UNIVERSITÄT STUTTGART

INSTITUT FÜR STATIK UND DYNAMIK DER
LUFT- UND RAUMFAHRTKONSTRUKTIONEN
PROFESSOR DR.-ING. BERND-H. KRÖPLIN

MASTER THESIS

Soundappan Ramanathan

Knowledge Incorporation into Evolutionary Algorithms to Speed up Aerodynamic Design Optimizations

July 2009

PFAFFENWALDRING 27, 70569 STUTTGART

Knowledge Incorporation into Evolutionary Algorithms to speed up Aerodynamic Design Optimizations

Soundappan Ramanathan

Institute of Statics and Dynamics of Aerospace Structures

University of Stuttgart, Germany

A thesis submitted for the degree of

Master of Science in Computational Mechanics of Materials and Structures

Supervised by

Dipl.-Ing.Lars Graening, HRI-EU.

PD Dr.-Ing. Habil. Stephan Rudolph, ISD.

January 2009 - July 2009

Declaration

I hereby declare that the efforts of my master's thesis are done independently and this thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of this thesis.

Stuttgart, July 27, 2009,

Soundappan Ramanathan

Acknowledgements

Although there is a single name on the cover of this thesis, it would not be accomplished in its current form without the help of many individuals. The efforts of this thesis work carried out between January - July 2009 at “Institut für Statik und Dynamik der Luft-und Raumfahrtkonstruktione”(ISD), university of Stuttgart in collaboration with Honda Research Institute, Germany (HRI-EU).

Firstly, I would like to thank PD Dr.-Ing. habil. Stephan Rudolph, for giving me the opportunity to do my Master Thesis at ISD. I also thank all the professors from the COMMAS program for sharing their knowledge and providing me the opportunity to learn. I also would like to thank Dipl.-Ing. Daniele Rosato, our course director for his extended support in all the possible ways throughout the study period.

I would like to express my sincere thanks to Prof. Bernhard Sendhoff for providing me an interesting task for the thesis along with a vibrant atmosphere to work in his team. I also would like to thank Dr. Markus Olhofer for the valuable suggestions and guidance during my thesis work at HRI-EU.

This thesis could not have been written without Lars Graening, my supervisor at HRI-EU who not only gave me impassioned and intensive guidance but also encouraged and challenged me throughout the period. The tireless motivation from Lars always led to constructive discussions and fueled the desire for achieving the best results in me. Also my special thanks to all the scientists of EL-Tec (Evolutionary and Learning Technology) team and the fellow students at HRI-EU for helping me out in various ways with the stimulating discussions.

Finally, I would like to thank my family members and my colleagues who have been a constant source of support and encouragement in all respects during my stay in Germany.

Abstract

This work deals with making the search process with Evolution Strategy with Covariance Matrix Adaptation (CMA-ES) more efficient. Methods are proposed to initialise the optimisation with predefined knowledge instead using a random initialisation in the context of aerodynamic design optimisation. In aerodynamics, the search for better shape is always in research. Conventional numerical optimisation processes results in huge amount of data sets as a result of each optimisation. But most often only the best designs are taken into consideration. Graening, (2) developed a framework to extract the meaningful information about the shape from those data sets. This approach extends by means of developing a common framework to incorporate the knowledge extracted from the designs, so that the search process can be initialised in order to achieve the new outperforming designs.

This thesis is split into two parts: (i) Development of frame work for quantifying interaction effects between the design and the performance. (ii) Developing an optimal strategy to initialise the search process. Firstly, for quantifying interaction effects linear as well as non-linear interaction effects are analysed. Statistical techniques such as multiple regression and information theory are applied to quantify the parameter interaction effects. Second part deals with developing a common framework to initialise the optimisation by means of incorporating the knowledge into the covariance matrix. The developed framework is applied to 2D Gas turbine blade optimisation for the validation of the knowledge incorporation technique.

Contents

List of Figures

Chapter 1

Introduction

'Evolution is an optimisation process -Ernst W. Mayr. '

With the increased availability and affordability of high speed computers in the recent decades, evolutionary computational techniques emerged successfully as a potential tool in solving broad range of practical problems in optimisation, machine learning and design. These techniques include *genetic algorithms(GAs), evolution strategies(ESs), and evolutionary programming(EPs).* As the Evolution provides inspiration for computing the solutions to problems that have previously appeared intractable, in recent years along with parallel machines the activity of research on evolutionary optimisation has been increased. This work Primarily deals with evolutionary strategies that have been initially introduced by Rechenberg (1965, 1973) and Schwefel (1965, 1977) at the Technical University of Berlin and in particular one of its variant called *evolution strategy with covariance matrix adaptation* (CMA-ES) (1), as it provides good convergence rate even for small population sizes.

Optimisation, *the act of achieving best results under any circumstances* is always a part of every product design process. There exists, huge amount of datasets as a result of any design optimisation process. Graening (2) proposed a method to extract useful knowledge from those datasets. Integration of this knowledge into the optimisation process will certainly lead to a more efficient design process. Thus the motive of this work is to incorporate the priori knowledge acquired from existing designs which are the results of previous manual or computational optimisations into the algorithm. More precisely, this work is aiming towards: *finding an optimal shape of the given geometry regarding its aerodynamic performance by setting up the strategy parameters of the evolutionary design optimisation run.*

1.1 Motivation

1.1.1 Search for an Optimal Shape

Shape of the design plays a vital role in any design process as most of the key design variables that control performance are related to shape. In many complex engineering design problems, particularly in aerodynamics, (optimisation of aircraft wing cross sections,Turbine Blade Design optimisation, etc..) traditionally an approach of numerical optimisation models are applied to increase the performance of the design. In this High-fidelity analysis models are employed in which each function evaluation requires Computational Fluid Dynamics (CFD) simulation which always has the huge economic impact due to enormous amount of computing time using parallel machines. This numerical approach typically involves many repetitive function calls to the high-fidelity analysis codes to obtain the merits of the different combinations of design variables. These methods may reach global optimum if the objective function and the constraints are differentiable and convex. But in most real world optimisation problems such as, non-convex problems obtaining the global optimum with the traditional methods is a tedious task.

Alternatively, stochastic evolutionary design optimisation models like *ES, GAs* have shown considerable success in locating the global optimum even for non-convex and continuous domain tasks with high robustness. However, due to stochastic components and the need for large number of evaluations, this search strategy is always more expensive with respect to computational resources. Recent researches in evolution strategies use the self adaptation techniques like *covariance matrix adaptation*, where the strategy parameters of the optimisation strategy are object to the optimisation as well. This shows the advantage of combining evolutionary design optimisation with high fidelity simulations. The Basic approach will be replacing the high fidelity simulation codes with the miniature model such as Target Shape Design, as described in (3) and (4) along with the covariance matrix adaptation. Cumulative step size adaptation, an inspired property of CMA-ES which enables the extraction of information from past generation, fastens the adaptation of the strategy parameters, which inturn results in high convergence rate even at a small population size.

1.1.2 Search for an Optimal Strategy

Setting various parameters of an evolutionary algorithm such as *population size, probability of mutation, etc...* is crucial for a good performance of the optimisation algorithm. This work deals with the setting the strategy parameters *as they define the initial search distribution or mutation distribution*, with the knowledge acquired from huge amount of hetrogenous design data during the previous optimisation runs in order to find the optimal shape. As CMA-ES modifies both the objective as well as strategy parameters through only one stochastic source (Covariance Matrix), stochastic influence of the mutation is greatly reduced. Adapting the precise values of these parameters greatly influences search strategy of the algorithm.

Comprehensive knowledge observed (during the numerical optimisation methods) on local differences in design shows the amount of direction of surface modifications. Graening's (2) framework was able to extract the meaningful knowledge on local differences in design. This work extends with developing a framework for adapting them as strategy by reducing the uncertainty about the knowledge extracted and incorporating it into the algorithm so that further optimisation can be initialised in order to achieve optimal solutions. Statistical techniques such as *regression techniques and concepts from information theory* are investigated and adapted to identify and reduce uncertainty about the knowledge.

1.2 Outline

Chapter 2: In this preliminary chapter brief discussion on evolutionary computation and in specific, fundamental mathematical principles which are needed for further explanations are outlined. Therein, the decomposition of covariance matrix and the relation of its geometric interpretation to the multivariate normal distribution are discussed. Furthermore, the evolution strategy with covariance matrix adaptation is described in detail along with the self adaptation in general.

Chapter 3: Main focus of this chapter is to obtain an overview of knowledge incorporation in general and the methods applied for incorporation of the knowledge into evolutionary algorithm. An approach adapted in this thesis for knowledge incorporation for the initialisation of the initial optimisation strategy is also briefly discussed.

Chapter 4: In this section important statistical concepts which are core to this work are briefly explained. Methods for identifying the parameter interaction effects are also explained. Furthermore, the model for quantification of interaction effects using multiple regression and information theory also discussed here.

Chapter 5: In this chapter, a method on incorporating knowledge into CMA-ES is proposed. The framework for setting up the strategy parameters is derived here using linear as well as non-linear interaction effects. Also, different models for initialising the optimisation are discussed.

Chapter 6: This chapter demonstrates the capabilities of incorporated knowledge by applying the described techniques for the optimisation of 2D Gas Turbine Blade. The results are compared in detail in terms of initialisation, adaptation of the strategy and their impact on the convergence rate of the algorithm.

Chapter 7: Finally, the work done in this thesis is summarised and concluded.

Chapter 2

Evolutionary Computation

Nature has been always the best source of inspiration for problem solving. While considering most powerful problem solvers, there are two promising candidates:

(i) The Human Brain (that created “the wheel, New York, wars, and so on ”)

(ii) The evolutionary process (that created the human brain)

Attempts to design the problem solvers inspired by second candidate led to the field of Evolutionary Computation (5). As discussed in previous chapter, there exists several techniques in evolutionary computation. The *Darwinian theory* is the common underlying principle behind them. It can be stated as *given a population of individuals, the environmental pressure causes the natural selection (Survival of the fittest), which causes a rise in the fitness of the population.* Fitness is viewed as adaptation capability to the environment which in technical systems is estimated by defining a suitable quality function as an abstract of the fitness so that some of the better candidates are moved to successive generations in the search for the global optimum. Figure (2.1) represents the general flow of the evolutionary computation (1).

In evolutionary computations, the initial population of μ individuals (parents) are generated randomly. The objective parameters are encoded into the chromosomes of the individuals as real valued parameters. Hence, stochastic components such as mutation and recombination constructs the ancestors that build the offspring population (λ). The evaluation of the individuals in their environment is done by assigning a fitness value to the individuals of the offspring population and based on that the parents for the next generation are calculated. Thus the selection process defines the parents for the next generations. The evolutionary cycle continues till the required criterion is met. The mutation which generates the offsprings (possible solutions) and the quality measure which defines the problem are influential components of the algorithm. As the population extends from the initial set of individuals, that also plays alternatively an important role, when

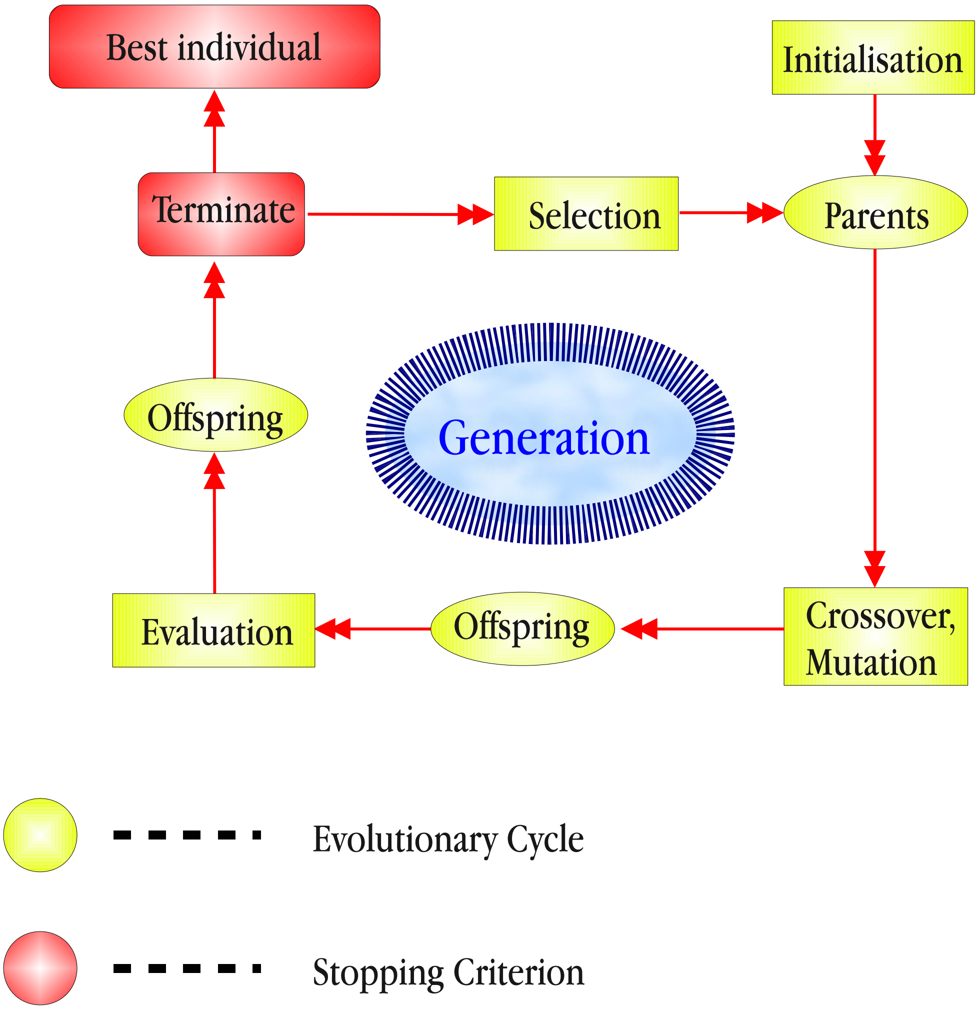


Figure 2.1: Picture depicting the general flow of the evolutionary optimisation.

the priori information about the the problem is given.

