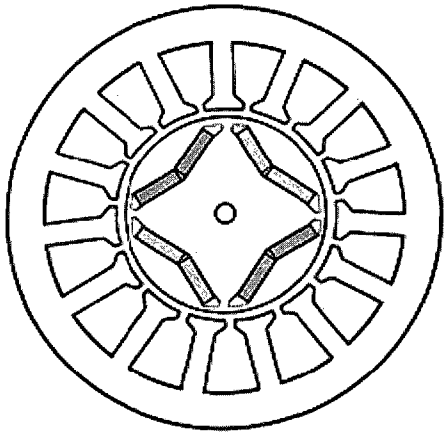

Figure 1: Interior Permanent Magnet Motor [2]

A typical IPM has the stator surrounding the rotor. The stator contains the windings and the permanent magnets are embedded in the rotor as shown in Figure 1. Unlike a conventional synchronous motor, which requires two sources of current to bring it into synchronism, excitation of an IPM is only on the stator side thereby reducing copper

loss. The IPM runs at synchronous speeds as the stator current supply is usually controlled by an external Pulse Width Modulated (PWM) inverter circuit. Based on the speed of the IPM,

$$\omega = 120\, f/P \tag{2.1}$$

where ω is the speed in rpm, f is the supply frequency in Hz and P the number of poles. The external circuit varies the supply frequency and also controls the direct axis current by phase advancing the stator supply current in order to achieve the maximum torque to current ratio at all speeds and an effective flux weakening operation (explained in detail in the next section) (Eqn.(2.1) shows how frequency and speed are related). The torque-speed plot of a typical IPM as shown in Figure 2 has a constant torque or increasing power region from zero to the base speed and a constant power or decreasing torque, also known as flux-weakening region, from the base to the maximum operational speed.

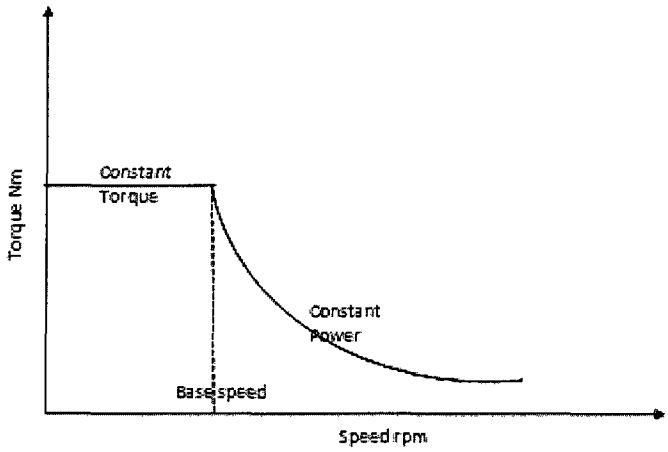

Figure 2: Torque-Speed Plot

2.1 IPM: Theory

The principle of operation of an IPM is similar to that of a conventional synchronous machine. Due to the polyphase supply to the stator a rotating magnetic field effect is produced on the rotor. The rotor magnetic poles are attracted to the field induced on the stator due to the windings and its connections, and start rotating along with the stator magnetic field. As the rotor starts rotating, while synchronizing to the rotating magnetic field, it induces an emf in the stator known as the "back emf" or the "magnetic friction" (generally sinusoidal in nature depending on the stator winding layout) that opposes the terminal or input voltage,

$$E_f = V - IR \tag{2.2}$$

where V is the terminal voltage, I is the armature current, R is the armature resistance and E_f is the back emf. The back emf is directly proportional to the angular velocity of the rotor,

$$E_f = k_e . \omega \tag{2.3}$$

where k_e is the back emf constant, which is dependent on the air-gap flux density, number of turns of the coil, and size of the motor, and ω is the angular velocity of the motor in rpm.

In order to understand the functioning of the IPM, it can be represented by two equivalent circuits as shown in Figure 4 & Figure 5; one along each of the two axes as shown in, the d axis (passing from the center of the motor cross-section through the center of the magnet) and the q axis (axis passing from the center of the motor cross-section through the space between two magnets); Figure 3. The angular difference between the axes is always 90° elec. The basic idea to represent the motor as a two-axis

model is to take care of the variable reluctance path caused by the presence of the embedded permanent magnets.

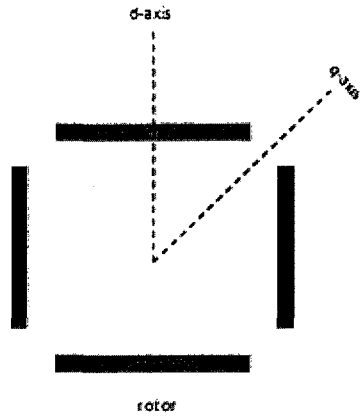

Figure 3: d-q axis of IPM

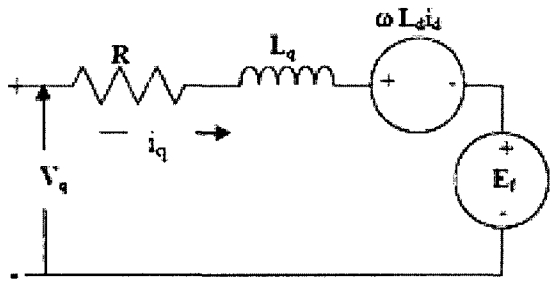

Figure 4: Equivalent q-axis circuit [14]

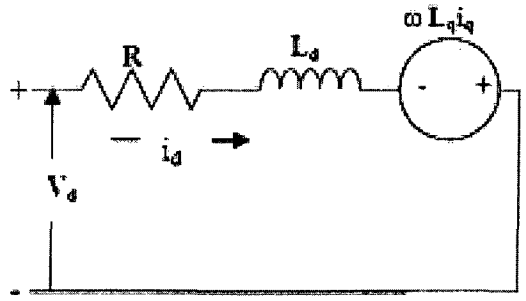

Figure 5: Equivalent d-axis circuit [14]

where R is the stator armature resistance, L_q is the q-axis inductance, L_d is the d-axis

inductance, ω is the speed of rotation of the rotor, i_d is the d-axis current, i_q is the q-axis current, E_f is the back emf, V_q is the q-axis terminal voltage and V_d is the d-axis terminal voltage.

Based on the assumptions like, 1) saturation is neglected and 2) back emf is sinusoidal, the equations derived from the equivalent d & q axis circuits are,

$$V_q = i_q R + L_q \frac{di_q}{dt} + \omega L_d i_d + E_f \tag{2.4}$$

$$V_d = i_d R + L_d \frac{di_d}{dt} - \omega L_q i_q \tag{2.5}$$

During steady state operation $\frac{di_q}{dt} = \frac{di_d}{dt} = 0$ i.e.

$$V_q = i_q R + \omega L_d i_d + E_f \tag{2.6}$$

$$V_d = i_d R - \omega L_q i_q \tag{2.7}$$

Solving for i_d & i_q while neglecting R, we get,

$$i_q = -\frac{V_d}{\omega L_q} \tag{2.8}$$

$$i_d = \frac{V_q - E_f}{\omega L_d} \tag{2.9}$$

The equivalent phasor diagram is

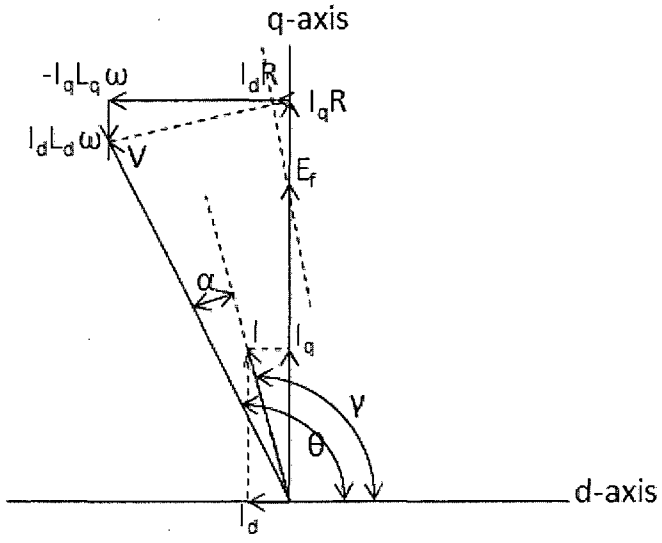

Figure 6: Phasor diagram of IPM during flux weakening operation

where,

$$I = I_d + I_q \tag{2.10}$$

$$V = E_f + (I_d + I_q)R - I_qL_q\omega + I_dL_d\omega \tag{2.11}$$

$$\gamma = \theta - \alpha \tag{2.12}$$

where γ is the angle that I makes with respect to d-axis, θ is the angle that V makes with respect to the d-axis and α is the angle by which V leads I. Referring to Figure 6, all voltage sources have been considered as negative and potential drops as positive.

A typical IPM is attached to an external inverter circuit, which supplies a Pulse Width Modulated wave as input to the motor, to control γ for flux weakening purposes and the supply frequency for it to run at synchronous speeds. Referring to Figure 6, during high speeds (speed above the base speed), if γ is increased, while holding I_q constant

by allowing I to increase, I_d increases resulting in an increase of $I_d L_d \omega$. As a result, E_f decreases since the IPM is a voltage controlled motor, thereby enabling the IPM to achieve high speeds. This phenomenon is known as flux weakening operation.

In general for a polyphase motor,

$$mT = \frac{mP_{out}}{\omega} \tag{2.13}$$

$$P_{out} = P_{in} - I^2R - core\ loss \tag{2.14}$$

$$P_{in} = VI\cos\alpha \tag{2.15}$$

where m is the number of phases, T is the torque, P_{out} electrical output power, P_{in} is the electrical input power and I^2R is the copper loss.

Simplifying (2.15) vectorially from the phasor diagram,

$$P_{in} = V_q I_q + V_d I_d \tag{2.16}$$

$$P_{in} = I_q(E_f + I_q R + I_d L_d \omega) + I_d(I_d R - I_q L_q \omega) \tag{2.17}$$

$$P_{in} = I_q E_f + (I_q^2 + I_d^2)R - I_q I_d(L_q - L_d)\omega \tag{2.18}$$

Assuming that there are no core losses and replacing P_{in} by (2.18), (2.14) can be written as,

$$P_{out} = I_q E_f - I_q I_d(L_q - L_d)\omega \tag{2.19}$$

Substituting in (2.13), we get

$$mT = m\frac{I_q}{\omega}E_f - mI_q I_d(L_q - L_d) \tag{2.20}$$

Replacing i_d & i_q by Eqn.(2.8) & Eqn.(2.9) respectively and L_d & L_q by the corresponding reactances, i.e.

$$X_q = \omega L_q \tag{2.21}$$

$$X_d = \omega L_d \tag{2.22}$$

we obtain,

$$mT = -m\frac{V_d}{X_q \omega}E_f + \frac{mV_d(V_q - E_f)}{X_q X_d \omega}(X_q - X_d) \tag{2.23}$$

Replacing $V_d = Vcos\theta$, $V_q = Vsin\theta$, $\delta = \theta - \frac{\pi}{2}$ in (2.23) and then simplifying, we get

$$mT = m\frac{VE_f sin\delta}{X_d} + m\frac{V^2}{2}\left(\frac{1}{X_q} - \frac{1}{X_d}\right)sin2\delta \tag{2.24}$$

The first term in (2.24) is the magnetic torque and the second is the reluctance torque which is produced due to the presence of the embedded magnets causing a variation in the reluctance path having a saliency ratio $\frac{X_q}{X_d} = \frac{L_q}{L_d} > 1$. The contribution of reluctance torque to the main torque along with the magnetic torque enables IPM to achieve a high performance. From the above equation, it can be deduced that the reluctance torque is independent of the back emf and thus, if there is no back emf, the only torque component that would act on the rotor is the reluctance part.

2.2 Parameterized Torque Equation

Having derived the analytical torque equation, let us analyze which geometrical parameters affect torque.

From (2.3), replacing E_f by $k\omega\psi_m$ in (2.24) where ψ_m is the peak flux linkage due to the permanent magnets and k is a constant, we get

$$T = \frac{mVk\omega\psi_m sin\delta}{X_d} + \frac{mV^2}{2}\left(\frac{1}{X_q} - \frac{1}{X_d}\right) sin2\delta \qquad (2.25)$$

From [13], referring to Figure 7,

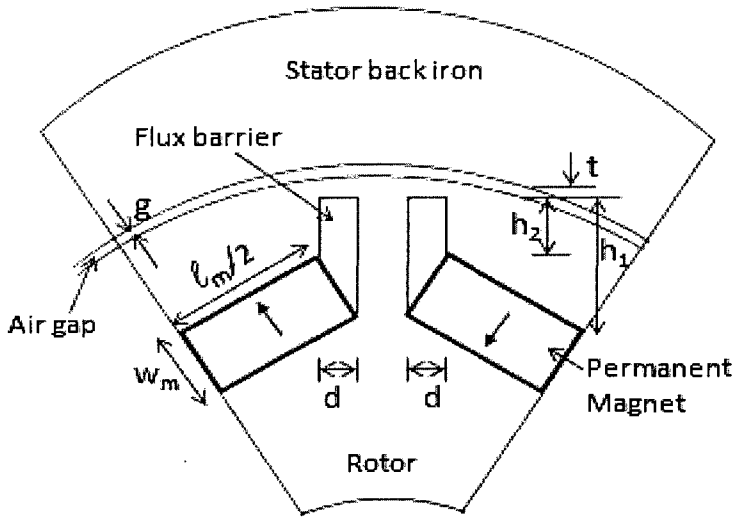

Figure 7: IPM parameters [13]

and defining the symbols in Table 1,

C_ϕ	flux concentration factor as given by (2.32) [15]
B_r	remanence of the magnet
μ_{rec}	recoil permeability [15]
K_c	Carter coefficient [15]
g	width of the air gap
w_m	width of the magnet
d	width of the flux barrier
h_1	inner flux barrier height
h_2	outer flux barrier height
l_m	length of the magnet

A_m	cross-sectional area per pole of the magnet
A_{mm}	cross-sectional area of the thin iron bridge above the flux barriers
B_s	limit of the leakage flux density in the bridge due to saturation
t	thickness of the iron bridge
A_g	cross-sectional area per pole of the air gap
r_r	outer radius of the rotor
D	inner diameter of the stator
l	lamination stack length of the motor
α	half of the magnet pole-arc angle or half spread angle

Table 1: Symbol reference table

the average air gap flux density B_g is expressed as,

$$B_g = \frac{C_\phi}{1+\beta(1+2\eta+4\lambda)} B_r \tag{2.26}$$

where,

$$\beta = \frac{\mu_{rec} K_c g}{w_m} C_\phi \tag{2.27}$$

$$\eta = \frac{w_m(h_1+h_2)}{4d\mu_{rec} l_m} \tag{2.28}$$

$$\lambda = \frac{1+{}^{1}/_{\beta}+2\eta}{2\left({}^{A_m}/_{A_{mm}}\right)\left({}^{B_r}/_{B_s}\right)-4} \tag{2.29}$$

$$A_m = l_m w_m \tag{2.30}$$

$$A_{mm} = tl \tag{2.31}$$

$$C_\phi = {}^{A_m}/_{A_g} \tag{2.32}$$

$$A_g = {}^{g(2r_r+g)\alpha}/_{2} \tag{2.33}$$

After expanding and simplifying Eqn.(2.26) by using (2.27) to (2.33), we get

$$B_g = \frac{2l_m w_m / g(2r+g)a}{A} B_r \tag{2.34}$$

$$A = 1 + \mu_{rec} K_c \frac{2l_m}{(2r+g)a} \left(1 + \frac{(h_1+h_1)w_m}{2d\mu_{rec} l_m} + Z\right) \tag{2.35}$$

$$Z = \frac{2\left(1 + \frac{1}{\mu_{rec} K_c} \frac{(2r+g)a}{2l_m} + \frac{w_m(h_1+h_2)}{2d\mu_{rec} l_m}\right)}{\frac{w_m l_m}{tl} \frac{B_r}{B_s} - 2} \tag{2.36}$$

Representing ψ_m in terms of average flux linkage,

$$\psi_m = \frac{4Dl}{\pi} \left(\frac{K_{w1} N_{ph}}{p}\right) B_g \sin(a) \tag{2.37}$$

Substituting (2.37) in (2.25) and expanding, it can be deduced that, in terms of the geometric parameters, $l_m, w_m, g, a, h_1, h_2, d, t, D, l$ and r affect the torque provided that the material of the permanent magnet is not changed, and the number of poles and the number of turns of the stator coils are held constant. Out of the eleven parameters mentioned, h_1, h_2 & d are dependent on l_m, w_m & a and therefore can be omitted from the above list. Hence, the remaining eight independent geometrical parameters influence the torque out of which l_m, w_m & a are related to the permanent magnet.

Later in this thesis, in Chapter 5, we shall try to investigate these parameters with the help of a simulated model using *Finite Element Analysis*, while in the next two chapters we shall develop a multi-objective algorithm required to optimize the performance of the IPM, which is one of the goals, as stated in Chapter 1.

3 Evolutionary Algorithms

(terms in *italics* are explained in Appendix A)

Evolutionary algorithms (EAs) are *stochastic or non-deterministic* search processes that mimic the natural biological evolution process. They deal with a population of potential solutions and are therefore, by default, parallel in nature. Similar to the biological process, evolutionary algorithms use Charles Darwin's concept of survival of the fittest through a sequence of generations, to gradually produce better *individuals* suited for the environment. Evolutionary algorithms use the concepts of reproduction such as *crossover or recombination* and *mutation* to produce new individuals. The basic evolutionary cycle is shown in Figure 8. Figure 9 focuses on the family tree of evolutionary algorithms.

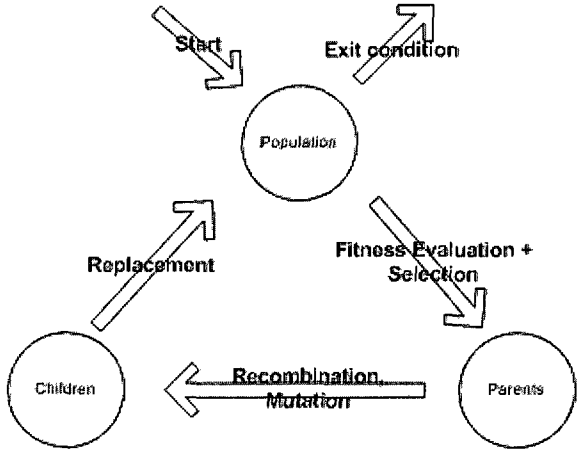

Figure 8: Evolution cycle in Evolutionary Algorithms

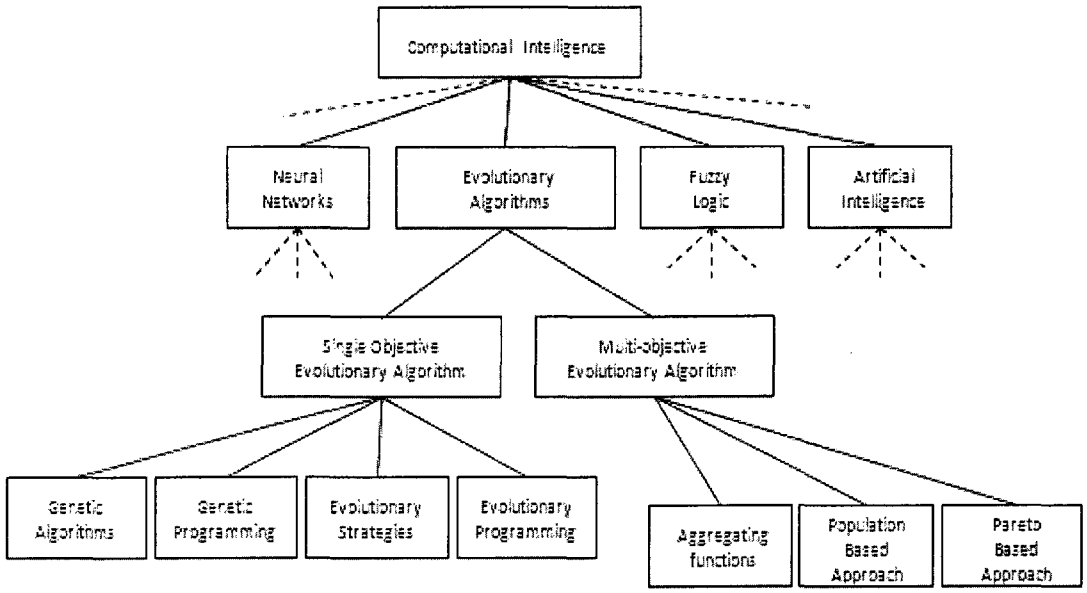

Figure 9: Family tree

The following sections provide a brief review of existing evolutionary algorithms.

3.1 Single Objective Evolutionary Algorithm

Genetic Algorithms (GAs): They are inspired by the natural evolution process. Traditionally, solutions are represented in binary format as an array of bits. Other variants treat the individuals as a list of real numbers but reproduction takes place at the bit level i.e. real numbers represented as 8-bit, 32-bit, etc. binary numbers depending on the problem (not IEEE standard for floating point number). Individuals reproduce based on *selection* of the fittest one. Selection procedures such as *rank selection*, *tournament selection*, etc. are commonly used. The weakest ranked individuals are replaced by the new offspring in order to maintain a constant *population*, the size of which is problem dependent.

Genetic Programming (GP): Like all EAs, it also follows the natural evolution process. The

difference between GAs and GPs is that GPs evolve computer programs and not *chromosomes*. These computer programs are represented by tree type data-structures having nodes and terminals which contain information about the individual as shown in Figure 10. *Fitness* assignment is problem dependent, similar to that in GAs. All operations of GPs are based on the tree type structure. Crossover happens by swapping one of the nodes with a node of another individual. Mutation occurs by either replacing the information of a node with another similar node or by replacing a sub-tree with another similar sub-tree of the same individual.

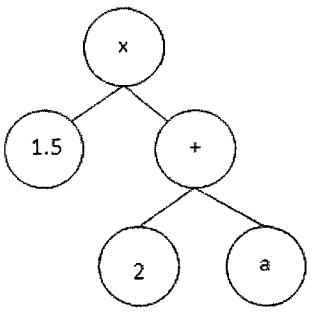

Figure 10: For example, individual represented as 1.5x(2+a)

Evolutionary Strategy (ES): It is similar to GAs, the differences being that the individuals are encoded as real numbers and it mostly relies on mutation for reproducing. Some variants also use cross-over for reproducing. An individual is comprised of objective and strategy parameters. The objective parameters are vectors of real numbers representing the variables to be optimized and the strategy parameters are also vectors of real numbers but control the objective parameters during mutation.

$$chrom = (op, sp) \tag{3.1}$$

where op is an objective parameter, sp is a strategy parameter and

$$op = (o_1, o_2, \ldots, o_n) \tag{3.2}$$

$$sp = (s_1, s_2, \ldots, s_n) \tag{3.3}$$

where o_i and s_i are real numbers. Mutation happens both for the objective

parameters and the strategy parameters. Common ES schemes are $(\frac{\mu}{\rho},\lambda)$ or $(\frac{\mu}{\rho}+\lambda)$, where μ is the total number of parents, ρ is the number parents to mutate, λ is the number of offspring, 'comma' denotes that selection takes place only among the offspring and 'plus' denotes selection involving both offspring and parents.

Evolutionary Programming (EP): It uses real valued vector represented individuals where each position in the vector represents a feature of the individual (Figure 11). For reproduction, only the mutation operator is used.

