DEVELOPMENT OF NON-DOMINATED SORTING GENETIC QUANTUM ALGORITHM FOR STRUCTURAL AND TOOL SHAPE OPTIMIZATION

By

Amirreza Khorsand

A Thesis

Submitted to the Faculty of Graduate Studies in Partial Fulfillment

of the Requirements for the Degree of

MASTER OF SCIENCE

Department of Mechanical and Manufacturing Engineering

University of Manitoba

Winnipeg, Manitoba

Development of Non-Dominated Sorting Genetic Quantum Algorithm for Structural and Tool Shape Optimization

By

Amirreza Khorsand

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

Of

Master of Science

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION

DEVELOPMENT OF NON-DOMINATED SORTING GENETIC QUANTUM ALGORITHM FOR STRUCTURAL AND TOOL SHAPE OPTIMIZATION

BY

Amirreza Khorsand

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

of

MASTER OF SCIENCE

To my mother, my father, and my sister

ACKNOWLEDGMENT

I would like to express my gratitude to my advisors Dr. Raghavan Jayaraman and Dr. Gaofeng Gary Wang for their valuable guidance and financial support throughout my program.

Very special thanks to my mother, father, and sister for their words of encouragement and support over the course of my study.

ABSTRACT

Structural optimization is important for product design in order to improve the performance and decrease the production time as well as total cost of a product. In the past, engineers used costly and time consuming trial-and-error methods to find the optimum for their structural design problems. Today, with steadily increasing computational power, computers are increasingly used to design, manufacture, and evaluate the performance of a product without having to physically building it to reduce the cost of development. The finite element analysis (FEA) used in such developmental process are expensive and hence, often hinders the direct use of conventional optimization techniques such as the mathematical programming and most heuristic approaches. When there are multiple objectives and constraints, the difficulty compounds. Efficient optimization of structural design problems with computationally intensive FEA processes has been studied intensely by many researchers for the past decade. Only recently, researchers have started to focus on optimization of multiple objective problems using these expensive FEA. This thesis aims to develop a multi-objective optimization algorithm which can handle multiple design variable, objectives, and constraints of expensive FEA models. A new algorithm, called non-dominated sorting genetic quantum algorithm (NSGQA), is developed by integrating a newly developed evolutionary algorithm, genetic quantum algorithm (GQA), and a multi-objective sorting mechanism. The performance of NSGQA is intensively evaluated using inexpensive optimization problems and compared with state-of-the-art meta-heuristics multi-objective optimization algorithms. Subsequently, NSGQA is applied to a real structural optimization problem with 47 trusses and many constraints. From the tests and application to structural optimization, NSGQA demonstrates promising performance. The NSGQA is integrated with a FEA-based process model for composite processing and is applied to optimize tool shape to reduce the process-induced warpage. This preliminary benchmarking was done to pave the way for further research on multi-objective tool shape optimization for composite processing.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENT

CHAPTER 1
INTRODUCTION

1.1 Challenges in Structural Optimization

Structural optimization includes a wide range of applications from aerospace industry to auto industry. Depending on the type of problems, the optimization goals often seek to minimize one or more of the following, cost, weight, drag force, fuel consumption, and others. For example, a body of wing can be optimized to achieve minimum drag. Hague et al. [1] proposed a technique, composed of nine separate search strategies, which use parametric representation of airfoil surface and two- and three-dimensional situations, to minimize drag.

The field of structural optimization presents an array of challenging problems, usually characterized by their computational intensity, multi-modality, non-convexity, high dimensionality, and multi-objectivity. As a result, only a limited number of such problems are fairly amenable to conventional mathematical programming approaches, for which the design space is continuous and convex, and the problem dimensionality is low. However a significantly larger class of structural design problems is characterized by unknown non-convexity in the design space, and a mix of continuous, discrete, and/or integer design variables, as well as multiple objectives.

Conventional mathematical programming techniques rely heavily on a smooth and continuous design space and the starting point; they also yield locally optimal solutions only. Alternatively, exhaustive enumeration and random search methods are among the simplest and most robust strategies for automated structural design. They

are applicable to a wide range of problems and are not so severely limited by discontinuous design spaces such as in techniques derived from mathematical programming principles. However they are saddled with heavy computational requirement. Population-based stochastic search mechanisms, such as genetic algorithms (GA), also do not need gradient information and are in general more robust and present a better global behavior. However, similar to their competing random search approaches, GA may still also suffer from a slow rate of convergence towards the global optimum. Consequently, while it is promising, it is also challenging to develop an evolution-based search method that is computationally efficient for structural optimization problems, which is the aim of this research.

1.2 Mathematical Programming (or Gradient-Based Optimization)

Many practical design problems such as the structural and tool shape design problem are characterized by high computational cost, mixed continuous and discrete variables, as well as discontinuous and often non-convex design space. These features hinder the direct application of traditional gradient-based mathematical programming techniques. Some issues with gradient-based mathematical programming methods are:

- Only a single design point for starting the procedure and the choice of the starting point is blind and often has significant effect on the quality of the converged optimum.
- Derivatives of objective function are required, which in practice are often unavailable or unreliable, and they have difficulty in solving discrete and integer programming problems.
- Most of the gradient-based techniques are local optimization techniques and can get trapped in a local optimum of mediocre performance.

In order to overcome these difficulties various non-gradient based optimization methods have been proposed. Next, we will briefly describe one of the most widely recognized categories of non-gradient based methods.

1.3 Metaheuristic

In recent years, some optimization methods that are conceptually different from the traditional mathematical programming techniques have been developed. These methods are based on certain biological, molecular, and neurological phenomena. Methods known as genetic algorithms[2], or in general evolutionary optimization, are based on the principles of natural genetics and natural selection. Simulated annealing [3]is based on the simulation of thermal annealing of heated solids. Both genetic algorithm and simulated annealing are stochastic methods that can find the global minimum with a high probability and are naturally applicable to the solution of optimization problems. Particle Swarm Optimization (PSO) [4] attempts to simulate the choreographed, graceful motion of swarms of birds as part of a socio-cognitive study investigating the notion of "collective intelligence" in biological populations. Ant colony [5], inspired by the behavior of ants, is that of a parallel search over several constructive computational threads based on local problem data and on a dynamic memory structure containing information on the quality of previously obtained result.

All of the above mentioned methods are called metaheuristics because they are not problem-specific, so that a method which works for one problem can be used to

solve a different one. Extended from regular metaheuristics, multiple objective metaheuristics (MOMH) are methods that aim to provide a good tradeoff between an approximation of a set of efficient solutions and the time to obtain it. These methods may manipulate a complete or incomplete single solution or a collection of solutions at each iteration. While most real world problems require the simultaneous optimization of multiple, often competing, criteria (or objectives), the solution to such problems is usually computed by combining them into a single criterion to be optimized, according to some assumption. In many cases, however, the assumption is not well known prior to the optimization process. The whole problem should then be treated as a multi-objective problem with objectives that cannot be reasonably combined into a single objective. In this way, a number of solutions can be found which provide the decision maker with insight into the characteristics of the problem before a final solution is chosen.

Metaheuristics (either single objective or multi-objective) methods provide us with the following advantages:

- Capability to search for global optimum
- Being independent of types of design space
- Allowing multiple starting points
- No need for derivatives of objective function

Despite all these advantages of metaheuristics methods (for both single objective and multi-objective) they suffer from slow rate of convergence and they are computationally expensive. In other words expensive fitness evaluation is when the

time for evaluating the objective function is at least on magnitude greater than the optimization run. For example, if the convergence criterion is to stop after reaching 1000 function evaluations and if the time for each design analysis or function evaluation is 1 hour, we need 1000 hours to find the optimal design by metaheuristics methods. Therefore, many researchers are working towards decreasing the number of function evaluations so that these methods can be applied for highly expensive function evaluation like the tool shape design while maintaining their other advantages.

Since, in structural design, we have to take into account different aspect of design like strength, vibration, and drag of airplane wing, considering single objective for all these different phenomena often does not lead to desired optimal design. Using multi-objective optimization (MOO) algorithm for this type of structural design problem is expected to result in better optimal design and this defines the main goal of thesis. Detailed research objectives are organized in the next section.

1.4 Research Objectives

The objectives of this thesis include:

a) to develop a metaheuristics multi-objective optimization algorithm that can efficiently solve multiobjective structural optimization problems with reasonable accuracy

b) to integrate the developed algorithm with FEM simulation for design automation

c) To test the developed algorithm using two applications

- design of a truss tower, which was used as a bench mark problem

- tool-shape optimization to minimize warpage in autoclave cured composite parts; This includes developing FEM based process model incorporating tool-part interaction that occurs during autoclave manufacturing.

1.5 Organization of Thesis

CHAPTER 2 presents a review of published research on multi-objective optimization and structural optimization. The multiobjective optimization algorithm developed as a part of this thesis is presented in CHAPTER 3. Performance evaluation of this algorithm and benchmarking with state-of-the-art multiobjective optimization algorithms are presented and discussed in CHAPTER 4. Preliminary evaluation of applicability of this algorithm for tool shape optimization with an objective to minimize process-induced warpage is described in CHAPTER 5. Conclusions will be given in CHAPTER 6.

CHAPTER 2

LITERATURE REVIEW

A review of the relevant literature is provided in this chapter. It details the existing related research and identifies useful ideas and concepts that could contribute to the success of this work. Basic concepts of multiobjective optimization (MOO) are introduced first, followed by a brief review of MOO with a focus on metaheuristic approaches. Structural optimization and its categorization are described. Detailed review on shape optimization is conducted. A summary of the literature review is then given at the end of this chapter.

2.1 Multi-Objective optimization

A design problem often involves several criteria or design objectives. If the objectives are conflicting, the problem reduces to finding the best design trade-off among objectives. Consequently MOO is widely used in design and practical application. In MOO problems, the most important task is to find the values for the design variables, which optimize several objective functions simultaneously. A MOO problem is defined as follows,

$$\text{minimize } f(X) = (f_1(X), f_2(X), ..., f_k(X))$$

$$\text{Subject to} \qquad \text{2-1}$$

$$h_i(X) = 0; i = 1, \cdots, p$$
$$g_i(X) \le 0; j = 1, \cdots, m$$

where k is the number of objective function, p is the number of equality constraint, and m is the number of inequality constraint. $f(X)$ is k-dimensional vector of objective function. $h_i(X)$ is equality and $g_i(X)$ is inequality constraint. The solution of a MOO problem is often chosen from a so-called Pareto optimal set.

We say that a vector of decision variables $X^* \in F$ is Pareto optimal front if there does not exist another $X \in F$ such that $f_i(X) \leq f_i(X^*)$ for all $i = 1, \ldots, k$ and $f_j(X) \leq f_j(X^*)$ for at least one j. Here, F represents the feasible region of the problem.

Figure 2.1 illustrates the FON multiobjective problem $X \in [-4,4]$ and the objective functions are in Equation 2-2:

$$f_1 = 1 - \exp(-\sum_{i=1}^{3}(X_i - \frac{1}{\sqrt{3}})^2)$$
$$f_2 = 1 - \exp(-\sum_{i=1}^{3}(X_i + \frac{1}{\sqrt{3}})^2) \qquad 2\text{-}2$$

As depicted in Figure 2.1 there is no point in the objective space better than those square points, therefore these points in the objective space which correspond to points in the design space are Pareto optimal front. $X \in [-4,4]$ is the design space and f_1 and f_2 are objective functions here.

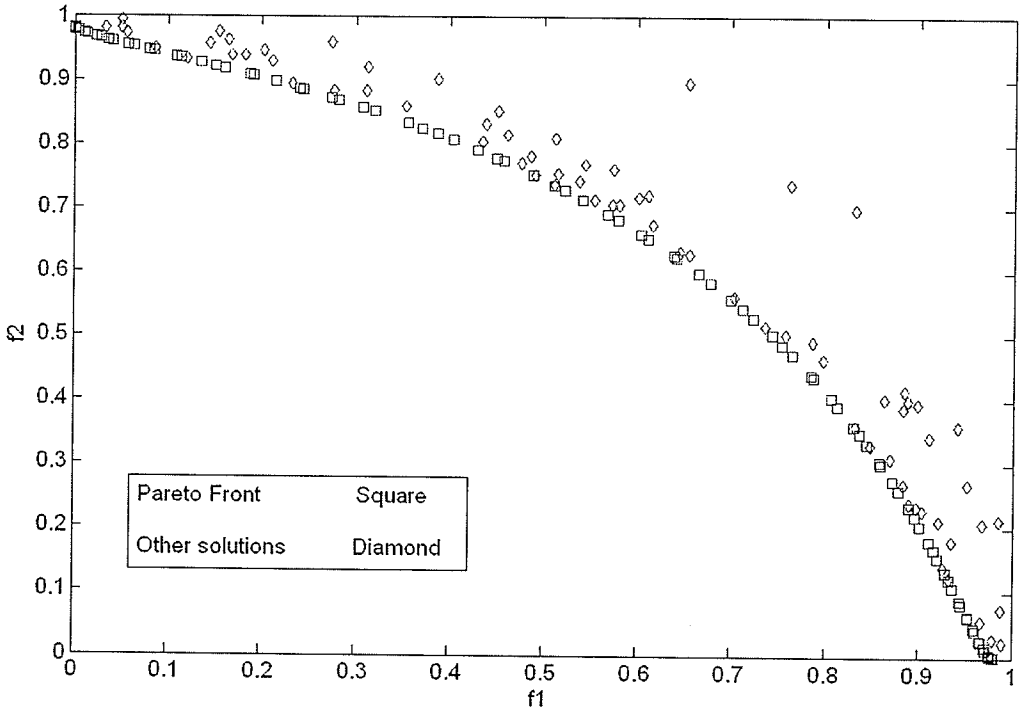


Figure 2.1 Illustration of Pareto front

MOO has been an intensively studied area. Given the difficulty of mathematical programming approaches for simulation based structural optimization, this review will focus on MOO methods based on metaheuristic.

The Niched Pareto Genetic Algorithm (GA) was introduced by Jeffrey Horn et al.[6] as an algorithm to find the Pareto optimal set. The performance and ability of Niched Pareto GA to find and maintain a diverse "Pareto optimal population" were demonstrated using two artificial problems and an open problem in hydro-systems. The study stated that the algorithm could be used to search any partially ordered space. A novel evolutionary approach to approximate the shape of the Pareto-optimal set of MOO problems was presented on the basis of a spatial predator-prey approach [7]. The main concern of the study was to find whether the predator-prey approach could be applied to MOO problems.

Knowles and Corne [8] proposed Pareto Archived Evolution Strategy (PAES) as the simplest possible non-trivial algorithm. The PAES uses a reference archive of previously found solutions to approximate dominance ranking of the current and candidate solution vectors. PAES is compared with Niched Pareto GA on a real world application from the telecommunications field. In Ref. [9], an extended 0/1 knapsack problem was taken as a basis to compare quantitatively four multi-objective evolutionary algorithms (EA). Furthermore, a new evolutionary approach to multi-criteria optimization, the Strength Pareto EA (SPEA), which combines several features of previous multi-objective EAs in a unique manner, was introduced. Simulation

results showed that SPEA could be very effective in sampling from the entire Pareto-optimal front and distributing the generated solutions over the trade-off surface. Moreover, SPEA clearly outperformed the other four multi-objective EAs on the 0/1 knapsack problem. Deb [10] investigated that features of a MOO problem may cause difficulties for an EA in converging to the True Pareto Front (TPF). Test problems containing features that were known to cause EAs difficulty in converging to the Pareto-optimal front were also developed. A systematic comparison of various EAs to MOO was carried out by Zitzler et al. [11] by using a design example,. This investigation concluded that elitism was an important factor for improving evolutionary multi-objective search. The mechanism of natural immune system and entropy principle in multi-objective evolutionary process were applied in Reference [12]. Furthermore, a strategy of preserving diversity in the population of multi-objective EA based on immune and entropy principles was introduced. Experimental results indicated that this strategy effectively preserved population diversity. Basic principles of evolutionary MOO algorithms were explained by Zitzler et al. [13] using an algorithm design example, which include issues such as fitness assignment, diversity preservation, and elitism in general. Moreover, a thorough discussion was undertaken on different techniques to implement these strongly related concepts. Other important aspects such as constraint handling and preference articulation were treated as well. In reference [14], an improved version, namely SPEA2, was proposed, which improved its predecessor with a fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method. A comparison between SPEA2 with SPEA and two other modern elitist methods, PAES and NSGA-II [15], demonstrated that SPEA2 perform better than SPEA and it has similar

performance like NSGA-II and in higher dimensional objective space SPEA2 is more successful

Non-dominated sorting-based multi-objective EA (MOEA), called non-dominated sorting genetic algorithm II (NSGA-II) was developed by Deb et al. [15]. A fast non-dominated sorting approach with low computational complexity was proposed in the work. The authors also proposed a selection operator which created a mating pool by combining the parent and offspring populations and selecting the best (with respect to fitness and spread) N solutions. The proposed NSGA-II, for most test problems, was able to find much better spread of solutions and better convergence near the TPF compared to existing methods.