This chapter further proceeds with the description based on important mathematical concepts and definitions which are required for the subsequent treatments, primarily in the context of CMA-ES. These include, the decomposition of covariance models, the gaussian distribution to adapt the covariance model for the continuous parameter optimization. Furthermore, theory behind CMA-ES and adaptation procedure to find the optimum are also explained in broader context.

2.1 Mathematical Preliminaries

2.1.1 Covariance Matrix

Covariance indicates the measure of strength of correlation between two or more random variables. The sample covariance for the two random variables X and Y each with a sample size of N is defined by the expectation value,

$$Cov(X,Y) = \Sigma_{i=1}^{N} \frac{(x_i - \overline{x_i})(y_i - \overline{y_i})}{N} \tag{2.1}$$

$Cov(X,Y) = 0$, is said to be uncorrelated. Positive covariance ($Cov(X,Y) > 0$), indicates that X and Y are proportional. While Considering $X = Y$,

$$Cov(X,Y) = \sigma_x^2$$
$$Cov(X,Y) = Variance(x)$$

in correlation terms,

$$r = \frac{cov(X,Y)}{\sigma_x \sigma_y}$$
$$r^2 = \frac{\sigma_{xy}}{\sqrt{\sigma_{xx}\sigma_{yy}}} \tag{2.2}$$

Hence the elements of the covariance matrix are the variances and covariances of the parameter.

2.1.2 Eigen Decomposition of Covariance Matrix

Positive Definite Matrix has a particularly simple expression for a class of matrices often used in multivariate analysis such as correlation, covariance, or cross-product matrices. A matrix $\mathbf{C} \in \mathbf{R}^{n \times n}$ is said to be positive semi-definite when for all non-zero vectors X, as $\mathbf{XX^T} > 0$

$$\mathbf{C{=}XCX^T} \tag{2.3}$$

This implies that a positive-definite matrix is always symmetric.

The eigen-decomposition plays a vital role as it is used to find the maximum (or minimum) of functions involving these matrices. Importantly, the positive semi-definite matrix $\mathbf{C}$ is characterized by the following properties :

1. Eigenvalues are always positive or null. $d_1 \ldots d_n > 0$.
2. Eigenvectors are pairwise orthogonal when their eigenvalues are different. $\mathbf{Q} = [\underline{q}_1 \cdots \underline{q}_n]$.

Because eigenvectors corresponding to different eigenvalues are orthogonal, it is possible to store all the eigenvectors in an orthogonal matrix $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$.

This implies the following equality:

$$\mathbf{Q}^{-1} = \mathbf{Q}^T$$

Therefore one can express eigen decomposition of the matrix $\mathbf{C}$ as follows,

$$\mathbf{C}{=}\mathbf{Q}\mathbf{D}^2\mathbf{Q}^T \tag{2.4}$$

where,
$\mathbf{Q}$ is an orthogonal matrix, ($\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^T\mathbf{Q} = \mathbf{I}$). Columns of $\mathbf{Q}$ forms the orthonormal basis of eigenvectors.
$\mathbf{D} = diag(d_1 \ldots d_n)$ is a diagonal matrix with square roots of the eigenvalues of $\mathbf{C}$ as the diagonal elements.

Similarly eigen decomposition of the $\mathbf{C}^{-1}$ can be computed as follows,

$$\mathbf{C}^{-1} = (\mathbf{Q}\mathbf{D}^2\mathbf{Q}^T)^{-1} = (\mathbf{Q}\mathbf{D}^{-2}\mathbf{Q}^T) = Qdiag\left(\frac{1}{d_1^2} \cdots \frac{1}{d_n^2}\right)\mathbf{Q}^T \tag{2.5}$$

from the above, we can further derive

$$\mathbf{C}^{\frac{1}{2}} = \mathbf{Q}\mathbf{D}\mathbf{Q}^T \tag{2.6}$$

and the inverse,

$$\mathbf{C}^{-\frac{1}{2}} = \mathbf{Q}\mathbf{D}^{-1}\mathbf{Q}^T \tag{2.7}$$

Covariance Matrix ($\mathbf{C}$) used in the algorithm is a symmetric positive definite matrix, $C \in \mathbf{R}^{n \times n}$ for all $x \in R^n$.

2.1.3 Multivariate Normal Distribution

The multivariate normal distribution also known as gaussian normal distribution is the generalization of $1D$ normal distribution to N Dimensions. The mean value is the modal value that corresponds to the distribution mean $\underline{m} \in \mathbf{R}$.

Multi-variate normal distribution $\mathcal{N}(\underline{m}, \mathbf{C})$ is estimated from the mean $\underline{m} \in \mathbf{R}^n$ covariance matrix $\mathbf{C}$. This can be shown as,

$$\mathcal{N}(\underline{m}, \mathbf{C}) = \frac{1}{(2\pi)^{N/2}|\mathbf{C}|^{1/2}} \exp\left(-\tfrac{1}{2}(\underline{x} - \underline{X})^\top \mathbf{C}^{-1}(\underline{X} - \underline{m})\right)$$

whereas,

The covariance matrix describes the shape of the distribution. This distribution is explained with its Geometrical interpretation into ellipsoids($X \in \mathbf{R} \mid X^T\mathbf{C}^{-1}X = 1$). The squared axes lengths of the ellipsoids relates to the eigenvalues and direction is defined by the eigenvectors vectors of the Covariance Matrix (2.2).

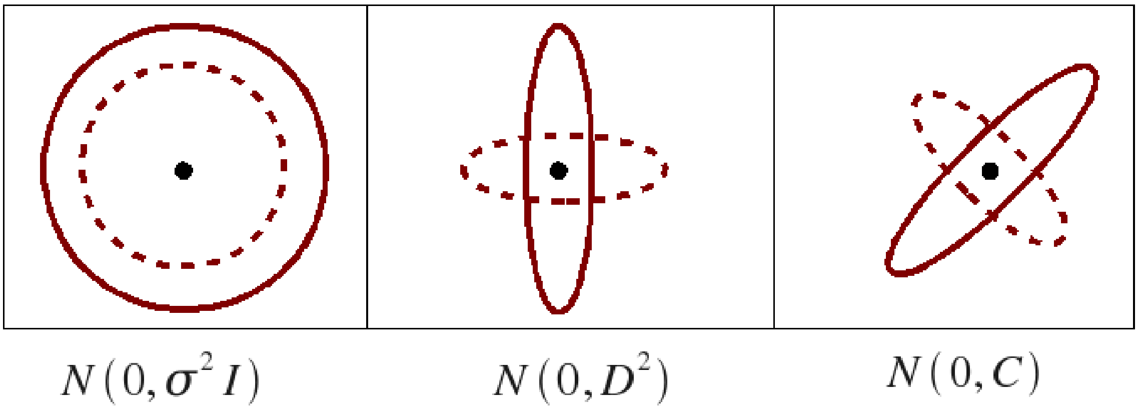


Figure 2.2: Multivariate Normal Distribution - modes of estimation

where as,

(1) $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, represents the distribution $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. The Identity produces the isotropic distribution.

(2) $\mathcal{N}(\mathbf{0}, \mathbf{D^2})$, represents the distribution $\mathcal{N}(\mathbf{0}, \mathbf{D}^2)$ The Diagonal matrix (eigenvalues) $\mathbf{D}$ scales the the spherical distribution with the coordinate axis (axis parallel oriented ellipsoid).

(3) $\mathcal{N}(\mathbf{0}, \mathbf{C})$, represents the distribution $\mathcal{N}(\mathbf{0}, \mathbf{C})$ (i.e with the full rank covariance matrix $\mathbf{C} = \mathbf{QDQ}^T$) $\mathbf{QQ}^T$ defines the orientation for the ellipsoid with $\frac{n^2-n}{n}$ degrees of freedom where n

2.2 The CMA Evolution Strategy

Self Adaptation of the strategy parameters during the search process, the motivation behind the application of CMA-ES for the optimisation problem as it enables the use of the information *possible solutions in the form of individuals* from the past generation. Strategy parameters are defined in the form covariance matrix which defines the multi variate normal distribution for the mutation operator. Self adaptation reduces the complexity in the dimension of the search process and enables faster adaptation of strategy parameters resulting in higher convergence speed of the optimisation algorithm.

2.2.1 Self Adaptation

Self adaptivity means strategy parameters of the algorithm are varied during the run in certain manner: these parameters are included in the chromosomes and coevolve with the solutions.

With the estimation of covariance matrix from the distribution, the search process initialised with the mutation of objective parameters. This enables the realisation of the correlated mutations and direction of the search space i.e. enables the rotation of the search space. In order to proceed, the adaptation of the covariance matrix is based on the population and the path of the evolution.

Thus the estimation of covariance matrix is done in two steps,

(i) adaptation of the estimated whole covariance matrix from the distribution i.e. self adaptation

(ii) observation of the evolution path.

Continuous parameter optimization *i.e.the search process* $\mathbf{R}^n \rightarrow \mathbf{R}$ begins with estimating the initial set of population of individuals (objective parameters). Then sampling them by multivariate normal distribution with the zero mean. The cycle proceeds with constructing the offsprings from the parents. *('In biological terms 'genotype-phenotype mapping').* Thus the next point in the search space is estimated as follows,

$$\underline{X}_k^{g+1} \sim \underline{m}^g + \sigma^g \mathcal{N}(0, \mathbf{C}^g) - k = 1 \dots \lambda \tag{2.8}$$

where,

g is the generation number $0, \dots, n$, the operator $\sim$ denotes the same distribution both sides of the formula. $\mathcal{N}(0, \mathbf{C}^g)$, denotes the multivariate normal distribution, $\underline{X}_k^{g+1} \in \mathbf{R}^n$, k-th offspring for the successive generations, $m^g \in \mathbf{R}^n$, mean value of the search distribution at generation g and $\sigma^g \in \mathbf{R}_+$, is the "overall" standard deviation, step size or the global step size at generation g. The number of offspring $\lambda \geq 2$.

Hansen and ostermeir (6) considers by choosing the distribution with zero mean value, parents are regarded as the best approximation to the optimum known so far and describes the distribution with "non-zero mean value", as extrapolation *i.e.moving the population from one parametric space to another* also the effect of extrapolation are small.

In order to move the search to the next generation, the algorithm extends in estimating move operator (distribution mean) $\underline{m}^{g+1}$, the covariance matrix $\mathbf{C}^{g+1}$, and the global step size σ^{g+1}.

2.2.2 Selection and Recombination

The move operator or mean of the next search distribution $\mathbf{g}^{+1}$) is given as the weighted average of $w_i = \frac{1}{\mu}$ be the selected parent.

Let $x_{i:\lambda}$ be the i-th ranked solution, such that $f(x_{i:\lambda} f x_{\lambda:\lambda})$

$$\underline{m}^{g+1} = \Sigma_{i=1}^{\mu} w_i x_{i:\lambda}^{g+1} \tag{2.9}$$

The survivor selection *i.e survivor of the fittest* is deterministic. It is only based on fitness rankings. The Survivors having the fitness value with closer proximity to existing parents are only allowed to become the parent for the next generation. Selection of new parents, μ, is done in two ways,

– selection of individuals from the offsprings the best of them becomes the parent of the next generation while the current parent is always disregarded. This is termed as μ, λ *or Non-Elitist strategy.*

– selection of individuals from the union set of parents and offsprings, (i.e. the generated offsprings competes with the parents). This is termed as, $\mu + \lambda$ *or Elitist strategy.*

Recombination scheme involves two parents that create one child. To obtain λ offsprings, the recombination is performed λ times. CMA-ES adapts intermediate recombination scheme by averaging the values of the parent alleles. Generally, the recombination scheme is estimated as $\frac{\lambda}{2}$.

2.2.3 Covariance Matrix Adaptation

The covariance matrix from the distribution mean and the selected parents from the survivor population is estimated by,

$$C_{\mu}^{g+1} = \Sigma_{i=1}^{\mu} w_i ({x_i}^{g+1} - \underline{m}^{g+1})({x_i}^{g+1} - \underline{m}^{g+1})^T \tag{2.10}$$

where, C_{μ}^{g+1} is an estimator for the distribution of *i.e. successful steps*.

In order to achieve good convergence speed, the population size λ has to be small which in turn causes trouble in calculating the reliable estimator for a good covariance matrix. The inclusion of information from the previous generations for the estimation of the covariance matrix helps to overcome this problem while retaining the convergence speed even at small population sizes. With the inclusion of information from the certain no.of previous generations, the mean of the estimated covariance matrices from all generations are calculated as follows,

$$C^{g+1} = \frac{1}{g+1} \Sigma_{i=0}^{g} \frac{1}{\sigma_i^2} C_{\mu}^{i+1} \tag{2.11}$$

CMA-ES initialises the search process with the covariance matrix as unity ($C^0 = I$).

Figure 2.3 shows how the distribution changes with the update of the covariance matrix. In the left the ellipsoid represents the initial covariance matrix. Ellipsoids in the middle and the right side shows the change in search direction and the distribution as a result covariance matrix updation.