1.2	3.4	5.0	6.7

Figure 11: Real-valued vector

Unlike ESs that calculate fitness based on the evaluated objective functions, EPs give a chance to every parent to reproduce. However, after reproduction they assign the fitness according to a pre-determined criterion, like evaluating the performance in a tournament selection method between a certain number of individuals, and then selecting the best individuals required for the next generation. Variants of EPs can have a dynamic population.

3.2 Multi-Objective Evolutionary Algorithm (MOEA): Definitions

Multi-Objective Evolutionary algorithms have evolved as the preferred choice for solving MOPs since MOEAs deal with multiple possible solutions simultaneously, which enables it to find the trade-off surface [6]. The basic concepts used in most MOEAs have been derived from the Single Objective Evolutionary Algorithms, for example, crossover, mutation, fitness, *strength*, etc. In general, MOEAs can be divided into Aggregating functions, Population based and Pareto based approaches. Before getting into the

details about the above mentioned types, let us first go through the vocabulary used in MOPs.

Dominated and Non-Dominated solutions: [6] In a multi-objective problem where there is more than one objective, say f_1, f_2,..., f_n, there can be three possible solutions, either solution $x^{(1)}$ dominates solution $x^{(2)}$ or vice versa or neither dominates.

$x^{(1)}$ is said to dominate $x^{(2)}$ if both the following conditions are satisfied,

- Solution $x^{(1)}$ is no worse (the operator $\prec$ to denote worse and $\succ$ to denote better) than $x^{(2)}$ in all objectives, i.e. $f_j(x^{(1)}) \not\prec f_j(x^{(2)})$ for all $j = 1,2, \ldots n$.
- The solution $x^{(1)}$ is better than $x^{(2)}$ in at least one objective, or $f_j(x^{(1)}) \succ f_j(x^{(2)})$ for at least one $j = 1,2, \ldots, n$

it can then be said that $x^{(2)}$ is dominated and $x^{(1)}$ is non-dominated. However, if $x^{(1)}$ is worse than $x^{(2)}$ in one of the objectives but better than $x^{(2)}$ in others or if $x^{(1)}$ is not worse than $x^{(2)}$ in any of the objectives but is also not better than $x^{(2)}$ in any of the objectives then for both the cases neither of the solutions dominates the other i.e. both are non-dominated with respect to each other.

Pareto Optimal Set: The set of non-dominated solutions is called the Pareto optimal set.

Global Pareto Optimal Set: If there exists no solution in the *search space* that dominates any member of the Pareto optimal set, then the Pareto optimal set is known as the Global Pareto Optimal set.

Pareto Front: Members of the Global Pareto Optimal set when plotted in the *objective space* form the Pareto front having the shape of a curve. It is referred to as a "front" since it represents the boundary of the dominated solutions. It is also known as the

trade-off front.

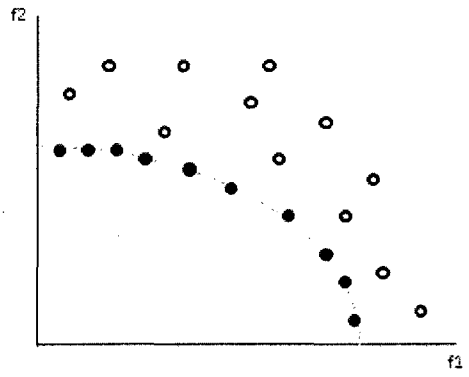

Figure 12: The Pareto front representing the boundary of all the dominated solutions

The above diagram, Figure 12, displays a two-dimensional Pareto front having the shape of a concave curve. Theoretically speaking, a Pareto front can have an infinite number of solutions if the solution is in real space, but is of finite size when represented in binary. In other words, by making it finite, i.e. of known resolution, it can be said that the search space is not continuous which means that the total search space is limited. In a way, this acts as an advantage (explained in Chapter 4) but with it comes the problem of tackling duplicate chromosomes. Evaluating a chromosome which is almost similar or identical to one from a previous generation can lead to a waste of computational time especially in the case of field evaluations of physical entities. Nevertheless, resolution is just a matter of accuracy and only depends on the requirements of the decision maker.

3.2.1 MOEA: Review of some of the common Algorithms

Having gone through the vocabulary, we shall now review the types of MOEAs in detail and get to know the commonly used algorithms.

Aggregating functions: A MOP is converted into a SOP using weighting functions where

the aggregate of the weights is equal to unity (1.1) & (1.2). This is the simplest way of solving MOPs. The drawback being that the designer has to know the weights *a priori*. Figure 13 shows decision making as *a posteriori* method.

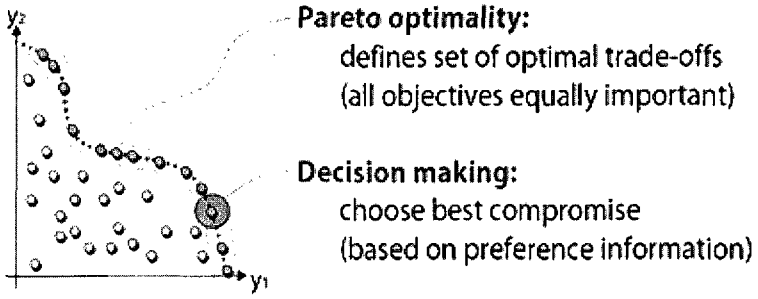

Figure 13: Decision making as *a posteriori* process in a maximization problem [16]

Population based approach:

The Vector Evaluated Genetic Algorithm (VEGA) [17] is a classical example of this type of approach. A VEGA, for mating purposes, divides the original population into sub-populations of equal size at each generation on the basis of proportional selection [4] according to each objective function in turn. For example, if there are n objectives and the population size is N then each sub-population is of size N/n. [4]. These sub-populations are shuffled among each other to obtain new sub-populations of size equal to that of the original sub-populations. Afterwards, individuals crossover and mutate within their respective sub-populations formed from shuffling, in effect, achieving mating between different sub-populations. VEGA's weakness is that it works against the concept of Pareto dominance since it discards all those compromise solutions which may not be the best solution for at least one of the objectives. It also has a bias towards certain solutions and fails to find good compromise solutions in the case of concave surfaces [7]. The solution to this is obtained by either of the two methods known as the "non-dominated selection heuristic" and the "mate selection heuristic". In the non-dominated selection heuristic, the dominated individuals are

penalized by a fixed penalty and the total penalty for the dominated chromosomes is divided among the non-dominated individuals. However, the non-dominated selection heuristic fails when there are few non-dominated individuals, resulting in a large fitness value for those non-dominated chromosomes, leading to a high selection pressure [18]. In the mate selection heuristic, an individual is selected as a mate to a randomly chosen individual based on the maximum distance in the solution space from its mate. However, it fails due to the inability to prevent participation of poorer individuals [18].

Pareto based approaches:

The Multi-Objective Genetic Algorithm (MOGA) [5] is based on the concept that all individuals are ranked on the basis of how many chromosomes in the current population dominate them. All the non-dominated individuals get the same rank while all the dominated ones get penalized on the basis of the area of the corresponding region of the Pareto front. Figure 14 shows the ranking scheme.

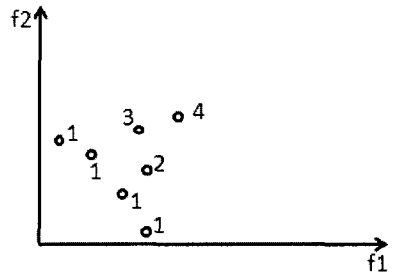

Figure 14: Ranking

The number of points sampling a certain area of the Pareto front is calculated based on the choice of the sharing factor. Sharing is done on the objective functions instead of the variables on the assumption that it leads to a uniform distribution of the global Pareto front. Fitness assignment is done in the following way [5],

1. Sort the population according to rank
2. Assign fitnesses to individuals by interpolating from the best to the worst according to some function; usually linear, but not necessarily.

3. Average the fitnesses of the individuals with the same rank so that all of them are sampled at the same rate.

MOGA also involves the decision maker during the selection operation, in order to guide the Pareto front towards the compromise solution, in case the decision maker has certain knowledge about the kind of region in the Pareto front he wants to zoom-in and hence save time. The main disadvantage of MOGA is its dependency on the choice of sharing factor (explained in Chapter 4) and the fitness assignment scheme that can lead to a large selection pressure and therefore to a premature convergence.

The Non-Dominated Sorting Genetic Algorithm (NSGA) [18] ranks the whole population on the basis of non-domination unlike MOGA, which ranks according to domination. It classifies the entire population into several fronts. The non-dominated chromosomes form the first front. NSGA assigns a large dummy fitness value proportional to the population size to all the non-dominated individuals. Diversity of the population is achieved by fitness sharing between two individuals of the same front in the *variable or parameter space*. Fitness sharing is achieved by performing selection on the basis of degraded fitness which is calculated by dividing the original fitness of an individual by a quantity proportional to the number of neighbors. After sharing between the non-dominated individuals, NSGA processes the rest of the population in order to identify the next front and continues until the entire population is classified into fronts. While processing the next fronts NSGA assigns an individual a dummy fitness value less than the minimum shared dummy fitness of the previous front so that the better individuals get more copies during reproduction thereby achieving *elitism*. NSGA is slightly better than MOGA in the context that it can find more non-dominated solutions [19] and is faster than MOGA [7]. The main drawback of NSGA is its inability to get out of local optimum points [6] and its sensitivity to the sharing factor is similar to that of the

MOGA.

The Niched Pareto Genetic Algorithm (NPGA) [20] uses a tournament selection scheme known as a "Pareto dominance tournament" in which two individuals are selected at random. A comparison set of individuals is also chosen randomly. Each of the two individuals is compared with each in the comparison set and the one which remains non-dominated is chosen for reproduction unlike in a binary tournament scheme where the best individual is that which dominates the other and therefore effectively ends up with a single solution after a certain number of generations. However, in the case of a tie during a Pareto dominance tournament i.e. both individuals remaining non-dominated, fitness sharing in the objective space is used to determine the winner. Unlike fitness sharing as done in NSGA on the basis of degraded fitness within the same niche or area, in NPGA the best non-dominated individual in a particular generation is that which has the least number of individuals in its niche i.e. the least number of neighbors. It must also be noted that the radius of a niche is pre-determined by the user depending on the density of the solutions desired. NPGA is fast and produces good Pareto fronts. Its main drawback is the optimal choice of the comparison set that determines the pressure of selection and therefore its performance.

The Strength Pareto Evolutionary Algorithm (SPEA) [21] is different from the other algorithms discussed so far. It maintains an external set of non-dominated solutions evolved through the previous generations. Selection is based on the binary tournament scheme. The fitness of an individual is calculated only from the external non-dominated set and is a two stage process. Each non-dominated individual in the external set is assigned a value called "strength", proportional to the number of current population members it dominates. The fitness is the same as the strength for the non-dominated individuals. In case of individuals of the current population, the fitness is the total

strength of all the non-dominated individuals of the external set that dominate this individual plus one. Adding one to the total fitness is a way to ensure that the non-dominated ones have a better fitness (an individual is said to be fitter if its fitness value is less than the one it is compared with). In other words, the more the number of neighbors of a particular individual the less fit it is due to the large strength value of the associated non-dominated individual. Thus, SPEA uses Pareto based niching and therefore doesn't require any knowledge of the niche radius. The size of the external non-dominated set influences the behavior of SPEA. However, maintaining a large non-dominated set means a slower and a biased search. Therefore, SPEA uses a finite sized external set i.e. whenever the set reaches a certain size limit, the algorithm prunes it down by *clustering*. The idea of incorporating a fixed sized external set helps in maintaining selection pressure [22]. The main drawback of SPEA is its time complexity [22] due to the external set, however, the quality of the Pareto front is in general better than NSGA, MOGA, etc. [23].

The Micro-Genetic Algorithm for Multi-Objective Optimization (micro-GA) [24] uses two memories: the "external memory" for storing non-dominated solutions similar to that in the SPEA and "population memory" which has a replaceable and a non-replaceable part. The non-replaceable part remains unchanged throughout the run and provides diversity. The population size of both the parts can be regulated by the designer. The initial population is formed by including individuals from both the parts. During each generation, two non-dominated individuals are selected and compared with the external set. If either remains non-dominated then it is included in the external set and the dominated member is eliminated from the external set. These two individuals are also compared with two members belonging to the replaceable memory. If either remains non-dominated then it replaces the dominated member of the memory so that

gradually the replaceable memory is filled up with a greater number of non-dominated individuals. In order to tackle the problem of the increasing size of the external set, micro-GA uses the concept of an adaptive grid. Once the external set reaches a certain size limit, then any newly generated non-dominated solution is included in the external set only if that individual, which is assigned a certain coordinate according to the grid, belongs to a less populated location, while the less important chromosomes are discarded. The algorithm utilizes a very small population size and has a low computational cost.

In this chapter we have reviewed the basic concepts of MOEA & EA *per se*, the advantages and drawbacks of some of the algorithms we discussed and had an opportunity to familiarize ourselves to the common linguistics dealt in MOP. In the next chapter we shall discuss a new algorithm and why is it suitable for the kind of application we intend to optimize.

4 Multi-Objective Optimization

In the previous chapter, when we reviewed some of the commonly used algorithms we came across concepts such as fitness sharing, archiving, elitism, selection for reproduction, etc. After going through these ideas, the obvious question that arises is "what is their use ?". In the following sections we shall examine their effect on the problem of interest i.e. the Interior Permanent Magnet Motor (IPM).

The aim of a MOEA algorithm is to find a well-distributed and accurate Pareto front as shown in Figure 15, so that, the decision maker or the designer has the information to understand the trade-offs. Algorithms which are purely deterministic in nature converge very quickly and accurately if started from a carefully chosen design. However, in cases where there can be possibilities of multiple local minima or non-convexity of the front or where an initial idea of the global minima isn't possible, stochastic search mechanisms might be efficient, robust and easy to use.

Any stochastic optimizer embeds two distinct phases in its operation: exploration and exploitation. In exploration, the goal is to search the objective function space in order to find possible areas for the optimal solution; in exploitation, local knowledge is used to determine the exact location of the optimal solution as shown in Figure 15. All algorithms provide some form of balance between these two phases and how it is applied depends on the shape of the objective space. In a MOP, this has to be applied for each objective. This brings us to the critical issues that MOEAs face, which are,

1. How to guide the search towards the Global Pareto front, and
2. How to achieve a well distributed trade-off front

Some other aspects but not as critical as the ones mentioned above are,

3. What is the correct population size to start with, and
4. When to terminate the search process

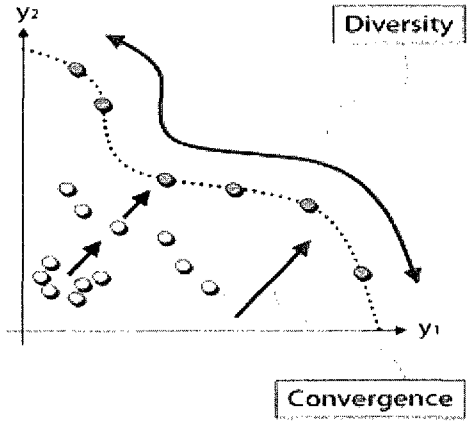

Figure 15: A well distributed Pareto front in a maximization problem [16]

1. How to guide the search towards the Global Pareto front: Any search algorithm might get stuck in a local Pareto optimal front due to deceptive fronts [6] also referred to as multimodality (if there are more than two local optima). There can be cases where the search space is flat and the true front is isolated. In order to tackle these criticalities, ranking and selection strategies are generally used. The basic idea is to enhance the exploration capability. For example, as mentioned in Chapter 3, in NSGA [18] the whole population is classified into several fronts on the basis of non-domination. The non-dominated solutions are grouped into a single category so that they have equal chances of reproducing and have more offspring than the rest of the population. When visualized with respect to the parameter space, the non-dominated individuals might be scattered all over the search space. During reproduction since the fitter ones have more copies, effective exploration of the surrounding space is carried out in the context of the parameter space. However, it doesn't stop the weaker individuals from being used in the reproduction process, assuming that the weaker ones can also produce good individuals but with a lower probability. In other words, by enforcing elitism i.e. giving more chance

to the fitter individuals since they have higher probability of finding good individuals in the parameter space, the population evolves to find the true Pareto front.

2. How to achieve a well distributed trade-off front: To determine a useful Pareto front, it is necessary to find solutions which are as diverse as possible. If all the solutions are clustered together on the front, the purpose of Multi-Objective Optimization (MOO) is not served. However, in trying to maintain diversity, Multi Objective Optimization (MOO) faces difficulties such as convexity or non-convexity. If the fitness is proportional to the number of solutions dominated as implemented in MOGA, then the sampling of convex shaped functions is more biased than in the case of non-convex functions [6]. There can also be situations where the true Pareto front is not continuous [6], instead, it is discretely spaced and non-uniform, i.e. some regions having higher density of solutions [6]. Such difficulties are generally tackled by fitness sharing and elitism. The goal of fitness sharing is to distribute the population over a number of peaks or optimum points or niches in the search space proportional to the height of the peak or the fitness. For example, in NSGA, fitness sharing between two individuals having identical rank is done by performing a selection procedure using degraded fitness values which are obtained by dividing the original fitness value of the chromosome by a quantity proportional to the number of chromosomes around it. In other words by doing fitness sharing, multiple optimal points are able to co-exist and therefore when they take part in reproduction, it leads to a more diverse search, [18]. The goal of elitism in terms of diversity is to give more chance to the fitter individuals since they have a higher probability of finding more non-dominated points along the front. For example, in SPEA, by maintaining an external archive of non-dominated solutions, which interact with the main population during the binary tournament selection procedure, elitism is achieved [21]. In a way, the basic approach to attaining diversity in the Pareto front is by exercising effective exploitation

of the front.

3. What is the correct population size to start with: In general, most MOEAs tend to guess the initial population. However, if the population is too small there can be a possibility of premature convergence i.e. the algorithm might terminate before finding the true Pareto front or, if it is too large, then it might require excessive computational time. Instead of guessing the optimal population, algorithms such as the Incrementing Multi-objective Evolutionary Algorithm (IMOEA) [25] employ the concept of a dynamic population with an adaptive mechanism helping it to adapt the population size based on the discovered trade-off surface and the desired population density along the trade-off surface. As the number of non-dominated solutions increases with generations, the algorithm also adapts itself to an increased population size.

4. When to terminate the search process: In multi-objective problems, the greater the number of points along the trade-off front, the better it is for the designer. Halting a Multi-Objective Algorithm (MOA) means a choice between having enough solutions and the desire to have more solutions. Most algorithms either halt the process after a fixed number of generations or monitor the population at certain intervals and interpret it visually, if the decision maker is human, to determine whether or not to halt the search. Other variants calculate the relative distance from the actual trade-off front, which require an idea of the final Pareto front *a priori*, to determine when to halt. Algorithms such as the IMOEA, propose a convergence assessment based on population domination. IMOEA calculates the progress ratio based on the relative fractional change in the number of non-dominated chromosomes dominated in generation $n-1$ by that in generation n.

4.1 Proposed Algorithm

Over the last two decades there have been an overwhelming number of optimization algorithms that do the same job but in different ways. Most of them differ in selection and fitness strategies, for example some have fitness sharing, some implement elitism, some have an external population, some use ranking, etc. As mentioned earlier, all of these are variations of exploration of the search space and exploitation of the explored space. However, they may not perform the same way for all problems. In fact, according to the "No free lunch theorem" [26] there can be algorithms which outperform some of the conventional ones for a particular type of problem but might underperform in others. The intention here is to propose an algorithm which will perform better than existing algorithms for our type of problem. The question that comes to mind is what makes our problem so special.

A major issue in optimizing devices that operate subject to the laws of physics is that accurate evaluations of the performance require computationally expensive solutions. For an electromagnetic device, the cost of the full solution of the field may be orders of magnitude greater than the overhead of the optimization process. Thus, any algorithm in this area must attempt to minimize the number of accurate function evaluations. For example, it takes nearly 2.5 min to run a single static 2D analysis of an IPM machine in MagNet (version 6.22.1) - Finite Element Software from Infolytica Corp., in a machine configured with a single core 1.6GHz Intel Pentium M processor and having 1.5GB of RAM. Hence, by preserving information so that in a future generation an extremely similar or even the same individual is not evaluated again can lead to an efficient algorithm. Algorithms like the SPEA or the NSGA fail to do that although SPEA preserves a part of the population.

A second issue is that, in a manufacturing environment there is a statistical spread of values for the input parameters i.e. the input data has a level of uncertainty or a tolerance associated with it. Algorithms such as the Real Coded Clonal Selection Algorithm (RCSA) [27], by working with parameters which are expressed as real numbers assume that the limiting factor in candidate device representation is the finite word length of the computer. This is, of course, correct in solving the physics problem – a low number of bits to represent the floating point numbers leads to numerical inaccuracies due to round-off. However, finding an optimal point having an unprecedented accuracy which is undesired by the decision maker is a waste of computational resource. For example, if a device has a length dimension of 3.81 cm but a known tolerance of +/- 0.04 cm, a representation as 3.8 cm would be appropriate – any more accuracy is unnecessary. In other words, by having a finite resolution based objective space, which can be done by encoding the floating point number into a finite length binary string can lead to a more efficient optimization algorithm. However, limiting the parameters to a predetermined resolution can lead to the problem of duplicates that may arise during reproduction.

A third issue is the simplicity aspect that most algorithms tend to overlook. Tweaking fitness sharing in order to ensure an optimal selection strategy, determining the optimal crossover and mutation rate for a particular application, etc. can be considered as added burden on the user who intends to use the algorithm as a tool and is not interested in the second order performance details.

The basic ideas in our algorithm are a combination of various concepts discussed so far. It can be described as an elitist, Pareto archived, progress measured, dynamic population and restricted resolution based multi-objective evolutionary algorithm or to

give it a name we can call it a mixed Elitist Restricted Resolution Multi-objective Evolutionary Algorithm (mERR-MOEA).