Recently, Iorio and Li [16] incorporated a cooperative co-evolutionary algorithm idea into the non-dominated sorting procedure. The algorithm has demonstrated its ability to obtain well spread PF in comparison with the NSGA-II on some multiobjective optimization problems. A new MOO method was presented by Keerativuttitumrong et al. [17], which was based on the integration between two types of genetic algorithm: a multi-objective genetic algorithm (MOGA) and a co-operative co-evolutionary genetic algorithm (CCGA). The comparison between two GA's showed that MOCCGA was superior to MOGA in terms of the variety in solutions generated and the closeness of solutions to the TPF. Zeng et al. [18] applied the orthogonal and statistical optimal designs to speed up the computation of EA. Jin and Wong [19] proposed a novel E-dominance concept, an Adaptive Rectangle Archiving

(ARA) strategy, which does not need prior knowledge with this archiving technique. Coello and Becerra proposed a novel cultural algorithm, which uses evolutionary programming, Pareto ranking, and elitism to solve MOO problems [20]. Cultural algorithm also demonstrated comparable performance with NSGA-II. Reference [21] introduced a novel Pareto–frontier Differential Evolution (PDE) algorithm. Sato and his colleagues [22] enhanced multi-objective evolutionary algorithms with a distributed search based on local dominance. They chose NSGA-II and SPEA2 as two representatives' of multi-objective evolutionary algorithms and then enhanced NSGA-II and SPEA2 with distributed search based on local dominance. A new algorithm was also developed based on the idea of emulation of the immune system [23] behavior . This algorithm was also compared with NSGA-II on three test problems. Tran [24] developed the elitist non-dominated sorting GA for multi-objective optimization as a parameter-less version of NSGA-II. Subsequently, he investigated the performance of the parameter-less NSGA-II against the original NSGA-II. He concluded the parameter-less NSGA-II performed well compared to original NSGA-II.

The evaluation of different Pareto solutions has been a challenging task. Ang and his colleagues [25] investigated the multi-objective evolutionary algorithm performance measurement. The results indicated that while some performance indicators were conclusive and consistent, in some cases the 'diversity' indicator in a benchmark test should be included. Lu and Yen [26] proposed the rank-density-based genetic algorithm (RDGA). Statistical results show that RDGA is competitive with four other representative MOEAs in terms of keeping the diversity of the individuals along the tradeoff surface, tending to extend the PF to new areas and finding a well-

approximated Pareto optimal front. From the literature, it is also found that NSGA-II developed by Deb et al. [15] has been used intensively for comparison.

Archive-based hYbrid Scatter Search (AbYSS) [27] is a hybrid metaheuristic algorithm which follows the scatter search structure but using mutation and crossover operators coming from evolutionary algorithms. This algorithm combines ideas of three state-of-the-art multiobjective evolutionary algorithms. On one hand, an external archive is used to store the non-dominated solutions found during the search, following the scheme applied by PAES, but using the crowding distance of NSGA-II as a niching measure instead of the adaptive grid; on the other hand, the selection of solutions from the initial set to build the reference set applies the density estimation of SPEA2.

CellDE [28] present a new hybrid cellular genetic algorithm. They take MOCell as starting point, a multi-objective cellular genetic algorithm, and typical genetic crossover and mutation operators are replaced by the reproductive operators used in differential evolution. An external archive is used to store the non-dominated solutions found during the search process and the SPEA2 density estimator is applied when the archive becomes full.

Fast Pareto genetic algorithm (FastPGA) has been recently developed by H. Eskandari and C.D. Geiger [29]. FastPGA uses a new fitness assignment and ranking strategy for MOOs where each solution evaluation is relatively computationally- and/or financially-expensive. This is often the case when there are time or resource

constraints involved in finding a solution. A population regulation operator is introduced to dynamically adapt the population size as needed up to a user-specified maximum population size.

Optimized Multi-Objective Particle Swarm Optimization (OMOPSO) [30] is a particle swarm optimization algorithm for solving MOOs proposed by Reyes and Coello. Its main features include the use of the crowding distance of NSGA-II to filter out leader solutions, the use of mutation operators to accelerate the convergence of the swarm, and the concept of epsilon-dominance to limit the number of solutions produced by the algorithm.

Strength Pareto Evolutionary Algorithm (SPEA2) [31] was proposed by Zitler, Laumanns, and Thiele. In this algorithm, each individual has assigned a fitness value that is the sum of its strength raw fitness and density estimation. The algorithm applies the selection, crossover, and mutation operators to fill an archive of individuals; then, the non-dominated individuals of both the original population and the archive are copied into a new population. If the number of non-dominated individuals is greater than the population size, a truncation operator based on calculating the distances to the k-th nearest neighbor is used. In this way, the individuals having the minimum distance to any other individual are chosen.

In summary the research on metaheuristics-based MOO has been intensive and many good algorithms have been developed recently. All these methods, however, are developed for inexpensive objective functions. While for optimization involving expensive objective functions, new MOO methods need to be developed. In this work,

2.3 Size optimization

In the first approach, also known as "automatic dimensioning of structures", the only design variables are cross-sectional dimensions or transversal thicknesses. For example in trusses, rods cross section area is design variables, and weight minimization is the objective which satisfies a set of constraints.

2.4 Shape optimization

In shape optimization, the design variables are geometrical parameters that define the shape of the structure. These shape parameters are the coordinates of specific points, the poles. In 2D, these poles define the contour of the structure as a set of curves, for instance by using Lagrangian, Bézier or B-splines interpolations.

We can also model the geometry using lengths of segments, radii, angles, etc. For example in the case of a support structure: the objective is mass minimization and the constraint is critical Von Mises stress. Figure 2.3 shows load and boundary conditions. 10 independent design variables are used, modeling the support geometry through lengths and arcs of circles.

2.5 Topological optimization

In topological optimization, the aim is to determine the optimal shape of a structure by starting from a bulk of material, and gradually taking off removing the material which carries less stress. And the final structure must still satisfy the user-defined constraints.

the developed new algorithm, NSGQA, will be compared to the best of these recently developed to gauge the performance of NSGQA.

2.2 Structural optimization

Numerical method has greatly enhanced the design of mechanical components since the 1960's. Now Finite Element software's has an application in aeronautical, mechanical, naval and civil engineering. Meanwhile different fast and efficient optimization algorithm has been developed for wide range of mathematical problem. Structural optimization was introduced base on building a model in Finite Element software and using optimization algorithm to find the optimum for it. The design variables are usually geometrical, material and/or topological parameters, and the design should meet the constraint. Structural optimization is traditionally classified in three families following the nature of the variables involved.

- in size optimization, design variables are cross-sectional dimensions or thicknesses
- in shape optimization, the design variables are parameters which defines geometry of the structure
- in topological optimization design variables can modify the shape and the topology of the structure.

These categories are briefly discussed below.

Figure 2.4 illustrate topological optimization benchmark, the Michell truss problem, which is proposed by Reynolds [33] et al. They optimized it with the reverse adaptivity technique, the algorithm is as follows: first modeling the initial finite element problem, the method continues with a refinement of low (Von Mises) stress regions of the mesh by element subdivision. Then, low stress subdivided elements are removed and the process is repeated. The structures obtained after respectively 6, 42, 75 and 120 iterations of this process are represented in Figure 2.5. At the 120th iteration, only 8.8% of the whole (initial) area remains.

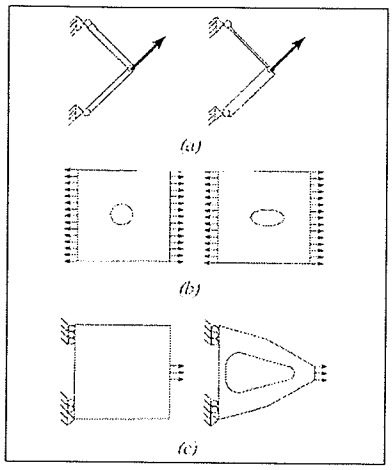


Figure 2.2 Distinction between design (a) shape (b) and topology (c) optimization [32].

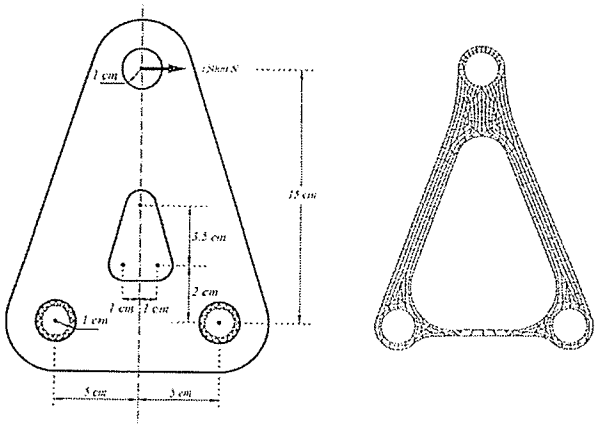


Figure 2.3 Shape optimization of a support: definition of the initial geometry (left) and solution obtained after 5 iterations of the optimization process (right) [32].

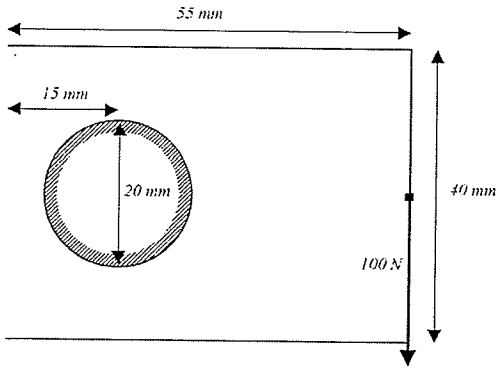


Figure 2.4 Definition of the Michell truss problem [Boundary condition the inner circular hole is fixed] [32].

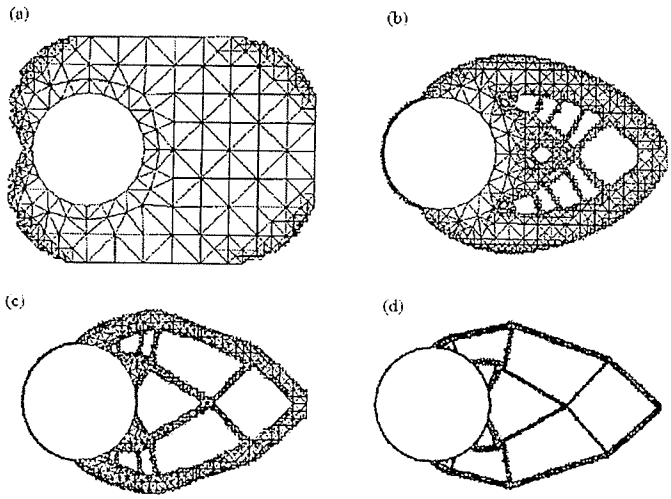


Figure 2.5 . Topological optimization applied to the Michell truss problem: results at iterations 6 (a), 42 (b), 75 (c) and 120 (d) [32].

From all three categories of structural optimization problems, this work targets at sizing and shape optimization problems. The sizing problem is conceptually simple. Therefore next section will be devoted to review of research in shape optimization.

2.6 Review of Research in Shape Optimization

Shape optimization was first originated from ship building. On June 20, 1809 [34]: Sir Georgw Cayley proposed "Solid of least resistance," a boat with the trout shape to minimize resistance in water. This is the first known publication on shape optimization.

The most application of shape optimization today is in aerospace industries, where the body or wing of an airplane is optimized in order to achieve minimum drag. Hague et al. [1] used nine separate search strategies on a parametric representation of airfoil surface to minimize drag for two- and three-dimensional situations. They also included comparisons with known variational solutions. In all cases compared results are in close agreement with previously obtained solutions. The nine separate search strategies are sectioning, pattern, magnification, steepest-descent, adaptive creeping, quadratic, Davidon's method, random point, and random ray search.

Vanderplaats[35] investigated multidisciplinary shape optimization of combined structural and aerodynamic design. For instance the optimum structural and aerodynamic designs of aircraft wing are not the same; therefore the optimum design is not the summation of optimum design of each part. Applications of shape

optimization in addition to structural design were investigated in [35]. Then he discussed mathematical and numerical aspect of multidisciplinary shape optimization. He concluded that optimization in multidisciplinary design needs distributed computing to make process time reasonable.

Fully automatic mesh generation and design model based on design-oriented geometric primitives are used in new optimization approach which is proposed by Botkin [36]. Subsequently Botkin combined shape optimization with fully automatic mesh generation. His study's shown that the algorithm exhibited good convergence in less than 10 finite element analyses after making design models and obtaining adequate mesh sensitivities from the geometric model operations for design optimization.

A finite element-based shape optimization program for large shape changes of three-dimensional shell structures was developed [37]. Adaptive mesh generation, substructuring, and linear and nonlinear optimization techniques along the commercial finite element analysis program (MSC/NASTRAN) are building blocks of this new shape optimization program. The program has the capability of optimizing shapes by allowing multiple edges to move.

The authors in reference [38] described a different approach to shape optimization based on the use of high-order p-type finite elements tightly coupled to a parameterized computational geometry module. Accurate results with much fewer finite elements, large shape changes without remeshing, and automatic adaptive analysis for accurate results at each step of the optimization process are the

advantages of their approach. They implemented their approach on sizing and shape optimization and compared with previous results from the literature.

Before 1995 topology and the shape optimization were molded separately. Maute [39] and his colleagues in their paper proposed an integrated model for topology and shape optimization. An analysis model was built based on the material distribution in the design model. Finite element was used for the parameterization of the analysis for topology optimization. Lagrangian curve or Bezier splines parameterization of the contours was used for shape optimization. The advantages of this algorithm were capability to find the optimum layout of a structure by using material based topology optimization, to include directly boundary variation techniques, and to improve the quality of the optimization results and the numerical efficiency of the optimization process.

Continuous optimization of the mechanical structural behavior using a flexible optimization model that may be changed during the optimization process is called interactive optimization. In reference [40] a concept for interactive shape optimization of plane and axisymmetric continuum structures is presented. This algorithm integrates the analytical formulation of hybrid mixed finite elements for structural and sensitivity analysis, adaptive mesh generation scheme for the automatic derivation of the analysis model, and non uniform rational B-splines (NURBS) curves for changing boundary representation.

Shape optimization of flat or curved 3D shell structures is presented by Lindby et al. [41] and it is composed of the geometric modeling and automatic meshing

capabilities of an existing parametric/associative CAD system. The calculation of analytical sensitivities with the discrete method and the implementation of the used shell elements are described in reference [41]. The validity of their method is demonstrated with the optimization of two complex 3D shell structures.

A general two-dimensional shape optimization by an improved growth-strain method was investigated by Han [42] . His study showed that growth-strain method could not provide reasonable optimized shapes for structures with two or more free surfaces such as structures with holes inside. Using the finite element method he built an automatic shape optimization system by the improved growth-strain method. This new method was verified by some examples in [42] in order to show the performance and application of the developed shape optimization system.

Lin and Chao [43] proposed a fully automated configuration optimization system for two-dimensional structures. Their algorithm is as follows: first creating a grey level image of a structure with minimum compliance by homogenization or material distribution methods, then performing image interpretation that converts the grey level image to a parameterized structural model ready to undergo shape optimization. Finally shape parameters are treated as design variables in a shape optimization that search for the optimum that satisfies all constraints. They concluded that this automated method increases the efficiency of the integrated topology and shape optimization for configuration designs.

In reference [44] an automated approach for simultaneous shape and topology optimization of shell structures is presented. They optimized shape and material

distribution of the shell structure simultaneously to maximize the stiffness of the shell. This algorithm uses a variable ground structure for topology optimization, since the shape of the shell is modified during process. The efficacy of this method was demonstrated using several examples.

Zhou and his colleagues [45] proposed simultaneous sizing, shape, and topology optimization in a single process because they believe that separate optimization methods pose a limitation on the design space and therefore prevents finding possible better designs since the interaction of sizing and shape variables with topology modification is excluded.

Free-form deformation technique suitable for aerodynamic shape optimization which is independent of grid topology is proposed in reference [46]. This method can be used on structured and unstructured computational fluid dynamics grids in the same manner. Better control of surface shape changes and decreasing the number of design variables by an order of magnitude are advantages of this method. The proposed technique is simple, compact, and efficient. They provided the complete formulation and aerodynamics shape optimization results.

Kegl and Marko [47] presented an approach to parameterization based shape optimization of statically loaded structures. A rational Bezier body is used to serve as the design element which is used to retrieve the nodal geometrical data of finite elements for shape parameterization. They concluded that their method can be applied for shape optimization of both continuous and discontinuous structures.

Shape optimization of two-dimensional anisotropic structures is done in reference [48] by the integration of boundary-element method with an optimization algorithm. The objective was to maximize stiffness by finding the optimum shape of a structure and to minimize the elastic compliance of the structure subject to constraints upon stresses, weight, and geometry. Since compliance, weight, and stresses are behaving nonlinearly they used feasible direction method along golden section method as the numerical optimization algorithm for the one-dimensional search. Hermitian cubic spline functions are used to represent boundary shapes that offer considerable advantages in fitting a wide range of curves and in the automatic remeshing process.

Sayed et al. [49] employed numerical optimization techniques for the shape optimization of fluid flows using CFD. To show the efficacy of the developed tool and its ability to produce results with reasonable CPU time, the shape optimization of an airfoil and S-shaped duct are studied with different numbers of design variables. Then they discussed the optimization and CPU time results.

Reference [50] presents a method for the structural optimization of variable thickness plates and free-form shells. First they considered topology, sizing and shape optimization independently. Then these tools are combined to form a robust and reliable fully integrated design optimization tool to obtain optimum designs. They implemented this method for flexible integrally stiffened plate and shell formulation to the design of stiffened plates and shells.