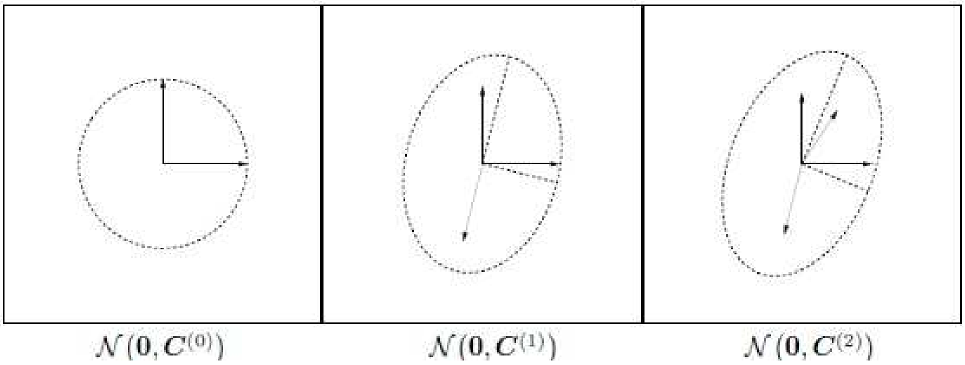

$\mathcal{N}(\mathbf{0}, \boldsymbol{C}^{(0)})$
$\mathcal{N}(\mathbf{0}, \boldsymbol{C}^{(1)})$
$\mathcal{N}(\mathbf{0}, \boldsymbol{C}^{(2)})$

Figure 2.3: Change in the distribution with the update of Covariance Matrix. The vectors indicate change in direction of the search with the update from each generation.

Thus search process with CMA-ES can be summarized as follows. Initially information about the objective parameters are encoded into parents. Then with the random mutation distribution CMA-ES maps the information to descendants (local optima). The selection for the next generation is done such that the best parents according to objective function or fitness function are ensured as parent for next generation. The estimation of covariance matrix defines the strategy for the offsprings of the next generation which is explained by the geometrical interpretation of the mutation distribution. Self adaptation takes place by with the updation of the learning and evolution steps into the covariance matrix.

The priori knowledge about the search is included as strategy parameters. This enables the intialisation of the search with predefined mutation distribution instead random This will refine and speed-up the search process. Thus this work proceeds with estimating and analysing the initialisation of the search the process with the knowledge at hand.

Chapter 3

An Overview of Knowledge Incorporation

Generally, problems seldom emanate due to the isolation of the system. Approaching it with predefined knowledge or the information about the history of the problem at hand results in further development of the system. In practice, often evoultionary algorithms have been applied where experience and knowledge about the problem available but not used. In such applications, a performance benefit can be achieved by utilising the existing knowledge, provided that knowledge doesn't heavily deviate the search from the generation of optimal solutions. In the context of design optimisation, knowledge of the existing designs are applied to drive further optimisations with the motive of achieving increasingly efficient or optimal designs. This section analyses the different ways to incorporate the knowledge into the evolutionary algorithm.

There exists two approaches to incorporate knowledge into the evolutionary algorithm. The first one adds knowledge as priori before running the algorithm. The later one extracts information from the previous generations and feeds it back into the algorithm. This is described as self-adaptation in CMA-ES. Thus this chapter proceeds with analysing different modes of knowledge incorporation methods. Generally prior knowledge can be incorporated in the following phases of the algorithm. This is summarised in three phases as follows (Fig.3.1). (5)

- Initialisation phase
- Crossover and mutation phase
- Fitness evaluations phase

3.1 Initialisation Phase

The common and obvious approach is to incorporate the knowledge during the initialisation phase. Incorporating the knowledge at this phase can be advanta-

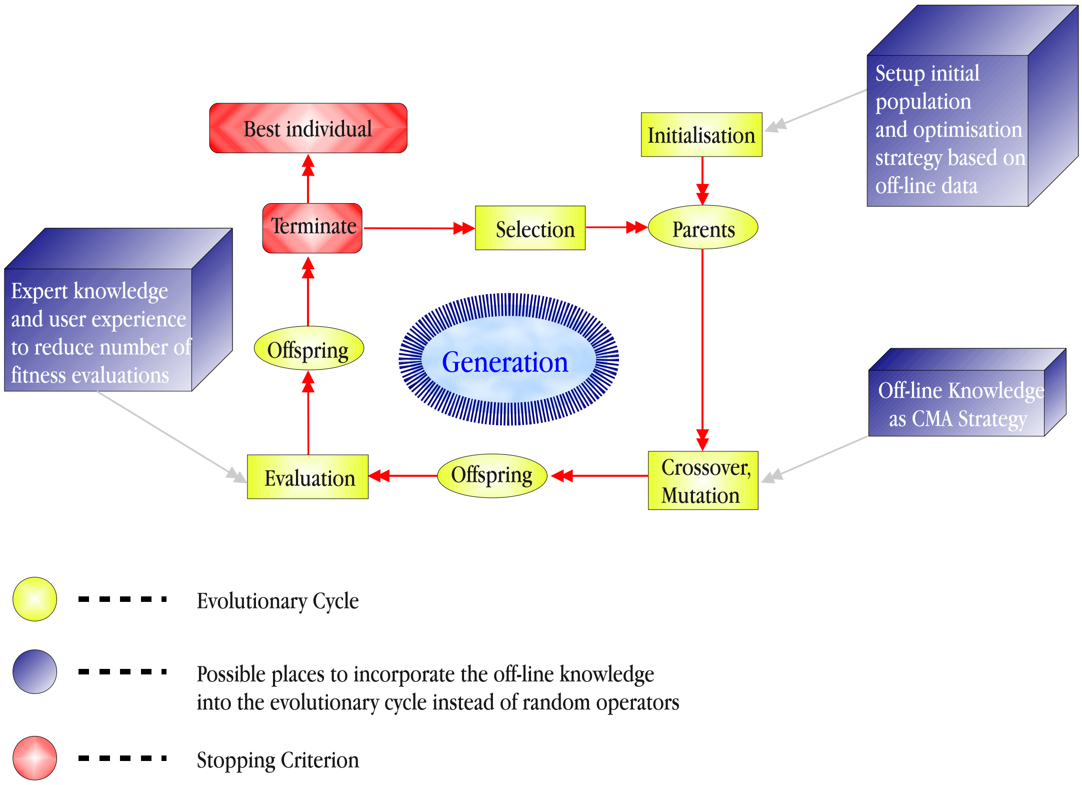

Setup initial population and optimisation strategy based on off-line data
Best individual
Initialisation
Terminate
Selection
Parents
Expert knowledge and user experience to reduce number of fitness evaluations
Offspring
Generation
Off-line Knowledge as CMA Strategy
Evaluation
Offspring
Crossover, Mutation
Evolutionary Cycle
Possible places to incorporate the off-line knowledge into the evolutionary cycle instead of random operators
Stopping Criterion

Figure 3.1: Figure indicating possible places to incorporate knowledge into evolutionary cycle with CMA-ES

geous due to the following considerations,

* Eliminating the need for *reinvention of the wheel* by using existing solutions inturn this also reduces the computational costs by means of increased convergence rate.

* A non-random initial population can route the search in the needed regions of the search space which has the high probability to find the new outperforming solutions.

* Considering as a whole, this deterministic search process with the prior knowledge at the initialisation phase could lead to better results than the the randomly initialised evolutionary search.

In the following some of the existing methods in which initialisation function can be changed from simple random creation (5).

Seeding the population with one or more previously known good solutions arising from other optimisation techniques. These vary from trial and error to the use of greedy constructive heuristics using instance-specific information. Application for this can be found in operations research problems such as *schedule hardest first* on scheduling and planning or *nearest neighbour-heuristics* for Traveling Salesman -like problems.

In **Selective Intialisation** a large number of random solutions are created and considerably a good set of initial population of individuals are choosen to undergo the evolution. This can be done as a series of N *k-way* tournaments instead of selecting the best N from $K.N$ solutions. Alternatively, selecting a set based not only on fitness but also on diversity so as to maximise the coverage of the search space. Executing a local search with each member of the initial population containing a set of points that are locally optimal with respect to some move operator (7).

Applying such methods to identify possibly best solutions then to proceed with the high mutation rate. This is done to generate the individuals in the needed space and direction. With this approach of initialising with the existing knowledge, proportion of the solutions are increased along with the mean performance. However the variance in performance is decreased. It can be understood as there exists no occasional good runs resulting from the EA completely searching entirely new regions of the space and coming up with novel solutions.

3.2 Crossover and Mutation Phase

Alternatively, parallel researchers proposed so-called *intelligent* variation operators to incorporate problem or instance specific knowledge. In a simple case, these might take the form of introducing bias into the operators. for example, a binary coded algorithm is used to select features for use in another classification algorithm, efforts can be done to deviate the search process towards more compact feature sets by using higher probability for mutating from the allele value *use to don't use* instead vice versa. Smith (8), suggests a related approach in which genes encode for microprocessor instructions, which clusters obviously into sets with similar effects. Hence the mutation operator was biased to incorporate this expert knowledge. This incorporation directs the mutations such that it happens more likely between the instructions in the same group instead shuffling between groups.

Another approach which varies slightly from the above uses problem specific instead instance specific knowledge which can be noticed in the modified one-point crossover operator for protein structure prediction as in (9). The authors suggest that heritable features being combined by recombination were folds, or fragments of three-dimensional structures. A property of the problem is that during folding protein structures can be free to rotate about peptide bonds. Effective application of this knowledge by the modified operator by testing explicitly maximum possible different orientations of the two fragments. *This is accomplished by trying all the possible allele values in the gene at the crossover point* in order to find energetically highly favorable ones. When there is no feasible confirmation, different crossover points were selected and the process has been started from the beginning. It can be noticed how the incorporation of local search phase can be incorporated.

On the other side, the operators are modified to incorporate the instance-specific knowledge. This usage of instance-specific knowledge works by maintaining the diversity within the population to restrict the premature convergence. Diversity is maintained by ensuring that the offspring inherits all of the edges common to both parents, but none of the edges that are present in only one parent. The intelligent part of the operator emerging from nearest neighbour heuristic to join together with the inheritants from the parents. This exploits the instance-specific edge length information.

In the context of optimization, use of surrogate models are suggested by Rasheed (10). The author suggests two approaches for setting up and applying the surrogate models to speed up genetic-algorithm based optimisations. In these approaches functional approximations of the fitness function which helps to fasten the search. One method fastens the search by keeping genetic operators more informed. The informed operators (IO) replaces the pure randomness with decisions that are driven by the surrogate models. Evolution operators such as initialisation, mutation and crossover are replaced by informed intialisation, informed mutation and informed crossover respectively.

In **Informed Initialisation**, for generating an individual in the initial population, a number of uniformly random individuals in the design space has been generated and then the best among them is selected according to the surrogate model. The number of random individuals is a parameter of the method with a default value of 20.

In **Informed Mutation**, several random mutations are generated among the base point. Mutation is randomly generated by randomly selecting the proper parameters for the mutation. The mutation which seems best according to the surrogate model is returned as a result of the mutation. The number of random parameter of the method has five as the default value.

Informed Crossover is done by randomly selecting two parents according to the usual selection strategy. These two parents are fixed and will not change as the effect of informed crossover operation. Multiple crossovers are conducted by randomly selecting its internal parameters and applying it to those two parents in order to produce a potential child. Internal parameters are governed by the chosen crossover method. Hence informed mutation is applied to every potential child and result of this informed crossover is the best point as considered in the context of surrogate models.

The other one fastens the search by genetically engineering some of the individuals instead of applying a Darwinian approach. This generates the new individuals using crossover/mutation, with iteration in which individuals are generated by running mini-optimisation using the approximations and returning the best point found therein. The application of informed operators framework to aircraft design optimisations and engineering design domains did consistently produce effective results.

3.3 Fitness Evaluations Phase

Computationally efficient meta models or approximate models find the common use in evolutionary optimisations when the original fitness function is computationally expensive for problems like 3D aerodynamic design optimisations. These surrogate models are applied to assist the algorithm to obtain acceptable solution using a limited number of evaluations based on the expensive original fitness functions and by applying meta-models to fitness functions. These approaches can be interpreted as how the knowledge from the model can be applied to guide the evolutionary search. In these approaches individual and generation based control can be done to ensure that the these approximate fitness functions will converge to the true optimum. Jin (11), suggested a framework for managing these approximate models with the generation or individual based control of the evolution.

The following main constraints must be addressed while applying these models which are based on expert and the user knowledge. The main constraint is to improve the quality of the model with the given limited number of training data. Another one is predicting how often the approximate model should be used which is widely interpreted as model management or evolution control. Dennis (12), suggests a framework for model management based on his investigation with combining approximate models with conventional optimisations. Researches in this direction shows that managing approximate models requires a strong interaction among the optmisation and fidelity of the applied model which is primarily based on the trust-region method. With this, one can be assured that the search process converges to the acceptable solution of the original problem.

Generally, individual or generation based evolution control are applied to combine the approximate models with the original fitness function. With the individual-based evolution control, the approach is that all the individuals are first evaluated using the approximation or surrogate model. Then some of the individuals among them are chosen so that they undergo the re-evaluation using the original fitness function. The following criterias are analysed for the selection of individuals.

Choosing the individuals randomly for re-evaluation considered as not efficient in reducing number of expensive evaluations. Its reasonable to choose the best individuals from previous optimisations which are suggested based on the approximate models which might result in better approximation of the fitness values. Selection of most uncertain individuals, leads to high exploration of the unknown regions in the search space as that space is searched very less. K-means clustering algorithm helps to identify the most representative individuals. This is done by dividing the population by number of clusters and the individuals which are at close proximity to the cluster centres are selected for re-evaluation for each cluster. This has the advantage of good distribution of the number of samples which inturn improves the performance such as high online learning in neural networks. Furthermore, it encourages the exploration of unknown regions because individual which is located far from the current search space usually will get selected for re-evaluation. This selection of most uncertain and the best individuals can also be done by the hybrid strategies.

Generation based evolution control is done by evaluating all the individuals for specified number of generations and also the re-evaluation is done for all the individuals for one or more generations. This generation based control with a fixed frequency of using the original fitness function is not highly preferred. Hence frequency of applying the original fitness function has to be adapted to ensure the effectivity of the algorithm.