4.1.1 The Algorithm: Details

Representations used in the following algorithm:

P_m denotes the main population

P_{nd} denotes the external non-dominated set (this set is empty initially)

P_d denotes the external dominated set (this set is empty initially)

N_m - Number of chromosomes in P_m

N_{nd} - Number of chromosomes in P_{nd}

N_d - Number of chromosomes in P_d

x, y, z - Genes according to GA language or known as variables mathematically

A typical multi-objective problem without any constraints, when represented mathematically (suppose having two objectives and three parameters), is

$$Minimize \quad f_1(x,y,z) \ \& \ f_2(x,y,z)$$

$$given \quad a \leq x \leq b, c \leq y \leq d, e \leq z \leq f$$

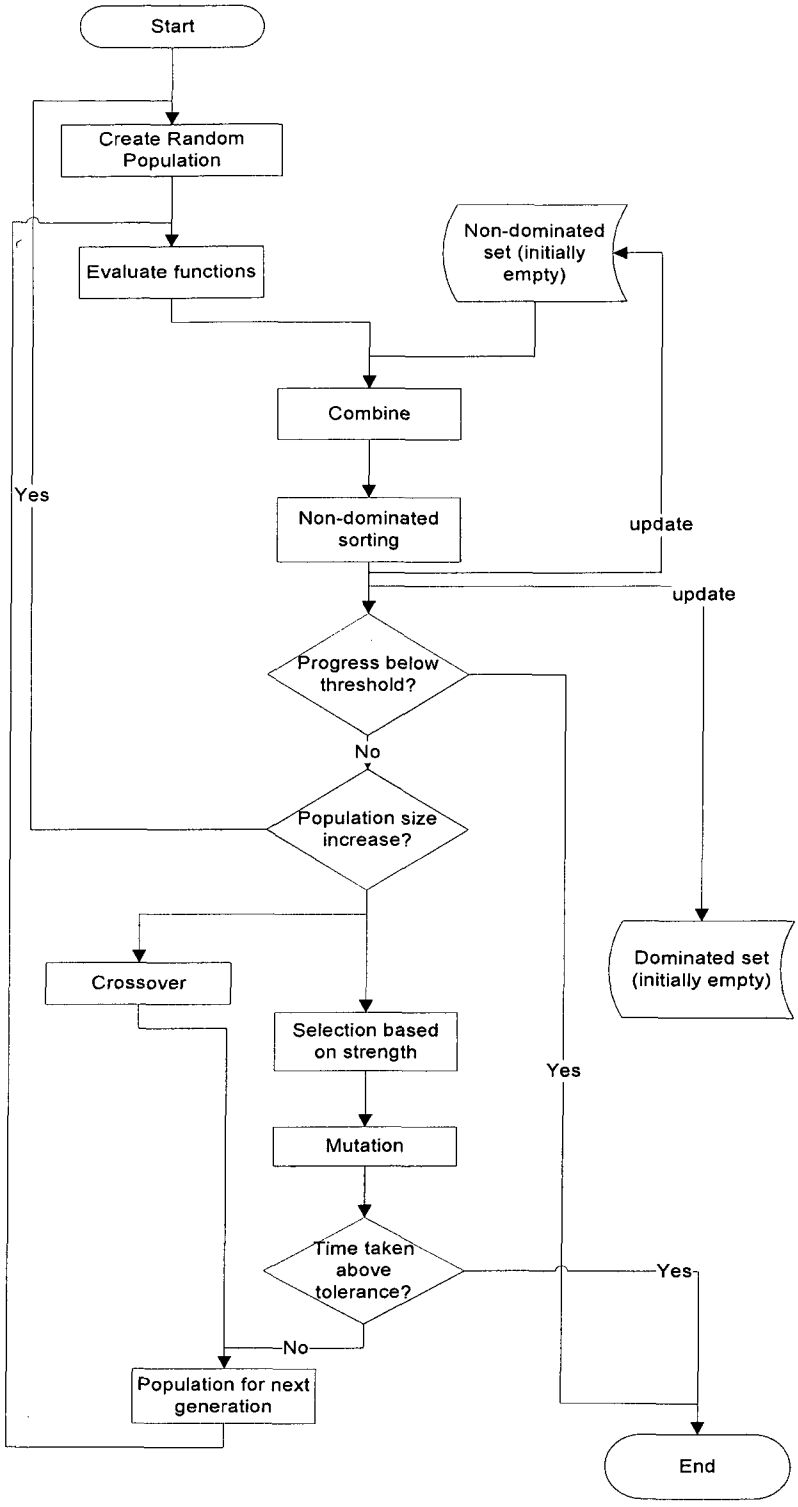

Figure 16: Simplified Flow Chart

Referring to Figure 16, following are the steps used to solve this problem.

1. Create a population of N_m chromosomes by randomly creating $x, y \& z$ *genes* within the predetermined ranges of each variable such that none of the chromosomes have any duplicate copies in P_{nd} and P_d which are empty during the first generation. For example, Chrom 1 might have three genes, namely $[x_1, y_1, z_1]$.

2. Evaluate the objective functions for each of the chromosomes in P_m. For example if there are two objectives, then their values are $f_1(x_1, y_1, z_1)$, $f_2(x_1, y_1, z_1)$.

3. Add the external set of non-dominated chromosomes to P_m resulting in a size of N_m+N_{nd}. Suppose P_m has chromosomes $C1, C2, C3, C4, C5$ and P_{nd} has $C6, C7$ from the last generation then, after this step P_m has $C1, C2, C3, C4, C5, C6, C7$. Figure 17 depicts this step.

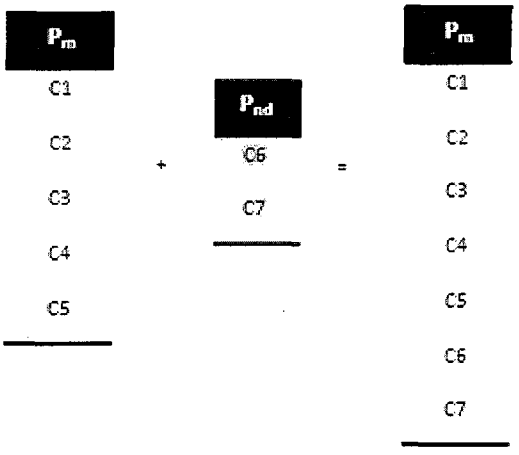

Figure 17: Step 3 is depicted here

4. Perform non-dominated sorting on P_m based on the values of the objective functions obtained in Step 2, which means that the population in P_m can be divided into two

parts after non-dominated sorting, i.e. the dominated and the non-dominated sections. Let the number of non-dominated chromosomes be N'_{nd} and dominated chromosomes be N'_d such that $N_m + N_{nd} = N'_{nd} + N'_d$. Suppose after this step, $C1, C2, C3$ are non-dominated and the remaining are dominated as shown in Figure 18.

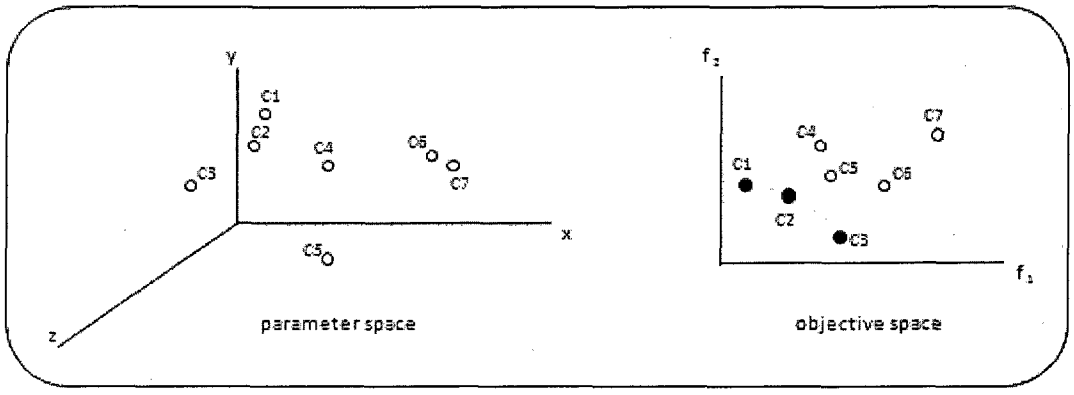

Figure 18: Non-dominated individuals represented in function space as filled circles, remaining are dominated

5. Calculate the strength 's' of each chromosome in P_m. The strength of a chromosome is calculated based on $s = \frac{n_d + \frac{1}{z} n_{nd}}{N'_{nd} + N'_d}$, $0 \leq s < 1$, where n_d is the number of chromosomes in P_m dominated by that chromosome and n_{nd} is the number of chromosomes in P_m it neither dominates nor is dominated by.

6. Store a copy of each of the N'_{nd} non-dominated chromosomes obtained from Step 5 in P_{nd} by replacing the current set. The new population size of P_{nd} is N'_{nd}. For example, since $C1$, $C2$, $C3$ are non-dominated, P_{nd} will contain $C1$, $C2$, $C3$.

7. Store a copy of each of the N'_d dominated solutions in P_d. The new population size of P_d is $N_d + N'_d$. Suppose P_d had $C8$, $C9$ from last generation, after this step it contains $C4$, $C5$, $C6$, $C7$, $C8$, $C9$ as shown in Figure 19.

P_{nd}	P_d	P_m
C1	C4	C1
C2	C5	C2
C3	C6	C3
	C7	C4
	C8	C5
	C9	C6
		C7

Figure 19: Members of the external and main population sets after Step 6

8. Randomly create as many possible pairs of genes of a particular type from the non-dominated chromosomes of P_m for single-point binary *crossover*.

9. Convert the two randomly chosen x genes from the non-dominated chromosomes of P_m into a k-bit binary representation depending on a pre-determined precision. Suppose, the two chosen genes are x_1 from $C1$ and x_3 from $C3$. The process is shown in Figure 20.

10. Convert the new genes obtained from cross-over back to real numbers. Say, the new genes are x_{10} and x_{11}.

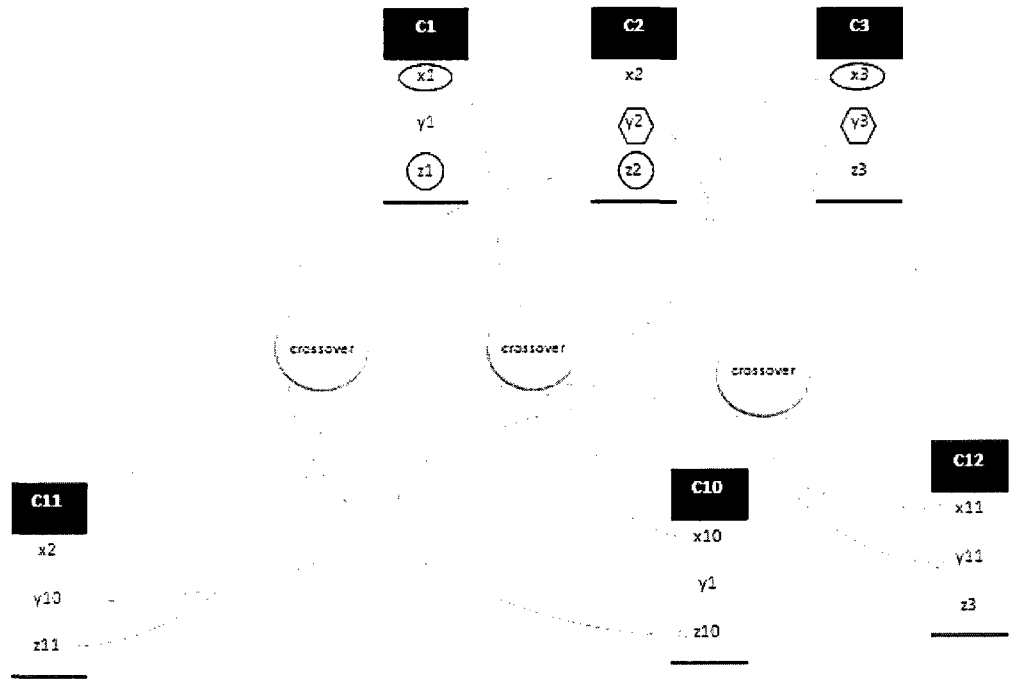

Figure 20: Crossover process in the gene level

11. Repeat Steps 8 & 9 for all the other pairs of x-genes.

12. Repeat Steps 8 to 10 for the y gene. Suppose the two chosen genes are y_2 from $C2$ and y_3 from $C3$. After cross-over they form, say, y_{10} and y_{11}.

13. Repeat Steps 8 to 10 for the z gene. The two chosen genes are z_1 from $C1$ and z_2 from $C2$. After cross-over they form, say, z_{10} and z_{11}.

14. Combine the new x, y & z genes to form new unique chromosomes. The population of the new chromosomes after cross-over is same as N'_{nd}. After combining we get, $C10 = [x_{10}, y_1, z_{10}]$; $C11 = [x_2, y_{10}, z_{11}]$; $C12 = [x_{11}, y_{11}, z_3]$.

15. If any copy of an existing individual in P_{nd} or P_d is found in P_m after cross-over, it

is deleted. During the mutation process the vacant spaces created due to deletion are filled up.

16. Based on a randomly generated number $0 \leq s' < 1$, chromosomes whose strength is less than s', starting from the best strength to the worst are selected for mutation, until the mating pool of size $N_m - N'_{nd}$ is filled. If the mating pool doesn't get filled up, which is possible if the selection starts too far down in the ranked list, then a new s' is generated and the process continues until the mating pool is filled. As far as the example is concerned, $N_m - N'_{nd} = 2$ provided there are no duplicates. Let us assume that after this step $C3$ & $C4$ get selected.

17. Convert the x, y & z genes of the chromosomes selected in the mating pool into the k-bit binary representation.

18. Mutate the genes in order to get unique chromosomes. Suppose, after mutation x_3 becomes x_{13}, y_3 becomes y_{13} and, z_3 remains unchanged. Similarly, in case of $C4$.

19. Convert the new chromosomes obtained from mutation into real numbers. The population of the new chromosomes after mutation is same as $N_m - N'_{nd}$. We then get $C13 = [x_{13}, y_{13}, z_3]$; $C14 = [x_{14}, y_{14}, z_4]$. Figure 21 depicts this process.

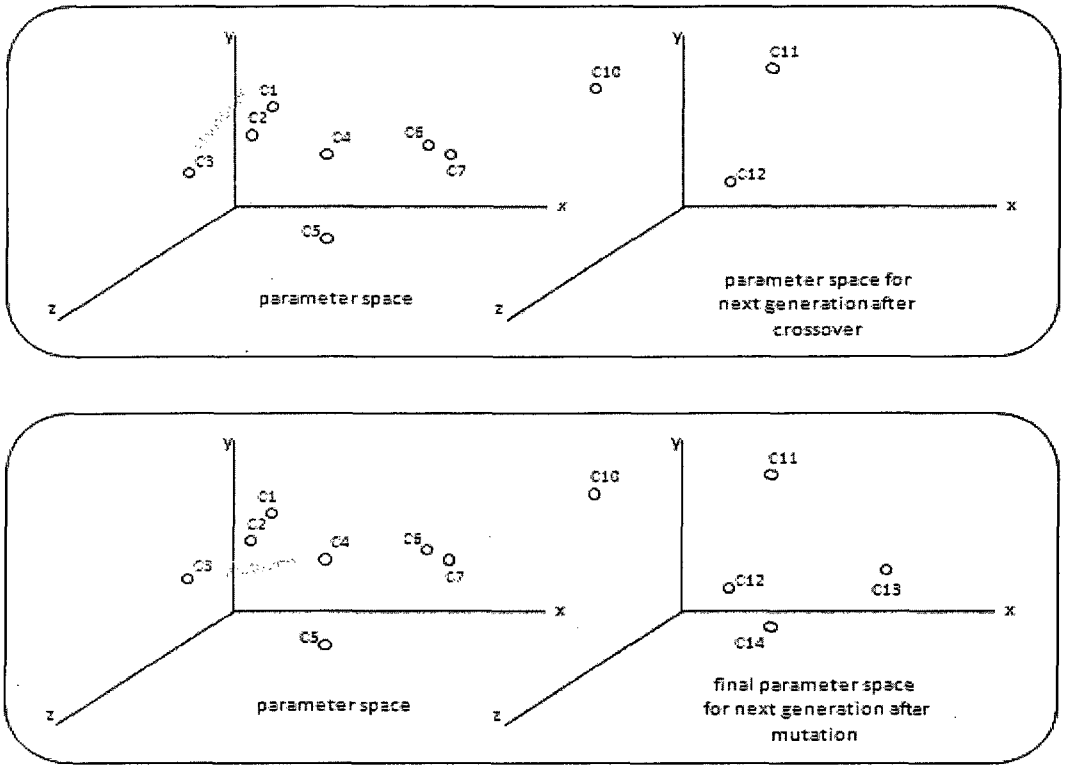

Figure 21: Crossover and mutation processes represented in parameter and function space

20. Combine the population from Step 14 with that in Step 19 to form the new P_m for the next generation. Repeat from Step 2 until the exit condition is triggered. On combining we get $C10$, $C11$, $C12$, $C13$, $C14$ as the new members of P_m.

P_m
C10
C11
C12
C13
C14

Figure 22: Members of the main population for next generation

21. If the relative change in the size of P_{nd} over six consecutive generations compared to the current size of P_{nd} remains below a cutoff for a couple of successive generations, then the program terminates.

22. If the current size of P_{nd} is greater than or equal to a certain percentage of P_m then the capacity or size of P_m is incremented by an amount equal to the initial population of P_m, i.e. by N_m, and the program continues from Step 1.

23. If the time taken in Step 1 or in Step 18 for producing new unique chromosomes is more than a certain predetermined amount of time, and if this occurs more than once throughout the life time of the program, then the program terminates.

4.1.2 The Algorithm: Salient Points

1. It maintains two external population sets (external means not part of the main population but part of the memory of the algorithm), one for the non-dominated chromosomes and another for the dominated chromosomes, and a main population. The genes are represented in terms of real numbers but reproduction happens at the bit level in the binary system i.e. a floating point number having a precision of up to 6 decimal places can be represented as a 20 bit binary number.

2. The external set of non-dominated chromosomes, also known as the Local Pareto front, along with the remaining chromosomes of the main population of the current generation take part in non-dominated sorting. The external set of non-dominated chromosomes is updated with the new set of non-dominated chromosomes in every

generation, in order to gradually lead to the Global Pareto Optimal set.

3. The other external set of dominated chromosomes serves the purpose of keeping track of the number of chromosomes already evaluated. There is no restriction on the size of the external sets.

4. The size of the external non-dominated set can decrease in a particular generation.

5. Only the non-dominated chromosomes take part in cross-over. Cross-over is done at the gene level, shown in Figure 20 i.e. instead of choosing a pair of non-dominated chromosomes randomly, a pair of genes, belonging to the same type, say the x gene, is chosen randomly. Any gene can take part in crossover only once in each generation. It is possible to have chromosomes born out of cross-over which have an existing copy in the external sets.

6. For the purpose of mutation, all the chromosomes of the main population, which includes the non-dominated chromosomes of the current generation, are sorted based on their strengths. The strength formula used here is different from the SPEA.

7. All the chromosomes formed from mutation are unique (none of the x, y & z genes of a chromosome match the respective genes of any other chromosome), with respect to both the other newly created individuals and to the external sets of non-dominated and dominated chromosomes.

8. The population is dynamic in nature. However, it never decreases even if there is a momentary decrease in the size of the external non-dominated set.

9. The algorithm implements multi-level stopping criteria. The progress measurement

followed here is different to that in IMOEA.

10. It implements elitism in two ways and therefore is called mixed i.e. preserves the non-dominated individuals to make them available during non-dominated sorting and implements rank based selection for mutation.

11. It can work for both maximization and minimization problems by just changing the non-dominated sorting.

12. It can work for any number of objectives and any number of variables.

4.1.3 The Algorithm: Discussions

The purpose of maintaining the external non-dominated set is to include them during each generation, so that elitism is maintained. The other external set is used to preserve information instead of discarding the dominated individuals, so that none of those points in the variable space, which have been already evaluated, get re-evaluated in future generations thereby saving computational time. The increase in the non-dominated set serves a major requirement of placing more emphasis on exploration and less on exploitation during the initial stages and is inherent in the formulation of this algorithm, and therefore pruning the external set by *cluster analysis* as done in the SPEA may not be desirable here. Moreover, the cost of ranking of the individuals for mutation is negligible compared to that of field evaluations.

The purpose of cross-over is to exploit the space in the nearby vicinity of the parent

chromosomes in the context of the parameter space. Since the non-dominated chromosomes are the ones which make up the Global Pareto front, by using the cross-over operator only on them, the nearby optimal points in the function space, can be searched for, leading to a faster convergence due to elitism. The notion of not classifying the non-dominated chromosomes, similar to that done in the NSGA, is related to the need to achieve diversity. The cross-over strategy of selecting any pair of the same type of genes from the non-dominated ones and not just selecting any pair of chromosomes, leads to a more uniform and diverse search for new non-dominated chromosomes along the front. Also, cross-over between different gene types for example $x \& y$, is restricted since they have different ranges. During cross-over, duplicates are not avoided as there can be a possibility of the algorithm getting trapped in an infinite loop as some of the non-dominated chromosomes, belonging to the Global Pareto set, may turn out to be members of the mating pool in every generation.

The purpose of mutation is to provide an ability to move away from local optimum points and explore possibilities that lie beyond the neighborhood of the parents in the parameter space so that diversity can be achieved in terms of the objective space. In the proposed algorithm, for the purpose of mutation, the non-dominated chromosomes are classified, similar to the SPEA, so that elitism can be maintained. The selection scheme for mutation allows the dominated individuals to also take part based on the strength calculated considering both domination and non-domination, so that the search process doesn't bias towards certain regions of the front. Similar to the SPEA, the strategy ensures that the non-dominated individuals will always have greater strength than the dominated ones although a different strength formula is used here. It is based on a scheme like the rank based selection, but is also dependent on a chance factor brought in by a randomly generated number, as mentioned in Step 16 of Section 4.1.1, which is

independent of the problem. In other words, the selection scheme is partly random unlike the rank selection method. However, the randomness is partly guided due to the fact that chances of a non-dominated individual getting selected increases as the percentage of non-dominated ones in the main population increases with generations, which is desirable here. Moreover, the starting point of the selection procedure might be random but the act of selecting an individual for mutation is based on elitism. The reason for making the selection mechanism partly random in nature is due to the fact, that the algorithm might get trapped in an infinite loop if some of the non-dominated chromosomes, belonging to the Global Pareto set, turn out to be members of the mutation pool in every generation.

We know, that as the number of non-dominated chromosomes increases, the chances of having a greater number of non-dominated individuals in the mating pool for mutation also increases, leading to a faster convergence. However, since the dominated individuals form the major share of the mating pool initially, exploration of new fronts is given priority at the beginning. With the increase of the non-dominated chromosomes, the importance of cross-over gradually increases which means better exploitation of the explored region, which is desired, but having mutation fill up the vacancies created due to deletion of duplicate individuals, leads to a balancing act between exploration and exploitation, thereby not letting the algorithm be fooled by deceptive fronts but gradually increasing the speed of convergence towards the true front. However, since the population is dynamic, the gradual increase in the importance of exploitation is restricted to the current population size. Once the population size increases, the importance of exploration increases again, but momentarily, and the cycle continues.

The strategy of a dynamic population adds weight to the exploration capabilities

because through this approach a biased search can be taken care of and new chromosomes from uncharted territory in the parameter space can be probed. In our algorithm a small initial population is desired. The idea of intelligent adaptation of the population as mentioned in [25] has its drawback, such as filtering out of certain chromosomes. In other words throwing away information is an unwanted strategy here.

The idea of implementing multi-level stopping criteria is to make sure that the search process doesn't terminate prematurely but stops when there are enough options along the Pareto front for the DM. Tracking the change in the overall value of the objective functions in the non-dominated set is not suitable in the case of multiple objectives having different tolerance levels or due to the nature of the physical device. The proposed scheme of tracking the percentage change in the size of the non-dominated set, with respect to the current size of that set, over a couple of generations, is a way of measuring progress towards the Pareto front. Also, enforcing a time limitation is a way to decide whether to continue searching more points or to terminate the search.

4.2 Optimization Framework

Having discussed the algorithm in detail in the previous few sections, we shall now present the automation framework designed for the optimizer.