Reference [51] investigated aerodynamic shape optimization of aircraft wings under the effect of aeroelastic deformations at supersonic regime. Their approach is

composed of a high-fidelity aeroelastic analyzer and an aerodynamic optimizer. The shape optimizer is based on a "CAD-free" approach and an exact gradient method with a single adjoint state. They concluded that this method reduces the sonic boom production, while preserving the aerodynamic performances of flexible wings.

From the literature review, there are many approaches for shape optimization and a growing interest in integrating sizing, shape, and topology optimization into one optimization problem. The shape optimization on tooling, however, has not been found in the literature.

2.7 Summary of Literature Review

In section 2.1 different metaheuristic multi-objective optimization algorithms are reviewed. These studies highlight fitness assignment, diversity preservation, and elitism mechanism for better spread of solutions and better convergence near the TPF. None of these reviewed papers investigated the convergence to TPF for expensive optimization with a small numbers of function evaluations. Hence, this thesis is focused on this knowledge gap using a new metaheuristic method called genetic quantum algorithm, which provides diversity preservation. Non-dominated sorting mechanism, which presents fitness assignment and elitism mechanism, is integrated with genetic quantum algorithm.

Structural optimization, its categorization, and a detailed review on shape optimization have been reviewed. The literature review shows that multi-objective

metaheuristic optimization has not been applied to structural optimization with expensive objective functions.

Based upon the existing knowledge, this research is focused on developing a computationally efficient optimization algorithm for multiobjective structural optimization problems with expensive objective functions. The genetic quantum algorithm and the non-dominated sorting method are integrated to yield NSGQA. The performance of NSGQA is evaluated on classical multi-objective benchmark problems, a structural optimization problem, and a tool shape optimization problem.

CHAPTER 3

NON-DOMINATED SORTING GENETIC QUANTUM ALGORITHM

Based on the concept of Genetic Quantum Algorithm (GQA) [53]- [54], this chapter will discuss in detail the developed non-dominated sorting (NS) genetic algorithm (NSGA). The performance measurement of this algorithm and comparison with the state-of-the-art algorithms will be presented in Chapter 4, along with an application of NSGQA to a challenging structural optimization problem. Before introducing NSGQA, basic concepts of GQA will be described.

3.1 Genetic Quantum Algorithm

3.1.1 Representation

Genetic Quantum Algorithm [53] is inspired from the principles of quantum computation, and its superposition of states is based on qubits, the smallest unit of information stored in a two-state quantum computer. A qubit could be either in state "0" or "1", or in any superposition of the two as described below,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad 3\text{-}1$$

where α and β are complex numbers, whose square values denote the probability of the appearance of their corresponding state, "0" or "1", as determined by the constraint given below,

$$|\alpha|^2 + |\beta|^2 = 1 \qquad 3\text{-}2$$

This probabilistic representation implies that if there is a system of m qubits, the system can represent 2^m states simultaneously. At each observation, a qubit's quantum state collapses to a single state as determined by its corresponding probabilities. Consider the *j-th* individual in the *i-th* generation as defined by an m-qubit as below,

$$Q_j^i = \begin{bmatrix} \alpha_{j,1}^i & \alpha_{j,2}^i & \cdots & \alpha_{j,t}^i & \cdots & \alpha_{j,m}^i \\ \beta_{j,1}^i & \beta_{j,2}^i & \cdots & \beta_{j,t}^i & \cdots & \beta_{j,m}^i \end{bmatrix}, \qquad 3\text{-}3$$

where m is the number of qubits, i.e., the string length of the qubit chromosome, for $j = 1, 2, \ldots, n$, $t = 1,2,\ldots,m$, where n is population size. Since a qubit is a probabilistic representation, any superposition of states is simultaneously represented. For instance, if there is a three-qubit ($m = 3$) individual such as below,

$$Q_j^i = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} & \sqrt{\frac{2}{3}} & \frac{\sqrt{3}}{2} \end{bmatrix} \qquad 3\text{-}4$$

The possible states of the individual can alternatively be represented as,

$$\frac{1}{2\sqrt{6}}|000\rangle, \frac{1}{2\sqrt{2}}|001\rangle, \frac{1}{2\sqrt{3}}|010\rangle, \frac{1}{2}|011\rangle, \frac{1}{2\sqrt{6}}|100\rangle, \frac{1}{2\sqrt{2}}|101\rangle, \frac{1}{2\sqrt{3}}|110\rangle, \frac{1}{2}|111\rangle \quad 3\text{-}5$$

The number left to each state is the product of α and β values of corresponding bits in Equation 3-3, whose square is the probability of achieving such a state. For instance, states $|000\rangle$,$|001\rangle$,$|100\rangle$, and $|010\rangle$ can be realized with the probabilities 1/24, 1/8, 1/24 and 1/12 respectively. Consequently, the three-qubit system of Equation 3-4 carries the information of all eight states simultaneously.

Evolutionary computing with the qubit representation has a better characteristic of diversity than classical approaches since it can represent superposition of states. Only one qubit chromosome such as in Equation 3-4 is sufficient to represent all possible states in the initial stages of evolution, whereas in classical representation eight chromosomes are needed.

The binary state of the individual is first converted into an integer using the following Equations 3-6 and 3-7.

$$\text{Integer} = b_1 2^{m-1} + b_2 2^{m-2} + \ldots + b_{m-1} 2 + b_m \quad 3\text{-}6$$

$$\text{design variable} = \text{Integer} \times \left(\frac{UB - LB}{2^m - 1} \right) + LB \quad 3\text{-}7$$

The value of design variable is calculated using the above equation where *UB* is the upper limit of the design variable and *LB* is the lower bound limit of the design variable, and $b = b_1, b_2, b_3, .. b_m$ are either 0 or 1.

Hence, if the binary state of the individual is 101, and design variable's range is $0.0056 \leq$ radius ≤ 0.0222,

$$\text{Integer} = 1\times 2^2 + 0\times 2^1 + 1 = 5$$
$$\text{Design variable} = 5\times\left(\frac{0.0222-0.0056}{2^3-1}\right) + 0.0056 = 0.0175\,(\text{m}) \qquad 3\text{-}8$$

3.1.2 Algorithm of GQA

The general structure of GQA is described in [53]. In the initialization step, $P_Q(i)|_{i=0}$, $\alpha_{j,t}^0$ and $\beta_{j,t}^0$ of all Q_j^0 are initialized to be $1/\sqrt{2}$. This implies that each qubit chromosome $Q_j^i|_{i=0}$ in the initial population represents the linear superposition of all possible states with equal probability. A population of randomly generated binary instants $P_X(i) = \{X_1^i, X_2^i, ..., X_j^i, ..., X_n^i\}$ is then created by observing their corresponding qubit chromosome in $P_Q(i)$. In other words, each binary instant, X_j^i, of length m, is formed by selecting each bit using the probability pairs of qubit $\left(\left|\alpha_{j,t}^i\right|^2, \left|\beta_{j,t}^i\right|^2\right)$ of Q_j^i. Fitness of each instant X_j^i is then evaluated, and the best instant solution $X_b^i = \max_{j=1}^{n}\{F(X_j^i)\}$ is then selected and stored from among the binary instants of $X(i)$. Next a mating pool is generated from preserving a population of elites. In the next step, 'Update $P_Q(i)$,' the ith population of qubit chromosomes $P_Q(i)$ is updated by applying quantum rotation gates $U(i)$. $U(i)$ manipulates the population of qubit chromosomes $P_Q(i)$ and steers them towards the qubit with best instant solution. The appropriate quantum gate is usually designed in accordance with problems under consideration. This process is repeated in a "while" loop until convergence is achieved.

3.1.3 Quantum Gates Assignment

A quantum-inspired crossover operator, quantum rotation gate $U(\theta)$, updates the qubit values α and β towards the best performing individual in the population, so that better individuals will have higher probability of survival in the next iteration. Specifically, the t-th qubit value $(\alpha_{j,t}^{i}, \beta_{j,t}^{i})$ is updated as

$$\begin{bmatrix} \alpha_{j,t}^{i} \\ \beta_{j,t}^{i} \end{bmatrix}' = \begin{bmatrix} \cos(S \times |\Delta\theta|) & -\sin(S \times |\Delta\theta|) \\ \sin(S \times |\Delta\theta|) & \cos(S \times |\Delta\theta|) \end{bmatrix} \begin{bmatrix} \alpha_{j,t}^{i} \\ \beta_{j,t}^{i} \end{bmatrix} \qquad 3\text{-}9$$

where $\Delta\theta$ (magnitude of rotation) and S (sign of rotation angle) are determined from Table 3-1 and Table 3-2.

In Table 3-1 here $P(X)$ is the fitness, and $X_{b,t}^{i}$, where $X_b^i = \max_{j=1}^{n}\{P(X_j^i)\}$, and $x_{j,t}^{i}$ are the t-th bit of the best instant and the binary solution X_j^i respectively [53]. In Table 3-2, $x_{b,t}^{i}$ and $x_{j,t}^{i}$ are the t-th bit of the best instant solution X_b^i and the t-th bit of j-th binary instant X_j^i, respectively. And $\Delta\theta$ is the rotation angle that controls the speed of convergence. In order to demonstrate how the rotation gate helps GQA find better solutions, let us consider Figure 3.2. For example, in the case of $x_{j,t}^{i}$ =1, $x_{b,t}^{i}$ =0, if $F(X_j^i) \geq F(X_b^i)$ is false, in order to get a better chromosome, the probability of current solution $x_{j,t}^{i}$ for becoming a "0" should be larger, i.e. if the qubit $(\alpha_{j,t}^{i}, \beta_{j,t}^{i})$ is in the first or third quadrant, θ should rotate clockwise and if $(\alpha_{j,t}^{i}, \beta_{j,t}^{i})$ is in the second or forth quadrant, θ should rotate counter clockwise, as plotted in Figure 3.2.

Table 3-1 Look up table of $\Delta\theta^i_{j,t}$ [53].

$x^i_{j,t}$	$X^i_{b,t}$	$P(X^i_j) \geq P(X^i_b)$	$\Delta\theta^i_{j,t}$
0	0	false	$\|\Delta\theta_1\| = \pi$
0	0	true	$\|\Delta\theta_2\| = 0.73332\pi$
0	1	false	$\|\Delta\theta_3\| = 0.5608\pi$
0	1	true	$\|\Delta\theta_4\| = 0.76471\pi$
1	0	false	$\|\Delta\theta_5\| = 0.21177\pi$
1	0	true	$\|\Delta\theta_6\| = 0.90196\pi$
1	1	false	$\|\Delta\theta_7\| = 0.9294\pi$
1	1	true	$\|\Delta\theta_8\| = 0.87844\pi$

Table 3-2 Sign of $\Delta\theta^i_{j,t}$ is determined as below [53].

$S(\alpha^i_{j,t}, \beta^i_{j,t})$					
$x^i_{j,t}$	$X^i_{b,t}$	$\alpha^i_{j,t} \times \beta^i_{j,t} > 0$	$\alpha^i_{j,t} \times \beta^i_{j,t} < 0$	$\alpha^i_{j,t} = 0$	$\beta^i_{j,t} = 0$
0	0	1	-1	1	0
0	0	-1	1	1	0
0	1	-1	0	0	1
0	1	1	-1	0	1
1	0	-1	1	1	1
1	0	1	1	0	1
1	1	-1	1	0	1
1	1	0	1	1	1

1 *Initialize* a population of qubit chromosomes, $P_Q(i)\big|_{i=0} = \{Q_1^i, Q_2^i, ..., Q_j^i, ..., Q_n^i\}$, where n = population size , j = 1, 2, … , n and i = generation number.

2 *Determine* instants of qubit chromosomes in the population i, $P_X(i) = \{X_1^i, X_2^i, ..., X_j^i, ..., X_n^i\}$, Where X_j^i is an instant of j^{th} individual in generation i.

3 *Analysis step:*

a. Determine fitness $F(X_j^i)$.

b. Store the best instant $X_b^i = \max_{j=1}^{n}\{F(X_j^i)\}$.

c. Insert the pair set $\left(Q_j^i, F(X_j^i), X_j^i\right)$ to a mating pool up to population size, while preserving a population of elites.

4 *Update* $P_Q(i)$ using quantum gates $U(i)$, increment $i = i + 1$.

5 *While* not converged, go to Step 2; If convergence conditions satisfied, stop.

Figure 3.1 Algorithm of GQA

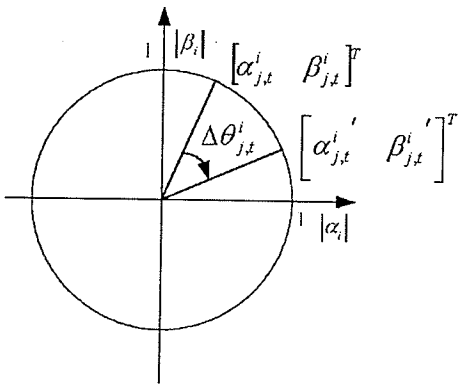


Figure 3.2. Consider a case in which $x_{j,t}^i$ =1, $x_{b,t}^i$ =0. So $(\alpha_{j,t}^i, \beta_{j,t}^i)$ should move toward higher probability of 0 state (increasing α) and consequently it should rotate clock wise.

The developed NSGQA method uses the non-dominated sorting concept originally from the non-dominated sorting GA (NSGA) method in [15]. NSGQA is thus to be compared with NSGA, together with a few other recently developed methods in the field. The next section will briefly introduce NSGA.

3.2 Non-Dominated Sorting Genetic Algorithm

NSGA-II [15] was advanced from its original version, NSGA. In NSGA-II, a non-dominated sorting approach is used for each individual to create a Pareto rank; and a crowding distance assignment method is applied to implement density estimation. In a fitness assignment between two individuals belonging to the same front, NSGA-II prefers the point with a lower rank value, or the point located in a region having fewer points. Therefore, by combining a fast non-dominated sorting approach, an elitism scheme and a parameter-less sharing method with the original NSGA, NSGA-II claims to produce a better spread of solutions in some testing problems [15]. In below section some key features of NSGA-II were described, as they are integrated into the proposed NSGQA.

3.2.1 Fast Non-dominated Sorting Approach

First sorting a population into different non-domination levels is described and next the modified method of sorting in order to decrease the computational time is presented. In conventional approaches, in order to identify solutions of the first non-dominated front in a population, each solution should be compared with other

solutions in the population to find if it is dominated or not. This requires comparisons of all solution pairs. At this stage, all individuals in the first non-dominated front are found. In order to find the individuals in the next non-dominated front, the solutions of the first front are discounted temporarily and the above procedure is repeated for the next rank. In the modified method, all ranks are recorded in the ascending order. In the next generation, which is expected to have a better average solution than the previous generation, all solutions in the first rank will be compared with new individuals. If the new individuals dominate all solutions in the first rank, then comparison will be stopped and these solutions will be stored as a new first rank. Otherwise these individuals will be added to the current first rank. Then the same procedure is repeated for fast rank assignment, or sorting.

3.2.2 Crowding Distance Assignment

Deb and his colleagues [15] presented a new operator called crowded-comparison, which has two main advantages: it has low computational complexity and it does not ask for any user defined parameter. The crowded-comparison operator and the density-estimation are the two parts of the crowding distance assignment which are described as follows.

3.2.2.1 Density Estimation

For each objective, the average distance of two points on either side is calculated to obtain the density of solutions around a particular solution in the population. In Figure 3.4 the average side length of the cuboids is the crowding

distance of the i-th solution in its front. A population is sorted for each objective function in the ascending order of the objective function value. Then the maximum and minimum values of each objective function are given an infinite distance value. The rest of solutions' absolute distance value with two neighboring solutions is then calculated. The same procedure is repeated for the other objective. So for each individual, the crowding-distance value is the sum of its objectives distance values. It should be noted that all objectives are normalized for the calculation of the crowing distance. Figure 3.4 shows the algorithm of calculating the crowding-distance for all solutions in arbitrary fronts.

For each individual i, set $dis\ (i) = 0$, $i = 1,2,3,\ldots,n$ (population size)

For each objective L_j, j=1,2,...,k, where k is the number of objectives

L' = sort (L_j), and L_j is an objective function value

$dis\ (1) = dis(l) = 0$, where 1 and l are boundary solutions for all objectives for the current population.

For i=2 to (l-1)

$dis\ (i) = dis\ (i) + (L_j'\ (t+1) - L_j'\ (t-1)\ )/(\ f_{Lj}^{max} - f_{Lj}^{min})$, where t is the location of L_j in sorted L', t=1,2,...,n

End

End

End

Figure 3.3 Crowding-distance calculation algorithm.

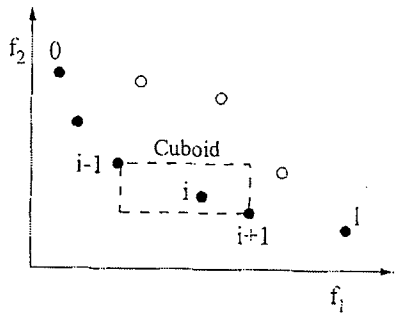


Figure 3.4 Crowding-distance calculation (points marked in filled circles are solutions of the same non-dominated front) [15].

3.2.2.2 Crowded-Comparison Operator

Selection is done by the crowded-comparison operator to achieve a well-spread Pareto frontier. There are two parameters for each individual in the population:

1) Non-domination rank *rank* (i), and

2) Crowding distance *dis* (i).