However, with the incorporation of knowledge in evolutionary optimisations by fitness approximation certainly improves the convergence speed of the optimisation. On the other hand particularly, in multi modal fitness landscapes this has the high probability for premature convergence which can worsen the optmisation process. Ulmer (13), suggests an approach called *model assisted evolution strategy* to address this problem which uses the Gaussian process approximation model to preselect the most promising individuals before applying expensive true fitness functions. Hence this preselection process has been refined by identifying the likelihood of each individual to improve the overall best found solution. Gaussian processes are applied due to its advantage of providing probabilistic interpretation of the model prediction.

Hence this strategy attempts to find the trade-off between the concerns of optimisation and the generation of suitable approximation model of the fitness landscape. This was done by introducing *probability of improvement* (POI) criterion to detect the highly promising individuals. POI controls the balance between exploitation and exploration of by making use of probabilistic interpretation of Gaussian models. This approach of model assisted ES with POI pre-selection has considerably increased capacity to sample in unexplored spaces.

However, eventhough with the increasing research efforts, the application of approximate, surrogate or meta-models for optimisation problems, have found only limited success in the usage of real world optimisation problems. Primarily the limitations are due to,

- The Curse of Dimensionality which results in hurdles to construct the accurate approximate model.
- Lack of massive parallelism with the existing strategies
- Considerably less emphasis on global convergence properties of these combination of evolutionary optimisation techniques.

Thus this work proceeds with developing a common framework to incorporate the knowledge in the following manner. Efforts are made to incorporate the domain knowledge at hand possibly from existing best designs, wind tunnel experiments etc. to the covariance matrix so that the search process can be initialised with the definition of search direction and search space. Knowledge from previous generations is taken care by CMA-ES in the form of self adaptation. Incorporation of knowledge in the fitness function through approximate model for this work is followed as described in chapter 6.

Chapter 4

Quantification of Parameter Interactions

In the context of Design Optimization, knowledge can be expressed as *coherence between design variables and the influence of them on the overall performance of the design.* Especially in aerodynamic design optimization, identifying and modeling those coherences or interaction effects is a challenging task. Conceptually two different techniques from regression analysis and information theory are described. Hence, the models applied to detect and quantify the parameter interaction effects which are analysed for this work are also explained.

4.1 Interaction Effects Using Multiple Regression

Regression techniques uses the moderated causal relationships 4.1 to model and detect the interactions. Generally regression models assume a functional relationships between dependent and independent variables.

Considering, the performance variable Y which is dependent on the value of the independent design parameters $X_1 \dots X_r$. The simplest type of relationship between the performance and the design parameters $x_1 \dots x_r$ is a linear relationship. Multiple regression for the given constant coefficients $\beta_1 \dots \beta_r$ is defined as follows,

$$Y = \alpha\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_r X_r + e \tag{4.1}$$

whereas e *is the residual term.* α *is the least square estimate for the performance variables.* $\beta_1 \dots \beta_r$ *are the regression coefficients for* $X_1 \dots X_r$ *respectively.* Equation 4.1 describes the linear regression of the performance variable for the set of design variables with the following assumptions (15),

- * The design variables are fixed in nature and have positive variance.
- * The rank of the sample data matrix equals the number of columns and is smaller than the number of parameters.

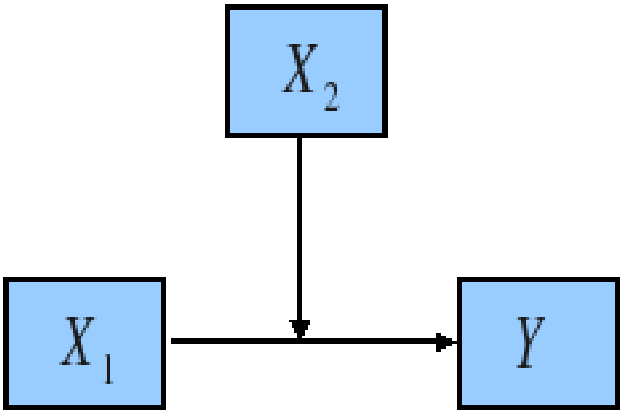


Figure 4.1: Illustration of moderated causal relationship - the nature relationship between X_1 and Y varies, depending on the value of X_2

* The residual at a given set of fixed values of X are normally distributed and have the variance that is equal to the residual value at any other fixed set of values of X.

The regression coefficient β_i provides the knowledge on the influence of each design variable X_i on the performance variable Y. Commonly, the *product terms* (X_1X_2) are used to model the interaction effect in the regression analysis [12].

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \gamma X_1 X_2 + e \tag{4.2}$$

It is interpreted from equation 4.2 that for every unit of change in the moderator design variable X_2 the value of β_1 is expected to change by β_3 units.

Product Regression coefficient γ provides the knowledge on combined influence of the design variable and the moderator design variable on the performance variable Y

4.2 Interaction Effects Using Information Theory

4.2.1 Preliminaries

4.2.1.1 Joint Probability Density Function

Probability density function (PDF) can be described as density of probability of random variable at each point in sample space. PDF for discrete random variables $X_1 \dots X_n$ is termed as joint PDF. The *Joint PDF maps each possible combination of attribute values into probability of its occurrences.* Considering the collection of independent mutually exclusive attributes, $A = a_1, a_2, \dots a_n$ and $B = b_1, b_2, \dots b_n$. The Joint PDF can be written as follows,

$$P_{A,B}(X) = P[(A,B) \in X], X \subseteq \mathbf{R^2} \tag{4.3}$$

4.2.1.2 Entropy

In the words of Statistics, *Shannon entropy* is said to be the amount of uncertainty associated with the random variable. This work employs entropy in order to reduce the uncertainty about the knowledge on performance, in order to achieve optimal solutions. For the random variable A, the Shannon entropy is given by,

$$H(A) \equiv -\Sigma_{a \in A} P(a) log_2 P(a) \tag{4.4}$$

$H(A)$ describes the amount of uncertainty about A which means higher the entropy level indicates that the knowledge is less reliable.

4.2.2 Mutual Information

Application of multiple regression techniques to determine the interaction effects is always based on the assumption that the relationship between the parameters. Alternatively, *Information Theory* is applied as it is purely a data driven approach without the need for any assumption about the kind of relationship between the parameters (15).

This quantification approach for the probabilistic interactions is based on how well one can approximate the joint probability distribution without admitting that there are interactions described as in (15). The calculation of information quantities for analysing the dependencies between variables are related to the estimation of Shannon entropy, which is then reduced to estimate marginal and joint probability distributions (15). This work proceeds with estimating two-way interaction information known as Mutual Information.

Considering the performance variable F and the design variable X , the joint probability distribution can be observed as, $P_{FX}(F, X)$. Then the uncertainty about F with the knowledge of X is estimated as follows,

$$H(F, X) \equiv -\sum_{i=1}^{N_F} \sum_{j=1}^{N_X} P(f_i, x_j) log_2 \frac{P(f_i, x_j)}{P(f_i)P(x_j)} = H(F, X) - H(X) \tag{4.5}$$

whereas,

$P(f_i, x_j)$, is the joint PDF by observing both the performance and design at the same time.

From the above equation, the *two-way interaction or mutual information* is obtained as follows,

$$I(F; X) = H(F) - H(F \mid X) \tag{4.6}$$

$H(F \mid X)$, is the conditional entropy which is a measure of what X does not say about F. This can also be explained as remaining uncertainty about the performance after X is known. Hence, $I(F;X)$ is also interpreted as *information gain* i.e. the knowledge achieved about the influence of the particular design variable (X) on the performance. $I(F;X) = 0$, indicates that both the design and performance variables are independent and there is no interaction between them.

Chapter 5

Setting up the Strategy Parameters

As discussed earlier, defining a suitable strategy always plays a crucial role in design optimisation to achieve the global optimum. This results in increased performance of the design. With the expert knowledge about existing designs from previous optimisations, the primary efforts of this work centred on setting up the strategy parameters before initialising the actual optimisation. These parameters greatly determine whether the algorithm will find an optimum or near optimum solution in an efficient manner by means of low number of fitness evaluations. However, choosing the correct parameters is s challenging task as it varies with the problem at hand. Instead parameter tuning, this work mainly focusses on defining the initial search points and the search direction. This is done with the motive of defining the common framework suitable for different optimisation problems by considering the computational costs and the small population size also in mind. This section describes the different models for defining initial strategy of the optimisation for initialising the optimisation using already existing designs and performance measurements.

In CMA-ES, the covariance matrix primarily defining strategy for the search, the knowledge incorporation efforts are made to define the components of the initial covariance matrix. This is done with the motive of achieving the a better optimum than using the existing standard initialisation process.

Considering the covariance matrix $\mathbf{C} \in \mathbf{R}^{n \times n}$ which has the form (refer section 2.1.2) of

$$\mathbf{C} = \mathbf{Q}\mathbf{D}^2\mathbf{Q}^T \tag{5.1}$$

Whereas $QQ^T = RR^T$ rotation matrix represents the eigenvectors. D^2, is the diagonal matrix represents the eigenvalues. With the standard initialisation process, the initial covariance matrix has the form of

$$\mathbf{C} = \mathbf{D}^2 = \mathbf{I} \tag{5.2}$$

Thus the focus is centred towards replacing the initial covariance matrix (identity matrix) either only with the eigenvalues or with the full rank covariance matrix by means of additionally initialising the eigenvectors *i.e. with eigenvalues and eigenvectors.* The linear and non-linear interaction effects are applied to estimate the eigenvalues and the rotation matrix is applied to estimate the eigenvectors. Based on the estimation of the components of the covariance matrix, the attempts are categorized as follows,

* Estimation of the eigenvalues with,
  - Pearson correlation coefficient
  - Mutual information
* Estimation of the eigenvectors with,
  - Multiple linear regression analysis

5.1 Estimation of the Eigenvalues Using Pearson Correlation Coefficient

Pearson product moment correlation coefficient (PMCC) quantifies the linear dependency between two variables X and F. This provides the measure of how well the future outcomes are predicted by model. In the context of the design optimisation, the estimation of this coefficient can be seen as efforts to quantify the linear relationship between the design parameter and the performance. While applying it to bivariate normal distribution, the correlation coefficient characterises the joint distribution as long as the marginal means and variances are known. However, the correlation coefficient is highly informative about the degree of linear dependence between two random quantities, regardless of whether their joint distribution is normal. There exist several suggestions for the interpretation of the correlation coefficient. The following interpretation suggested by Cohen is widely accepted. However, such criteria are in some ways arbitrary and should not be observed too strictly. This is because the interpretation of a correlation coefficient depends on the context and purposes. A correlation of 0.9 may be very low if one is verifying a physical law using high-quality instruments, but may be regarded as very high in the social sciences where there may be a greater contribution from complicating factors.

The correlation coefficient is estimated as the ratio of the sample covariance between each of the design variable X_i and the corresponding performance variable F their sample standard deviations. With N being the problem dimension or number of parameters, PMCC is given by,

$$r = \frac{\Sigma_{i=1}^{N}(x_i - \overline{x})(f_i - \overline{f})}{\sqrt{\Sigma_{i=1}^{N}(x_i - \overline{x})^2}\sqrt{\Sigma_{i=1}^{N}(f_i - \overline{f})^2}}, i = 1 \dots N \tag{5.3}$$

Where $\overline{x}$ and $\overline{f}$ are the expected design parameter values and performance numbers respectively. The correlation coefficient estimated from samples is indicated by r, while the population based estimation is denoted as ρ. The values of the correlation coefficient ranges between $r = [-1, 1]$. The maximum value of $r = 1$ indicating that given a design parameter the performance number can perfectly be predicted using a linear model, with all data points lying on a line for which F increases as X increases. The minimum value of $r = -1$ implies that all data points lie on a line for which F decreases as X increases and 0 implies that there is no linear relationship between the variables.

The estimated linear dependence between design and performance variables is incorporated into covariance matrix as eigenvalues so that the knowledge can be applied to initialise the search process. *i.e the correlation coefficients replaces the identity matrix.* This is denoted as the correlation matrix $\mathbf{C^r}$

5.2 Estimation of the Eigenvalues Using Mutual Information

As an alternative approach the mutual information is applied to estimate the eigenvalues of the covariance matrix. This provides the measure of non-linear relationship between the design parameter and the performance. The mutual information quantifies how much information a design variable provides in order to reduce the uncertainty about the performance. Hence it can be termed as *two-way interaction effect.* Initially the data is grouped into ten bins and from that the marginal and discrete joint probability distribution are estimated. Then mutual information is estimated by calculating the relevant frequencies. Based on the marginal and joint probability distribution the mutual information is calculated by modifying the equation 4.2 as follows,

$$I(X; F) = \sum_{x \in R_x, f \in R_f} p(x, f) log_2 \frac{p(x, f)}{p(x)p(f)} \tag{5.4}$$

Where, x and f relate to the discrete instances of the design and performance numbers. The mutual information estimated is applied to replace the identity matrix which is denoted by $\mathbf{C}^{mi}$. The following figure 5.1 explains the estimation of mutual information as well correlation coefficient from the existing data set.