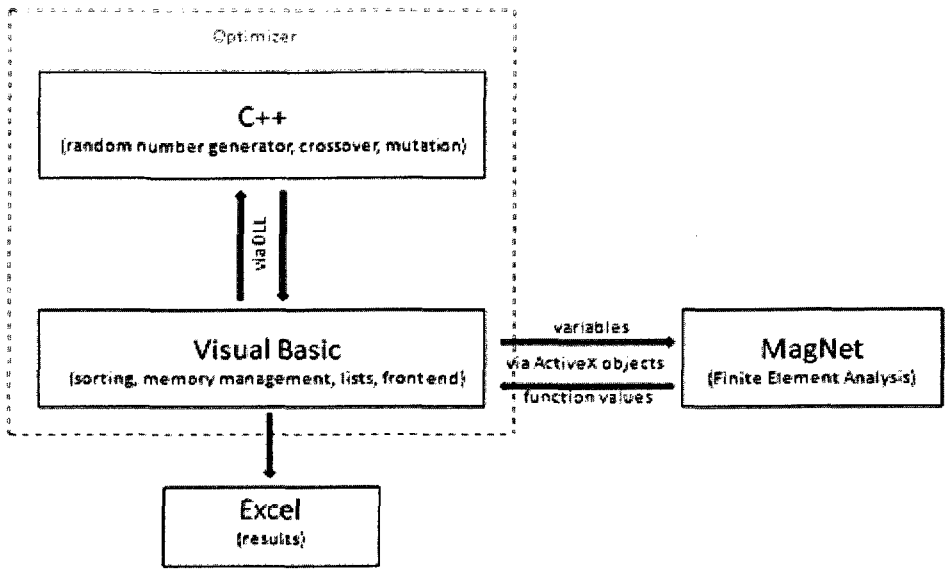

Figure 23: Optimization framework diagram

Visual Basic (VB) is being used here for memory management related operations like maintaining the external non-dominated and dominated sets, non-dominated sorting, etc. and C++ for data manipulation operations such as selection and reproduction. The advantage of VB is that it can be used as a connection mechanism between most low level languages and other high level application software tools. It provides a seamless connectivity with MagNet, used for FEA calculations of the electromagnetic devices. The purpose of including Excel as part of the automation framework is to display the dynamically changing Pareto front as the program runs so that the human decision maker can have a better understanding of the performance of the algorithm.

4.3 Analytical Test Functions

At the beginning of this chapter, we had stated that the major critical issues of MOEAs are that of maintaining diversity and guiding the search towards the true Pareto front. In

this section we shall use some standard functions to test whether the algorithm can identify the true Pareto front from a deceptive front, whether it can perform in a convex and a non-convex front, whether it can find diverse solutions and whether it can find solutions uniformly spread along the front. All the tests were carried out in a 1.6GHz Pentium M processor with 1.5GB RAM machine.

Test function 1:

The first test function is intended to determine whether the proposed algorithm can effectively find the true Pareto front. The function is designed to create a deceptive or local Pareto front where many existing algorithms will get trapped. The test function, as defined in [6] is,

$$\text{Minimize } f_1(x_1, x_2) = x_1 \text{ and } f_2(x_1, x_2) = \frac{g(x_2)}{x_1}$$

$$\text{where } g(x_2) = 2 - \exp\left\{-\left(\frac{x_2 - 0.2}{0.004}\right)^2\right\} - 0.8\exp\left\{-\left(\frac{x_2 - 0.6}{0.4}\right)^2\right\}$$

$$\text{and } 0.1 \leq x_1, x_2 \leq 1.0$$

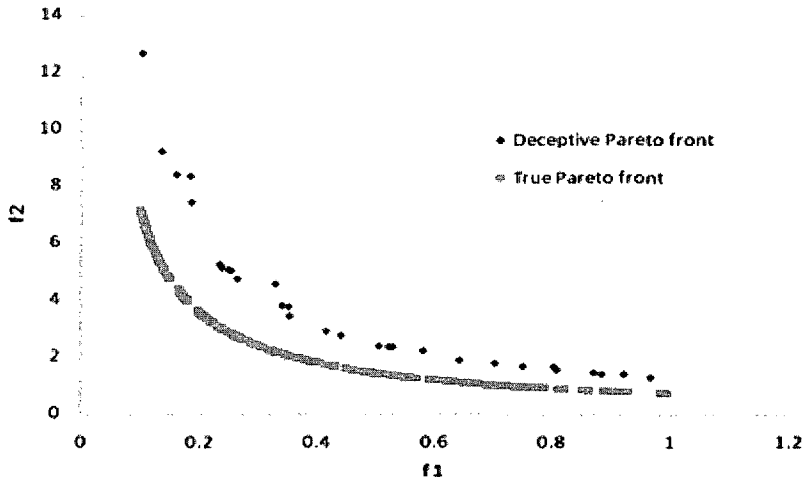

Figure 24: Deceptive Pareto front at iteration 1 and true front at iteration 22

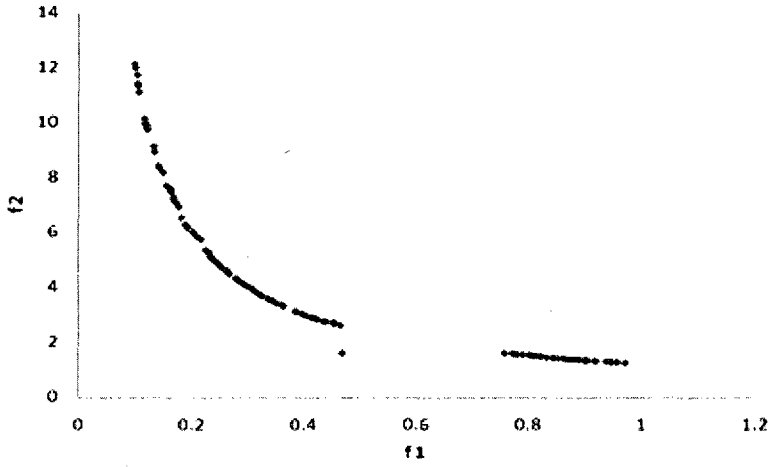

Figure 25: First discovery of a point along the true Pareto front at iteration 7

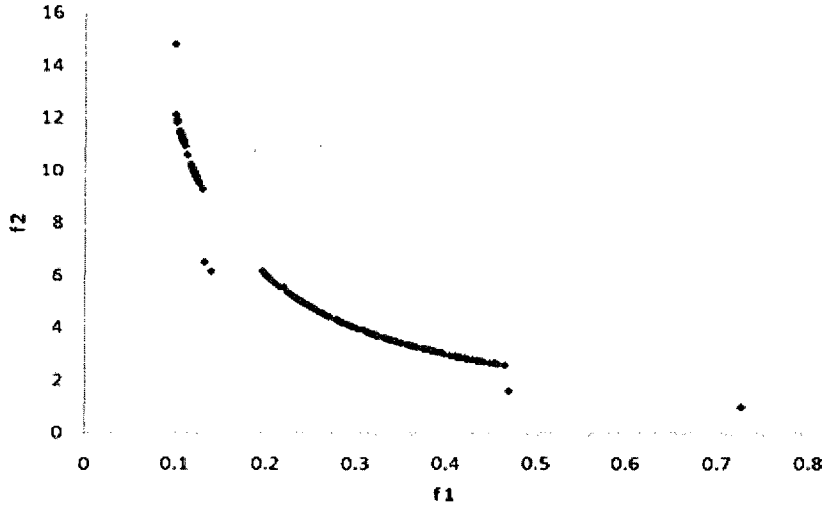

Figure 26: More points of the true Pareto front discovered at iteration 11

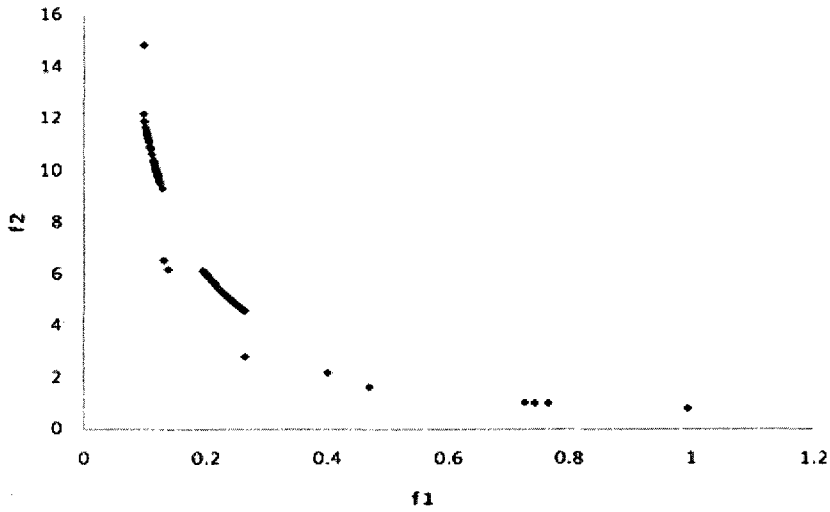

Figure 27: Shifting towards the true Pareto front discovered at iteration 12

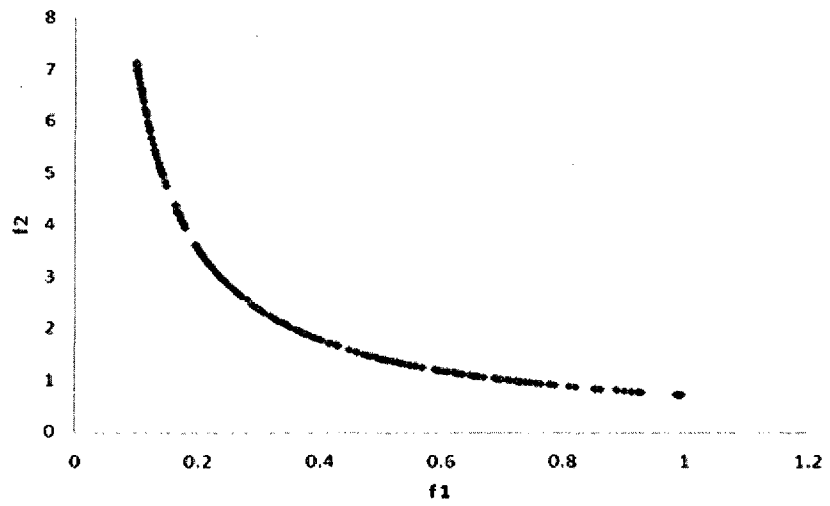

Figure 28: True Pareto front at iteration 22

From the above plots we can observe that in spite of getting trapped in the deceptive Pareto front initially, the algorithm soon managed to come out of the trap, which highlights the effectiveness of the exploration strategy. In terms of statistics, the algorithm got trapped in the deceptive front in 15 out of 20 runs, however, on all occasions i.e. all the 15 cases, it managed to come out of the deceptive Pareto front at around the 18^{th} iteration covering 5094 points out of a total search space of 81×10^{10} points in the variable space and having 257 non-dominated solutions. The initial

population considered was 60 and it grew to a size of 360. The final size of the external non-dominated set is same as the number of non-dominated solutions and the final size of the external dominated set is the total number of points covered minus the non-dominated solutions. A precision of up to 6 decimal places (20 bit binary string) as done in NSGA was considered. From [6] it appears that when NSGA, using a similar starting population and precision, was performed on the same function, it got trapped 51 out of 100 times but remained trapped even at the 100^{th} generation during one of its trial runs, thereby reflecting the fact that mERR-MOEA is highly effective for searching global optimums.

Test function 2:

The second test function determines the performance of the algorithm for convex Pareto fronts. The test function, as defined in [23] is,

$$\text{Minimize } f_1(x_1, x_2) = x_1 \text{ and } f_2(x_1, x_2) = g(x_2)h(f_1, g)$$

$$\text{where } g(x_2) = 1 + 9x_2$$

$$h(f_1, g) = 1 - \sqrt{f_1/g}$$

$$\text{and } 0 \leq x_1, x_2 \leq 1.0$$

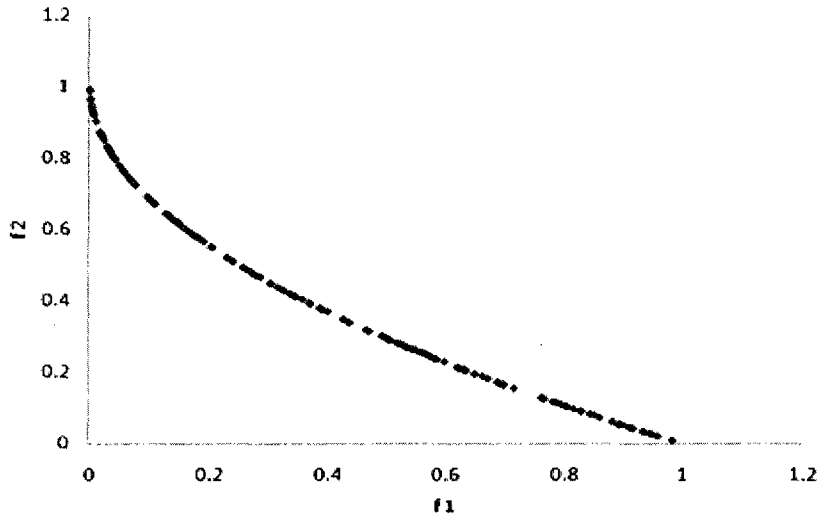

Figure 29: Pareto front

Similar to Test 1, a precision of 6 decimal places or a 20 bit binary number was considered here. The initial population was set at 100 and grew to a size of 300. The Pareto front having 201 non-dominated solutions was achieved after covering 2024 points out of a total search space of 10^{12} points in the variable space. For comparison, Figure 29 can be matched with Figure 30, although they are of different scales, however, Figure 30 reflects that SPEA is better than others.

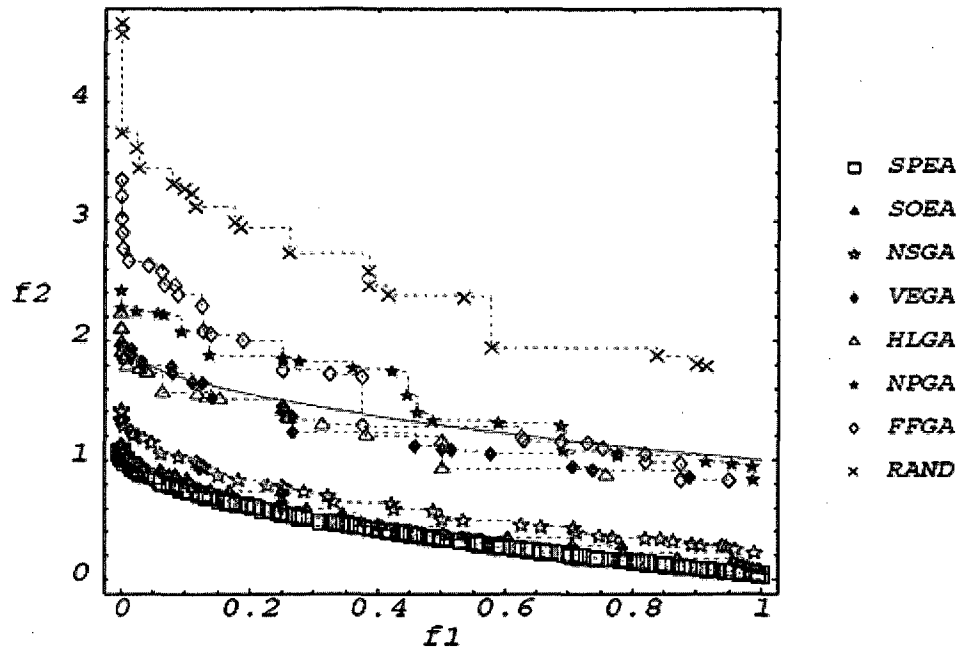

Figure 30: Pareto front of Test function 2 from [23]

Test Function 3:

Having successfully tested a convex Pareto front, we shall now test whether the algorithm works if the Pareto front is non-convex. The test function, as defined in [23] is,

$$\text{Minimize } f_1(x_1, x_2) = x_1 \text{ and } f_2(x_1, x_2) = g(x_2)h(f_1, g)$$

$$\text{where } g(x_2) = 1 + 9x_2$$

$$h(f_1, g) = 1 - (f_1/g)^2$$

$$\text{and } 0 \leq x_1, x_2 \leq 1.0$$

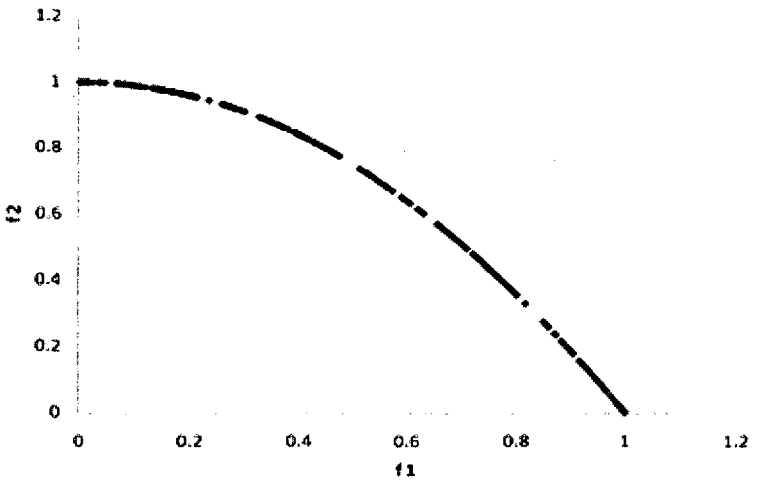

Figure 31: Pareto front

The Pareto front having 369 non-dominated solutions was achieved after covering 1927 points out of a total search space of 10^{12} points in the variable space with an initial population of 100 and a precision of 6 decimal places. The population grew to a size of 500. For comparison, Figure 31 can be matched with Figure 32, although they are of different scales, however, Figure 32 reflects that SPEA is better than others.

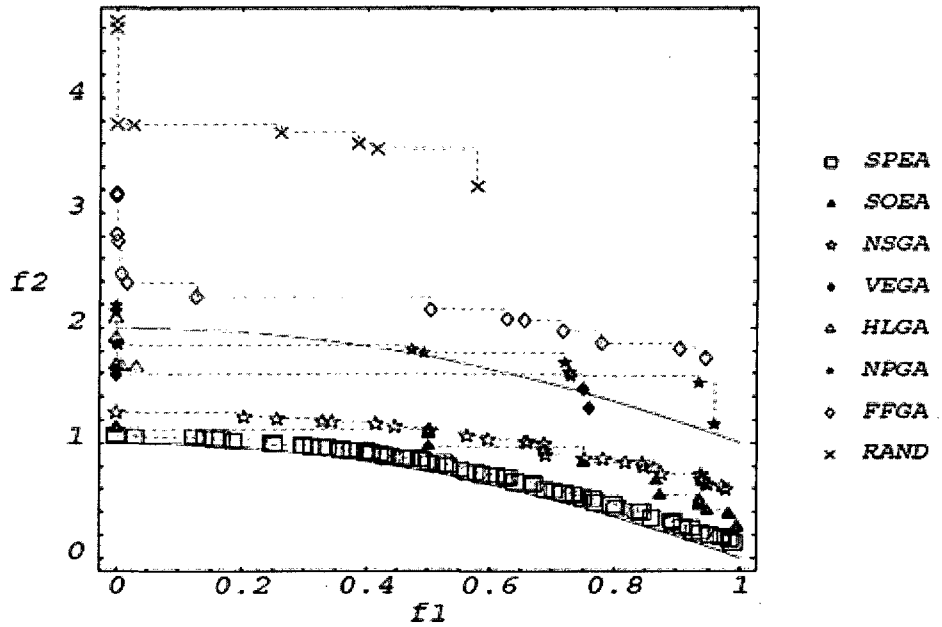

Figure 32: Pareto front of Test function 3 in [23]

Test Function 4:

The Pareto front of the following test function is discontinuous. Testing discontinuity is a way of measuring the algorithm's diversity implementation capability. The test function, as defined in [6] & [23] is,

$$\text{Minimize } f_1(x_1, x_2) = x_1 \text{ and } f_2(x_1, x_2) = g(x_2)h(f_1, g)$$

$$\text{where } g(x_2) = 1 + 9x_2$$

$$h(f_1, g) = 1 - \sqrt{f_1/g} - \left(f_1/g\right)\sin(10\pi f_1)$$

$$\text{and } 0 \leq x_1, x_2 \leq 1.0$$

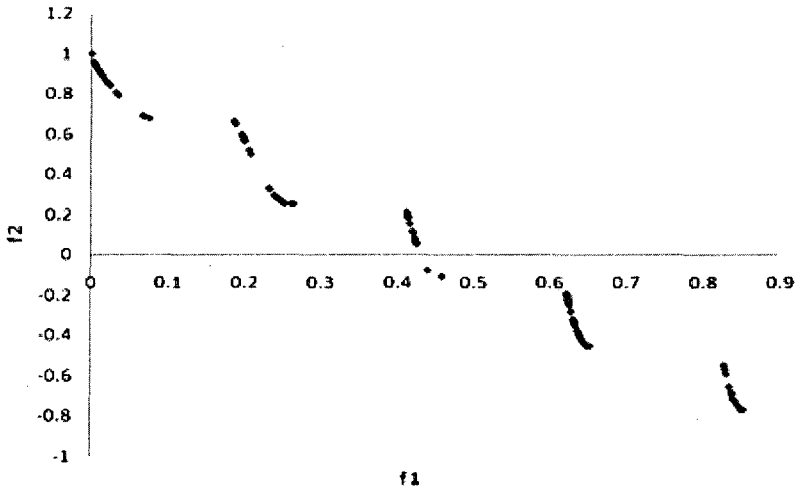

Figure 33: Pareto front of test function 4

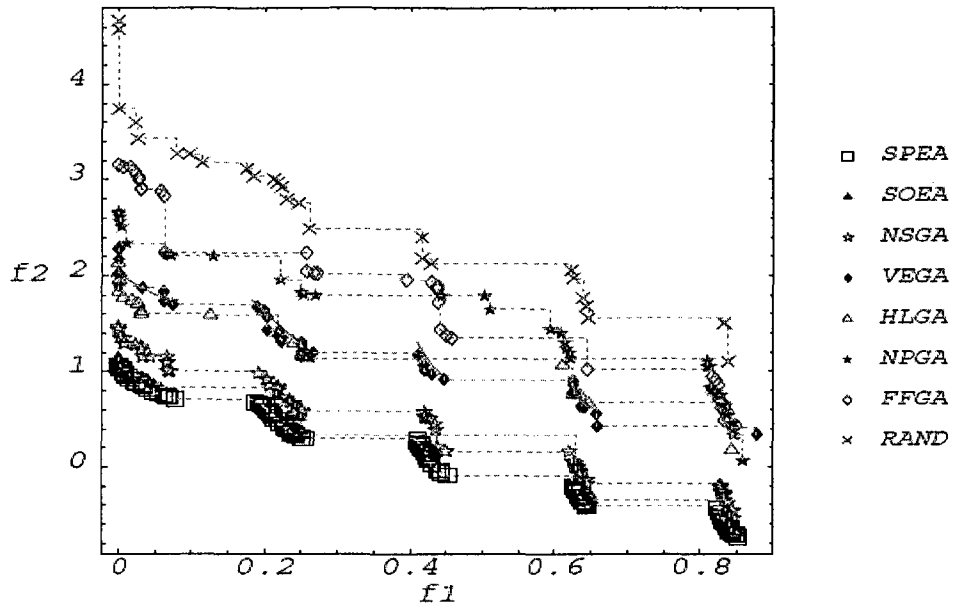

Figure 34: Pareto front of test function 4 given in [23]

The Pareto front having 105 non-dominated solutions was achieved after covering 2705 points out of a total search space of 10^{12} points in the variable space with an initial population of 100 and a precision of 6 decimal places. The population grew to a size of 200. For comparison, Figure 33 can be matched with Figure 34, although they are of different scales, however, Figure 34 reflects that SPEA is better than others.