Between two different individual solutions, the lower rank is preferred but in the same rank the solution with a smaller distance will be selected. All main operators of NSGA-II are defined above. The NSGA-II algorithm consists of following main steps:

- Generate an initial population, whose objective values are evaluated and then sorted based on the non-domination. Thereafter a children population is created by using binary tournament selection, recombination, and mutation operators.
- Combine parent and children populations to form R_t
- F = *fast-non-dominated-sort* (R_t); $F = (F_1, F_2, \ldots)$
- *Crowding-distance-assignment* (F_i) calculates the crowding distances in F_i
- Sort the R_t in the ascending order using the crowded-comparison operator
- Choose the first N elements of the sorted population
- Use selection, crossover and mutation to create a new population from the chosen N elements; repeat until convergence.

Details on coupling NSGA with GQA are presented in the next section.

3.3 Non-Dominated Sorting Genetic Quantum Algorithm

The probabilistic/super-positional containment of information in quantum bit representation may have provided a fertile ground for efficient and robust search by GQA. It has been shown previously in [53] that GQA can perform better than GA on some famous test beds. Since finding a well spread Pareto optimal front needs a powerful tool to explore and exploit design space, it is expected that implementing GQA for MOO will lead to a new and high performance MOO algorithm.

NSGQA uses the fast non-dominated sorting approach, crowding distance assignment, and crowded-comparison operator like NSGA. Figure 3.5 illustrates NSGQA in more detail. In the initialization step, $P_Q(i)|_{i=0}$, $\alpha_{j,t}^0$ and $\beta_{j,t}^0$ of all Q_j^0 are initialized to $1/\sqrt{2}$. This implies that each qubit chromosome $Q_j^i|_{i=0}$ in the initial population represents the linear superposition of all possible states with equal probability. A population of randomly generated binary instants $P_X(i) = \{X_1^i, X_2^i, ..., X_j^i, ..., X_n^i\}$ is then created by observing their corresponding qubit chromosome in $P_Q(i)$. In other words, each binary instant, X_j^i, of length m, is formed by selecting each bit using the probability pairs of qubit $\left(\left|\alpha_{j,t}^i\right|^2, \left|\beta_{j,t}^i\right|^2\right)$ of Q_j^i. Fitness of each instant X_j^i is then evaluated. In next steps the fast non-dominated sorting, crowding distance assignment, and crowded-comparison operators are performed and the best instant solution $X_b^i = \max_{j=1}^{n}\{F(X_j^i)\}$ is then selected and stored from among the binary instants of $X(i)$. Next a mating pool is generated while preserving a population of elites. In the next step, 'Update $P_Q(i)$,' the i^{th} population of

qubit chromosomes $P_Q(i)$ is updated by applying quantum rotation gates $U(i)$. Selection is performed in order to identify next generation parents. This process is repeated in a "while" loop until convergence is achieved, as shown in Figure 3.5.

In single objective optimization (SOO), the criterion to find the best parent for the next generation is elitism which is related to the objective function value. For example in minimization, the smaller the objective function value for an individual in a population, the higher the chance of having offspring in the next population. Thus, in SOO the process of ranking the parent from current population for offspring reproduction is straightforward with elitism mechanism. The challenges arise for MOO, where there are multiple objectives. The question is how we can rank the parents for offspring reproduction of the next generation in metaheuristic algorithm. Initially this problem was solved by weighting each objective and then adding them together. This rough approximation does not provide one with required information about the quality of answer. To overcome this issue, different methods and algorithms have been proposed to rank the parent for the next generation of offspring based on different criteria.

For NSGQA, non-dominated sorting will find all the Pareto fronts of the current population. As we know all points in the Pareto front do not dominate each other. In other words, none of the individuals in one Pareto front can outperform the other individual in the same Pareto front. The sorting method in NSGQA works first by finding different orders of domination in the current population. This mean finding Pareto fronts of different orders in one population. The Pareto front which is ranked

first in domination will eventually dominate the second, third, and other Pareto fronts. That process is done using *fast-non-dominated-sort* algorithm as stated earlier.

After having the levels of domination, we have to rank each individual in their corresponding Pareto front. *Density estimation,* the average distance of two points on either side around a particular solution is calculated to obtain the density of solutions in the population. Therefore those individuals that have lower density will be ranked higher in the Pareto front in order to raise their chance of getting more points in their neighborboods. Therefore, we rank individuals in a population based on their domination level and for those individuals that have the same level of domination, the individuals with lower density will be ranked higher. Now that we have a mechanism to rank our individuals in a population, Individuals with the highest rank in the population will then be fed to the quantum gate update mechanism and the best chromosome of the best individual will be chosen and the quantum gate will try to rotate the qubits of the rest of individuals toward the best individual chromosome. This process will continue till convergence criteria are met. In this work, the convergence criteria include the number of generations or reaching specific values for objective function.

GQA and non-dominated sorting algorithm are described in detail in this chapter. And developed NSGQA which is inspired from GQA and non-dominated sorting is proposed and explained. The performance evaluation of NSGQA algorithm in Figure 3.5 is in the next chapter. Four mathematical multi-objective test beds and six multi-objective algorithms are chosen from the literature and their results are compared for

500 function evaluation which means for expensive evaluation. At the efficacy of the developed algorithm is checked on a FEM problem.

1 *Initialize* a population of qubit chromosomes, $P_Q(i)|_{i=0} = \{Q_1^i, Q_2^i, ..., Q_j^i, ..., Q_n^i\}$, where n = population size, $j = 1, 2, \ldots, n$ and i = generation number.

2 *Determine* instants of qubit chromosomes in the population i, $P_X(i) = \{X_1^i, X_2^i, ..., X_j^i, ..., X_n^i\}$, Where X_j^i is an instant of j^{th} individual in generation i.

3 *Analysis step:*

Determine fitness $F(X_j^i)$ for all individuals in population

4 *Fast non-dominated sorting*

Find all PFpoints and their corresponding rank

5 *Crowding distance assignment*

Calculate distances among all the individuals in the same rank

6 *Crowded-comparison operator*

Assign single objective values to each individual according to their rank and distance

7 *Store the best instant* $X_b^i = \max_{j=1}^{n}\{F(X_j^i)\}$

8 *Mating Pool*

Insert the pair set $\left(Q_j^i, F(X_j^i), X_j^i\right)$ to a mating pool up to population size, while preserving a population of elites.

9 *Update* $P_Q(i)$ using quantum gates $U(i)$, increment $i = i + 1$.

10 *Determine* instants of qubit chromosomes in the population i, $P_X(i) = \{X_1^i, X_2^i, ..., X_j^i, ..., X_n^i\}$, Where X_j^i is an instant of j^{th} individual in generation i using Roulette wheel selection.

11 *While* not converged, go to Step 3; if convergence conditions satisfied, stop.

Figure 3.5 Algorithm of NSGQA

CHAPTER 4

APPLICATION AND VALIDATION

4.1 Performance Evaluation of Non-Dominated Sorting Genetic Quantum Algorithm

In order to evaluate the performance of NSGQA, test beds presented in reference [15] are used in this thesis and results are compared with results form 6 other algorithms. It should be noted that we ran jMetal [61] code for all 6 other algorithm and for performance comparison jMetal [61] is used as well. jMetal stands for Metaheuristic Algorithms in Java, and it is an object-oriented Java-based framework aimed at the development, experimentation, and study of metaheuristics for solving multi-objective optimization problems. jMetal provides a rich set of classes which can be used as the building blocks of multi-objective metaheuristics. Taking advantage of code-reusing, the algorithms share the same base components, such as implementations of genetic operators and density estimators, thus facilitating the development of new multi-objective algorithms. Table 4-1, Table 4-2, Table 4-3, Table 4-4, and Table 4-5, show the 4 test beds used and the performance comparison for 30 runs average for all test beds using 5 different comparison mechanisms. The number of fitness function evaluation is 500 for all methods and this number is chosen because the main purpose of this research is to develop a multiobjective algorithm for computational expensive problems. Here we describe that how we can compare two different MOO algorithms on a same multi-objective problem.

4.1.1 Performance Measurement

MOO performance measure is totally different from SOO. Because *MOO* optimum solution is a set of solutions, not a single one, we have to consider different aspects of Pareto front (PF) like spread, diversity and how close the solutions to true Pareto front (TPF) are. Below are five different criteria which have been widely used in literature.

4.1.1.1 Spread

Well spread PFs are desired for a set of solutions. To get that, Euclidean distance between any two neighbor solutions in non-dominated solutions is calculated and then the average of this distance, $\overline{d}$, can be obtained. Then we should find the extreme solutions (in the objective space). The non-uniformity in the distribution, Δ, is calculated using the following Equation 4-1 [15]:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} \left| d_i - \overline{d} \right|}{d_f + d_l + (N-1)\overline{d}} \qquad 4\text{-}1$$

Here, the parameters d_f and d_l are the Euclidean distances between the extreme solutions and the boundary solutions of the obtained non-dominated set or they mean minimum and maximum for objective of L_j respectively. The parameter $\overline{d}$ is the average of all distances d_i, which are $dis\ (i,L_j)$ in crowding-distance calculation, and i =1, 2, …, (N-1). With N solutions, d_i's are consecutive distances. Figure 4.1 demonstrates how this performance measure works. The lower Δ, the better the spread of Pareto front.

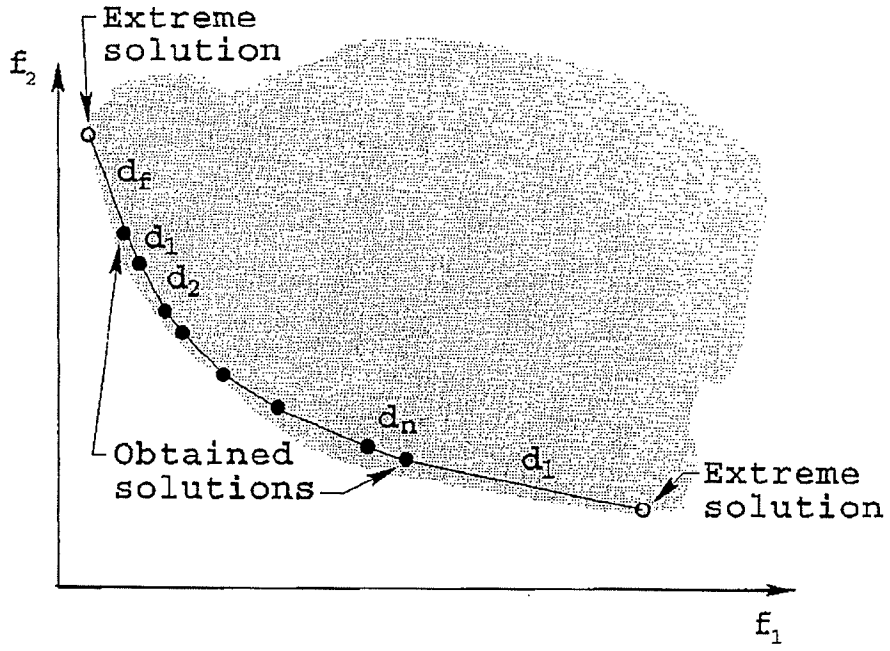


Figure 4.1 Diversity metric [15].

4.1.1.2 Generational Distance

The generational distance quality indicator was introduced by Van Veldhuizen and Lamont [58] for measuring how far the elements are in the found non-dominated pareto front from those in the TPF set. It is defined as:

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \qquad 4\text{-}2$$

where n is the number of vectors in the set of non-dominated solutions found so far and d_i is the Euclidean distance (measured in objective space) between each of these solutions and the nearest member of the TPF set. A value of $GD = 0$ indicates that all found non-dominated Pareto front are in the TPF. In order to get comparable results, non-dominated sets are normalized before calculating this distance measure. In other word this metric calculate the error between found PF and TPF.

4.1.1.3 Inverted Generational Distance

The inverted generational distance indicator was introduced by Van Veldhuizen and Lamont [58] for measuring how far the elements are in the TPF set from those in the set of non-dominated vectors found. It is defined as:

$$IGD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \qquad 4\text{-}3$$

where n is the number of vectors in the Pareto optimal set and d_i is the Euclidean distance (measured in objective space) between each of these solutions and the nearest member of the set of the non-dominated vectors found. A value of $IGD = 0$ indicates

that all the generated elements are in the PF and they cover all the extension of the Pareto front. In order to get comparable results, non-dominated sets are normalized before calculating this distance measure.

Here is the description to clarify the difference between *GD* and *IGD*. When we measure the Generational Distance (*GD*) of a obtained PF, we compute the distance between each solution contained into the PF and it closest solution contained in the TPF. While when we measure the Inverted Generational Distance (*IGD*), we compute the distance between each solution contained in the TPF and the closest solution in our PF (just the same but in both directions, from obtained PF to the TPF and from the TPF to obtained PF). Then, suppose that you have two obtained PF's of a given problem:

- A PF with only two points.
- A PF containing the two points of the previous PF and many other non-dominated points.

It is clear that from the multi-objective perspective the second PF is preferable to the first one. However, *GD* may not reflect it because it only measure the distances in the way from PF to TPF. Meanwhile *IGD* will reflect that the second one is better because the distances are measured from TPF to the obtained PF.

4.1.1.4 Hyper-volume

Hyper-volume quality indicator [9] calculates the volume (in the objective space) covered by members of a non-dominated set of solutions Q (the region enclosed into the discontinuous line in Figure 4.2, $Q = \{A,B,C\}$ for problems where all objectives

are to be minimized. Mathematically, for each solution $i \in Q$, a hypercube v_i is constructed with a reference point W and the solution i as the diagonal corners of the hypercube. The reference point can simply be found by constructing a vector of worst objective function values. Thereafter, a union of all hyper-cubes is found and its hyper-volume (HV) is calculated:

$$HV = volume\left(\bigcup_{i=1}^{|Q|} v_i\right) \qquad 4\text{-}4$$

Algorithms with larger values of HV are desirable. Since this metric is not free from arbitrary scaling of objectives, normalized objective function values are used to evaluate the metric.

4.1.1.5 Generalized Spread

The generalized spread metric calculates the distance between two consecutive solutions, which works only for two-objective problems. Reference [65] makes an extension by calculating Equation 4-5, the distance from a point to its nearest neighbor:

$$\Delta(PF,TPF) = \frac{\sum_{i=1}^{m} d(e_i,PF) + \sum_{X \in TPF} \left| d(X,PF) - \overline{d} \right|}{\sum_{i=1}^{m} d(e_i,PF) + |TPF|\overline{d}} \qquad 4\text{-}5$$

where PF is the current optimal Pareto Front or found optimal Pareto Front, TPF is the True Pareto Front, $\{e_1, \ldots, e_m\}$ are m extreme solutions in TPF and

$$d(X,S) = \min_{Y \in PF, Y \neq X} \left\| F(X) - F(Y) \right\|^2,$$
$$\overline{d} = \frac{1}{|TPF|} \sum_{X \in TPF} d(X,PF) \qquad 4\text{-}6$$

If the achieved solutions are well distributed and include those extreme solutions, $\Delta(PF, TPF) = 0$.

Here $|TPF|$ mean number of point in true pareto front and $d(e_{i,}, PF)$ is distance between m extreme solutions of TPF and PF solution.