5.3 Estimation of Eigenvectors Using Multiple Regression

Attempts to incorporate additional knowledge of the design into the algorithm led to the estimation of the eigenvectors using multiple regression techniques so that the search process can be initialised with the full rank covariance matrix. This helps in defining the search with the direction also. Estimation of eigenvectors is done as follows. Initially, as described in equation 4.2, the product regression coefficient γ is estimated from the design and the corresponding performance variables. The rotation angle is calculated using the product regression

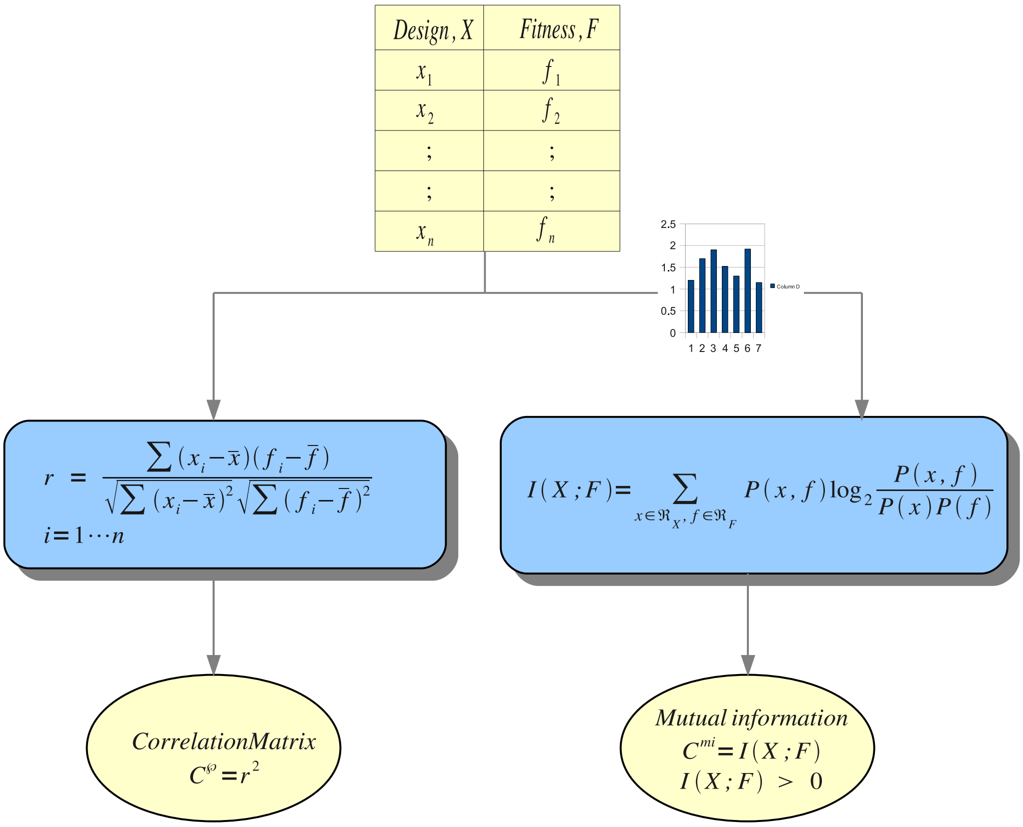


Figure 5.1: Figure explaining the estimation of correlation coefficient and the mutual information from the paired sets of design and performance variables

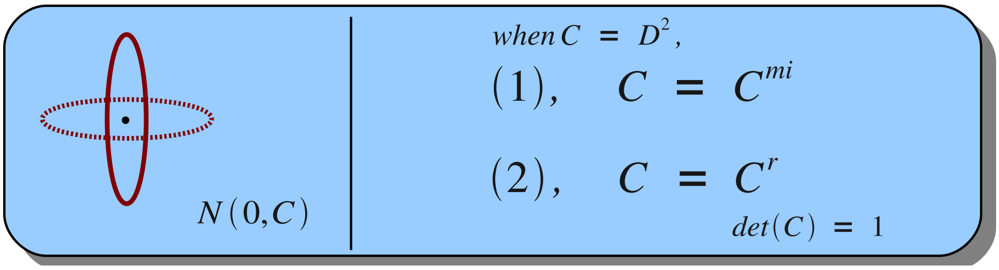


Figure 5.2: Figure explaining the estimation of the eigenvalues. Ellipsoids on the left indicates the geometrical interpretation of the initial covariance matrix

coefficients is used to estimate the complete rotation matrix for N-dimensional space. The figure 5.3 depicts the steps for the estimation of rotation matrix.

When discussing rotations, there exists two modes of rotation. One is to rotate the axes and the other one rotates the object related to the fixed axis. The efforts proceeds with initially estimating the two-dimensional rotation matrix and then rotating the object in N dimensions. This *rotation in N dimensional space* is realised by multiplying all the two-dimensional matrices. The rotation in two-dimensional space has the following matrix form which rotates the plane around the origin by an angle of θ and X axis is rotated towards the Y axis.

$$\mathbf{R}(\theta) = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix} \tag{5.5}$$

In order to estimate the rotation in N-dimensional space, the rotation matrix for each plane in N-dimensional space is estimated as follows,

$$\mathbf{R}(\alpha_{\mathbf{ij}}) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & cos(\alpha_{ij}) & 0 & -sin(\alpha_{ij}) & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & sin(\alpha_{ij}) & 0 & cos(\alpha_{ij}) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.6}$$

Hence the n-dimensional rotation is achieved by multiplying all the rotation matrices,

$$\mathbf{R} = \prod \mathbf{R}(\alpha_{\mathbf{ij}}) \tag{5.7}$$

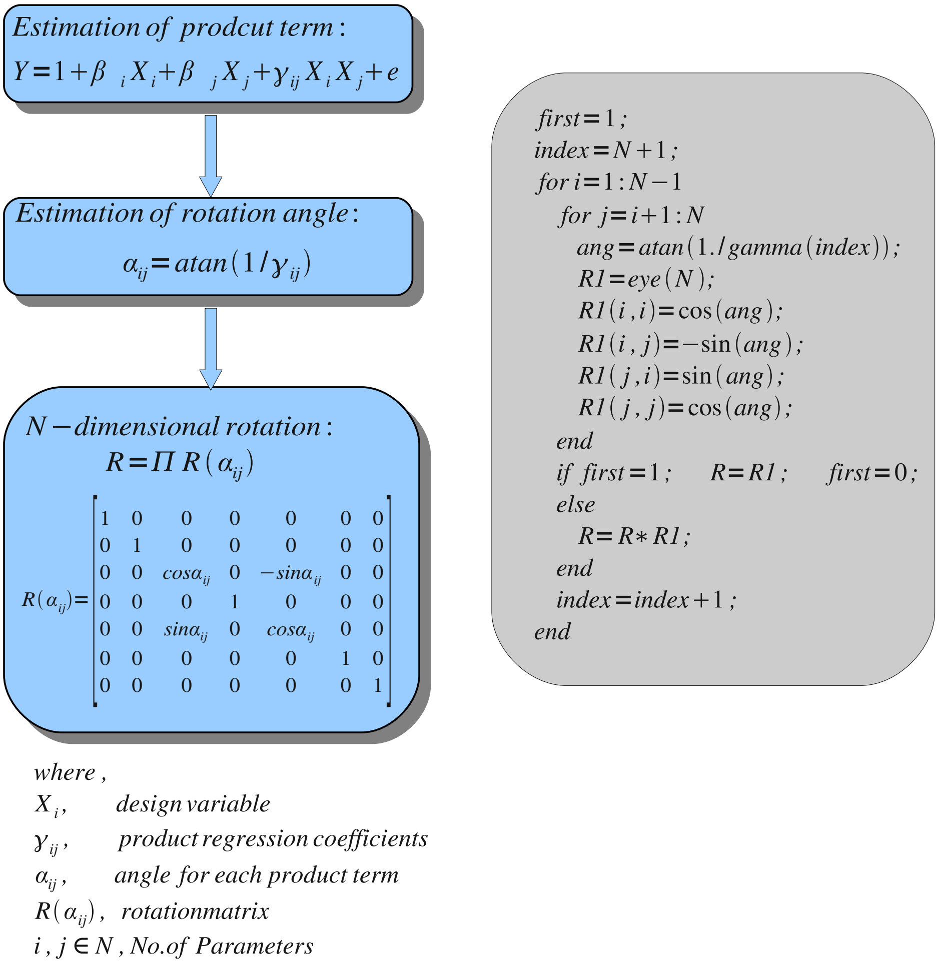
The properties of rotation matrices are,

$$det(\mathbf{R}) = 1; \quad \mathbf{R}^T = \mathbf{R}^{-1}; \quad \mathbf{R}^T\mathbf{R} = \mathbf{I} \tag{5.8}$$

In order to have an effective and fair comparison of the results the covariance matrix is normalised such determinant of the initial covariance matrix is constant. This is given by the following equation,

$$\mathbf{C}^{norm} = \mathbf{C} \cdot \mathbf{det}(\mathbf{C})^{(-\mathbf{1}/\mathbf{C_i})} = 1 \tag{5.9}$$

This enables keeping the volume of the distribution constant irrespective of the elements of the initial covariance matrix.


Estimation of prodcut term :
$Y=1+\beta_i X_i+\beta_j X_j+\gamma_{ij} X_i X_j+e$
Estimation of rotation angle :
$\alpha_{ij}=atan(1/\gamma_{ij})$
N − dimensional rotation :
$R=\Pi R(\alpha_{ij})$
$R(\alpha_{ij})=\begin{bmatrix}1&0&0&0&0&0&0\\0&1&0&0&0&0&0\\0&0&cos\alpha_{ij}&0&-sin\alpha_{ij}&0&0\\0&0&0&1&0&0&0\\0&0&sin\alpha_{ij}&0&cos\alpha_{ij}&0&0\\0&0&0&0&0&1&0\\0&0&0&0&0&0&1\end{bmatrix}$
where ,
X_i, design variable
γ_{ij}, product regression coefficients
α_{ij}, angle for each product term
$R(\alpha_{ij})$, rotationmatrix
$i, j \in N$, No.of Parameters

```
first=1;
index=N+1;
for i=1:N−1
  for j=i+1:N
    ang=atan(1./gamma(index));
    R1=eye(N);
    R1(i,i)=cos(ang);
    R1(i,j)=−sin(ang);
    R1(j,i)=sin(ang);
    R1(j,j)=cos(ang);
  end
  if first=1;    R=R1;    first=0;
  else
    R=R*R1;
  end
  index=index+1;
end
```

Figure 5.3: Steps in estimation of rotation matrix. Pseudo code given on the right calculates the rotation matrix.

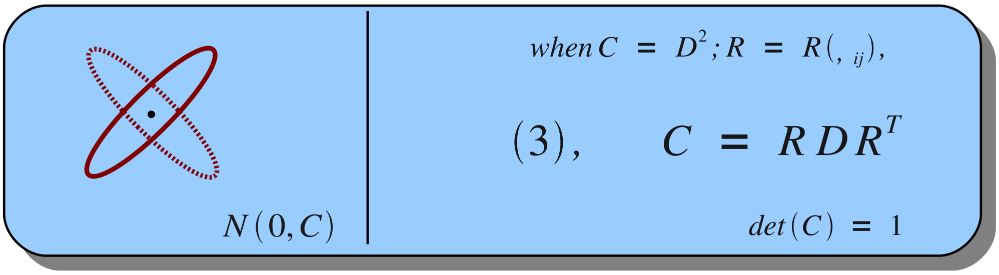


Figure 5.4: Figure explaining the estimation of full rank covariance matrix, ellipsoids indicate the geometrical interpretation of initial covariance matrix

Chapter 6

Knowledge Incorporation for 2D Blade Design Optimization

In order to investigate and proof the theoretical concepts discussed in chapter 5, the developed algorithms for the estimation of the initial covariance matrix in CMA-ES have been applied for the optimization of the stator blade design from the supersonic compressor cascade (17). The Shape of the design is represented by the B-Spline curve with 38 control points as shown in figure 6.2. Simulation model using Navier stokes 2D solver described as in (18) is applied for the experiments. As the motive is to derive a common framework for the shape optimization, the results are compared on the basis of behaviour of the incorporated knowledge with the fitness values and the algorithmic aspects. Thus this chapter describes the representation of the blade profile for encoding into the algorithm, the experimental setup and the results of the experiments.

6.1 Representation of the Design

The 2D profile of the chosen blade is represented by a B-Spline curve of 23 spline control points. Each control point is represented by its X and Y coordinate. As these control points represents the parameter vectors, there exists 46 parameters. By neglecting the two control points at the end, as they reflect the starting points to obtain a closed curve, totally 38 parameters to be optimised. Figure 6.2 is an example of the resulting geometry adapted in detail with control points and the construction of spline curve at the leading as well trailing edges.

In order to achieve the precise representation of the design, the profile of the design has to be represented adequately as it determines the optimum of designs along with the strategy by influencing the evolution path. Olhofer (19) describes the following as the main constraints that have to be fulfilled,

* Completeness

The encoding of the design must assure maximal degrees of freedom to represent the profile in order to avoid unnecessary constraints on the phenotype space. Representation with adequate degrees of freedom enables optimal representation of the design.

* Causality

Causality implies that small steps on the genotype space lead to small steps on the phenotype space. *This must be taken care of as it is important for the adaptation of strategy parameter in CMA-ES.*

* Compactness

It plays crucial role especially in population based algorithms due to high dimensional search spaces. The representation should be in such a way that it has to minimise the dimensionality of the search space so that the calculation time can be minimised.

The expert knowledge about the problem and possible designs of the blade helps greatly while devising the efficient representation of the design. So that the chosen model satisfies the strong demand for causality and balances between high degree of freedom for the structure generation and a low dimensionality of the genotype space. The upper and lower sides of the profile represented by five points fixing two fourth order polynomials.

6.2 Objective Function

The quality function applied for the optimisation of the 2D blade is given by 6.1. This description is based on (19).

$$f(x) = \eta_1 \omega + \eta_2 . f_2(\alpha_2) + \eta_3 . f_3(d_1) + f_4(d_2) \tag{6.1}$$

It can be seen that the quality of the design is described by four objectives. The first one is based on pressure loss ω which is a measure of energy loss which must find the minimum value of that. The second term measures the deviation of the averaged angle of the gas stream at the end of the blade α_2 from the specified value $\tilde{\alpha}$, which is the target of the optimisation. f_3 and f_4 represents the geometrical constraints considered. For the sake of stability and manufacturing purposes $d_1 > d_{min}, \ \ d_2 > d_{max}$ are set. Whereas d_{min} and d_{max} are the minimum and maximum given values of diameters respectively.

The non-linear functions f_i are introduced to include the given constraints concerning outlet angle and geometry into the fitness function. As long as the corresponding values are in the given tolerance range or larger than the minimal value the function is equal to zero.