Test Function 5:

In this test, on optimizing the functions, most solutions tend to bias towards $f_1 = 1$, which means that algorithms which are not good at maintaining diversity produce a biased Pareto front. The test function, as defined in [6] is,

$$\text{Minimize } f_1(x_1, x_2) = x_1 + x_2 \text{ and } f_2(x_1, x_2) = g(x_2)h(f_1, g)$$

$$\text{where } g(x_2) = 1 + 9x_2^{0.25}$$

$$h(f_1, g) = 1 - \left(f_1/g\right)^2$$

and $0 \leq x_1, x_2 \leq 1.0$

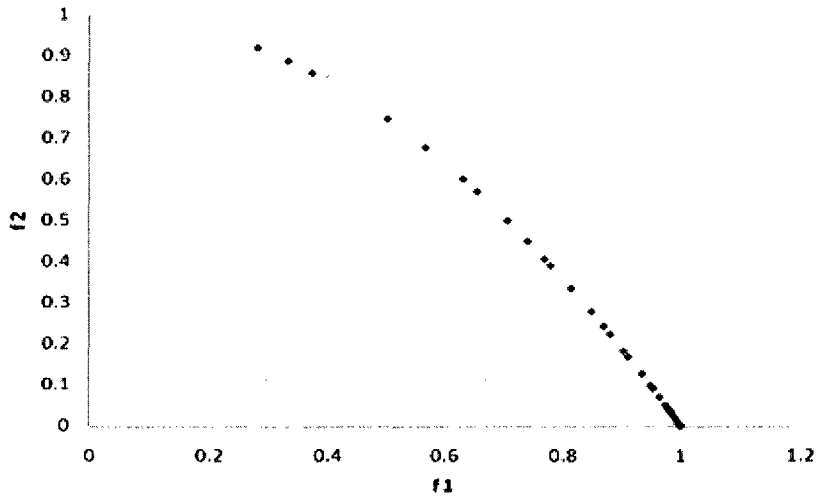

Figure 35: Pareto front of test function 5

The Pareto front having 50 non-dominated solutions was achieved after covering 602 points out of a total search space of 10^4 points in the variable space with an initial population of 50 and a precision of 2 decimal places. The population grew to a size of 150. As a test of bias, from Figure 35 we can determine that the front is more-or-less uniformly distributed thereby achieving diversity in the function space. The nature of the function is such that the value of x_2 from the non-dominated solutions is zero so that the minimum value of $g(x_2) = 1$. Hence, as a result, the objective function f_1 is only dependent on x_1. For better understanding, Figure 36 shows how the proposed algorithm achieves diversity in the parameter space i.e. the values of x_1 are spread throughout the function curve. In other words, the algorithm achieves diversity in both function space and parameter space without using any fitness sharing strategy as would have been in the case for the NSGA given in [6].

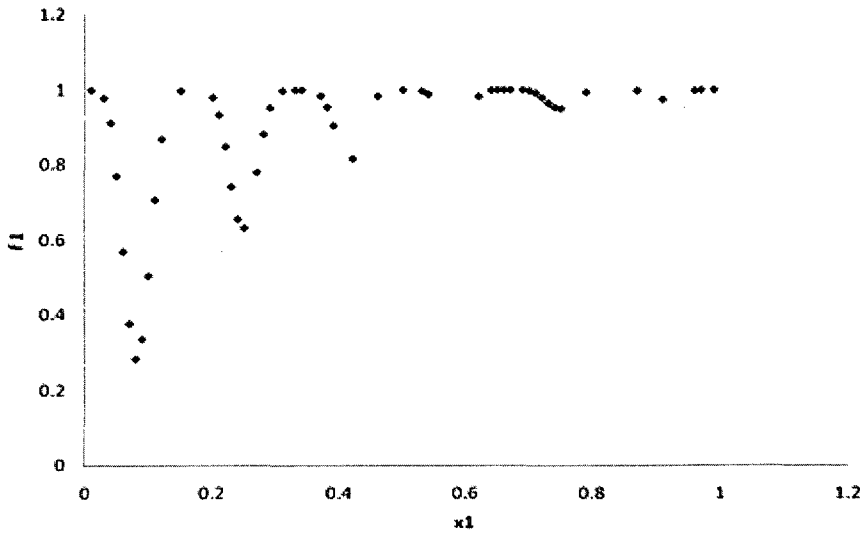

Figure 36: Plot showing diversity in parameter space

4.4 Benchmark Problem

Section 4.3 described tests to examine the performance of the algorithm in determining the Pareto front under specific, semi-artificial conditions by using a set of analytical functions. The next test involves a real device and is intended to verify the performance of the algorithm when compared to published results.

The TEAM benchmark problem 22 [28] deals with the optimization of a superconducting magnetic energy storage system (SMES). SMES devices store magnetic energy and therefore are mainly used in power grid networks for peak load supply. A typical SMES device is made up of a solenoid with superconducting coils. Since these devices generate stray fields across a wide area that can interfere with the operation of other nearby devices and also cause human safety related issues, a second solenoid is used as a shield for the main coil.

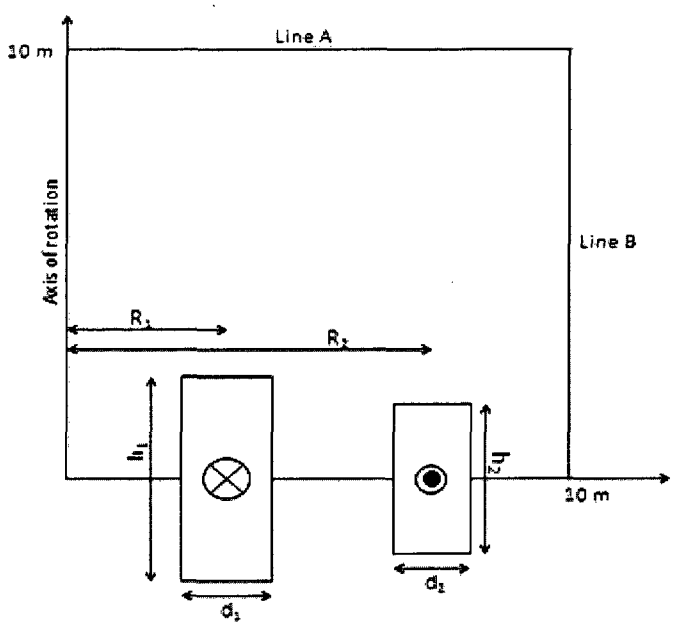

Figure 37: SMES device configuration

This optimization problem requires finding the correct geometrical parameters of the second device based on two objectives as described in [29] & [19], which are 1) to minimize the stray field and 2) to minimize the deviation from a predetermined stored energy value, while keeping the current density of both the coils below certain critical levels. Mathematically, this can be defined as:

$$Minimize \quad f_1 = \frac{|E - E_{ref}|}{E_{ref}} \; and f_2 = \frac{B_{stray}^2}{B_{norm}^2}$$

where $B_{stray}^2 = \frac{\sum |B_{stary_i}|^2}{22}, i = 1,2, \ldots, 22$ are 22 equidistant points along lines a & b

given $E_{ref} = 180MJ, B_{norm} = 2 \times 10^{-4}T$ [29]

$$R_1 = 2m, {}^{h_1}/_2 = 0.8m, d_1 = 0.27m$$

The parameters are: $R_2, {}^{h_2}/_2, d_2$

given $2.6 \leq R_2 \leq 3.4$

$$0.204 \leq {}^{h_2}/_2 \leq 1.1$$

$$0.1 \leq d_2 \leq 0.4$$

$$J_1 = 22.5MA/m^2$$

$$J_2 = -22.5MA/m^2$$

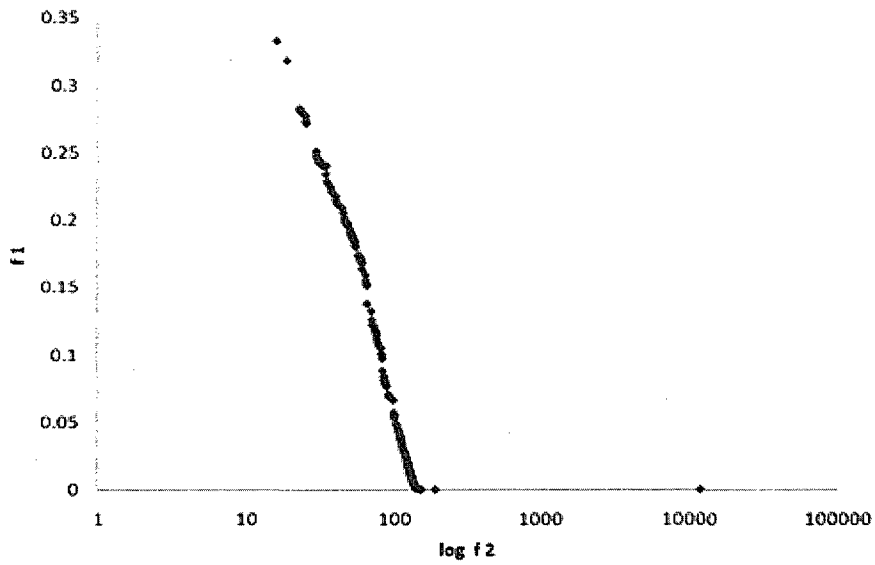

Figure 38: Pareto front of SMES

The number of points in the parameter space evaluated was 14564 out of a total search space of 108×10^9 points with a precision of 4 decimal places (16 bit binary string). It took 95 iterations to converge to the Global Pareto Optimal set. The Pareto optimal set has 195 optimal solutions. The initial population was 100 and it grew to a size of 300.

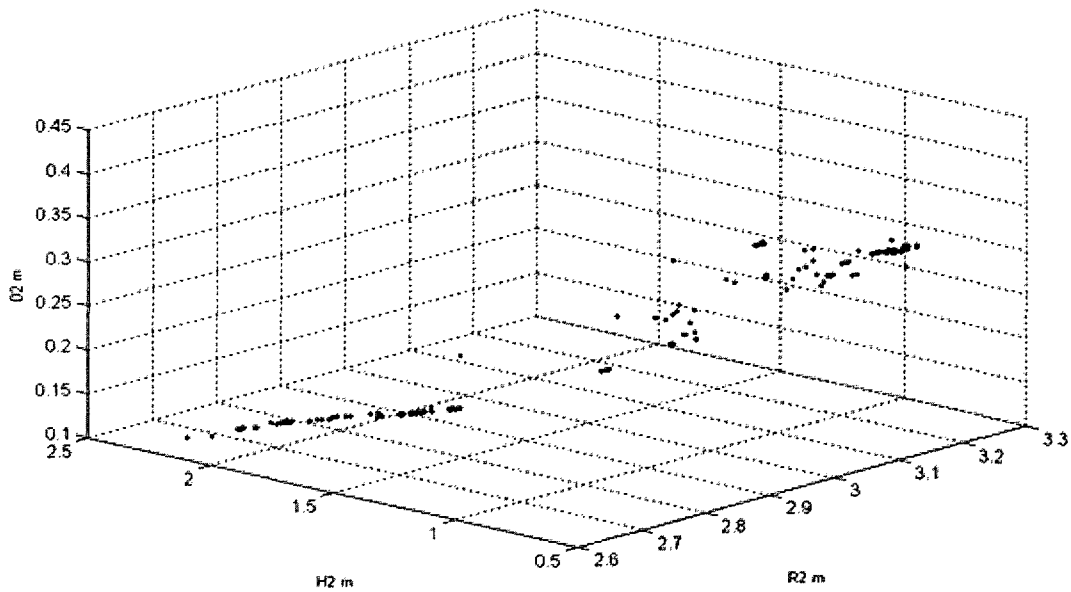

Figure 39: Parameter space (SMES)

In order to find the optimal solution, we followed the procedure given in [28], which is,

1. Eliminate the solutions that have $B_{stray} > 3mT$ and energy beyond the range of $\pm 18MJ$, i.e. having an error greater than 10%.
2. After the 1^{st} step, from the remaining solutions, the one that has the least volume ($V = 2\pi R_2 H_2 D_2$) is selected as the final optimal solution.

Using the above procedure, the final solution is,

R m	H m	D m	Energy MJ	$\lvert B_{stray} \rvert$ T	Volume m^3
3.1361	0.5200	0.3336	180.30	0.00237	3.4182

Table 2: Final optimal solution of SMES device optimization

The result can be compared with the solution from [28] which was achieved by using Multi-objective Clone Selection Algorithm (MOCSA),

R m	H m	D m	Energy MJ	$\lvert B_{stray} \rvert$ T	Volume m^3
3.4000	0.4397	0.2945	184.53	0.00256	2.7665

Table 3: SMES solution when using MOSCA

The result can also be compared with the best solution mentioned in [28],

R m	H m	D m	Energy MJ	$\lvert B_{stray} \rvert$ T	Volume m^3
3.0800	0.4780	0.3940	179.80	0.00089	3.6446

Table 4: SMES solution from [28]

The following diagram, Figure 40, shows the convergence progress of mERR-MOEA implemented in the SMES problem. It measures the relative change in the non-dominated population set over a block of six consecutive iterations as a percentage of the current non-dominated population.

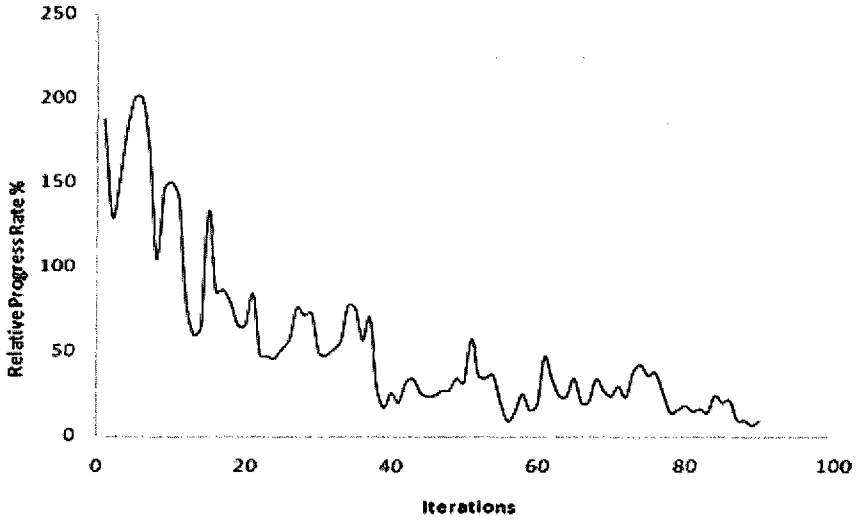

Figure 40: Progress measurement

4.5 Advantages of mERR-MOEA

Having tested the optimizer on all critical issues such as diversity, convex and non-convex functions and deceptive fronts as well as on a real physical device, we can summarize the advantages of mERR-MOEA as:

1. An ability to come out of deceptive Pareto front very quickly unlike NSGA.
2. An ability to save time during FEA by having a mechanism for keeping track

of all evaluated chromosomes unlike in SPEA or NSGA.

3. A mechanism to not to discard information, unlike in IMOEA.
4. A restricted resolution based search operation unlike the RCSA.
5. An ability to implement effective diversity without a fitness sharing method.
6. An effective mechanism to tackle both convex and non-convex functions without bias unlike the MOGA.

5 Optimizing the Motor

In Chapter 2 we discussed the concepts and general theoretical aspects of an IPM and also derived the torque equation and the parameters that affect it. In Chapters 3 & 4 we formulated the mERR-MOEA and tested it successfully on several analytical functions and on a benchmark device.

In this chapter, we shall fit the torque performance of an IPM to a given performance profile and find the optimal design. In addition, it is intended to investigate how the parameters affect the torque performance by analyzing the Pareto front. However, before that, we will first review the methods used for the calculation of torque from finite element analysis (FEA), and how the IPM has been modeled in MagNet.

5.1 Calculating Torque from Finite Element Analysis

In Chapter 2 while deriving the analytical torque equation we made certain assumptions such as no saturation, negligible iron losses, etc. However, by using 2D FEA, a more realistic model of the IPM can be simulated. The electromagnetic force on a body can be calculated by several procedures such as the Lorentz, Maxwell Stress Tensor and Virtual Work methods.

Lorentz Method: The Lorentz method can be used to predict the electromagnetic forces acting on current carrying structures only. Although it is easy to use with FEA, it cannot be applied to determine the forces acting on ferromagnetic structures. It is given by

$$F = BiL \tag{5.1}$$

where B is the magnetic flux density influencing the current carrying conductor, i is

the current through the conductor and L is the length of the conductor.

Virtual Work: It is based on the variation of the total magnetic energy of the system when its moving part is physically displaced in the direction of the component of the force being determined. At least two FEA field solutions are required, one before and the other after the displacement, which is also its drawback. If two or even three components of the force are required then the number of solutions needed at each point increases accordingly. The force is given by,

$$F = \frac{W_2 - W_1}{x_{12}} \tag{5.2}$$

where W_2, W_1 are the stored energy of the body at points 2 & 1 respectively and x_{12} is the displacement between points 1 & 2.

Maxwell Stress Tensor: Most FEA application software packages use this method to calculate torque. It requires only a single field solution and can be used to evaluate force on any part of the moving body. The accuracy of field calculations, which is critical near the interface regions of the air and the rotor, is enhanced by fine meshing. By integrating a simple force density expression over any closed surface surrounding the structure (ferromagnetic, winding, or both), the required force can be calculated. Choosing the correct path for integration determines the accuracy of the force calculations.

The force on a body is calculated from the following equation,

$$F = \frac{1}{\mu}\int_{V} \nabla . T dv = \frac{1}{\mu}\int_{S} T . ds \tag{5.3}$$

where

$$\boldsymbol{T} = \begin{bmatrix} (B_x^2 - \frac{1}{2}|B|^2) & B_x B_y & B_x B_z \\ B_y B_x & (B_y^2 - \frac{1}{2}|B|^2) & B_y B_z \\ B_z B_x & B_z B_y & (B_z^2 - \frac{1}{2}|B|^2) \end{bmatrix} \tag{5.4}$$

B_x, B_y, B_z are the flux densities along the x, y & z axes respectively. In the case of 2D analysis, $B_z = 0$. The torque exerted on a body is calculated by multiplying the force with the distance vector from the axis of rotation.

5.2 Modeling the IPM

As has been mentioned earlier, one of the objectives of this thesis is to fit the torque performance to a given torque-speed profile. We know that for flux weakening purposes and to run the IPM at synchronous speeds, the stator current and the supply frequency are controlled by an external electronic inverter circuit keeping the voltage supply a constant. The external circuit phase advances the stator current in order to achieve a maximum torque to current ratio. In our static 2D Finite Element Analysis of the simulated model in MagNet version 6.22.1 in a 1.6GHz Intel M processor with 1.5GB RAM, we bypassed the electronic circuit component, and instead, used a current that would have been supplied by the external circuit at that speed. We noted down the maximum torque and the corresponding phase angle and matched it with that in [2].

From Figure 2 we know that during constant power operation i.e. during flux weakening, torque is inversely proportional to speed. From Eqn.(2.2) & (2.3) we understand that for a given terminal voltage, back-emf is inversely proportional to the current and directly proportional to the speed, which means that current is inversely proportional to the

speed during flux weakening or constant power operation. Hence, it can be said that current and torque both change with speeds. Since the torque is what we are supposed to find, choosing different values of current is the same as choosing different speeds. Hence, the toque-speed profile can be represented by the torque-current profile.

A standard IPM has been selected as our reference model profile. The closest detailed set of specifications is given in [2]. The specifications cover major geometrical dimensions but have no mention of certain critical aspects, such as, the type of winding, spread angle of the embedded magnets, air gap length, magnet material, stator and rotor core material, etc. After several variations in the parameter sets in order to try to match the given torque-current profile given in [2], the best that was achieved is shown in Figure 41.

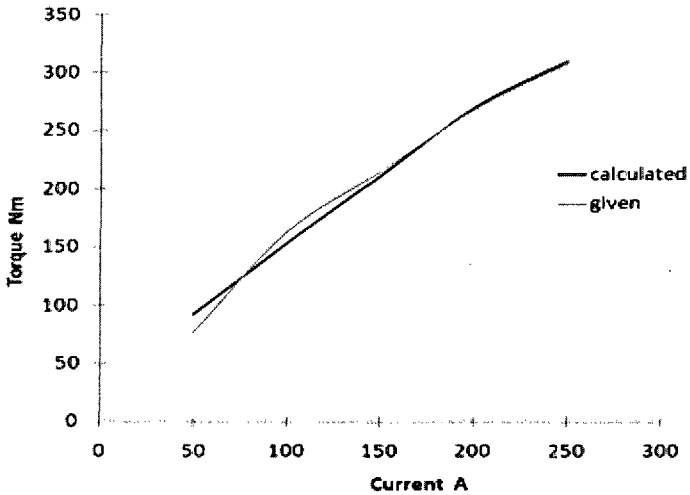

Figure 41: Torque-Current plot

The torque at a current of 200A came out to be 269.462Nm at an advance angle of 2^0 mech. against 268Nm at an advance angle of 8^0 mech. as given in [2].