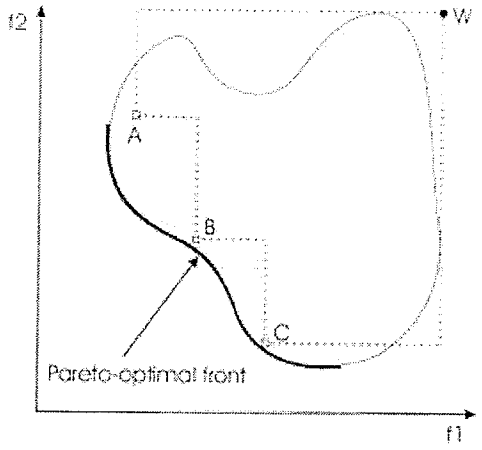


Figure 4.2 This illustrates how v_i is calculated [66]

Table 4-1 Test bed functions

Problem	n	Variable bounds	Objective function	Optimal solutions
FON	3	[-4,4]	$f_1 = 1-\exp(-\sum_{i=1}^{3}(x_i - \frac{1}{\sqrt{3}})^2)$ $f_2 = 1-\exp(-\sum_{i=1}^{3}(x_i + \frac{1}{\sqrt{3}})^2)$	$x_1 = x_2 = x_3$ $\in [-1/\sqrt{3}, 1/\sqrt{3}]$
KUR	3	[-5,5]	$f_1 = \sum_{i=1}^{n-1}(-10\exp(-0.2\sqrt{x_i^2 + x_{i+1}^2}))$ $f_2 = \sum_{i=1}^{n}(\lvert x_i\rvert^{0.8} + 5\sin x_i^3)$	Not available
ZDT6	10	[0,1]	$f_1 = 1-\exp(-4x_1\sin^6(6\pi x_1))$ $f_2 = g(x)[1-(f_1(x)/g(x))^2]$ $g(x) = 1+9[(\sum_{i=2}^{n} x_i)/(n-1)]^{0.25}$	$x_1 \in [0,1]$ $x_i = 0,$ $i = 2,\ldots,n$
SCH	1	$[-10^3, 10^3]$	$f_1(x) = x^2$ $f_2(x) = (x-2)^2$	$x \in [0,2]$

Table 4-2 FON test bed

FON 500	GeneralizedSpread			GenerationalDistance			Hypervolume			Inverted GenerationalDistance			Spread			Sum of Rank	Overall ranking
NSGQA	0.9368	0.0056	6	0.0097	0.0001	3	0.1761	0.001	4	0.0173	0	6	0.9146	0.0021	7	26	6
Abyss	0.9731	0.01	7	0.009	0	2	0.2371	0.0007	2	0.004	0	2	0.8887	0.0093	6	19	3
CellDE	0.7731	0.0076	4	0.0463	0.0003	6	0.1229	0.0018	6	0.0098	0	5	0.7045	0.0102	2	23	5
FastPGA	0.8307	0.0234	5	0.0057	0	1	0.2539	0.0002	1	0.0029	0	1	0.8255	0.025	5	13	2
NSGAII-Real	0.6669	0.0213	1	0.0215	0.0001	4	0.1888	0.0024	3	0.0048	0	3	0.6753	0.0104	1	12	1
OMOPSO	0.6687	0.0553	2	0.1575	0.005	7	0.0554	0.0072	7	0.022	0.0001	7	0.7554	0.0279	3	26	6
SPEA2	0.733	0.0312	3	0.0349	0.0002	5	0.1482	0.001	5	0.0069	0	4	0.7628	0.0108	4	21	4

Table 4-3 Kur test bed

KUR 500	GeneralizedSpread			GenerationalDistance			Hypervolume			Inverted GenerationalDistance			Spread			Sum of Rank	Overall ranking
NSGQA	0.8661	0.0246	6	0.0211	0.0003	4	0.6829	0.0108	5	0.003	0	4	0.8152	0.0135	5	24	6
Abyss	0.9471	0.0176	7	0.0053	0	1	0.7216	0.0048	6	0.0018	0	2	0.9165	0.0133	7	23	5
CellDE	0.657	0.0141	1	0.066	0.0009	6	0.5183	0.0021	2	0.0058	0	6	0.6637	0.0116	1	16	1
FastPGA	0.8324	0.0283	5	0.0068	0.0001	2	0.7305	0.004	7	0.0016	0	1	0.8311	0.011	6	21	4
NSGAII-Real	0.7413	0.015	4	0.0178	0.0001	3	0.6743	0.002	4	0.0027	0	3	0.7339	0.0216	4	18	2
OMOPSO	0.6762	0.0102	2	0.1031	0.0022	7	0.3691	0.0188	1	0.0101	0	7	0.6765	0.0031	2	19	3
SPEA2	0.6832	0.0209	3	0.0226	0.0001	5	0.6255	0.003	3	0.0035	0	5	0.7165	0.0106	3	21	4

Table 4-4 ZDT6 test bed

ZDT6 500	GeneralizedSpread			GenerationalDistance			Hypervolume			Inverted GenerationalDistance			Spread			Sum of Rank	Overall ranking
NSGQA	0.962	0.0014	6	1.9849	0.0544	1	0	0	1	0.1898	0.0001	1	0.9377	0.0008	1	10	1
Abyss	0.975	0.0035	7	3.7474	2.0379	9	0	0	1	0.1991	0.0002	2	0.9827	0.0009	6	25	6
CellDE	0.9385	0.0005	1	2.7512	0.068	3	0	0	1	0.23	0.0002	6	0.9371	0.0004	1	12	2
FastPGA	0.9568	0.0008	4	2.6974	0.0479	2	0	0	1	0.2103	0.0001	3	0.9511	0.0005	4	14	3
NSGAII-Real	0.9582	0.0017	5	2.823	0.2206	7	0	0	1	0.2254	0	4	0.9547	0.001	5	22	5
OMOPSO	0.9552	0.0006	3	2.917	0.2067	8	0	0	1	0.2416	0.0007	7	0.9509	0.0005	3	22	5
SPEA2	0.9434	0.0011	2	2.8149	0.1656	6	0	0	1	0.2269	0.0001	5	0.9412	0.0004	2	16	4

Table 4-5 SCH test bed

SCH 500	GeneralizedSpread			GenerationalDistance			Hypervolume			Inverted GenerationalDistance			Spread			Sum of Rank	Overall ranking
NSGQA	0.7473	0.1255	4	0.0223	0.002	1	0.3869	0.1398	1	0.0315	0.0006	2	0.7591	0.0964	3	11	2
Abyss	1.1184	0.0404	7	1.8178	31.8465	6	0.5886	0.0497	2	0.1456	0.1618	6	1.1513	0.0216	7	30	5
CellDE	0.6929	0.1116	3	0.4026	0.3674	4	0.3159	0.0716	5	0.0595	0.0023	4	0.7807	0.0589	4	20	4
FastPGA	0.5627	0.151	1	0.4857	0.736	5	0.2728	0.0737	6	0.0655	0.0038	5	0.6392	0.1136	2	19	3
NSGAII-Real	0.5822	0.1189	2	0.2957	0.0798	3	0.5123	0.029	3	0.0278	0.0001	1	0.6216	0.0668	1	10	1
OMOPSO	0.9578	0.008	6	67.8515	8998.4867	7	0.0727	0.0529	7	5.077	47.4228	7	0.9588	0.0101	6	33	6
SPEA2	0.7913	0.0794	5	0.1357	0.0577	2	0.4371	0.0594	4	0.0355	0.0003	3	0.8043	0.047	5	19	3

4.1.2 Test Results

Brief description for each of these algorithm are described in 2.1 and the code for all of these algorithm are available in jMetal [61].

In Table 4-1 test bed functions are described. Since development of this algorithm is for expensive function evaluation, 500 fitness function evaluation is chosen in order to measure the performance of proposed method for low number of function evaluation. Optimization and comparison results are listed in 3 columns under each criterion; the first is the 30 run average value for the criterion, next is the variance of these 30 runs and the last one is the ranking of this algorithm for that criterion among 7 available algorithms. As described earlier we have 5 performance measurements and 7 algorithms to compare with each other. Since it is not possible to find an algorithm which can be best with respect to all 5 performance criteria we decided to rank each algorithm for each performance measurement with 1 means the best and 7 the worst. By adding the ranks of performance measures for each MOO, we can evaluate the algorithm performance in comparison with others in the test bed.

In order to demonstrate the performance of these algorithms, we also show typical simulation results of NSGQA-II on the test problems KUR [15], FON [15], SCH [15], and ZDT6 [15]. Test bed characteristics are described in Table 4-6.

As one can see in Table 4-7, NSGQA ranked 1 in two of the benchmarks and ranked the lowest in the other 2. As structural optimization tends to involve relatively

large dimensions, ZDT6, which has 10 design variables, is of more relevance and importance than other problems.

For KUR and FON the NSGQA ranks the lowest, and a plot of the solution distribution for all algorithms is provided in Figure 4.3 and Figure 4.4 for a closer look at the solution quality. Problem KUR has three discontinuous regions in the Pareto-optimal front. As one would see in Figure 4.3, NSGQA and Abyss yields better optimal and denser Pareto solutions on the entire solution space than others do, which just covers part of the solution space. Figure 4.4 depicts the convergence ability of NSGQA, as compared to all other algorithms on the function FON. NSGQA, FastPGA, and Abyss have similar performance and converge close to TPF.

The solution distribution for the other two functions is also plotted in Figure 4.5 and Figure 4.6. As described in Table 4-6, ZDT6 is a unimodal non-uniformly distributed objective with a high number of design variables. One can see from Figure 4.5 that all the algorithms have difficulty to approach to TPF. But among all algorithms, NSGQA (visually) spreads well and close to TPF.

The SCH function is convex. NSGQA has found well spread solutions in comparison with all other algorithms.

From the above comparison, NSGQA demonstrates good performance over the state-of-the-art MOO algorithms from both the spread and solution quality perspectives. Next the performance of NSGQA will be evaluated on a practical structural optimization problem.

Table 4-6 Characteristics of chosen test bed functions

Test Beds	Characteristics
ZDT6	ZDT6 is a unimodal non-uniformly distributed objective space, both orthogonal and lateral to the Pareto-optimal front. It has been proposed to test the algorithms' ability to find a good distribution of points even in this case.
KUR	Kur is a multi-modal function in one objective and has pair-wise variable interactions in the other. The Pareto-optimal front is not connected and has an isolated point as well as concave and convex regions.
SCH	SCH is convex and there is a trade-off between the two objectives.
FON	Fon is non-convex.

Table 4-7 Summary of ranking for all test beds.

	FON 500	SCH 500	ZDT6 500	KUR 500
NSGQA	6	2	1	6
Abyss	3	5	6	5
CellDE	5	4	2	1
FastPGA	2	3	3	4
NSGAII-Real	1	1	5	2
OMOPSO	6	6	5	3
SPEA2	4	3	4	4

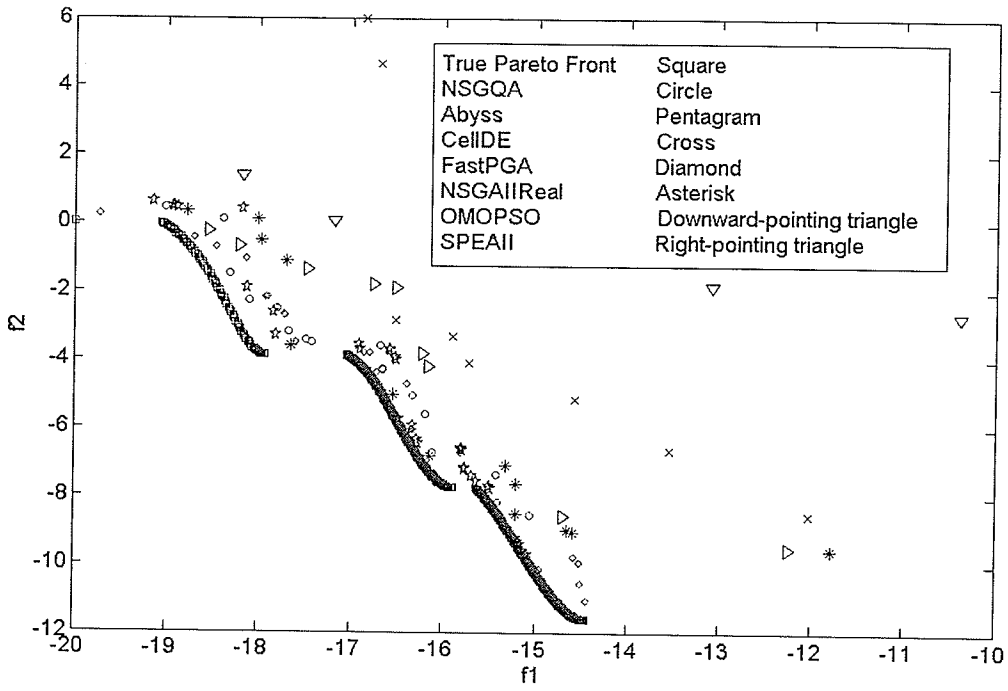


Figure 4.3 KUR test bed

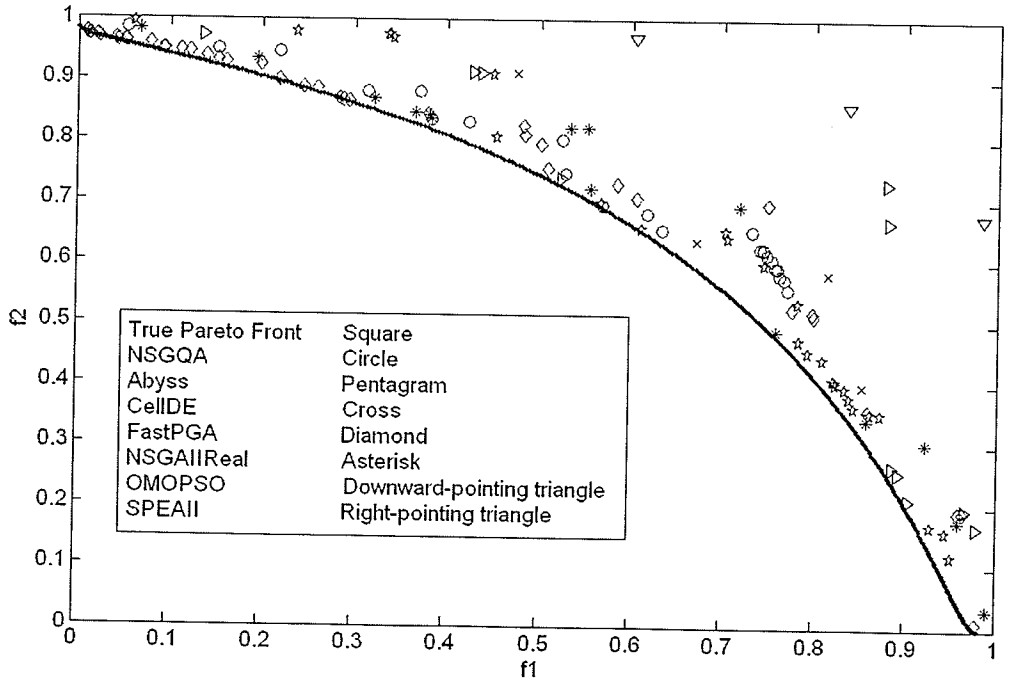


Figure 4.4 Non-dominated solutions of NSGQA and all other algorithm on FON.

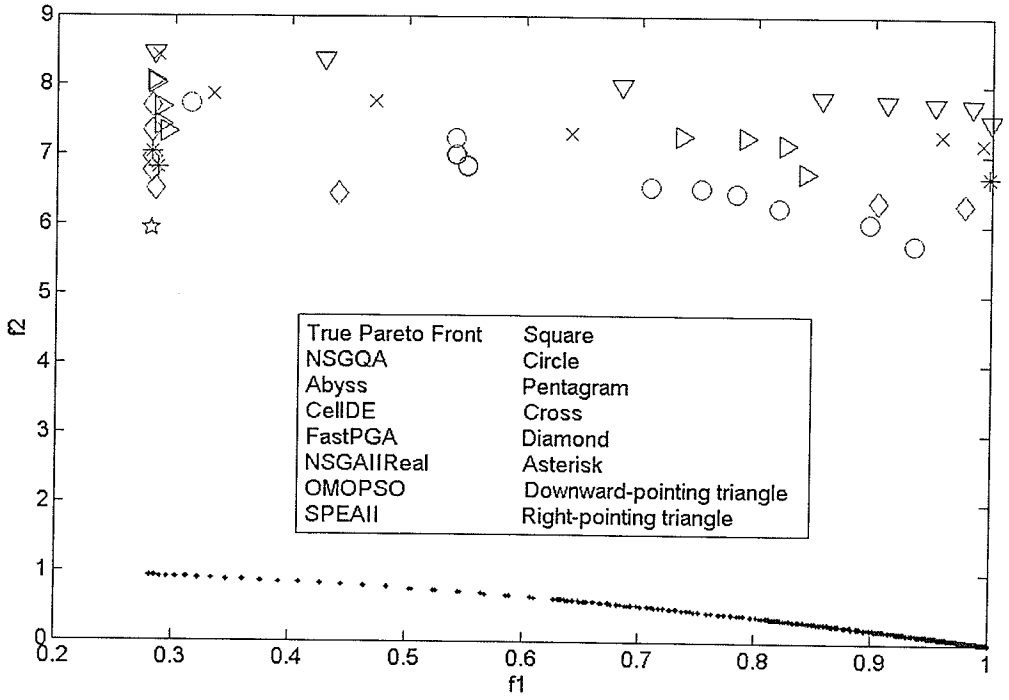


Figure 4.5 Non-dominated solutions of NSGQA and all other algorithm on ZDT6.

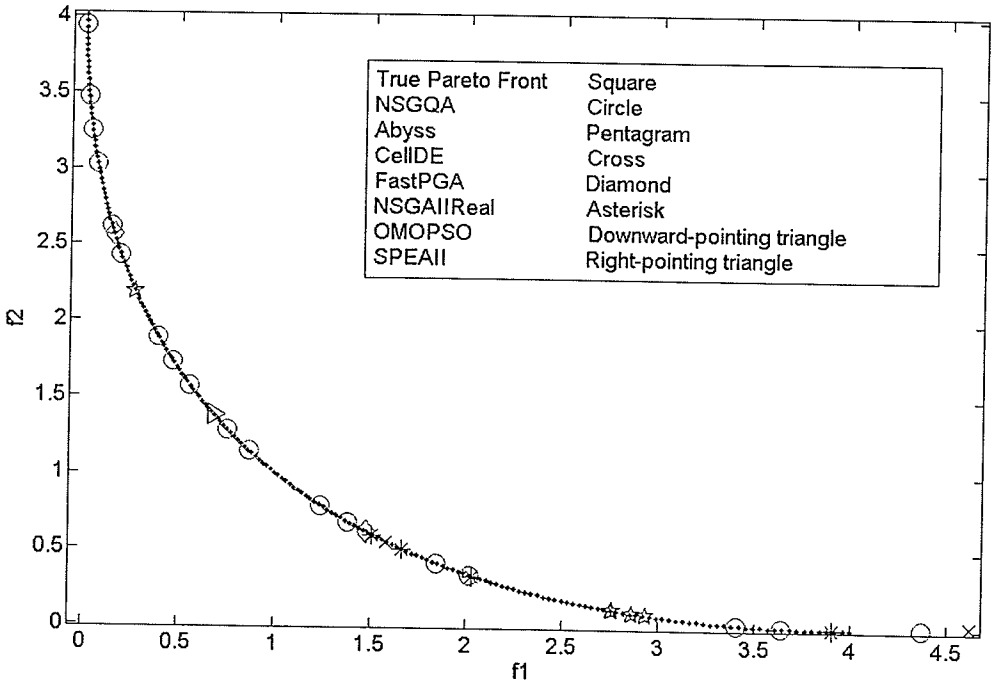


Figure 4.6 Non-dominated solutions of NSGQA and all other algorithm on SCH.