In equation 6.1,η_i are the weights of the different objectives which are essential to normalise the range of possible values for the different terms. Hence they allow prioritising certain criteria. Factor for the outlet angle is set to $\eta_2 = 10$, η_3 and η_4, the factors for the geometric constraints are set to 10^7. While considering the range of values of the angle and the thickness during optimisation, the relationship between different terms can be estimated. The maximum difference between outlet angle and the design value was $\alpha = 2.5^o$. The maximum difference between the stipulated and the measured minimal thickness at the thickest part and thinnest part were maximal 1% during the optimisation. Therefore the relation between the terms in the order given in equation 6.1 with the following initial setup of weights. $n_1 = 1, n_2 = 100, n_3 = 30, n_4 = 30$. During the optimisation due to low influence of pressure loss and the high influence of the constraints, the shape is modified to an allowed shape in the first step.

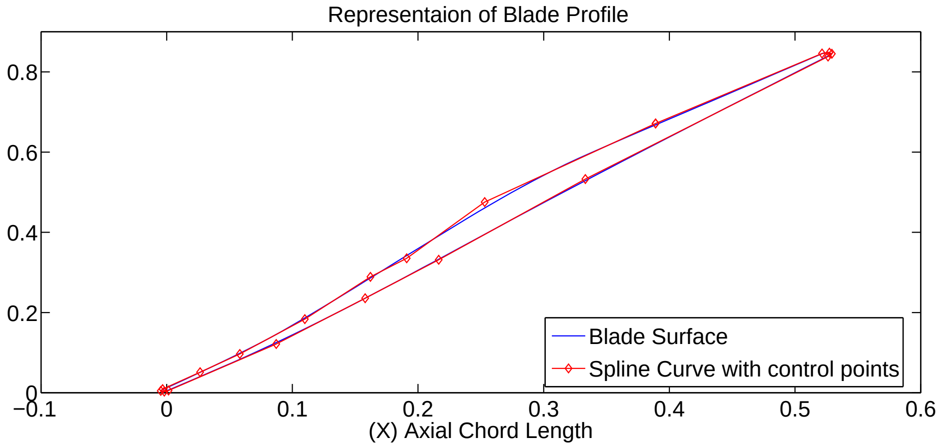


Figure 6.1: Representation of the initial 2D blade profile (blue) and its B-Spline representation (red) with 23 controls points. Each of the control point is defined by its x and y coordinates in 2D plane.

6.3 Experimental Setup

The initialisation models are implemented in C++ to incorporate the developed algorithm into the existing 2D Blade simulation code. The algorithm uses the Shark Library (20) for representing the genomes and performing genetic operators as well as for the selection method. PVM (21) is used to parallelize and accelerate the evaluation of individuals. Navier Stokes 2D solver with a low Reynolds number $k - \epsilon$ is used for flow simulations. As discussed in chapter 5, knowledge is incorporated by means of four different covariance models for optimisation and the results are compared with the standard initialisation. They are given as follows,

Initialisation of Eigenvalues with,

* correlation matrix

* mutual information matrix

Initialisation with full rank covariance matrix with,

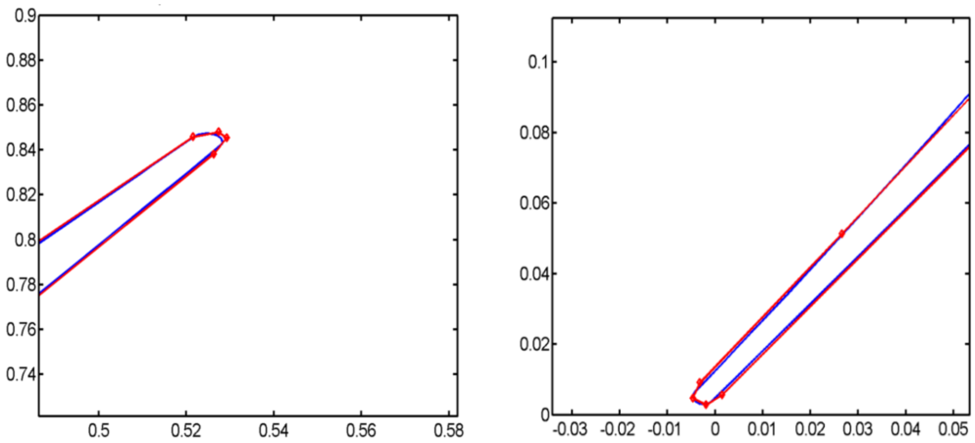


Figure 6.2: Representation of the trailing edge (left) and the leading edge (right) of the Blade profile

- * correlation and regression coefficient
- * mutual information and regression coefficient

The knowledge about the design is estimated for 70 generations and the optimisation is initialised from that point. The objective parameters are encoded into the parent population and the covariance matrix defines the strategy with the help of incorporated knowledge (Figure 6.3). Each experiment has been performed 10 times with different initialisation of the random number. The median of these runs is given in the following figures representing the results of the optimisation run.

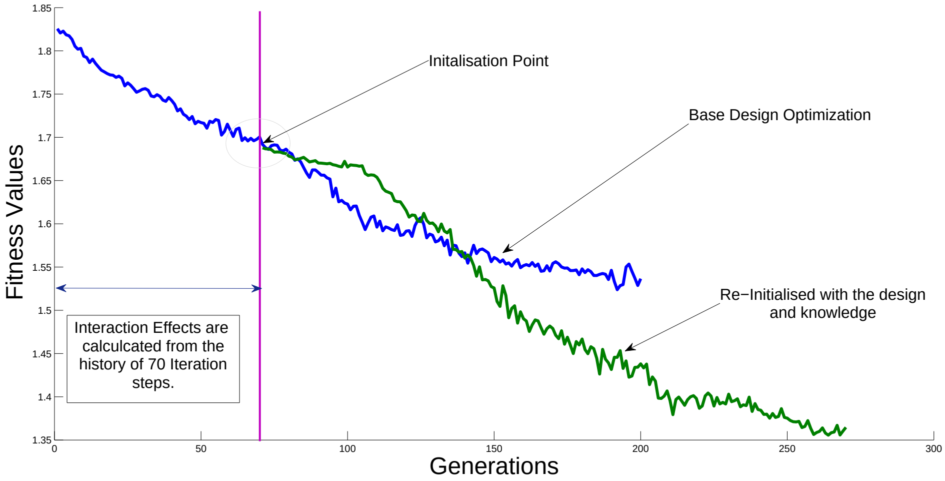


Figure 6.3: Experimental setup showing the extraction of knowledge from existing data and the initialisation with the extracted knowledge.

6.3.1 Initialisation with Eigenvalues

Initialisation is done by with replacing the initial covariance matrix (identity matrix) with correlation $\mathbf{C^r}$ as well as mutual information matrix $\mathbf{C^{mi}}$. As discussed earlier both the matrices are only with the eigenvalues. Figure (6.4) and (6.5) represents the initialised covariance matrix using correlation and mutual information as Eigenvalues respectively. It can be noticed from figure (6.4) that correlation matrix has the presence of high values. i.e. provides information on the existence of high level of interaction between the design and the performance variables compared to mutual information. The correlation and mutual information matrices represents linear and non-linear dependence of the design parameter over the performance parameter respectively. (Refer Chapter (4)).

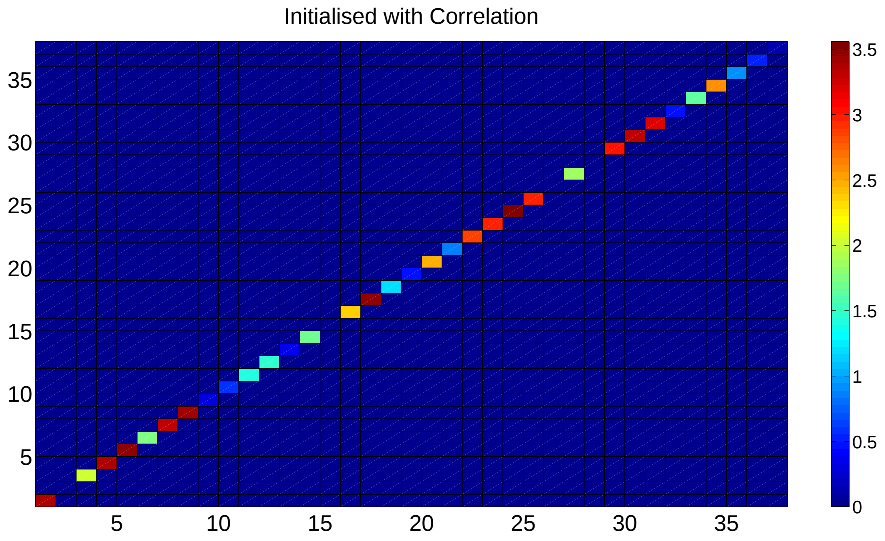


Figure 6.4: Initialisation of covariance matrix using correlation as eigenvalues. The size of the matrix represents the dimension of the search space e.g. the number of parameters considered for optimisation.

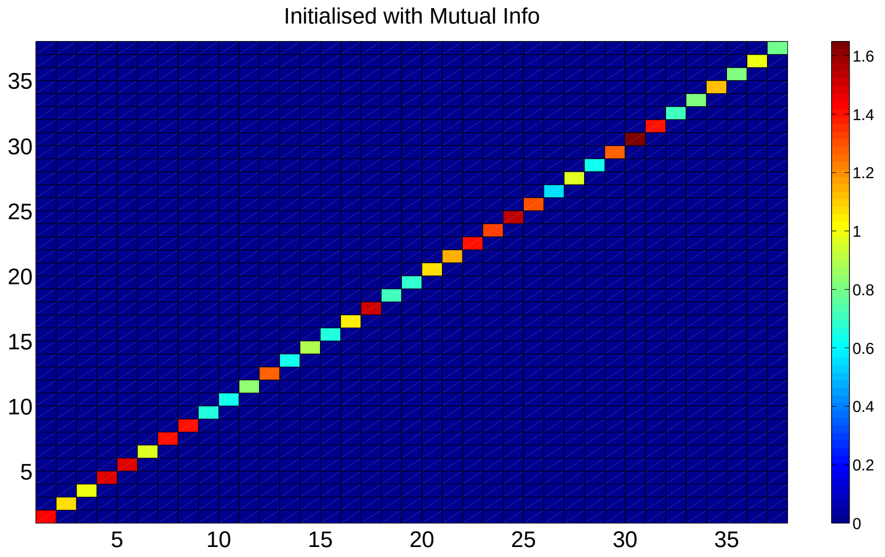


Figure 6.5: Initialisation of covariance matrix using mutual information as eigenvalues. The size of the matrix represents the dimension of the search space e.g. the number of parameters considered for optimisation.

Figure (6.6) shows the median fitness curves of the experiments conducted for 10 runs. The curve labelled baseline indicates the existing design optimization. Standard initialisation represents the experiment in which the covariance matrix is initialised as identity matrix which is a general procedure with CMA-ES. Initially the algorithm takes 10-15 generations to adapt the global step size. It can be noticed from Figure (6.14) that the effect of the covariance matrix comes into play after 10-15 generations. Hence, the initialisation with correlation shows good convergence behaviour of the fitness and it is almost similar to the baseline optimisation procedure. This performance is as good as the existing best but with better performing shape than the existing design. The initialisation with correlation matrix gives the advantage of around 60-70 generations by saving the computational time of 12 hours for the chosen 2D Blade optimisation problem. The good convergence with correlation can be explained as the effect of high value of interaction effects present in the correlation matrix. Refer (6.4). Initialisation utilising the mutual information which represents the non-linear interaction effects follows the standard initialisation and it does not show any significant improvement of the performance.

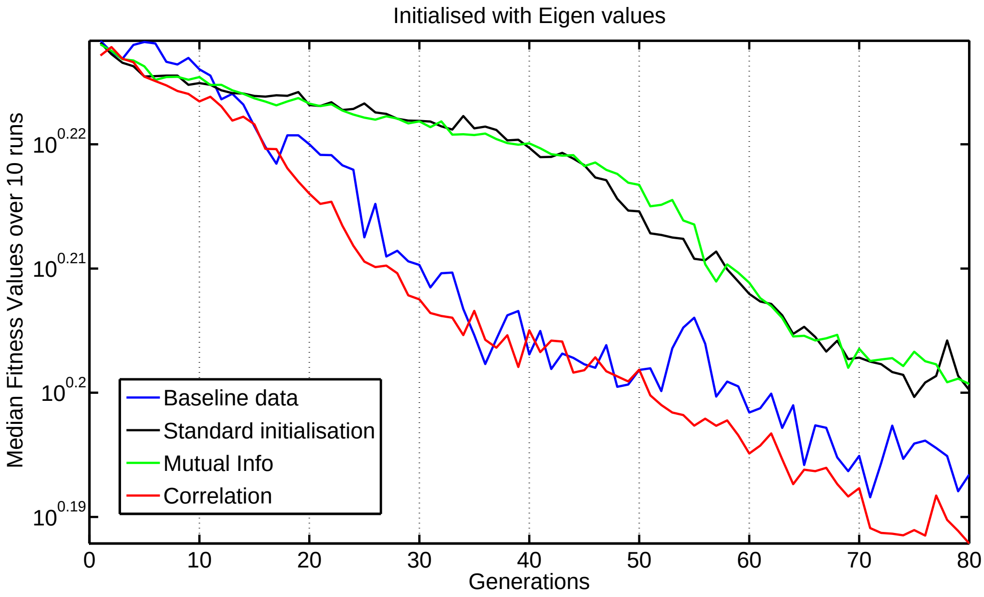


Figure 6.6: Comparison of the median fitness values for the initialisation with eigenvalues. The x-axis represents the generation and the y-axis represents the fitness values in logarithmic scale.

6.3.2 Initialisation with Full Rank Covariance Matrix

Figure 6.7 and 6.8 represents the initial covariance matrix using correlation and mutual information as eigenvalues respectively. It can be noticed that figure (6.7) has high values present in the matrix along the diagonal as well as the non-diagonal elements. when compared to diagonal elements the values of the non-diagonal elements in both the matrices are low.