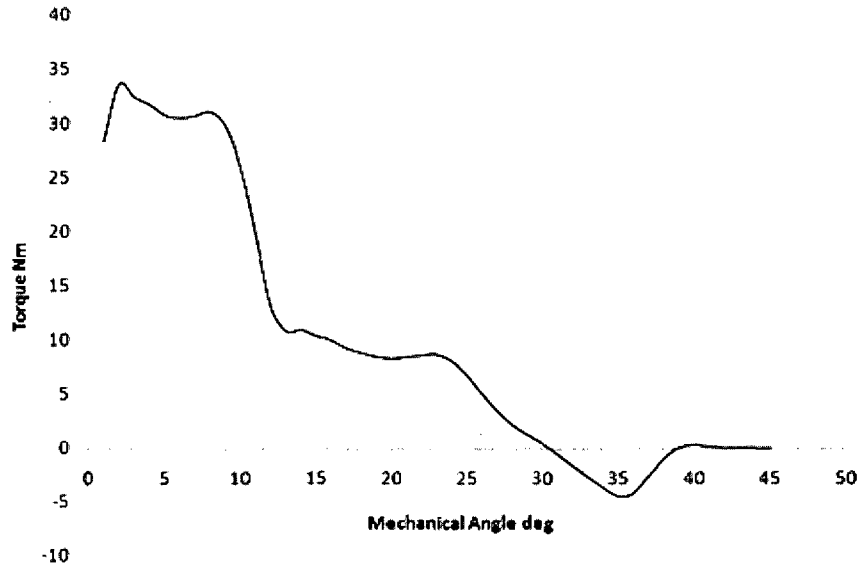

Figure 42: Torque plot at 200A at different positions of the rotor

The set of parameters that helped to achieve such a close match are, including the parameters that were deduced after several trials,

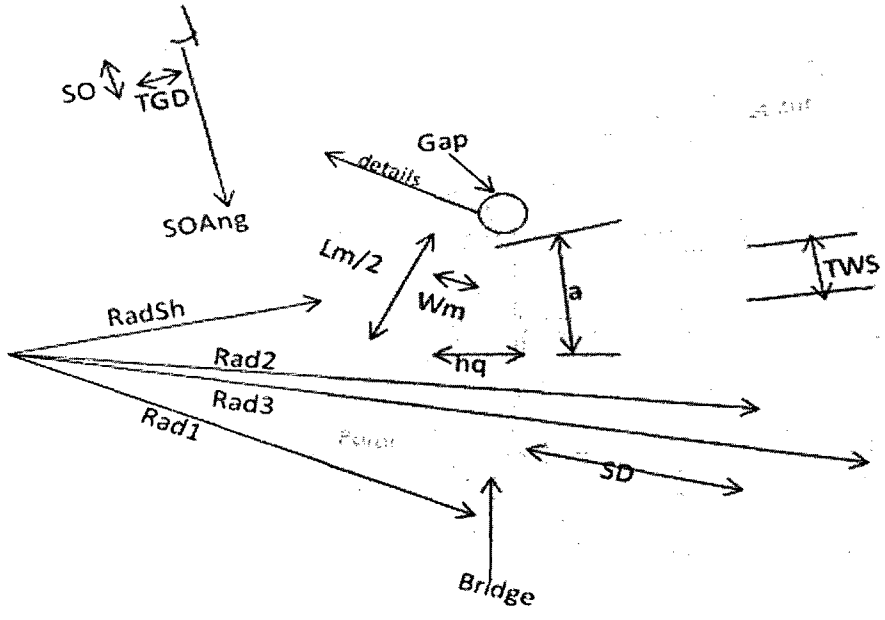

Figure 43: Detailed parametric diagram of the IPM

Parameter	Abbrev. in above diag.	Value mm
Stator outer radius	Rad3	134.6
Radial distance to end of slot	Rad2	116.2
Rotor outer radius	Rad1	79.0
Shaft radius	RadSH	50.5
Lamination stack length		83.5
Half magnet length	Lm/2	38.5
Magnet width	Wm	5.3
Bridge width	Bridge	1.4
Air gap	Gap	1.2
Half spread angle or half pole-arc angle	a	130^0
Stator tooth width	TWS	8.6
Slot depth	SD	35.2
Slot opening	SO	0.7
Radial depth of stator tooth tip	TGD	0.4
Undercut angle of stator tooth tip	SOAng	45^0
IPM web	Web	16.0
Radial depth of pole cap	hq	12.3

Table 5: IPM dimensions

Additional key design parameters:

Magnetic Poles	8
Slots per pole	6
Layers in stator slots	1
Coils per phase	8

Phases	3
Turns per coil	9
Coil connection	Series
Winding type	Distributed
Strands per coil	13
Wire type and Gauge	AWG 19
Magnet	Neodymium Iron Boron
Magnet Strength	1.13T
Lamination thickness	0.36mm
Stator & Rotor Material	M-19
Maximum Speed	5400rpm
Base speed	1040rpm
Peak Power Output	55kW

Table 6: Other information about the IPM to be optimized

5.2.1 Modeling the IPM: Key Aspects

It is a general procedure to enclose the computational domain within an air box having a larger size than the motor in order to simulate stray fields, accordingly. The air box was divided into stator and rotor regions so that the rotor air box rotates along with the rotor. Once the air box was drawn, the components of the motor were just embedded in it.

Force calculations by the Maxwell Stress Tensor method in and around sharp corners, in general, are erroneous [30]. In order to reduce the error, a virtual air gap was added. It has the same properties as that of air but is not used as a separation layer between bodies by MagNet [30]. For construction of the IPM, 2 layers of air and 2 layers of virtual

air were used. Multiple layers of air were implemented in order to divide the air gap into a stator and a rotor side, one facing the rotor and the other the stator. The arrangement is shown in Figure 44.

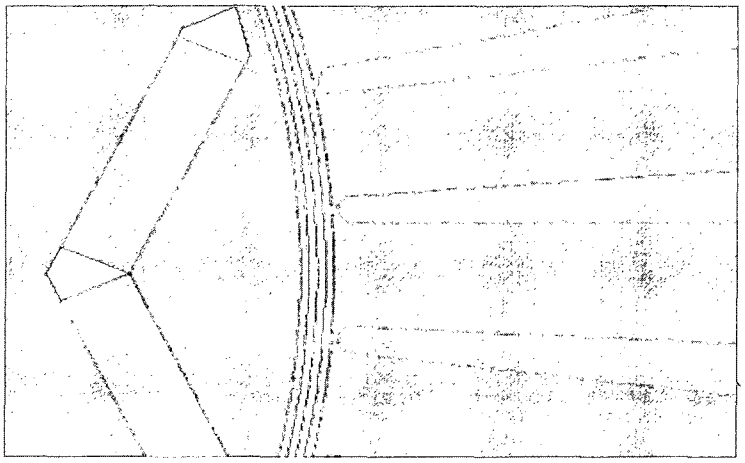

Figure 44: 4 layers of air-gap (outer two grey ones are virtual air and the inner two are air) in IPM

The stator coil was selected as a distributed, fully pitched winding connected in series. The three phase winding structure of the complete motor is shown in Figure 45 and Figure 46.

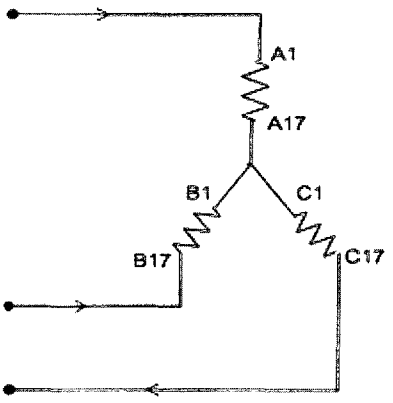

Figure 45: Coil connections of the IPM

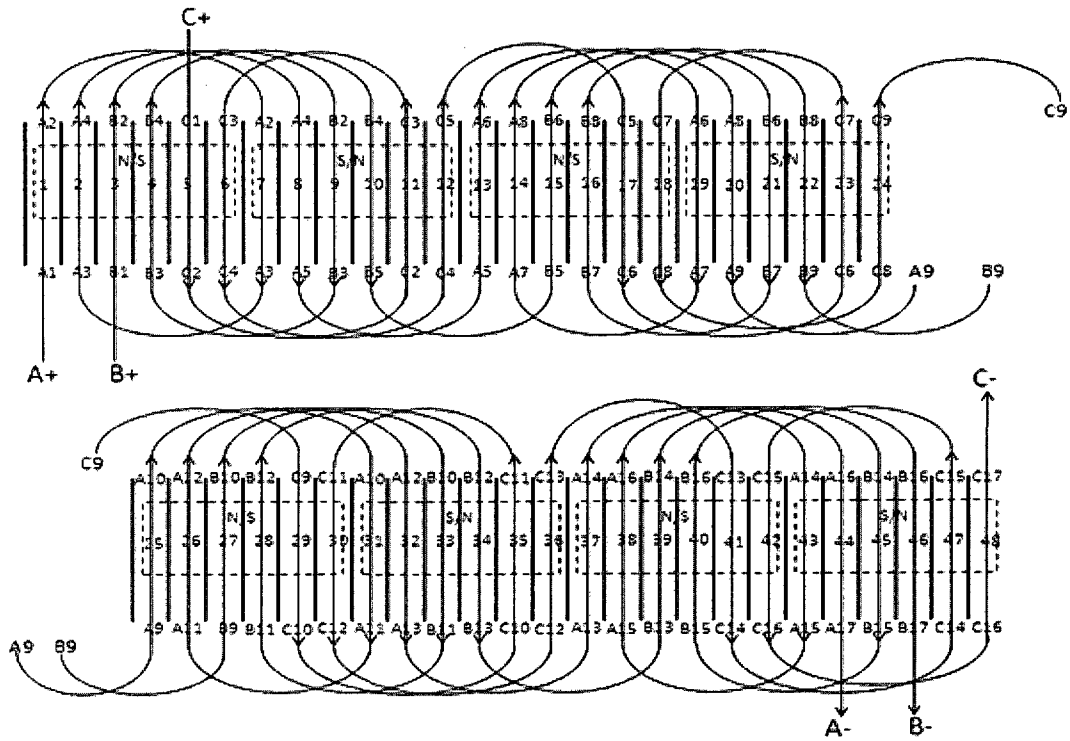

Figure 46: Detailed 3-phase distributed winding structure of the IPM

Due to the symmetrical structure, instead of simulating whole motor, $^{1}/_{8}th$ of it was used. In order to model the symmetry condition, the edges of the air box, should overlap the axes of symmetry as shown in Figure 47. In our case since we are simulating a single pole, we specified odd symmetry which means that the same coil connection and the magnetic polarity gets repeated alternately.

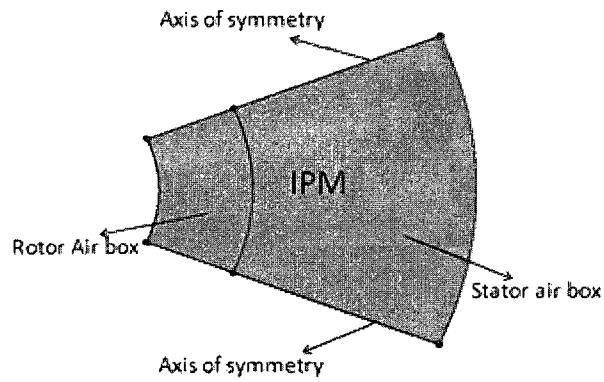

Figure 47: Symmetry axes of the IPM

We had mentioned in the previous section that there were some critical pieces of information that were missing in [2]. One of them is the slot width, which is important since it determines the resistance of the coil. However, the coil gauge, strands, turns and materials are known to us, from which we can calculate the area or volume occupied by the coils. The slot width was calculated assuming that the slots are nearly 100% filled (space is be left out due to round conductors and insulation). The final model is,

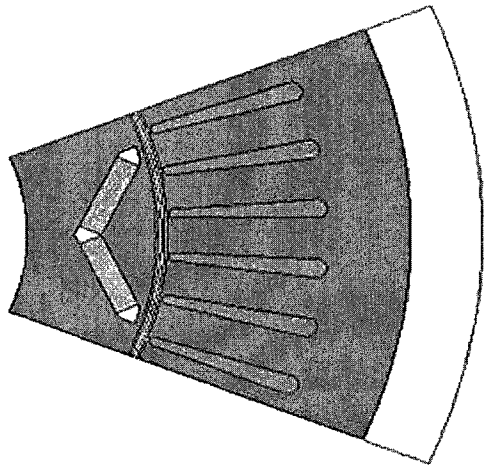

Figure 48: 1/8th part of the IPM for FEA

The direction of magnetization of the permanent magnets was set to be towards the air gap parallel to the width of the magnet in order to increase the density of flux in and around the air gap. The field distribution shown in Figure 49 shows how symmetry conditions affect the field lines.

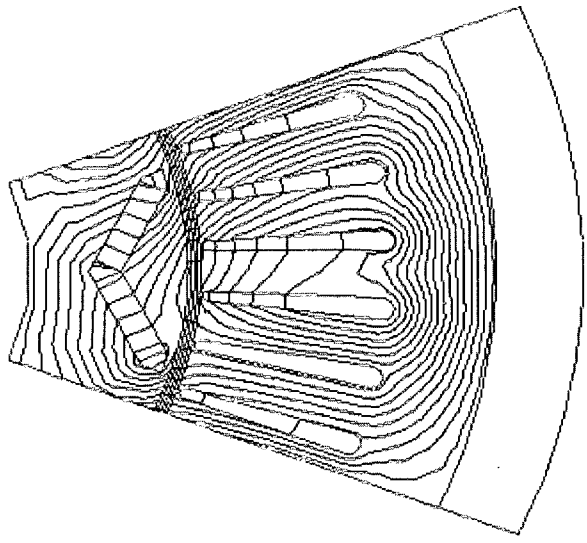

Figure 49: d-axis Field distribution of the IPM at 200A

The solution mesh in Figure 50 shows the crowding of tiny elements in and around the air gap and the flux barriers.

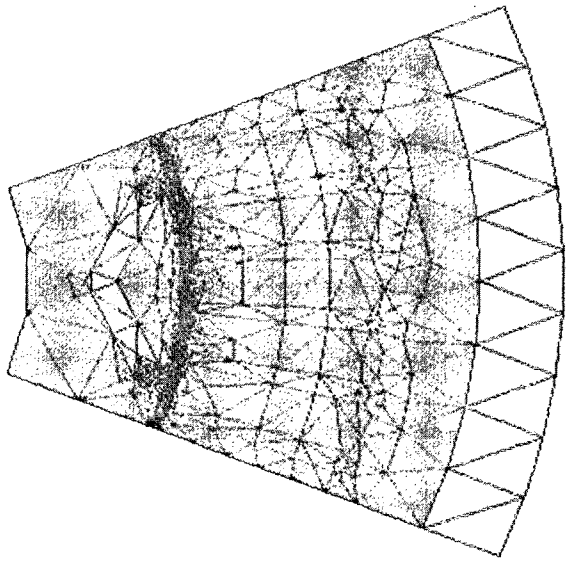

Figure 50: Solution mesh of the IPM

5.3 Optimizing the IPM

In Chapter 2, it was determined that about 8 geometrical parameters affect torque generation. Considering all eight parameters isn't feasible computationally, although the algorithm can do it, as that would mean an increased parameter search space and hence more time to achieve the Pareto front. Based on this and also since magnets are costly [2], it was decided to concentrate the optimization process on the magnet geometry and its location. In fact, we also investigated the effect of the air gap on torque.

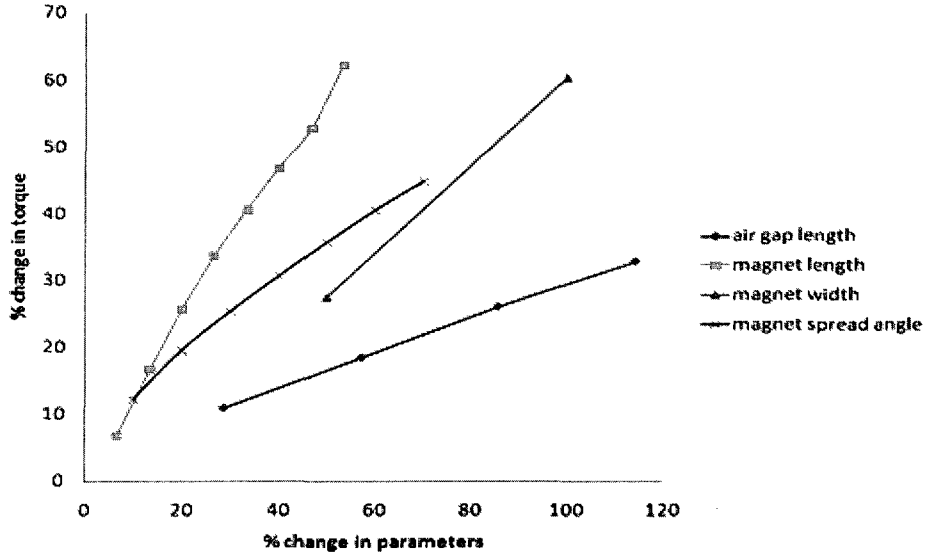

Figure 51: Effect of different parameters on torque

Figure 51 was obtained by assuming that the parameters are independent of each other. The steepness of the curve determines how sensitive the "% change in torque" is with respect to the "% change in parameters". In other words, a small change in the magnet length produces a huge change in the torque of the motor. From the above plot, it is evident that the air gap, although a critical parameter, may not be that important when compared to the location of the magnet and its size.

Having established a working model of the IPM and determined the parameters to be optimized, we will now consider optimizing the IPM. Three objectives were chosen so that the Pareto front can be represented by a 3D plot. As pointed out in section 5.2, torque at different current values can be considered equivalent to the torque at different speeds and thus we chose three different current values representing three different speeds i.e. a speed less than the base speed, the base speed and a higher speed. According to [2], 250A, 200A and 100A are approx. equivalent to 520rpm, 1040rpm and 3000rpm respectively.

$$Minimize\ f_1 = \frac{|T_1 - T_{200}|}{T_{200}}, f_2 = \frac{|T_2 - T_{250}|}{T_{250}}\ \ and\ f_3 = \frac{|T_3 - T_{100}|}{T_{100}}$$

$$Parameters \qquad 17.5 \leq \frac{L_m}{2} \leq 21.5$$

$$4.0 \leq W_m \leq 6.5$$

$$55 \leq a \leq 75$$

where T_{200}, T_{250}, T_{100} are the reference torques at 200A, 250A and 100A respectively; L_m is the length of the magnet, W_m is the width of the magnet and a is half of the pole-arc angle or half of the spread angle of the magnets (the following experiments were performed on a 1.6GHz Pentium M processor with 1.5GB RAM machine).

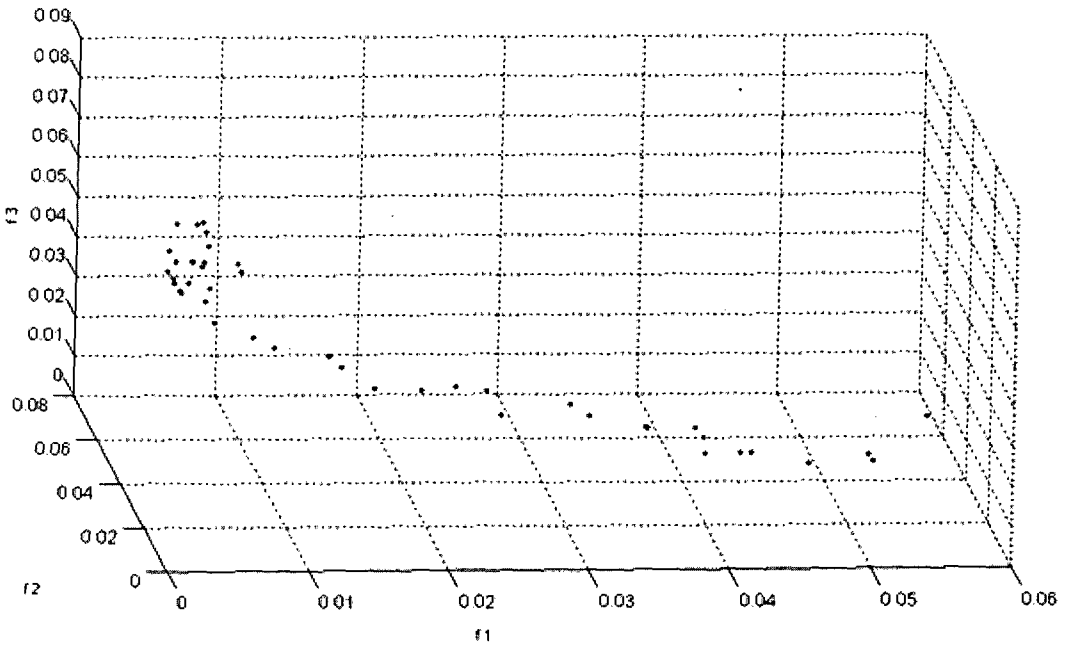

Figure 52: IPM optimization: Pareto plot

The Pareto front, having 44 non-dominated solutions, was achieved after covering 1403 points out of a total search space of 20000 points with an initial population of 20 and a precision of 1 decimal place. It took 33 iterations to converge. The population grew to a size of 60.

On careful observation of Figure 52, we may say that the Pareto front looks like a straight line, and, since the experiment was conducted considering the objectives to be absolute torque error, we cannot infer anything conclusive in terms of the actual torque values. For example, one of the objectives was to minimize,

$$f_1 = \frac{|T_1 - T_{200}|}{T_{200}} \tag{5.5}$$

where $\frac{|T_1 - T_{200}|}{T_{200}}$ is the absolute error. However, if we know the actual error we can derive the actual torque values. In order to do so, we recorded the actual error during the optimization process. Let the actual error be f_1',

$$f_1' = \frac{T_1 - T_{200}}{T_{200}} \tag{5.6}$$

Therefore,

$$T_1 = f_1' T_{200} + T_{200} \tag{5.7}$$

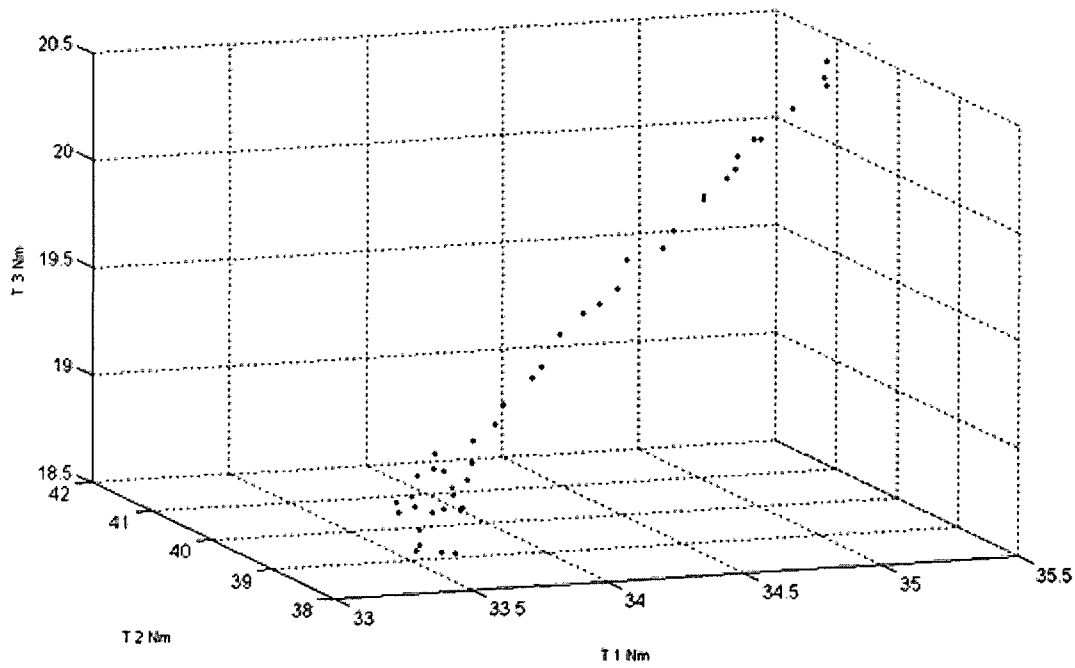

Figure 53: Actual Torque plot of the global non-dominated solutions

From Figure 53, we may say that current is proportional to the torque and therefore the front is slanted and unidirectional.

As mentioned earlier, the Pareto front allows the designer to make trade-off choices for the final "best" design according to his requirements. In order to do this, we chose the following steps, which are,

1. Filter out those absolute torque errors that are greater than 5%.
2. After filtering out, the best design is the one having the least *cogging torque*.

By following the above procedure, the best design achieved is,

w_m	$L_m/2$	a	T_1	T_2	T_3	$T_{cogging}$
5.6 mm	20.9 mm	55^0	274.43 Nm	313.78 Nm	157.36 Nm	6.5%

Table 7: Optimal design specifications

Now, we shall analyze the effect of the three parameters on the torque performance and thus make a decision on the 'optimal' machine design.

Figure 53 indicates that the torque, being a straight line, is not dependent on all 3 parameters as was assumed previously. It can also be observed from the plot that the parameters affect the torque in the same way at all currents, i.e. at all speeds which means that the torque is dependent on the ratios between the parameters. However, this observation is not totally conclusive from Figure 53, and therefore we need to study each parameter separately.