4.2 Performance of NSGQA on an Engineering Problem

A 47-bar truss transmission power tower [58] shown in Figure 4.7 is often used as a test problem in structural size and shape optimization (with single objectives). The truss structure consists of 22 nodes and 47 members, and the load is for the case of two power lines passing the tower at an angle. F_X= 26.68 kN and F_y= -62.27 kN on node 22 and 17. There is a stress constraint of 137.89 (MPa) in tension and -103.42 (MPa) in compression for each bar. The truss tower is symmetric with respect to Y axis. Nodes 1 and 2 are not moving along Y axis but they are free to move along X axis. The coordinates of nodes 15, 16, 17 and 22 are kept fixed. Therefore there are 17 independent shape variables and 27 independent size variables. More details of design variables are in Table 4-8.

The objectives for the 47-bar truss problem include the total weight of truss and the sum of deflections of node 22 in X and Y directions. Regarding to constraints, the maximum stress in tension and minimum stress in compression for all bars are determined as follows:

$$\begin{aligned} \sigma_{\max\, tension} &\le 137.89(MPa) \\ \sigma_{\max\, compression} &\ge -103.42(MPa) \end{aligned} \qquad 4\text{-}7$$

Table 4-8 lists detailed constraints among design variables, ranges of each variable, and the material property of the structure. For stress and deflection analysis, ANSYS is used. Hence for each fitness function evaluation a 47-bar ANSYS APDL code should be called from Matlab in order to run the multi-objective constrained optimization using NSGQA.

Table 4-8 Design variables detail for the 47-bar truss tower problem

Objectives	1 Minimization of Weight 2 Minimization of Node 22 Deflection under applied load
Shape variables	$X_2 = -X_1$, $X_4 = -X_3$, $Y_4=Y_3$, $X_6= -X_5$, $Y_6=Y_5$, $X_8= -X_7$, $Y_8= -Y_7$, $X_{10}= -X_9$, $Y_{10}=Y_9$, $X_{12}= -X_{11}$, $Y_{12}=Y_{11}$, $X_{14}= -X_{13}$, $Y_{14}=Y_{13}$, $X_{20}= -X_{19}$, $Y_{20}=Y_{19}$, $X_{21}= -X_{18}$, $Y_{21}=Y_{18}$
Size variables	$A_1=A_3$, $A_2=A_4$, $A_6=A_5$, $A_9=A_8$, $A_{11}=A_{12}$, $A_{13}=A_{14}$, $A_{16}=A_{15}$, $A_{17}=A_{18}$, $A_{19}=A_{20}$, $A_{21}=A_{22}$, $A_{23}=A_{24}$, $A_{25}=A_{26}$, $A_{29}=A_{30}$, $A_{32}=A_{31}$, $A_{34}=A_{35}$, $A_{37}=A_{36}$, $A_{39}=A_{40}$, $A_{42}=A_{41}$, $A_{44}=A_{45}$, $A_{47}=A_{46}$, A_{43}, A_{38}, A_{33}, A_{28}, A_7, A_{10}, A_{27}
Design sets	$A_i \in S = \{0.1, 0.2, 0.3, ..., 4.9, 5.0\} \quad i = 1,...,47$
Material properties	Modulus of elasticity, E 206.89 G (Pa) Density, ρ (8304) Kg/m^3

It takes about 10 hrs on a Pentium 4 with 1.5 GB RAM desktop computer for one optimization run, with the generation number of 200 and population size of 50. Hence the number of function evaluations is 10000 and time for each evaluation is 3.6 seconds. The length for each qubit is 16 and total number of points found in the PF is 67. The optimal results show that our algorithm is capable of performing MOO on a high dimensional engineering problem with 44 design variables and constraints, while high dimensional MOO problems are generally difficult for MOO algorithms. Figure 4.8 is the Pareto optimal front which was plotted with respect to weight and deflection. In Table 4-9, the weight, deflection, maximum tensile stress, and minimum compressive stress of three chosen Pareto solutions are listed. Among the chosen solutions, one has the maximum deflection, another has the maximum weight, and the last is at the middle of the Pareto front. The optimal tower designs of these solutions are shown in the last row of Table 4-9.

While attempting to find comparable problems and solutions from the engineering design literature, we could not find any literature that solves a MOO problem of this scale (44 design variables with many constraints). Nor could we find any literature solves the 47-bar truss problem with multiple objectives. This has brought difficulty in comparing our results with others, which may not exist at the moment.

In summary this chapter entitles the performance measurement of NSGQA on 4 classical benchmarks in MOO and one multi-objective structural optimization benchmark. The results of 4 classical benchmarks are compared using 5 measuring tool collected form 4 different papers.

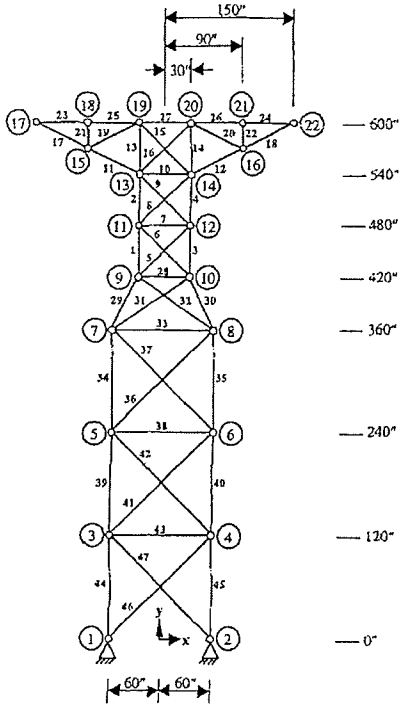


Figure 4.7 47-bar planar tower, initial geometry [58].

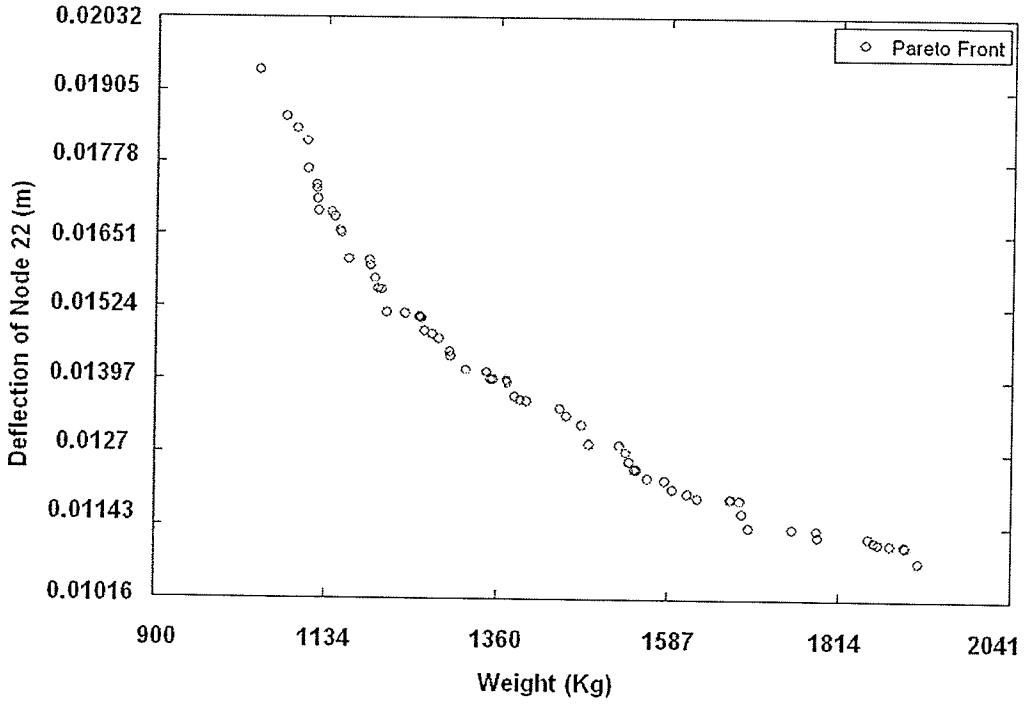
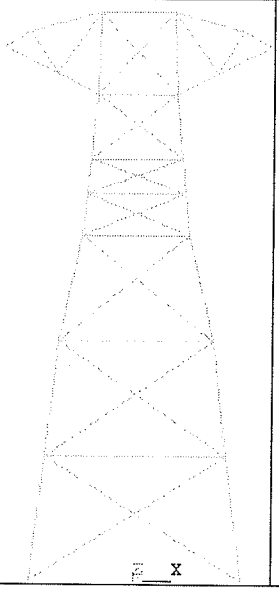
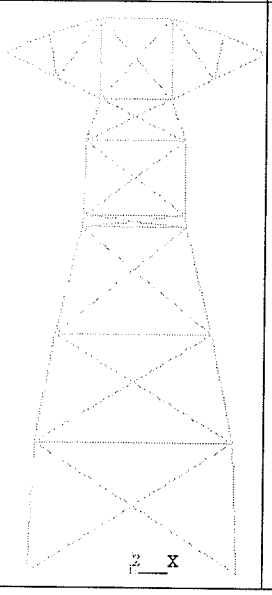
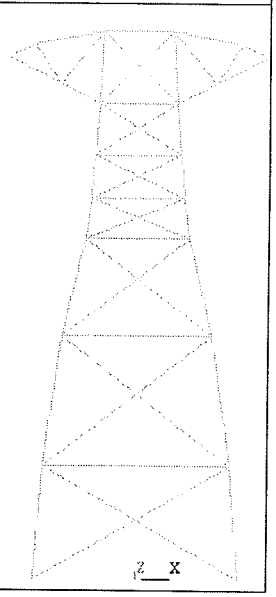


Figure 4.8 Non-dominated solutions of NSGQA for the 47-bar truss problem.

Table 4-9 Shape and performances of three chosen solutions for the 47-bar truss tower problem.

	Case1	Case2	Case3
Maximum Tensile Stress (MPa)	82.53	53.71	70.77
Minimum compressive Stress (MPa)	-94.71 M	-56.70 M	-83.55
Weight (Kg)	1041.6	1820.8	1372.7
Deflection of Node 22 (m)	0.01941	0.01155	0.01397
			

CHAPTER 5
WARPAGE TOOL SHAPE OPTIMZATION FOR MINIMIZATION OF WARPAGE IN AUTOCLAVE CURED COMPOSITE PART

5.1 Introduction

In this chapter applicability of shape optimization to composite manufacturing is evaluated. Composite parts used in aircraft, vary greatly in size and shape and are typically cured in batch loads in autoclaves. During autoclave manufacturing, composite prepreg, tape or fabric, consisting of a single layer of reinforcing fibers impregnated with B-staged thermoset resin, are cut to desired shape and stacked on a tool to form the desired part. The plies are oriented in pre-determined directions to yield desired mechanical properties. The staked plies are vacuum bagged as shown in Figure 5.1 and then loaded into an autoclave. The autoclave is subjected to a cure cycle illustrated in Figure 5.2. At time 0 the pressure is 0; as the process begins the pressure is applied and held constant (45 or 85 Psi) during the process and then reduced to 0 after the hold period. Temperature is initially the room temperature and it increases to 180 ^{o}C and it will cool down as desired degree of cure is achieved. Under the action of pressure and temperature, the polymer resin melts, flows, and cures resulting in a well consolidated composite part. The dimensions of the autoclave-cured composite parts often deviate from the intended values due to the warping of the composite parts. Figure 5.3 [68] shows one of the primary Autoclave Systems in use by Boeing and their partners to produce the new 787 Dreamliner.

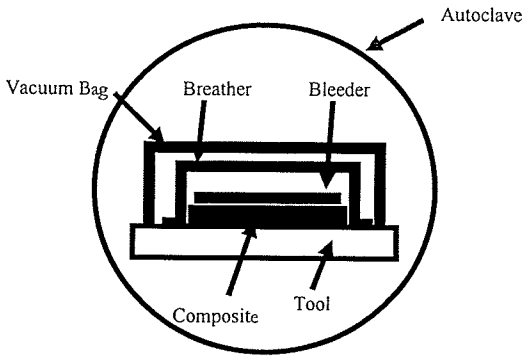


Figure 5.1 Schematic of composite and tool assembly in an autoclave

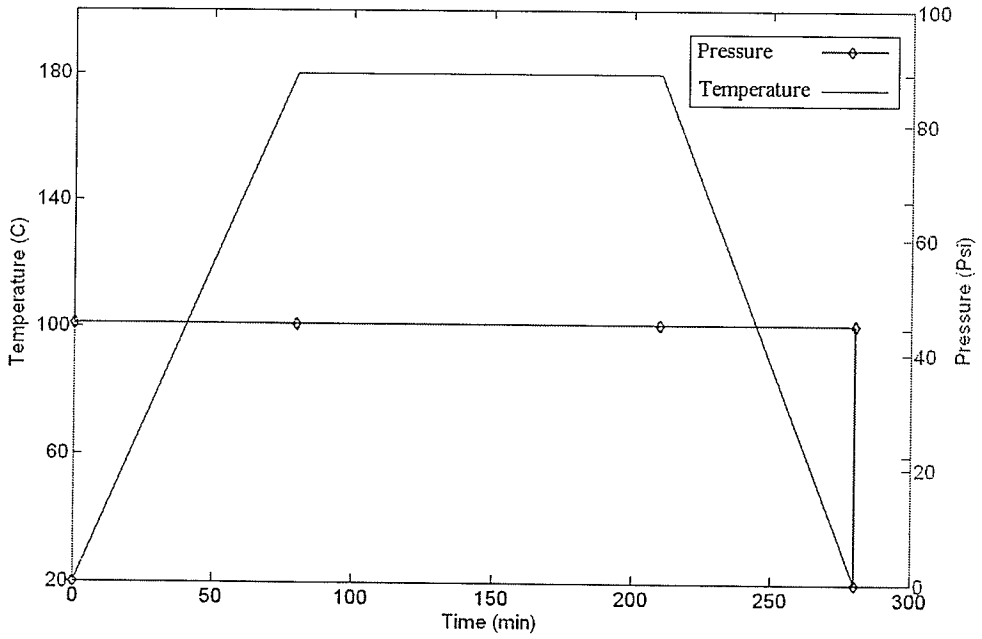


Figure 5.2 A typical autoclave cure cycle

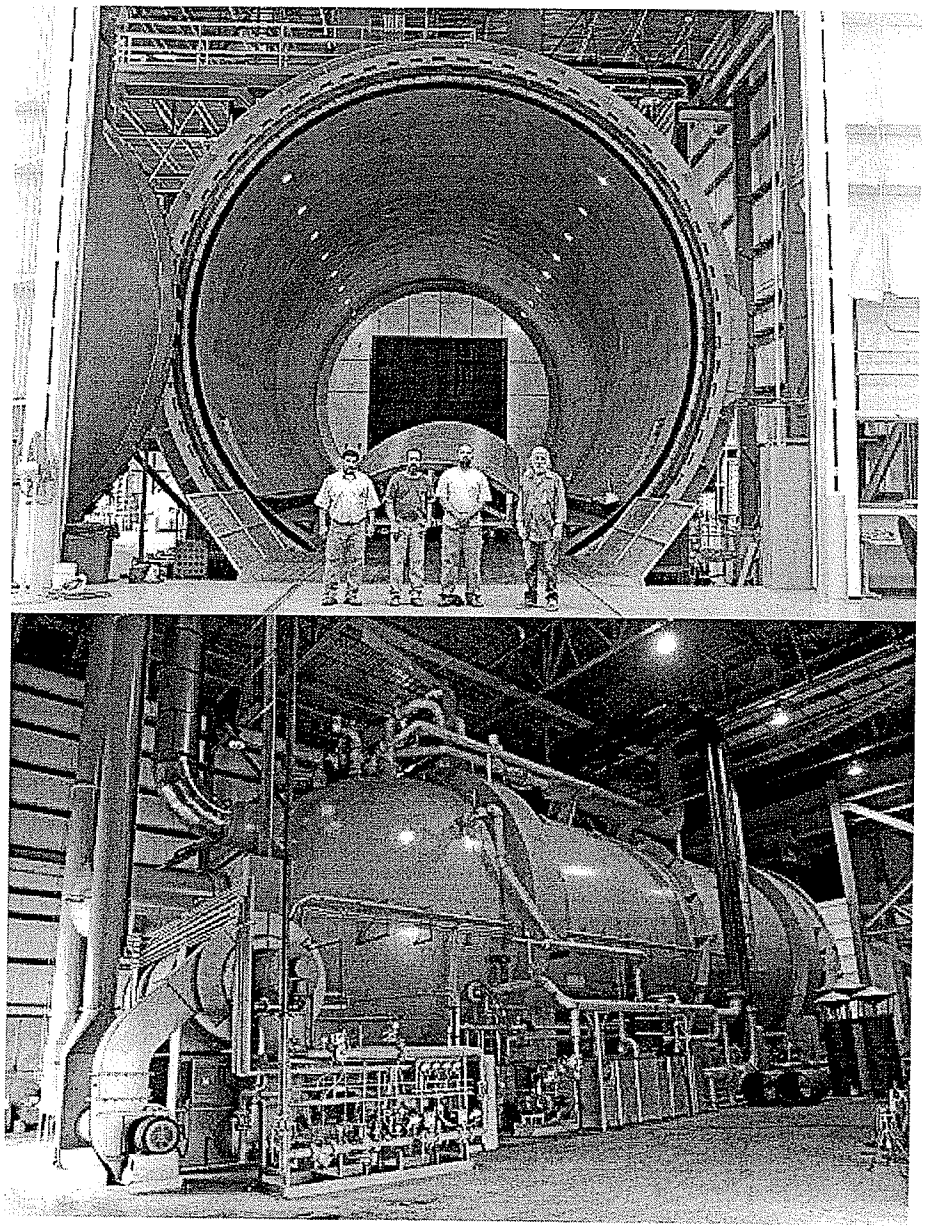


Figure 5.3 Two views of the autoclave TEC installed at Spirit Aerosystems in Wichita for Boeing 787 Program [68]. (Reproduced with permission from the web-site of Thermal Equipment Corporation (www thermalequipment.com))

A high quality autoclave cured composite part is expected to have a maximum degree of cure, uniform thickness, minimum voids, minimum residual stress, and minimum warpage. However, process-induced residual stress and warpage are two major current concerns during autoclave processing of large complex composite laminates. The selection of cure cycle parameters and design of a tool to produce a fully cured and void free composite structure without any warpage is a challenging task. Currently, the process-induced warpage is minimized through allowances in the tool. During the process development stages, the tool material selection, tool shape and the process are tweaked by trial-and-error to minimize the warpage. This process is costly and often requires repeated tweaking when the tool or the process is changed because warpage changes. Figure 5.4 shows a half-symmetry mesh of a Boeing aircraft part that has been processed in autoclave and Figure 5.5 shows the maximum warpage measured at the edge along the length of the part, after autoclave process.