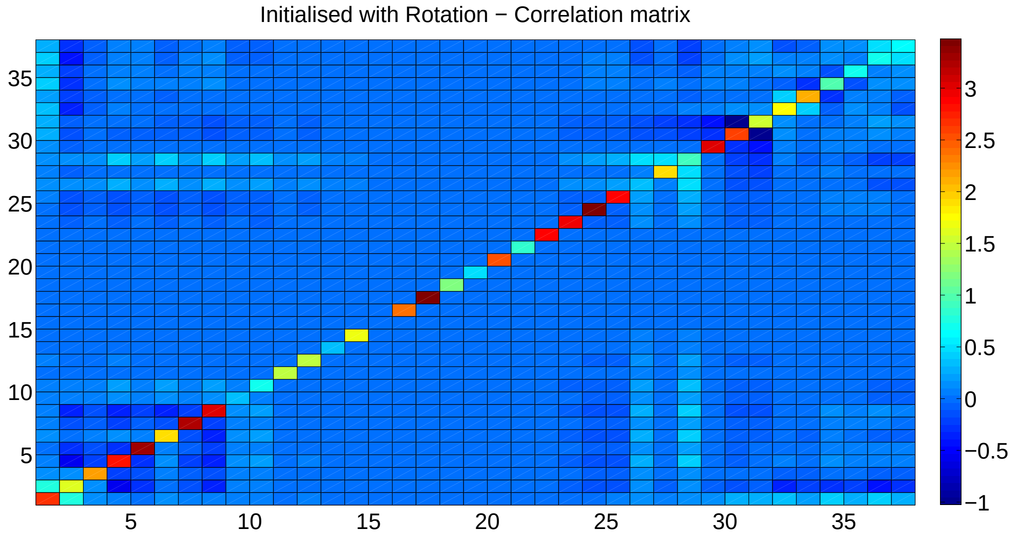


Figure 6.7: Initialisation of full rank using Correlation-Rotation matrix. Eigenvalues represent the correlation and eigenvectors represent the regression coefficient. The size of the matrix represents the dimension of the search space. e.g. number of parameters considered for optimisation.

The same setup extended for initialising with full rank covariance matrix (figure 6.9). It can be noticed that initially the algorithm takes 10-15 generations to adapt the global step size and then the effect of the covariance matrix comes into the play (figure 6.14). Hence here also initialisation with correlation shows good convergence values of the fitness and it is almost similar to the baseline optimisation procedure. But it is not showing good convergence than the baseline. Mutual information is converging better than the standard initialisation. Effect of the rotation didn't make considerable impact when compared to the initialisation with eigenvalues (Figure 6.9).

It can be seen clearly from the plots (6.9) the effect of the initialised covariance matrix lasts up to around 75 generations after that the direction of the search is changing.

Comparison of variances: The figures 6.10 and 6.11 represent the variances of fitness values over 10 runs for initialisation with eigenvalues. Statistical sig-

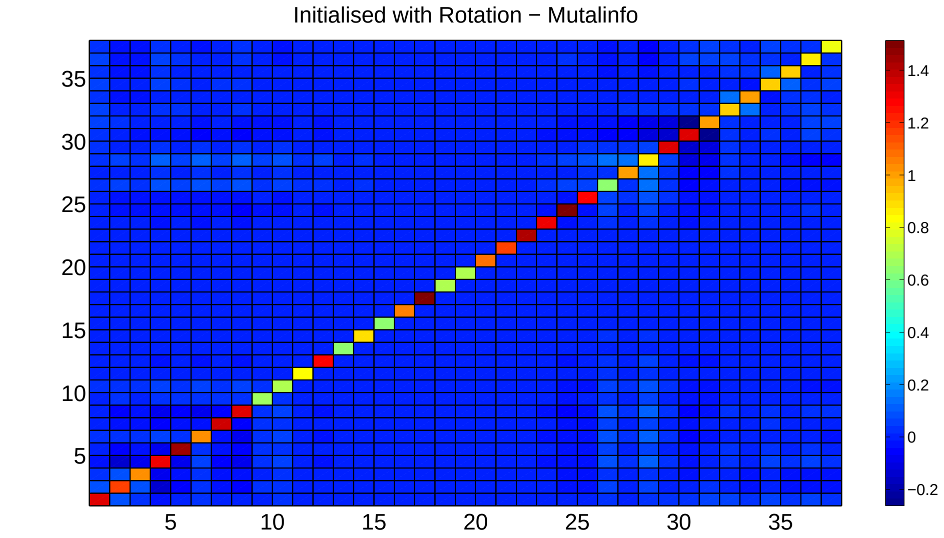


Figure 6.8: Initialisation of full rank using Mutual information - Rotation matrix. Eigenvalues represent the correlation and eigenvectors represent the regression coefficient. The size of the matrix represents the dimension of the search space. e.g. number of parameters considered for optimisation.

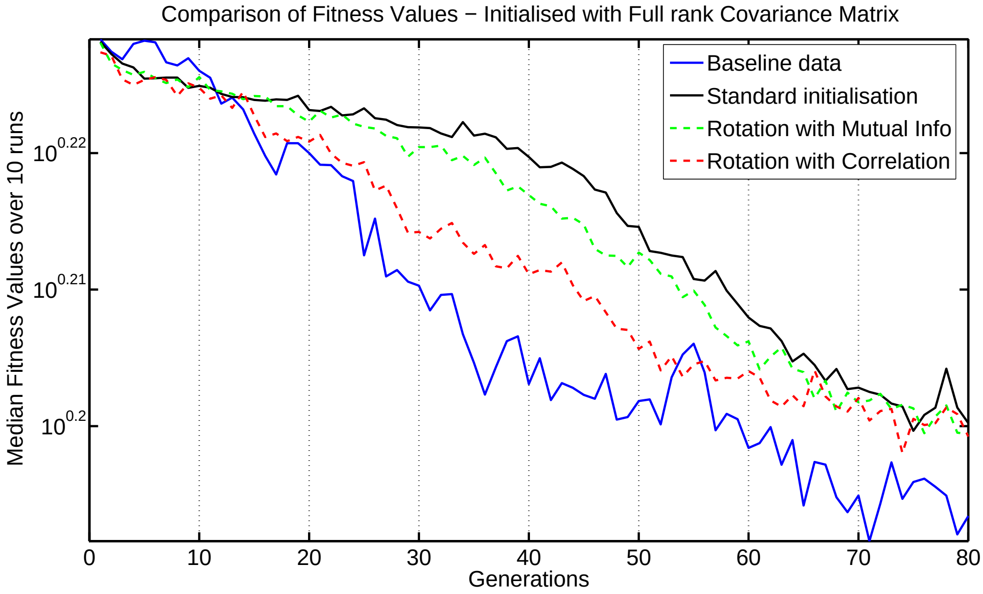


Figure 6.9: Comparison of the fitness values for the initialisation with full rank covariance matrix. x-axis represents the generation and the y-axis represents logarithmic values of median fitness values over 10 runs.

nificance of the correlation with the standard initialisation can be observed from figure 6.10. When compared with the mutual information, variance of the fitness values are also low. Statistical significance of the run initialised with correlation as eigenvalues resulted in faster convergence of the algorithm.

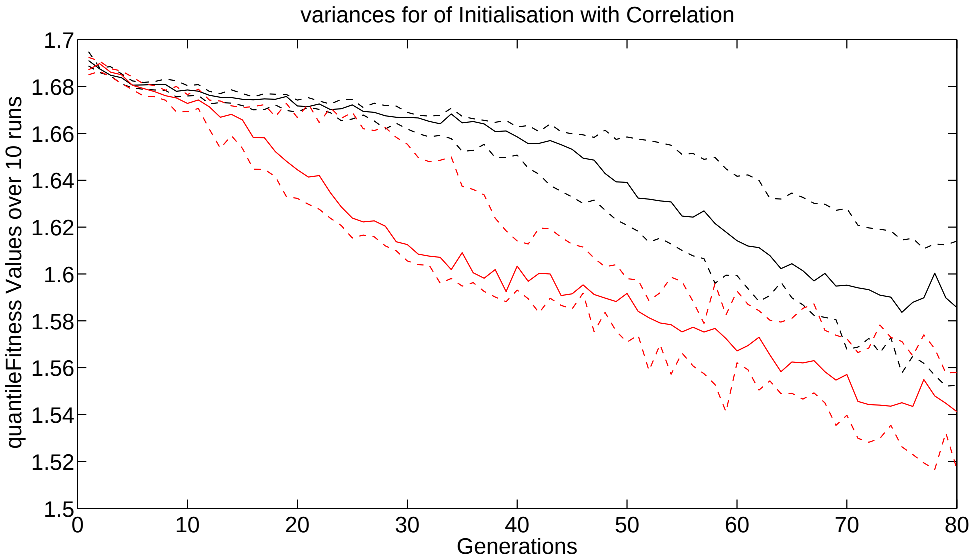


Figure 6.10: Comparison of the variance in fitness values of the initialisation using correlation as eigenvalues. Compared initialisation using correlation as eigenvalue (red) with standard initialisation (black): lower and upper dotted lines indicate 25% and 75% of the fitness values respectively. Solid line represents the median values.

Similarly, the initialisation using full rank covariance matrix using correlation as eigenvalues (Figure 6.12) shows high significance when compared with mutual information (Figure 6.13). Interestingly, in some of the runs mutual information showed good convergence similar to baseline optimisation run (lower dotted lines in Figure 6.13). There exists some chances of good convergence for the mutual information with rotation. In order to obtain the common procedure it will be preferable to test the mutual information-rotation model with another test problem.

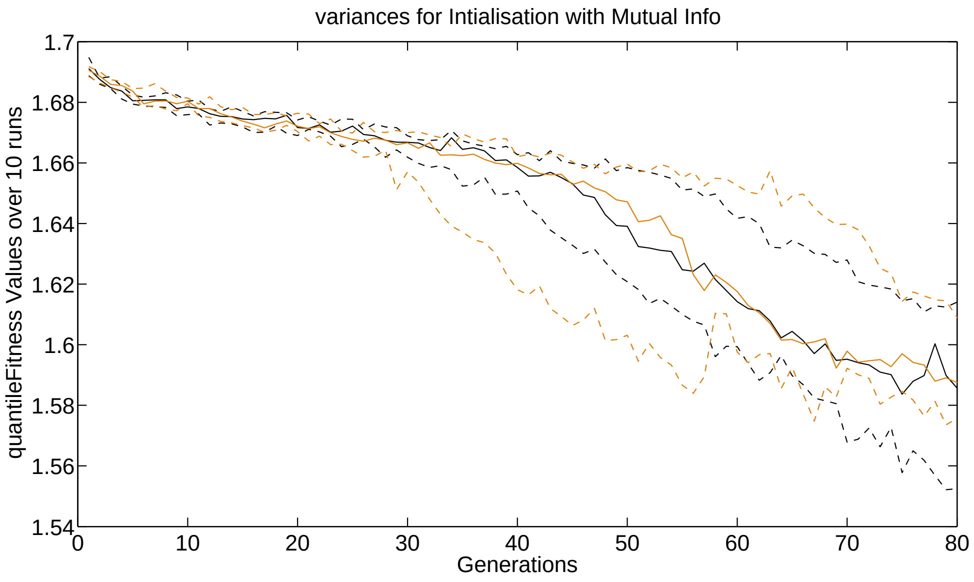


Figure 6.11: Comparison of the variance in fitness values of initialisation using mutual information as eigenvalues. Compared initialisation using mutual information as eigenvalue (brown) with standard initialisation (black): lower and upper dotted lines indicate 25% and 75% of the fitness values respectively. Solid line represents the median values.

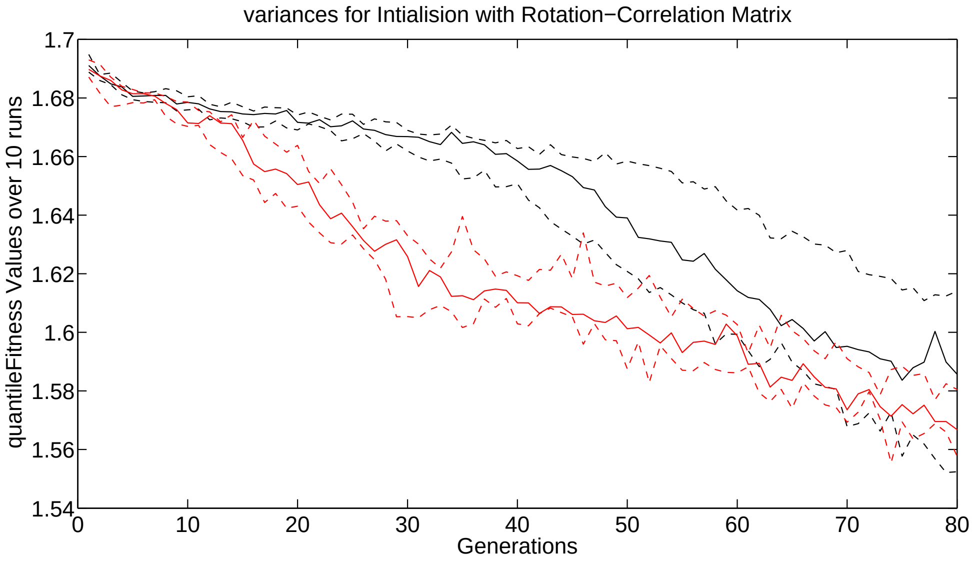


Figure 6.12: Comparison of the variance in fitness values of the full rank covariance matrix initialised with rotation-correlation matrix. Compared initialisation using correlation with rotation (red) with standard initialisation(black): lower and upper dotted lines indicate 25% and 75% of the fitness values respectively. Solid line represents the median values.