In Figure 54 consider the portion for one of the torques where the increase of 'a' from approx. 72^0 to 74^0 (circled region) is directly proportional to the increase in the torque. Fixing 'a' at 72-74^0 and looking at Figure 55, it can be said that when 'a' is between 72^0-74^0, W_m is approx 4.4mm (circled region). Now, by fixing 'a' at 72-74^0 and W_m at 4.4mm, by looking at Figure 56, it can be said that when 'a' is between 72-74^0 and W_m is 4.4mm, $L_m/2$ is approx 21.4mm (circled region). In order to get a better understanding of W_m and $L_m/2$, we now refer to Figure 57 and Figure 58. From Figure 57, it can be observed that when W_m is 4.4mm (circled region), the change in the torque is independent of Wm and this is the minimum value of W_m for a Pareto optimal solution. From Figure 58, it can be said that when $L_m/2$ is 21.4mm (circled region) given that W_m is held at 4.4mm, the change in the torque is not due to $L_m/2$ and this is the minimum value of $L_m/2$ for a Pareto optimal solution. In other words, change in the torque is dependent on the spread angle.

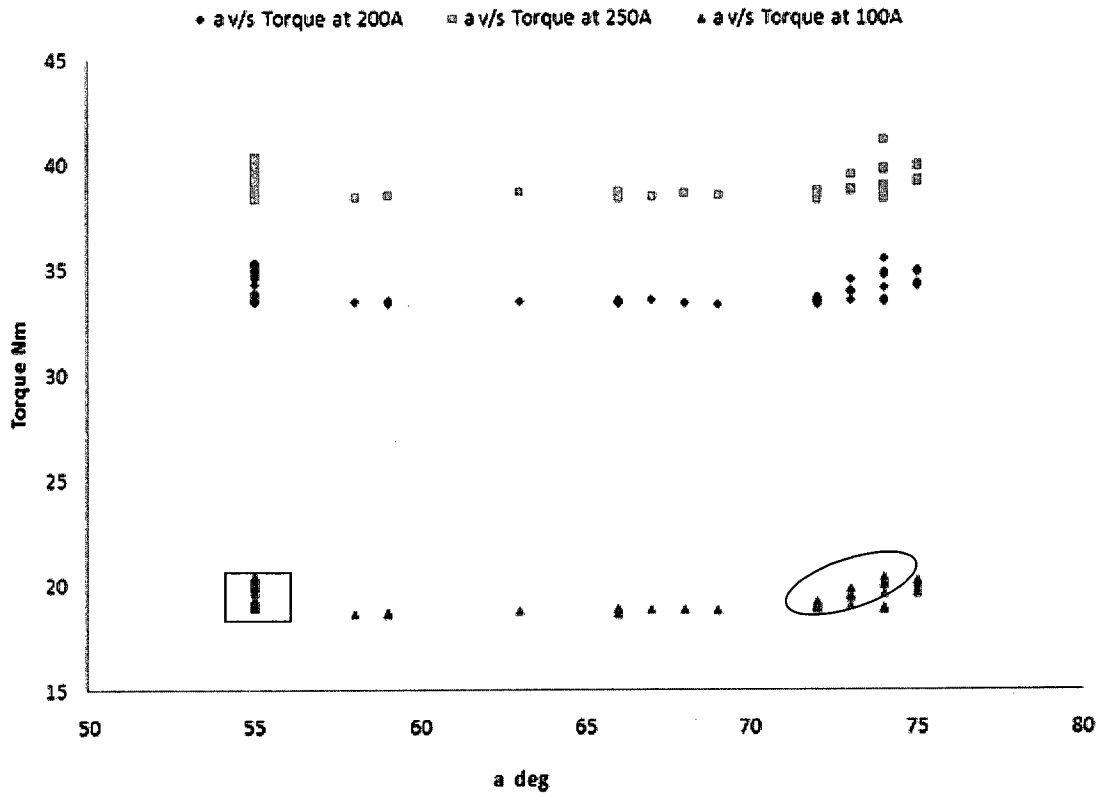

Figure 54: Half of the pole-arc angle from non-dominated solutions against the respective torques independent of 'Wm' and 'Lm/2'

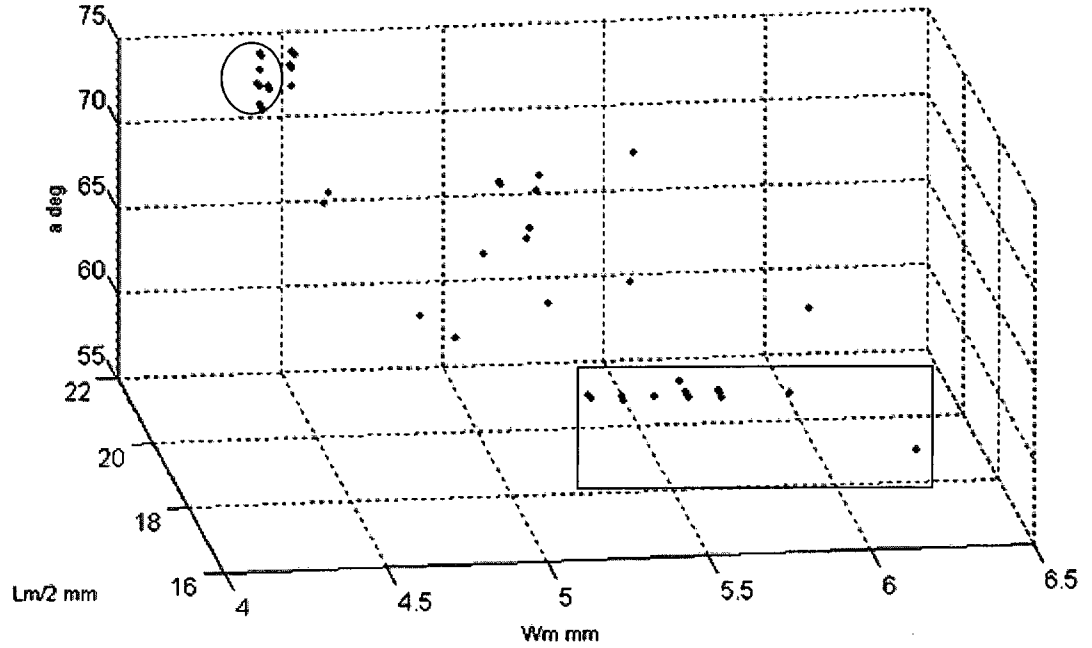

Figure 55: 3D plot of the non-dominated solutions in parameter space-view 1

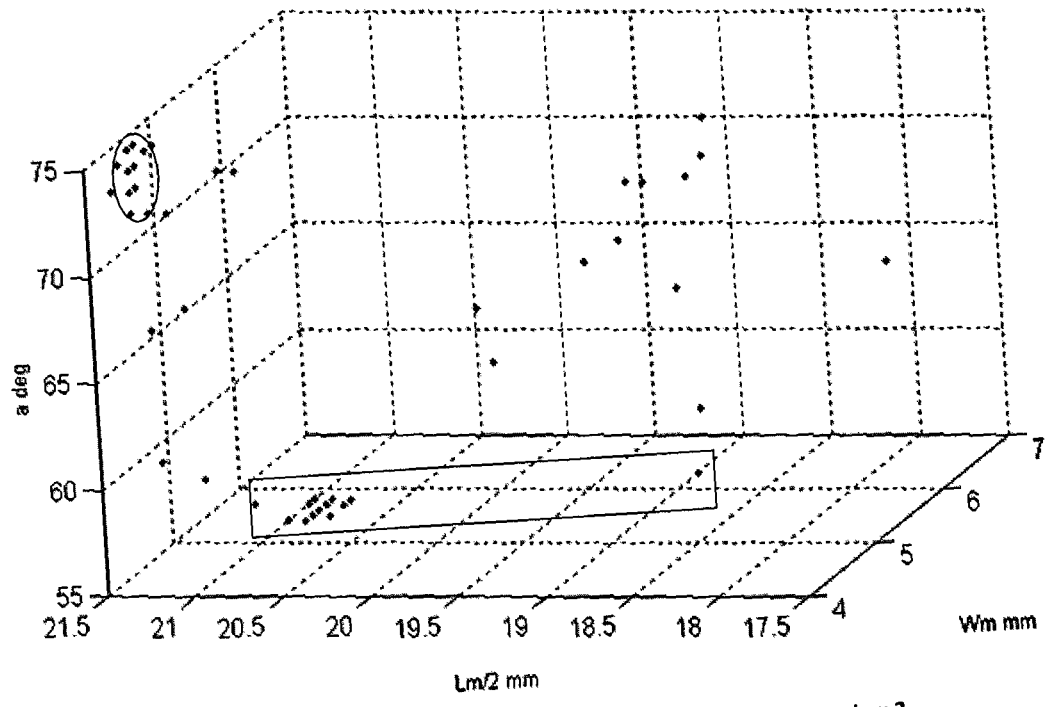

Figure 56: 3D plot of the non-dominated solutions in parameter space-view 2

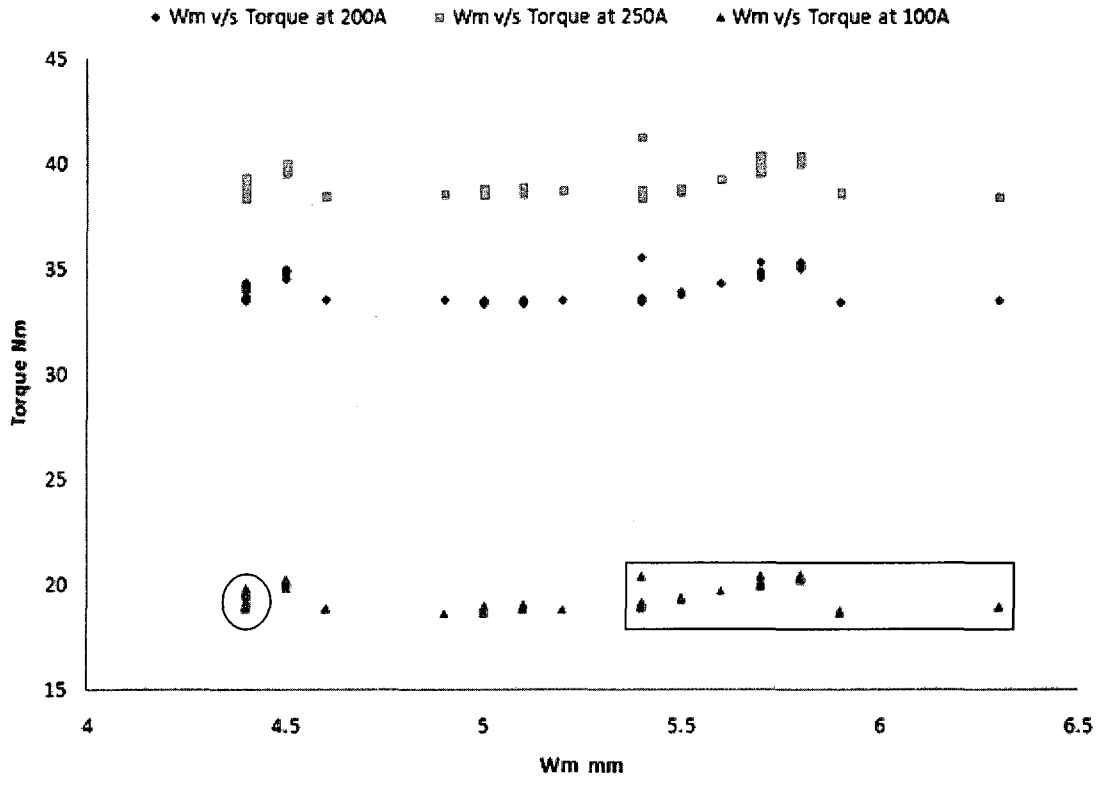

Figure 57: Width of the magnets from non-dominated solutions against the respective torques independent of 'a' and 'Lm/2'

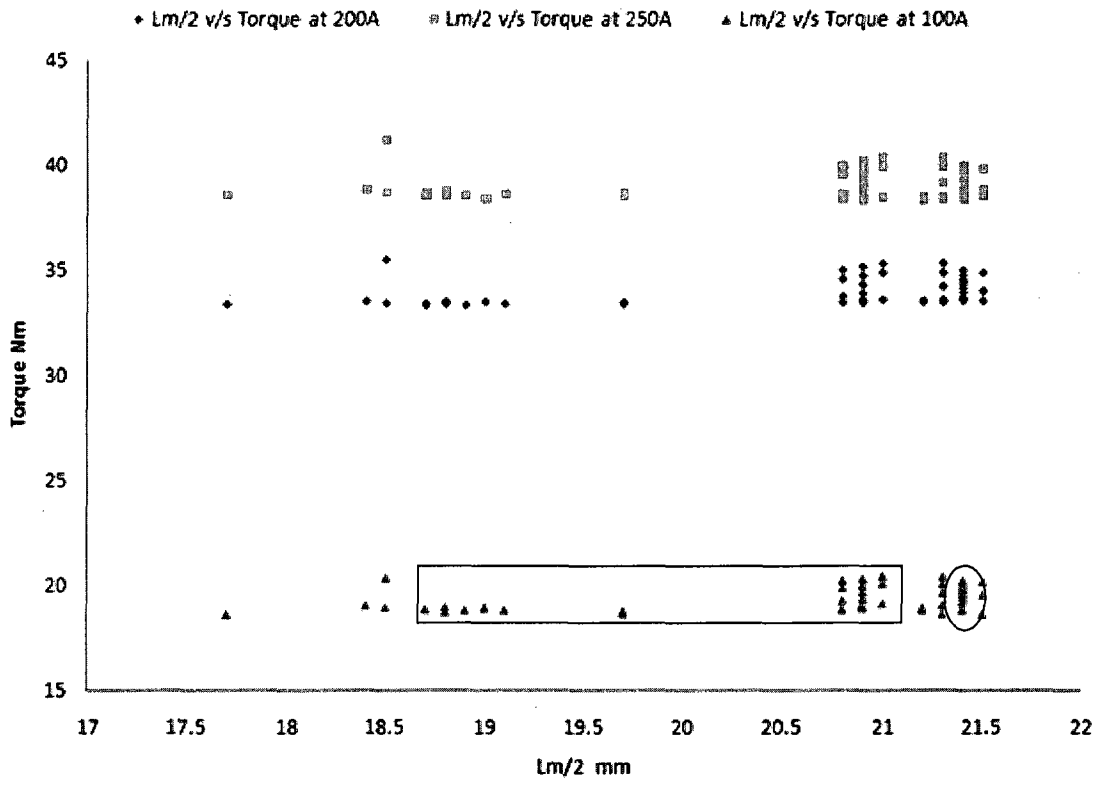

Figure 58: Half of the length of the magnet from non-dominated solutions against the respective torques independent of 'a' and 'Wm'

Next we shall investigate what happens if 'a' is fixed. In Figure 54 consider the portion where 'a' is approx 55^0 (region highlighted by a rectangle) and the change in the torque is independent of 'a'. Fixing 'a' at 55^0 and looking at Figure 55, it can be said that when 'a' is 55^0, W_m is approx. between 5.3-6.3mm (region highlighted by a rectangle). This implies that a reduction in 'a' to 55^0 results in an increase in W_m for a Pareto optimal design. Now, by fixing 'a' at 55^0 and W_m at 5.3-6.3mm, by looking at Figure 56, it can be said that when 'a' is 55^0 and W_m is between 5.3-6.3mm, $L_m/2$ is approx. between 18.7-21mm (region highlighted by a rectangle). In order to get a better understanding of W_m and $L_m/2$, we now refer to Figure 57 and Figure 58. Referring to Figure 57, when W_m is between 5.3-6.3mm (region highlighted by a rectangle), the torque is essentially independent of W_m with about 6% variations. However, from Figure 58, when $L_m/2$ is

between 18.7-21mm (region highlighted by a rectangle), the torque is not necessarily a constant.

Having studied the above diagrams, it might seem that there is a relationship involving W_m and $L_m/2$ which affects torque, i.e. the torque is dependent on the area of the magnet.

Let us now investigate how the area affects the torque (in effect, Lm and Wm both contribute to the volume of the magnet or area in 2D as the stack is of unit length).

From Figure 59 and Figure 60, we may observe that when 'a' is constant, the torque is directly proportional to the 'area' and when 'area' is constant, the torque is directly proportional to 'a', i.e. the torque is affected by two independent parameters instead of three. However, as 'a' is increased holding the area constant, the magnet moves towards the surface of the rotor, because the bridge width was considered to have a fixed value during the analysis, provided the aspect ratio remains unchanged. Similarly, as the 'area' is increased holding 'a' constant, which means a greater volume of the magnet, torque increases.

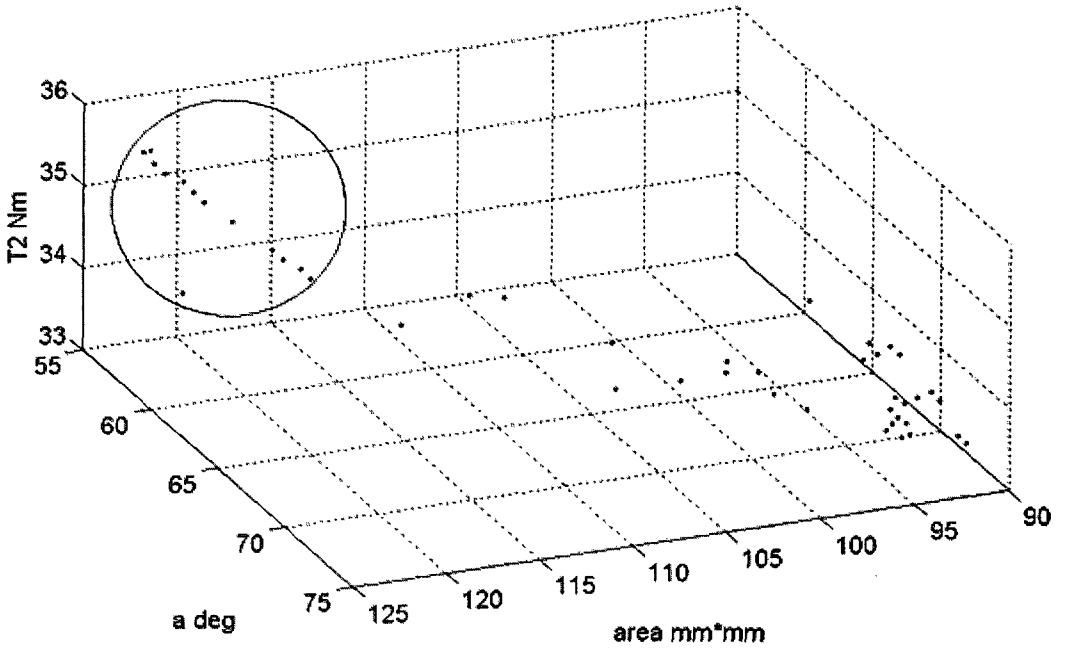

Figure 59: The circled region showing a constant 'a' and a varying 'area'

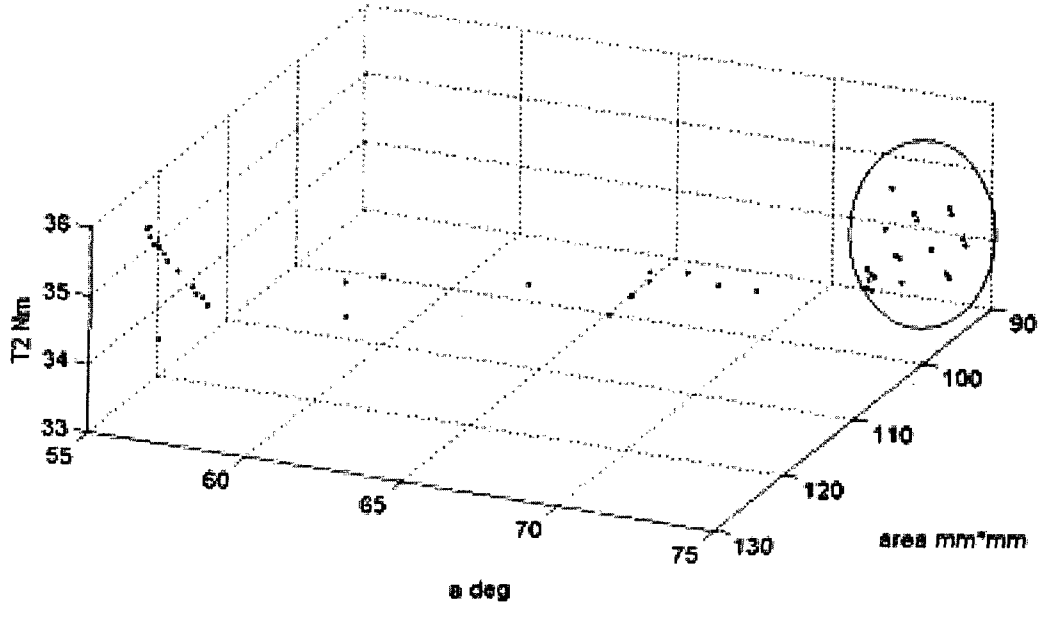

Figure 60: The circled region showing a constant 'area' and a varying 'a'

Further, on comparing the magnitude of the slope of the plot in Figure 61 when 'area' ranges from 113-122mm^2 (which is same as the circled region in Figure 59 having a constant 'a'), with the slope of the plot in Figure 62 when 'a' ranges from 72-74^0 (which

is same as the circled region in Figure 60 having a constant 'area'), we may conclude that the area of the magnet has greater effect than the spread angle, which vindicates our initial analysis shown in Figure 51. On trying to compare our observations with the initial analysis in Chapter 2, we may say that, analytically these facts cannot be verified.

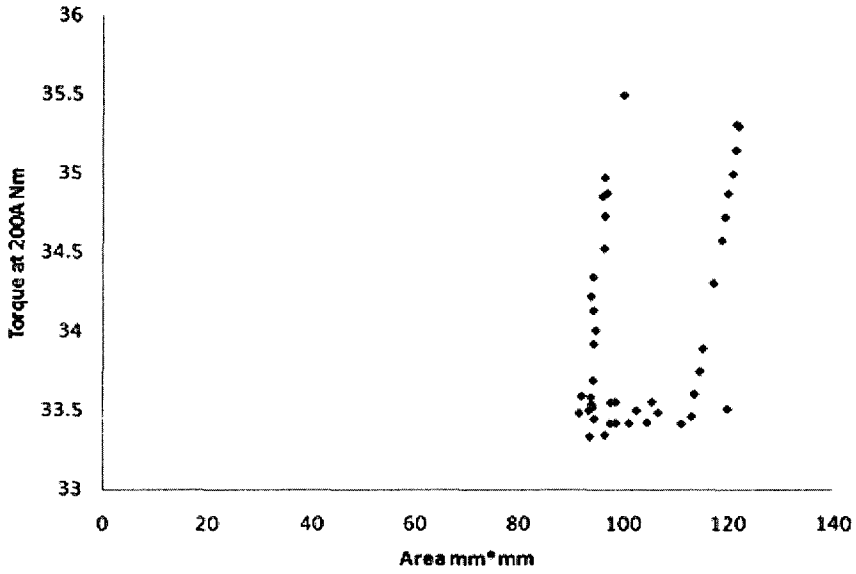

Figure 61: Effect of the area of the magnet on the torque at 200A

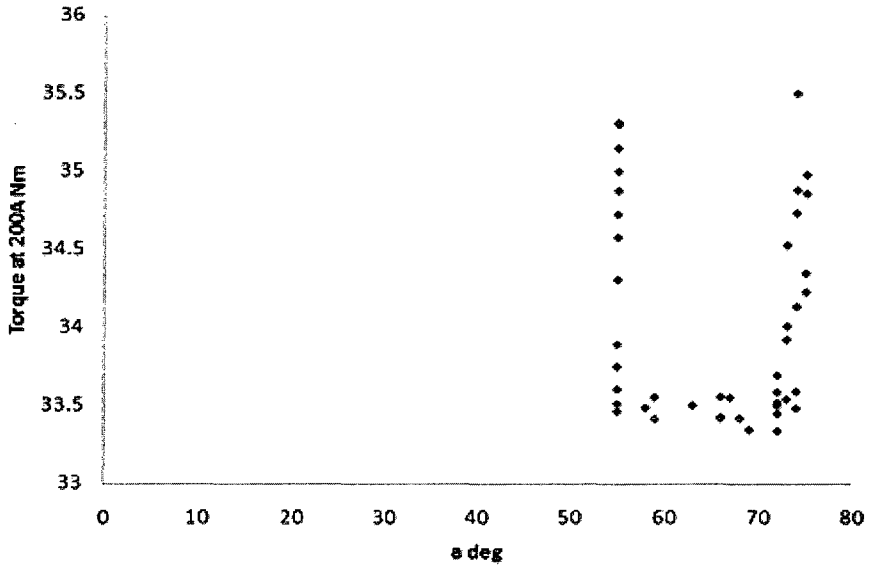

Figure 62: Effect of 'a' on the torque at 200A

In this chapter, first we discussed how to model the IPM in a Finite Element Software, and then designed a simulated prototype matching the given IPM in order to figure out all the required dimensions and other essential information. Next, we conducted an initial analysis to determine which parameters affect torque. Finally, we implemented the optimizer on the built IPM with our tested algorithm mERR-MOEA to achieve the best design and then examined the effect of each parameter, summarized in Table 8 and Table 9, on the final choice of a single Pareto optimal design to be built.