In order to eliminate the trial and error process of changing the tool shape, there is need for automated tool shape optimization, which results in desired part shape after the autoclave process. Since autoclave processed parts are 3-D complex shape, the objective of this thesis is to investigate the efficiency of single objective optimization for simple 2-D shape first, which is still expensive function evaluation. Results of this are evaluated to determine the applicability of NSGQA for optimization of complex 3-D shape of a tool.

This is achieved by interfacing a process model with GQA and validated using experimental results generated by Koteshwara [67]. A simple right-angled composite part, as shown in Figure 5.6 is chosen for this study. This part warps after

manufacturing with reduction in the included angle. Shape optimization problem is illustrated in Figure 5.7 in which design variables are the angle γ and the radius r. The objective function here is φ, which is the warpage angle after the autoclave process. While this is a single objective optimization problem, in case of more complicated shapes shown in Figure 5.4 the warpage cannot be defined by just a single value of φ. There will be multiple values, since the warpage varies along the length as illustrated in Figure 5.5. Hence, it is essential to develop a multi-objective optimization code, in the near future, to extend the current work on 2-D shape optimization.

The model framework is presented in Figure 5.8. The two main components, the GQA (Genetic Quantum Algorithm) and the process model, are discussed in sections 3.1and 5.2, respectively. A population consisting of a certain number of individuals is selected. Each individual, defined by GQA and assigned to a set of values for design variables, is input to the process model. Warpage data from the process model for these individuals are compared to select the best individual i.e. values for design variables. If the convergence is reached, this solution is taken to be the optimum. Otherwise, the next generation population with individuals is created using the best solution of the previous generation. The procedure is then iterated. The process model, details of optimization, and the results are discussed next.

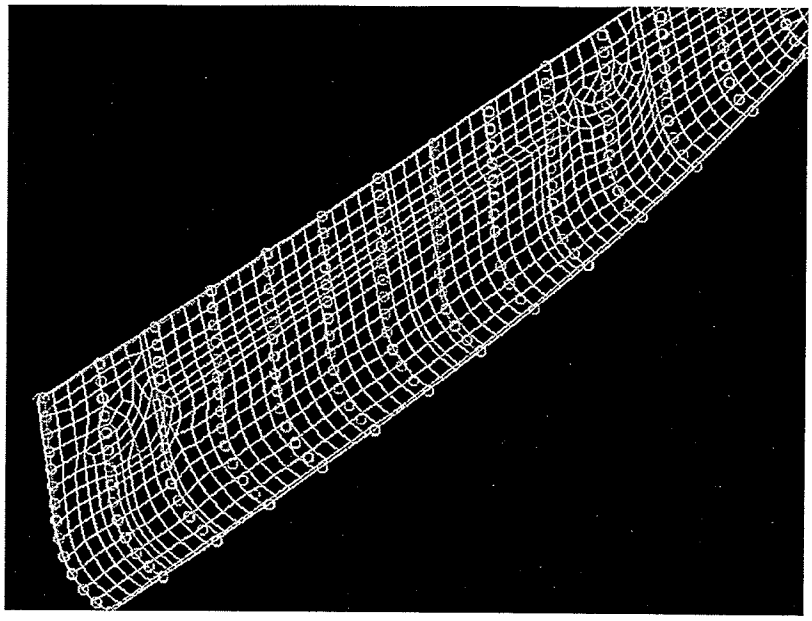


Figure 5.4 Half symmetry of a Boeing aircraft part [57].

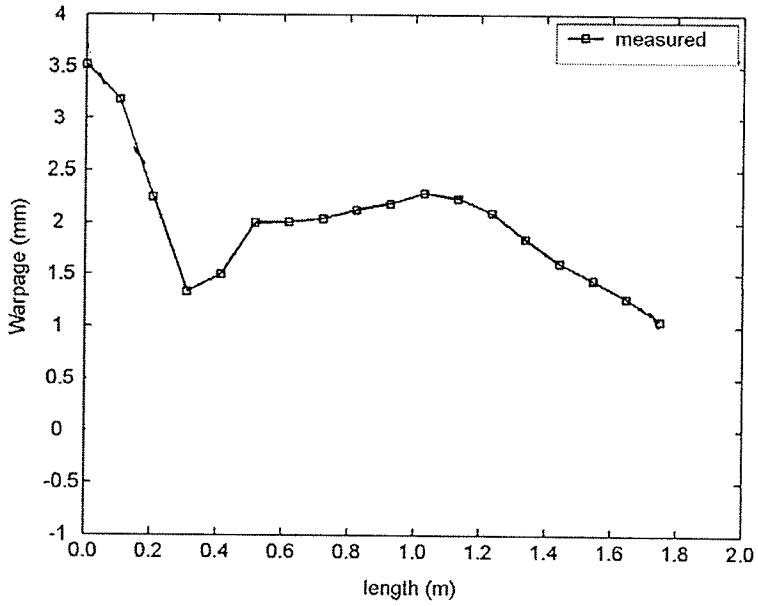


Figure 5.5 Maximum warpage measured at the edge of the part along its length.

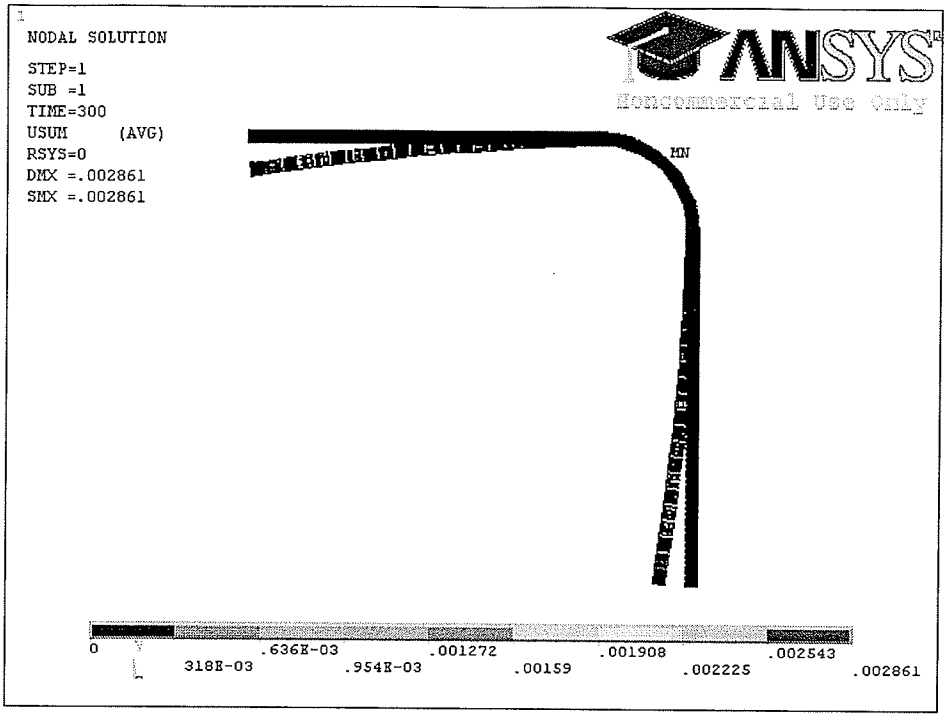


Figure 5.6 Observed warpage after tool removal.

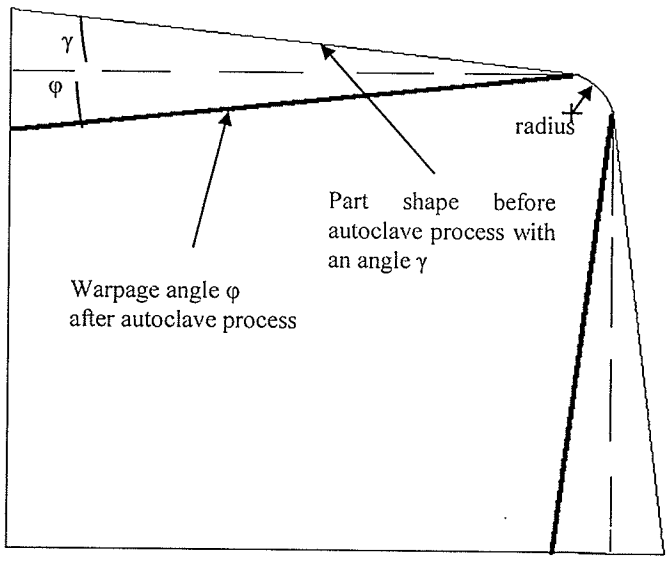


Figure 5.7 Tool shape design variables.

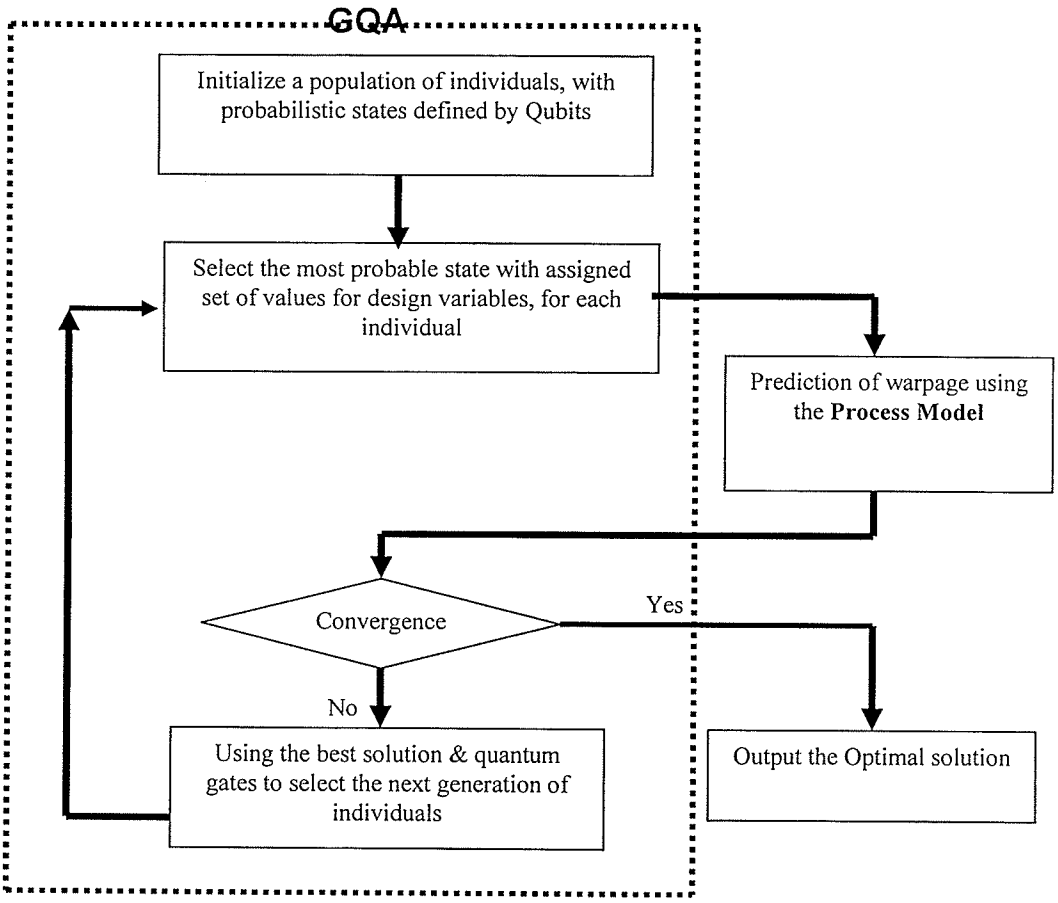


Figure 5.8 Optimization model frame work

5.2 Process Model

There are numerous publications on the causes for process-induced warpage in autoclave cured part as well as on process models for autoclave processing. A good review of this can be found in the reference [67]. The CMSRG (Composite Materials and Structures Research Group) have developed FEM based 2-D and 3-D process models and have recently validated the latter using a Boeing aircraft part [57].

In this thesis the 2-D model was used to demonstrate the applicability of NSGQA to warpage minimization through tool-shape optimization. It is hoped that this exercise would lay the foundation for extending this work in the near future to optimization of a complex part using the 3-D process model

- A brief introduction to the 2-D finite elements (ANSYS) based process model, developed at University of Manitoba for the prediction of process-induced residual stress and warpge of 2-D composite parts, is presented below. Similar to models available in the literature, UM model also has sub-module structure as shown in Figure 5.9. Input Module: Input tool-part geometry and mesh, tool material and composite material properties, cure cycle details, lay-up sequence, and boundary conditions
- Thermochemical Module: Predicts part temperature and resin degree of cure during the entire cure cycle.
- Material Module: Predicts evolution of composite properties during the entire cure cycle, and

- Stress Module: Predicts thermal stress development during the entire cure cycle and final part shape after tool removal

With the exception of part - tool geometry, mesh, and mechanical boundary conditions, which could be input through GUI of ANSYS, all other input parameters to thermochemical module, stress module and tool removal module requires special functions that are not available in ANSYS. Therefore these modules have been programmed using ANSYS Parametric Design Language (APDL) and interfaced with Frontal solver of ANSYS program. The 2-D model developed by Koteshwara [67] did not include tool-part interaction, which is a major factor that affects warpage. Hence, in this thesis tool-part interaction was added to the model and the accuracy of resulting warpage was compared with experimental results from reference [67]. This was done prior to optimization studies.

Since the ANSYS solver cannot account for resin cure shrinkage strain and elastic constitutive model for a laminate with a stacking sequence of $\pm\theta$, the thermomechanical properties of the plies and effective thermal strain of the composite were obtained using a separate material module. The material module has been written using FORTRAN 77 and interfaced with the stress module. The modules were solved sequentially in the following order thermo-chemical module, material module, and stress module.

5.3 Details on Composite Part and Simulation

Composite part details and simulation details are presented below.

5.3.1 Composite Part Details

The L-shaped composite part was made up of Cytec Fiberite's HMF5-322/34C pre-preg tape made of unidirectional carbon fibers and 934 neat resin. The pre-preg tapes were stacked such that the fibers in the pre-preg were oriented either parallel or perpendicular to the length of the part. The part had an included angle of 90° and the thickness and length of the parts were 0.00337 (m) and 0.14097 (m). The various input properties are composite pre-preg properties such as cure kinetics, thermo-mechanical properties and its dependence on degree of cure tool properties including modulus, coefficient of thermal expansion, specific heat capacity and thermal conductivity. Convective heat transfer coefficient as well as air temperature for the autoclave used in generating the experimental warpage results used in this study tool-part friction coefficient. The first three were generated by Koteshwara and were taken from reference [67]. The last one is discussed below.

Koteshwara [67] cured this part on an aluminum tool in an autoclave using the cure cycle shown in Figure 5-2. The warpage of this part measured by him were used in this study. As mentioned above, these experimental warpage results were compared with predictions from the modified 2-D process model developed in this study to validate the latter. Since Koteshwara did not generate any frictional data, those generated by Kaushik et al [57] was used in this study.

They investigated the tool-part interaction during autoclave processing and the factors that influence them. They used a modified friction test fixture originally designed by Boeing, to measure the static and dynamic friction coefficients at various times during the process, which are tabulated in Table 5-1 and Table 5-2 [57].