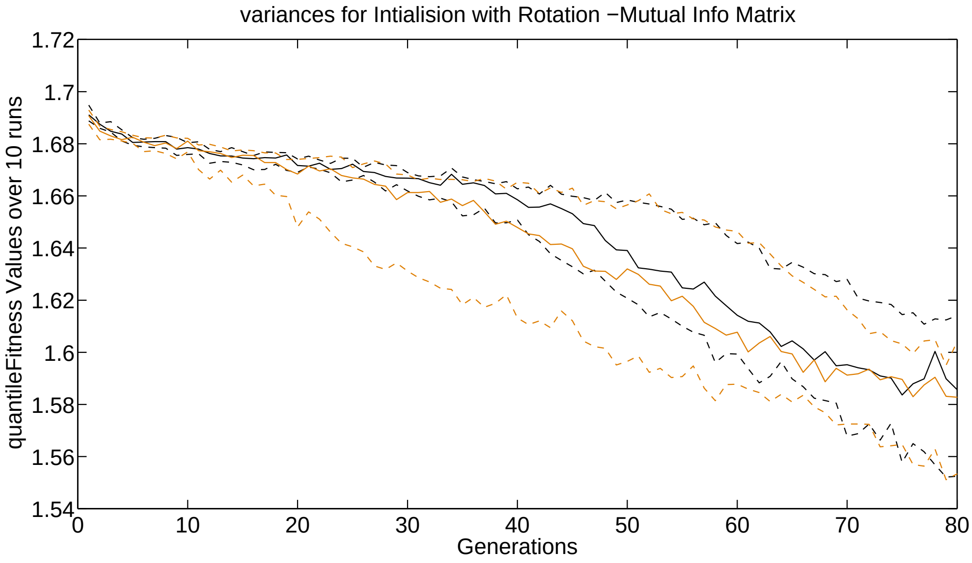


Figure 6.13: Comparison of the variance in fitness values of the full rank covariance matrix initialised with rotation-mutual information matrix. Compared initialisation using correlation with rotation (red) with standard initialisation(black): lower and upper dotted lines indicate 25% and 75% of the fitness values respectively. Solid line represents the median values.

6.4 Analysis of Step size Adaptation

The figures 6.14 and 6.15 visualizes the progress of global step size over the generations. Optimisation has been initialised with $\sigma = 0.0001$ for each optimisation run. It takes around 15 steps to adapt the strategy after that longer the steps leads to the faster convergence. It can be understood that the initialisation with correlation matrix as eigenvalues shows the faster adaptation of the strategy. This leads to better converging behaviour than all other initialised models.

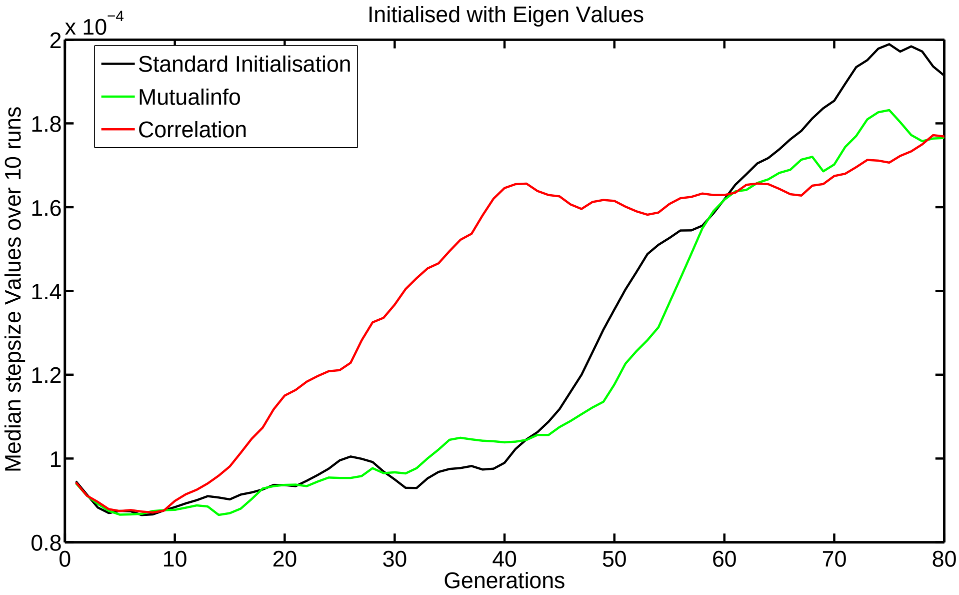


Figure 6.14: Comparison of the global step sizes for the covariance matrix initialised with eigenvalues.

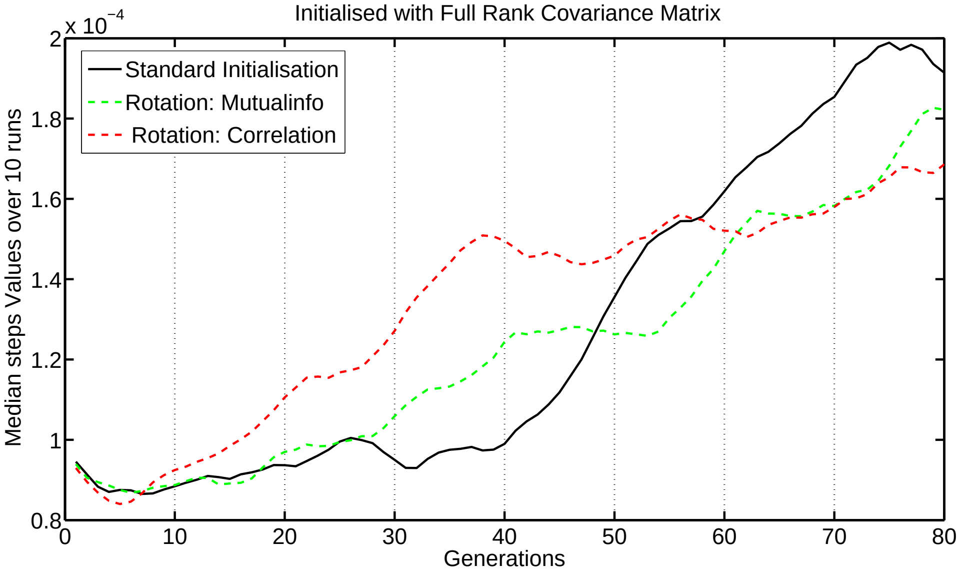


Figure 6.15: Comparison of the global step sizes for the initialisation with full rank covariance matrix.

Chapter 7

Conclusions

In this thesis different models for incorporating the knowledge to initialise the search process are analysed. Knowledge is estimated by means of quantifying the interaction effects. *i.e.Detection of linear as well as non-linear interaction effects between the design and the performance parameters.* More precisely, investigated different techniques for developing a common framework for estimation and initialisation of the covariance matrix using pre-knowledge in order to find the better shape of the given geometry with reduced computational time.

To incorporate the knowledge, two models of initialising the optimisation are applied. Firstly initialising search only with the eigenvalues is done. Second approach is based on initialising the search process by defining the full rank covariance matrix adding eigenvectors by means of regression coefficients to the covariance matrix. Correlation and mutual information are applied as eigenvalues to the covariance matrix representing linear and non-linear interaction effects respectively. Then both of them combined with eigenvectors to form the full rank covariance matrix.

In order to validate the applicability of the developed framework , it has been applied to the 2D Gas Turbine Blade with fluid dynamic simulation. The results are analysed on the basis of convergence speed of the algorithm, adaptation of the strategy. On the performance basis, initialisation done with the correlation as eigenvalues shows the good convergence speed and it is almost same as the convergence speed of existing best design but with the knowledge incorporated. This saves around 60 generations approximately, around 12 hours of computational time for this 2D Blade optimization. On the other hand, covariance matrix initialised using the mutual information eigenvalues did not show good convergence speed except some runs. This can be attempted with another complex problem to derive a common framework. Also the initialisation with full rank covariance matrix did not make any considerable impact on the search. It is preferred to initialise the search only with eigenvalues. Self adaptation of the strategy is also analysed.

In this search still there exists many interesting aspects to be studied in the future. They can be :

* Analysing the influence of global step size along with the incorporated knowledge.

* Generally CMA-ES takes around 10 initial generations to adapt the strategy. After that only the incorporated knowledge comes into effect. How to get the good convergence from the initialised point?

Appendix A

Nomenclature

This section provides the list of abbreviations and frequently used mathematical symbols.

Abbreviations :

EA	*Evolutionary Algorithms*
EP	*Evolutionary Programming*
ES	*Evolution Strategy*
GA	*Genetic Algorithm*
CMA	*Covariance Matrix Adaptation*
CPs	*Control Points*
MI	*Mutual Information*

Mathematical Annotations :

A, a	*common notation for scalars or sets*
$\underline{a}$	*common notation for vectors*
$\mathbf{A}$	*common notation for matrices*
α	*angle*
λ	*size of the offspring population*
μ	*size of the parent population*
$g \in \mathbf{N}_0$	*generation counter*
σ	*standard deviation, global stepsize*
σ^2	*variance*
σ^g	*stepsize at generation g*
$\underline{m}^g$	*mean value of the distribution generation g*
$f : \mathbf{R}^n \rightarrow \mathbf{R}$	*objective function, fitness function to be minimised*
$\mathbf{C} \in \mathbf{R}^{n \times n}$	*covariance matrix*
$\mathbf{C}^g$	*covariance matrix at generation g*
$\mathbf{I} \in \mathbf{R}^{n \times n}$	*identity matrix, unity matrix*
$\mathbf{R}_{ij}$	*Rotation matrix*
$n \in \mathbf{N}$	*search space dimension*
$\mathcal{N}(0, I)$	*multivariate normal distribution with zero mean and unity matrix*
$\mathcal{N}(m, C)$	*multivariate normal distribution with mean and covariance matrix*
$\underline{x}_k^{(g+1)}$	*k-th offspring/individual from generation g+1*
r	*pearson correlation coefficient*
$\mathbf{C}^{mi}$	*covariance matrix initalised with mutual information as eigenvalues*
$\mathbf{C}^r$	*covariance matrix initalised with correlation as eigenvalues*
cov	*covariance*

Bibliography

[1] Hansen, N. *The CMA evolution strategy: A tutorial.2008.* http://www.lri.fr/ hansen/ 1, 5

[2] Graening, L. Menzel, S. Hasenjger, M. Bihrer, T. Olhofer, M. and Sendhoff, B. *Knowledge extraction from aerodynamic design data and its application to 3D turbine blade geometries.* Journal of Mathematical Modelling and Algorithms, pages 329-350. Springer - Netherlands, 2008. iii, 1, 3

[3] Zhang, P. Yao, X. Jia, L. Sendhoff, B. Schnier, T. *Target shape design optimization by evolving splines.* IEEE Congress on Evolutionary Computation CEC 2007. pages 2009-2016. Singapore, 2007. 2

[4] Menzel,S. Olhofer,M. and Sendhoff,B. *Direct manipulation of free form deformation in evolutionary design optimisation.* PPSN IX, LNCS 4193, pages 352-361. Springer, 2006. 2

[5] Eiben, A.E. and Smith, J.E. *Introduction to evolutionary computing.* Springer, 2003. 5, 14, 16

[6] Hansen, N. and Ostermeier, A. *Completely derandomized self-adaptation in evolution strategies.* Journal of Evolutionary Computation, 9(2). pages 159-195. MIT press, 2001. 11

[7] Bramlette, M.F. and Kirby, S. *Initialisation, mutation and selection methods in genetic algorithms for function optimisation.* Fourth International Conference on Genetic Algorithms, pages 100-107. Morgan Kaufmann Publishers, 1991. 16

[8] Smith, J.E. Bartley, M. and Fogarty, T.C. *Microprocessor design verification by two-phase evolution of variable length tests.* In ICEC-97. pages 453-458. 17

[9] Unger, R. and Moult, J. *In proceedings of the 5th international conference on genetic algorithms.* In ICEC-97. pages 581-588. 1993. 17

[10] Rasheed, K. Ni, X. and Vattam, S. *Methods for using surrogate models to speedup genetic algorithm optimisation: informed operators and genetic engineering.* Knowledge incorporation in evolutionary computation. pages 103-122. Springer verlag, 2004. 17

[11] Jin, Y. Huesken, M. Olhofer, M. and Sendhoff, B. *Neural network for fitness approximation in evolutioanry optimisation.* Knowledge incorporation in evolutionary computation. pages 281-306. Springer verlag, 2004. 18

[12] Dennis, J. Torczon, V. *Managing approximate models in optimisation.* Mutlidisciplinary design optimisations: State-of-the-art. pages 330-347. SIAM publications, 1997. 19

[13] Ulmer, H. Streichert, F. and Zell, A. *Model assisted evolution strategies.* Knowledge incorporation in evolutionary computation. pages 333-358. Springer verlag, 2004. 20

[14] Jaccard, J. Turrisi, R. *Interaction effects in multiple regression* Sage publications, 2003.

[15] Roussas, G. *An introduction to probability and statistical inference* . Academic press, 2003. 21, 23

[16] Campbell, H. *Linear algebra with applications.* Appleton Century Crofts, 1971.

[17] Schreiber, H.A. *Experimental investigations on shock losses of transonic and supersonic compressor cascades.* AGRAD CP-401, 2001. 32

[18] Arima, T. Sonoda, T. Shirotori, M. Tamura, A. Kikuchi, K. *A numerical investigation of transonic axial compressor Rotor flow using a low-Reynolds-number K- turbulence model.* ASME journal of turbomachinery, 2006. 32

[19] Olhofer, M. Arima, T. Sonoda, T. and Sendhoff, B. *Optimisation of a stator blade used in a transonic compressor cascade with evolution strategies.* Adaptive computing in design and manufacture (ACDM 2000). Springer verlag, 2000. 32, 33

[20] Igel, C. Glasmachers, T and Meisner, H. *Shark.* Journal of machine learning research. pages 993 - 996. 2008. http://shark-project.sourceforge.net/ 34

[21] http://www.csm.ornl.gov/pvm/. 34