Case	a (deg)	W_m (mm)	$L_m/2$ (mm)	Conclusion
Circled regions in Figure 55 & Figure 56	Varies from 72 to 74 (torque is dependent on 'a', circled region in Figure 54	Constant at 4.4 (torque is independent of W_m, circled region in Figure 57)	Constant at 21.4 (torque is independent of $L_m/2$, circled region in Figure 58	Torque is dependent on 'a'
Regions in Figure 55 & Figure 56 highlighted by rectangles	Constant at 55 (torque is independent of 'a', region highlighted by a rectangle in Figure 54	Varies from 5.3 to 6.3 (torque is nearly independent of W_m, region highlighted by a rectangle in Figure 57)	Varies from 18.7 to 21 (no particular relationship can be deduced, region highlighted by a rectangle in Figure 58	Further investigation needed (refer to Table 9)

Table 8: (Summary-1) Analysis of parameters v/s torque

Case	a (deg)	area (mm*mm)	Conclusion
Circled region in Figure 59	Constant at 55	Varies from 113-122	Torque depends on area
Circled region in Figure 60	Varies from 72-74	Constant at 90	Torque depends on 'a'

Gradient comparisons	Observe the gradient of the plot in Figure 61 when area varies from 113-122	Observe the gradient of the plot in Figure 62 when 'a' varies from 72-74	Area has greater effect than 'a'

Table 9: (Summary-2) Analysis of parameters v/s torque

6 Conclusion

This thesis demonstrates a novel approach to optimizing the torque performance of an IPM and an innovative way to study the relation between the design input parameters and the output. It first performed a theoretical foundation of the parameterized torque equation thereby providing an initial idea of the various parameters that affect the torque. It then justified a case for the requirement of a specialized algorithm that can tackle optimization of electromagnetic devices by first reviewing some common MOEA algorithms and based on the ideas gathered from the various drawbacks of these algorithms, like inability to store all information, impractical level of accuracy, inability to be unbiased, inability to come out of deceptive front, not simplistic enough, dependency on fitness sharing, etc., formulated a novel algorithm, the mERR-MOEA, based on elitism, restricted resolution, archiving, dynamic population and progress measured. The thesis was tested on various aspects, such as diversity, deceptive front, convex and non-convex front, uniformity and a real physical device such as the SMES. The results allow us to conclude that mERR-MOEA is a robust and efficient algorithm when compared to others such as the NSGA, the SPEA, etc. for the kind of problem we intended to optimize.

The thesis also provided an insight into the modeling aspects of the IPM in a commercial Finite Element Software, MagNet. It demonstrated the process of optimization by first building a matching prototype IPM in order to deduce the missing information about the reference IPM, and then running the optimizer in order to derive from the Pareto solutions the best design based on an acceptable error range and least cogging torque. Finally it presented the curious reader a comprehensive analysis of the relationship between the input parameters and the output, highlighted at the start in Chapter 2 but could not be comprehended, with the help of the information obtained from

optimization and concluded that the torque is affected by the area and the spread angle of the magnet with the volume having higher effect. It has also been shown that the development of the Pareto front allows a designer to examine the effects of the various design parameters on the final 'optimal' design to be built (which was the original goal of this work).

6.1 Future Research

1. In Chapter 5 we had discussed the importance of preserving information in order to avoid unnecessary computation of duplicate individuals due to the time consuming nature of field computations in standard single processor machines. Algorithms that can save such time are considered efficient; however, they still have to conduct field computations for every desired individual. Innovative approximation techniques such as the response surface method can cut down the computational time enormously. By representing the computational domain in the functional space as an approximate analytical function by schemes like the *Method of Least Squares*, with the help of the available information about the constraints and the variables, the response surface method is applied. In other words, the optimization process starts with FEA but soon transits into the analytical form of the front thereby saving an enormous amount of computational time. However, response surface methods are not always accurate and therefore require further research.
2. While optimizing, we had simulated an artificial condition of running the IPM based on torque-current analysis instead of letting the supply be controlled by the external inverter circuit. Having said that, the obvious next step would be to build an inverter controlled IPM coupled with the optimizer as that would lead to the testing of the optimizer on a complete variable speed drive system.

Appendix A – Basic Definitions

Chromosome: An entire vector of input parameters can be called a chromosome. For example, any optimization problem, single or multi-objective, will have certain parameters or variables that are to be optimized. The function $f(x,y)$ has parameters x & y, therefore a chromosome is that individual which is made up of x & y. The parameters x & y are known as genes. Therefore a particular chromosome is made up of genes. For example, chromosome=[x,y]. Also, any chromosome is different from another chromosome if at least one of its genes is different.

Cluster Analysis: Each chromosome is treated as a subset or a cluster of the dataset. Subsets having common characteristics such as proximity according to some distance measure are merged with nearby subsets to form a large subset or cluster which can be represented by some trait, for example the centroid of that cluster.

Cogging torque: Owing to the interaction of the permanent magnet harmonics with the air-gap permeance variations due to the stator slots [31], a torque called the "cogging torque" is created. It acts even when the motor is not running. During motor operation it causes ripple and vibration especially at light load and low speeds. It is also known by the name of "ripple torque" when the motor is running. It is desirable to reduce the cogging torque in order to achieve a good design quality. A cogging torque of about 5-6% of the peak operational torque is considered to be a low value [2].

Crossover or **Recombination:** It is a way of reproducing by swapping parts of a gene with another. The new genes formed have the signature characteristics of both the parents. The type of crossover we have used in our algorithm is the single-point binary crossover, and is illustrated below,

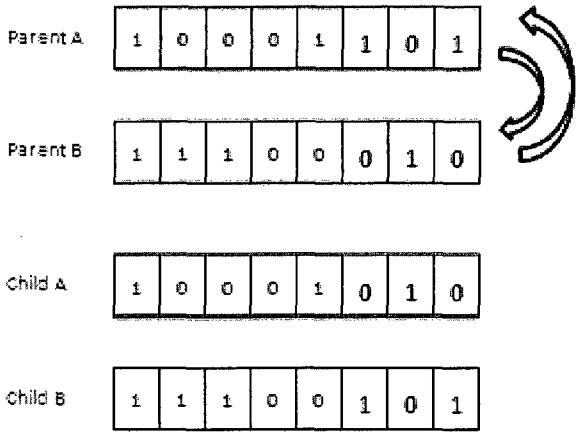

Figure 63: Crossover in GA

The position for swapping is chosen randomly.

Elitism: It is a phenomenon followed by nature that defines the strategy that the good individuals will produce good children more frequently than others which means that the greater the number of good individuals in the mating pool the better the chances of converging fast. Also, by preserving the non-dominated solutions of a particular generation and making them available to take part in future generations, elitism is achieved.

Finite Element Analysis (FEA) [32] is a numerical procedure for obtaining solutions to boundary-value problems which are represented by differential equations together with the conditions (such as Neumann or Dirichlet condition [32]) applied on the boundary that encloses the computational domain. In general, the procedure is to replace a complex continuous surface or domain by small sub-domains or elements which can be represented by simple functions called basis functions. Thus, the original boundary-value problem with an infinite number of degrees of freedom is replaced by a problem having a finite number of degrees of freedom, or in other words, the solution of the entire surface or domain is approximated by a finite pre-determined number of

unknown coefficients. Then, a system of algebraic equations is obtained by applying the Ritz or Galerkin procedure[32] and, finally, a solution of the boundary-value problem is achieved by solving the system of equations. The general steps in FEA can be summarized as,

1. Discretization or subdivision of the domain into elements (elements can be of any shape such as trianglar, quadrilateral or tetrahedral depending on the order of the basis-functions and type of analysis i.e 2D or 3D.
2. Selection of the basis-function formulation of the system of equations (S & T matrix assembly based on the method such as Galerkin or Ritz).
3. Solving the system of equations.

Fitness: It is a measure of the quality of an individual expressed in terms of a real number. It is a tool to classify or rank a population. In SPEA, fitness assignment for the non-dominated individuals is different from that of the main population. While classifying the non-dominated individuals, the chromosomes are assigned a value called the "strength" which is calculated on the basis of how many chromosomes of the dominated set are being dominated, say n, by a chromosome of the non-dominated set, divided by the total number of chromosomes, say N, in the main population plus one i.e. $s_i = f_i = \frac{n}{N+1}$. In the case of non-dominated individuals the fitness and the strength are same. Fitness for the dominated ones in SPEA is calculated by summing the strengths of all the external non-dominated individuals plus one i.e. $f_i = 1 + \sum s_i$.

Function or Objective space: A set of all possible values of the objectives of a problem achieved by evaluating the objectives as a function of the parameter space. In the multi-objective context the Pareto front is defined in the function space.

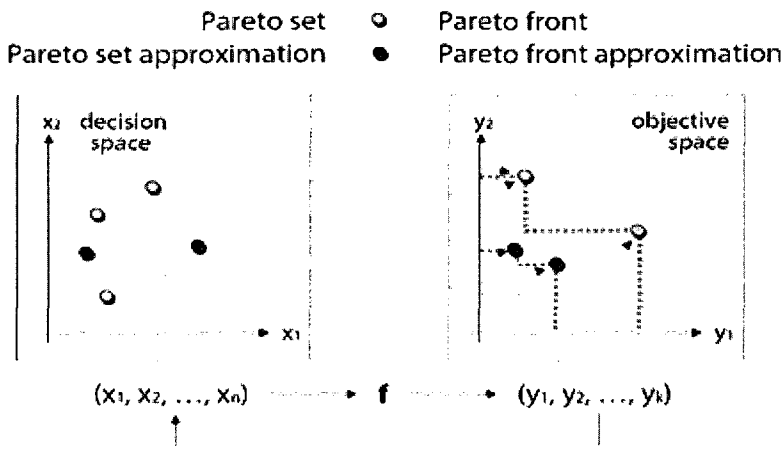

Figure 64: Relationship between decision space and function space [16]

Gene: The encoded version of the variables or parameters of a problem. For example, Gene-x = 10101101.

Individual: A chromosome can also be known as an individual.

Method of Least Squares: It is a mathematical procedure used to fit a given set of points along a curve by minimizing the sum of the squares of the distances between the points and the model curve function.

Mutation: It is one of the operators used in genetic algorithms for creating a new individual from an existing one. Changing only one of the bits of the existing individual represented in a binary format leads to a new individual. The one illustrated below has been applied in our algorithm and is known as binary mutation.

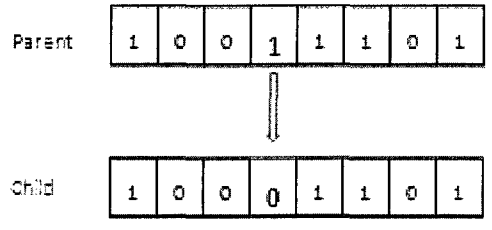

Figure 65: Mutation in GA

Through mutation a particular search operation can get out of local maxima or minima.

The bit that gets changed is randomly chosen, as a result of which, the new individual, can be anywhere in the search space i.e. it can be far-off from the parent or can be nearby.

Parameter or decision space: A set of all possible values of the parameters in a problem.

Population: A collection of one or more chromosomes make a population.

Rank Selection: The individuals are assigned selection probabilities based on their ranks determined by their strength or fitness.

Search Space: A set of all possible solutions to a problem is known as a search space.

Selection: It is a term that defines the different strategies involved in choosing the chromosomes for reproduction. Selection is performed after the population has been ranked or in other words, after the population has been classified. It is one of the main aspects of a multi-objective evolutionary search algorithm since, based on how selection is done, the quality of the Pareto front is determined. Selection procedures such as tournament selection, rank selection, etc. are commonly used.

Stochastic or Non-deterministic: It is a random process that deals with more than one possibility of how the process might evolve with time. The future evolution cannot be determined beforehand although the initial or starting condition might be known.

Tournament Selection: A set of individuals are chosen randomly and the best individual decided by its fitness or strength gets the opportunity to reproduce.

Appendix B – Parameterization of the IPM for automation

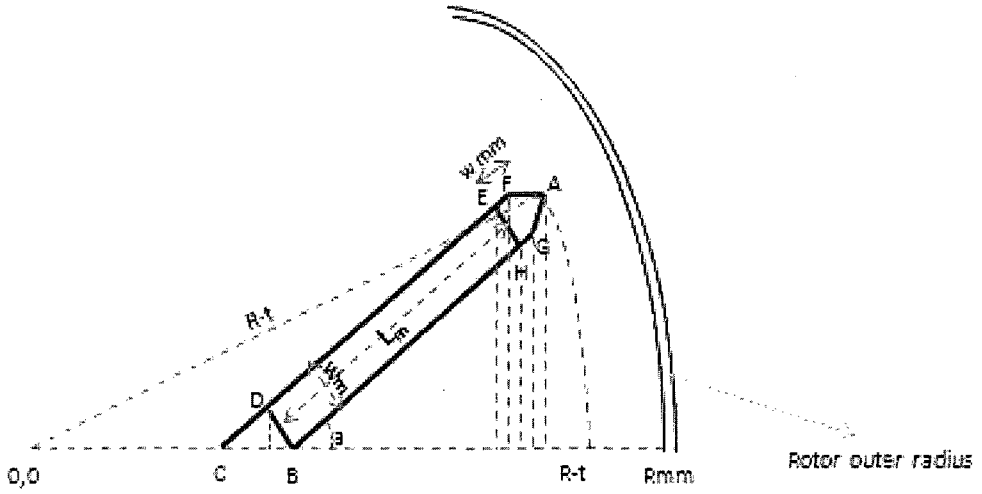

Figure 66: Parameterization of the permanent magnet for automation

$$A_x = (R-t)\cos\left\{sin^{-1}\frac{\left(\frac{w_m}{2}\tan(90-a)+L_m+w\right)sina}{R-t}\right\}$$

$$A_y = \frac{w_m}{2}\tan(90-a)+L_m+w)sina$$

$$B_x = A_x - \left(\frac{w_m}{2}\tan(90-a)+L_m+w\right)cosa + \frac{w_m}{2}coseca$$

$$B_y = 0$$

$$C_x = B_x - w_m coseca$$

$$C_y = 0$$

$$D_x = B_x - w_m\cos(90-a)$$

$$D_y = w_m\sin(90-a)$$

$$E_x = C_x + (w_m\tan(90-a)+L_m)cosa$$

$$E_y = (w_m\tan(90-a)+L_m)sina$$

$$F_x = C_x + (w_m\tan(90-a)+L_m+w)cosa$$

$$F_y = (w_m\tan(90-a)+L_m+w)sina$$

$$G_x = B_x + (L_m+w)cosa$$

$$G_y = (L_m + w)sin$$

$$H_x = B_x + L_m cosa$$

$$H_y = L_m sina$$

where t is the bridge width which is constant in our optimization case. The above figure is one half of the magnet, the other half being a mirror image about the x-axis. Since we were concerned with the dimensions and the position of the magnet for our optimization purpose, the stator and other remaining parts of the frame such as the air gap, coil slot width, etc. were held to be of constant dimensions as given in Table 5.

References

[1]Toyota Prius. *Wikipedia.* [Online] [Cited: June 3, 2008.] http://en.wikipedia.org/wiki/Toyota_Prius.

[2] Staunton, R. H., et al., *PM Motor Parametric Design Analyses for a Hybrid Electric Vehicle Traction Drive Application.* US Department of Energy, Oak Ridge National Laboratory. September 2004. ORNL/TM-2004/217.

[3] Bianchi, N. and Bolognani, S., "Design Optimization of Electric Motors by Genetic Algorithms." IEE Proceedings - Electric Power Applications, September 1998, Issue 5, Vol. 145, pp. 475-483.

[4] Coello Coello, C. A. and Lamount, G. B., "An Introduction to Multi-Objective Evolutionary Algorithms and their Applications." *Applications of Multi-objective Evolutionary Algorithms.* s.l. : World Scientific Publishing Company, 2004, Vol. 1, Chapter 1, pp. 1-23.

[5] Fonseca, C. M. and Fleming, P. J., "Genetic Algorithms for Multiobjective Optmization: Formulation, Discussion and Generalization." San Mateo, CA : Morgan Kaufmann, 1993. Proceedings of the Fifth International Conference. pp. 416-423.

[6] Deb, K., "Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems." Evolutionary Computation, Issue 3, Vol. 7, pp. 205-230.

[7] Coello Coello, C. A., "An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the Art and Future Trends." Proceedings of the Congress on Evolutionary Computation, s.l. : IEEE Press, 1999, Vol. 1, pp. 3-13.

[8] Honda, Y., et al., "Motor Design Considerations and Test Results of an Interior Permanent Magnet Synchronous Motor for Electric Vehicles." New Orleans, LA : s.n., 5-9 October 1997. IEEE Industrial Applications Society,Annual Meeting. Vol. 1, pp. 75-82.

[9] Qinghua, L., Jabbar, M. A. and Khambadkone, A. M., "Design Optimization of Interior

Permanent Magnet Synchronous Motors for Wide-Speed Operation." Indonesia : s.n., 2001. IEEE Power Electronics and Drive Systems. Vol. 2, pp. 475- 478.

[10] Sim, D. J., et al., "Efficiency Optimization of Interior Permanent Magnet Synchronous Motor using Genetic Algorithms." IEEE Transactions on Magnetics, Okayama, Japan : s.n., March 1997, Issue 2, Vol. 33, pp. 1880-1883.

[11] Bianchi, N. and Canova, A., "FEM Analysis and Optimisation Design of an IPM Synchronous Motor." 16-18 April 2002. IEE Power Electronics, Machines and Drives. Vol. Conference Publication No.487, pp. 49-54.

[12] Qinghua, L., Jabbar, M. A. and Khambadkone, A. M., "Design Optimization of Wide-Speed Permanent Magnet Synchronous Motors." 16-18 April 2002. IEE Power Electronics, Machines and Drives. Vol. 487, pp. 404- 408.

[13] Vaez-Zadeh, S. and Ghasemi, A. R., "Design Optimization of Permanent Magnet Synchronous Motors for High Torque Capability and Low Magnet Volume." Electric power systems research, 2005, Issue 2, Vol. 74, pp. 307-313.

[14] Otaduy, P. J. and McKeever, J. W., *Modeling of Reluctance-Assisted PM Motors.* US Department of Energy, Oak Ridge National Laboratory. January, 2006. ORNL/TM-2005/185.

[15] Hendershot, J. R., *Design of Brushless Permanent-Magnet Motors.* Oxford : Magna Physics and Clarendon Press, 1994.

[16] Zitzler, E., "A Tutorial on Evolutionary Multiobjective Optimization." [Power Point Presentation]. Swiss Federal Institute of Technology (ETH) Zurich : s.n., 2002.

[17] Schaffer, J. D., "Multiple Objective Optimization With Vector Evaluated Genetic Algorithms." 1985. Proceedings of the First International Conference on Genetic Algorithms. pp. 93-100.

[18] Deb, K. and Srinivas, N., "Multiobjective Optimization Using Non-Dominated Sorting in Genetic Algorithm." Evolutionary Computation, 1994, Issue 3, Vol. 2, pp. 221-248.

[19] Dias, A. H. F. and de Vasconcelos, J. A., "Multiobjective Genetic Algorithms Applied to Solve Optimization Problems." IEEE Transactions on Magnetics, Evian, France : s.n., March 2002, Issue 2, Vol. 38, pp. 1133-1136.

[20] Horn, J., Nafpliotis, N. and Goldberg, D. E., "A Niched Pareto Genetic Algorithm for Multiobjective Optimization." New Jersey : s.n., June 1994. Proceedings of the First IEEE Conference on Evolutionary Algorithm, IEEE World Congress on Computational Intelligence . Vol. 1, pp. 82-87.

[21] Zitzler, E. and Thiele, L., "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach." IEEE Transactions on Evolutionary Computation, November 1999, Issue 4, Vol. 3, pp. 257-271.

[22] Van Veldhuizen, D. A. and Lamont, G. B., "Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art." Evolutionary Computation, 2000, Issue 2, Vol. 8, pp. 125-147.

[23] Zitzler, E., Deb, K. and Thiele, L., "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results." Evolutionary Computation, 2000, Issue 2, Vol. 8, pp. 173-195.

[24] Coello Coello, C. A. and Pulido, G. T., "A Micro-Genetic Algorithm for Multiobjective Optimization." s.l. : Springer-Verlag, 2001. First International Conference on Evolutionary Multi-Criterion Optimization. Vol. 1993/2001, pp. 126-140.

[25] Tan, K. C., Lee, T. H. and Khor, E. F., "Evolutionary Algorithms with Dynamic Population Size and Local Exploration for Multiobjective Optimization." IEEE Transactions on Evolutionary Computation, December 2001, Issue 6, Vol. 5, pp. 565-588.

[26] Wolpert, D. H. and Macready, W. G., "No free lunch theorems for optimization." IEEE Transactions on Evolutionary Computation, April 1997, Issue 1, Vol. 1, pp. 67-82.

[27] Campelo, F., et al., "A Clonal Selection Algorithm for Optimization in Electromagnetics." IEEE Transactions on Magnetics, May 2005, Issue 5, Vol. 41, pp.

1736-1739.

[28] Guimaraes, F. G., et al., "A Multiobjective Proposal for the TEAM Benchmark Problem 22." IEEE Transactions on Magnetics, Shenyang, China : s.n., April 2006, Issue 4, Vol. 42, pp. 1471-1474.

[29] Cavaliere, V., et al., "Robust Design of High Field Magnets through Monte Carlo Analysis." COMPEL, 2003, Issue 3, Vol. 22, pp. 589-602.

[30] , Global Quantities-Force Maxwell Stress. *Infolytica Corporation.* [Online] [Cited: April 9, 2008.] http://www.infolytica.com.

[31] Zhu, Z. Q., et al., "Reduction of Cogging Torque in Interior-Magnet Brushless Machines." IEEE Transactions on Magnetics, September 2003, Issue 5, Vol. 39, pp. 3238 - 3240.

[32] Jin, J., *The Finite Element Method in Electromagnetics.* 2nd. s.l. : John Wiley & Sons, 2002, pp. 30-31.