However, these values were generated using Cytec Fiberite's MXB7701-7781-B3 – a plain weave glass fiber – epoxy composite, cured at 260 °F on steel tool. Despite this difference, the latter was used in this simulation assuming that the magnitude of friction coefficient and its dependence on the degree of cure was the same for both material–tool pairs. Implications of this are discussed later in this chapter.

During the early stage of curing process, i.e. before gelation when the material is in liquid state, the part can not sustain any load. This gel point for this material is 0.37 and the gelation occurred at a process time of 76 min when the part was ramped at the rate of 3 °F/ min. Gelation time for a ramp rate of 2 °F/ min is 96 min. Hence, the stress analysis was done only for the process cycle time beyond the above mentioned gelation times.

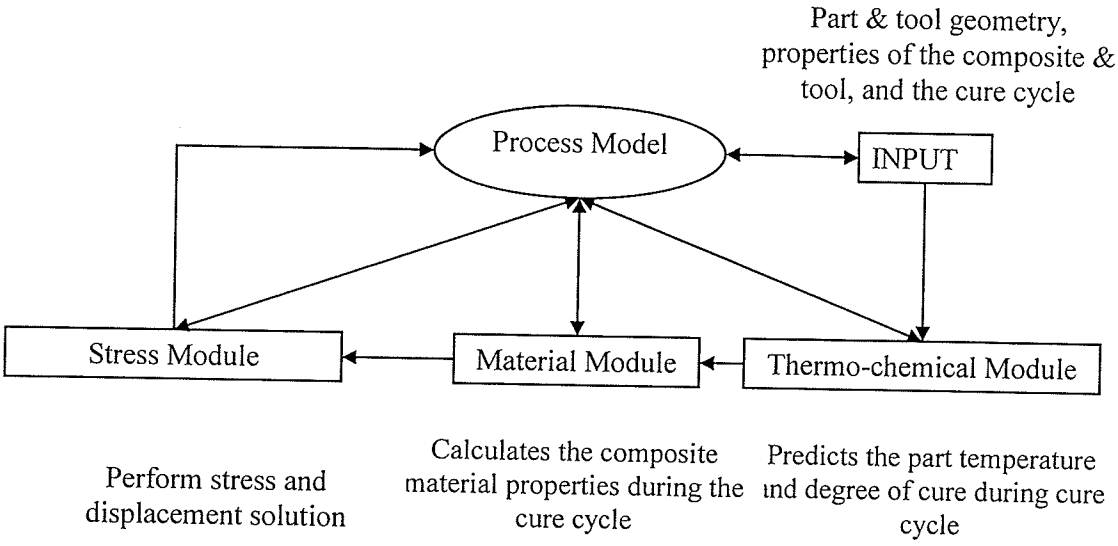


Figure 5.9 ANSYS based process model [67].

Table 5-1 Friction coefficient [57] for 3 deg F/min

α	μ (static)	μ (dynamic)
0	0.618	0.439
0.2	0.185	0.111
0.4	0.222	0.112
0.6	0.115	0.03
1	0.456	0.02

Table 5-2 Friction coefficient [57] for 2 deg F/min

α	μ (static)	μ (dynamic)
0	0.618	0.43
0.2	0.186	0.11
0.4	0.222	0.12
0.6	0.313	0.05
1	0.120	0.02

5.3.2 Simulation Details

In this section details on tool part interaction, contact analysis, and optimization are presented. A typical FEM mesh of the tool and part is given in Figure 5.10. The element type was changed through GUI depending on the modules. PLANE 55 (Thermal Solid) [52] was used for Thermochemical module and PLANE 42 (2-D-Structural Solid) was used for stress & tool removal module. During thermochemical analysis, both tool and part, as shown in Figure 5.10, were considered. Boundary conditions, as shown in the Figure 5.10 were applied to avoid rigid body motion. The process cycle used in the analysis is given in Figure 5.11 and the time step is 1 minute.

In order to simulate tool part interaction in FEA two types of elements should be defined between the two contact surfaces. Contact element CONTA172 [52], which is a 2-D and 3-Node Surface-to-Surface contact element, was used to define the surface of the tool. 2-D "target" surface of the tool was defined using TARGE169 Figure 5.12 and Figure 5.13 depict finite element definition of CONTA172 and TARGE169 respectively. Those lines representing the contact surface between tool and part were divided equally for contact element generation. Contact nodes were created on the part first. Subsequently, target nodes were generated on the tool. Finally, contact elements were defined by pairing of one contact node with one target node. The default values of the ANSYS program were chosen for all the contact parameters which are Normal penalty stiffness, Penetration tolerance, Tangent penalty stiffness, Contact cohesion, Maximum friction stress, Static/dynamic ratio, and Exponential decay coefficient, except the friction coefficient, which was taken from Table 5-1 and Table 5-2.

Simulation was done with and without tool and compared with the results of Koteshwara [67]. Details on optimization are presented next.

The design variables are the angle γ and the radius, as shown in Figure 5.7. The measured warpage is denoted as φ in Figure 5.7. GQA assigns a population of individuals with values for the radius and γ in the range of 0.01 to 0.0222 (m) and, 0-2.5° respectively. They are used by the process model to calculate the warpage, which is subsequently used by GQA to determine the optimized values for the design variable. GQA used a population size of 20, a generation size of 10, and a chromosome length (*m)* of 16.

In addition to GQA, non-linear programming in MATLAB was also used for optimization and compared with GQA. For optimization using non-linear programming *fmincon* [55], a routine to find a minimum of constrained nonlinear multivariable function, was used. The constraints used for the current problem were $0.01 \leq$ radius ≤ 0.0222 m, *and* $0 \leq \gamma \leq 2.5°$. Radius of the tool used in generating the experimental results in reference [67] was chosen as the upper limit for the radius. Limitation posed by ANSYS during meshing dictated the selection of 0.01 to be the lower limit. The optimization runs were completed using a Pentium 4 with 1.5 GB RAM desktop computer.

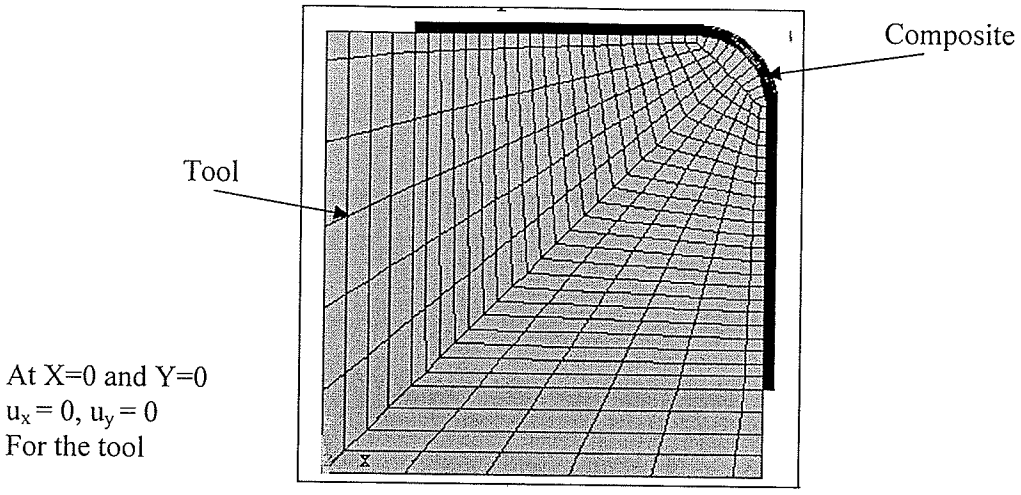


Figure 5.10 A typical finite element mesh for tool/part configuration generated using GUI of ANSYS

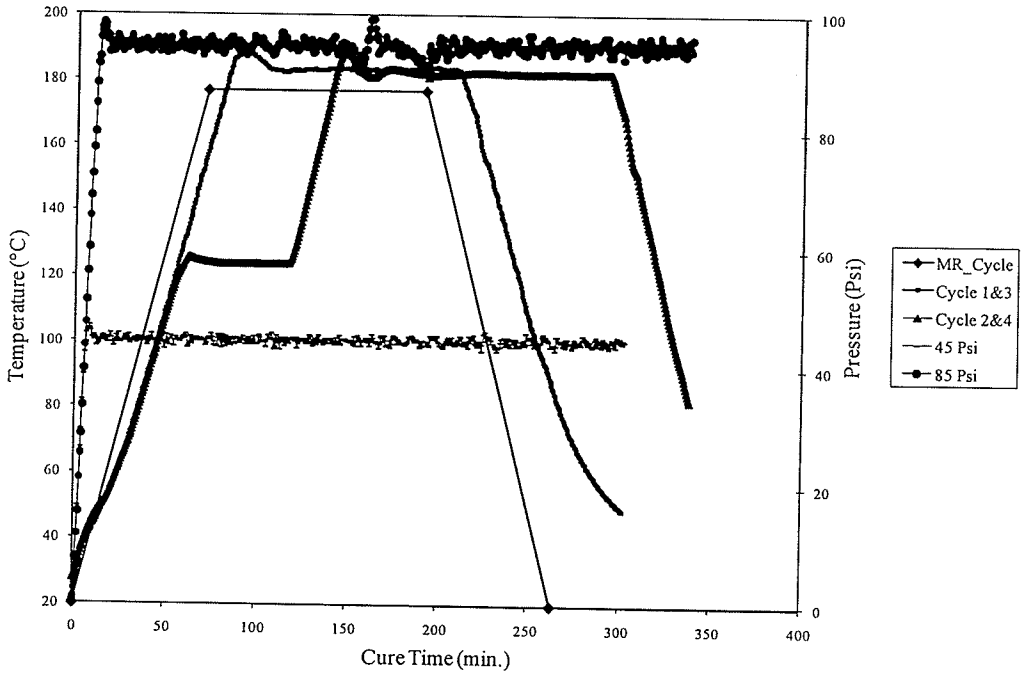


Figure 5.11 A typical autoclave cure cycle [67].

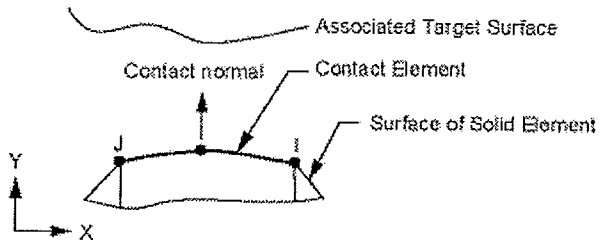


Figure 5.12 CONTA172 geometry [52].

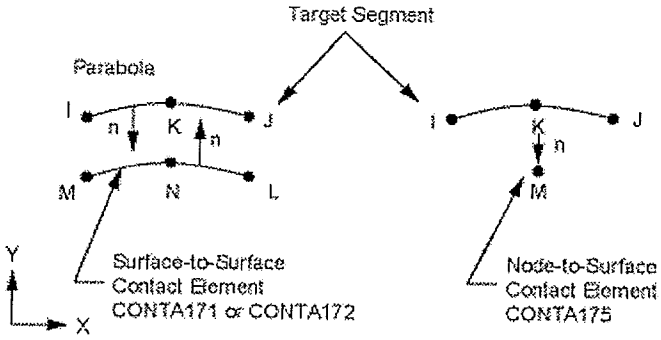


Figure 5.13 TARGE169 geometry [52].

5.4 Process Model Validation

As a part of this thesis, an existing ANSYS based 2-D process model was substantially upgraded by incorporating the tool-part interaction. Hence, prior to tool shape optimization, the model was verified by simulating the warpage of the chosen composite part and comparing with the results of Koteshwara [67]. Owing to the difference in friction data, initial comparison was done using predictions without tool. Warpage of the part processed without tool (appropriate boundary conditions were applied during simulation to prevent rigid body motion of the part) using the modified process model are compared with the predictions from reference [67] in Table 5-3. Warpage here is based on the slope of the line joining end point of the arm of the angle laminates.

Table 5-3 Comparison of predicted warpage in L-shaped part cured without any tool. Details of cycle are given in reference [67] .

CYCLE	Warpage from ref. [67] : (deg)	Warpage predicted by Current Process Model (deg)
1	2.00	1.97
2	2.72	2.74
3	2.12	2.15
4	2.68	2.57

Table 5-4 Comparison of predicted warpage (accounting for tool-part interaction) with experimental results.

Heat Ramp rate during cure	3 F/ min (deg)	2 F / min (deg)	Experimental [67] for 3.89 F / min (deg)
Predicted Warpage	1.80	1.87	1.69

The two predictions compared very well within an error of 3.3%. This small difference is attributed to the small changes made to the material models used by Koteshwara. The above results confirmed simulation capability of the modified model.

Subsequently, the process model was used to predict the warpage of the L-shaped part processed in contact with the tool. The predictions are compared with experimental results in Table 5-4. It can be observed that the predicted results are in close agreement with experimental results with 6.5% error for the cycle with 3 °F/ min ramp rate. Koteshwhara [67] used a ramp rate of 3.89 (deg F/ min) during autoclave manufacturing the L-shaped composite parts. It can be observed from Table 5-1 and Table 5-2 that the friction coefficients are sensitive to ramp rate during autoclave manufacturing. Using these two sets of data, the warpage was predicted for two rates (2 and 3 °F/ min) and tabulated in Table 5-4. It can be observed that the predicted warpage decreases with increase in ramp rate. Hence, for accurate comparison with experimental results, friction data at 3.89 °F/ min is required. Since it was not available, the prediction using friction data for 3 °F/ min is compared in Table 5-4. While this is one source of error, another source is the use of frictional data for a different material – tool pair. Despite this the error in prediction is within 6.5%. This appears to confirm the assumption on the magnitude of friction coefficient and its dependence on degree of cure. Further experiments are required to verify this. Nevertheless, the good correlation in Table 5-4 validates the predictive capability of the modified process developed in this thesis.

5.5 Results and discussion of Tool Shape Optimization

GQA optimization runs were completed using the developed software for an L-shaped part processed on an aluminum tool. The optimal tool shape that resulted in minimal warpage of the part after autoclave processing is shown in Figure 5.14. The variables corresponding to this optimal shape, as tabulated in Table 5-5, are $\gamma = 2.11°$ and r = 0.014 m. Please note that the tool shape is also the part shape since the pre-preg material before processing is pliable and conforms to the shape of the tool. The variables corresponding to this optimal shape, as tabulated in Table 5-5, are $\gamma = 2.11°$ and r = 0.014 (m). The predicted part shape after processing is shown in Figure 5.15 with a final warpage of $\varphi = 0.003°$ (degree) warpage. The negligible warpage obtained through this optimization demonstrates the potential of GQA for multi-objective optimization of complex tool shapes for composite manufacturing. The convergence to optimal solutions involved about 200 iterations and the time of this convergence is about 13.5 hours (i.e. roughly 4 minutes per iteration). It is believed that this computation time can be reduced by improving the programming.

The optimized tool shape obtained using non-linear programming functions of MATLAB and the same design variables and constraints used in GQA, is shown in Figure 5.16. The variables corresponding to this optimal shape, as tabulated in Table 5-5, are $\gamma = 0°$ and r = 0.022 m. The predicted part shape after autoclave processing is shown on Figure 5.17 and the warpage of was 1.15°. The warpage of the part optimized by GQA is much lower (almost negligible) when compared to warpage of the part optimized by non-linear programming. This comparison demonstrates the ability of GQA to find a better optimal than Nonlinear Programming.

Table 5-5 Optimized tool shape and final warpage for a L-shaped part processed on an aluminum tool.

	Warpage (°)	Optimal radius (m)	Optimal design γ (°)
GQA	φ = 0.003°	r = 0.014	γ = 2.11°
Matlab-NLP	φ = 1.15°	r = 0.022	γ = 0°

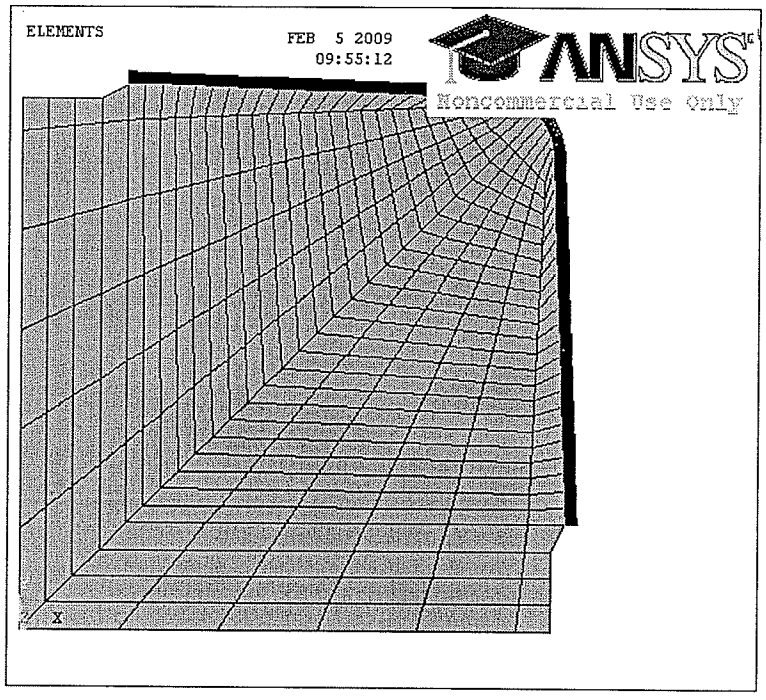


Figure 5.14 Optimized tool / part shape using GQA

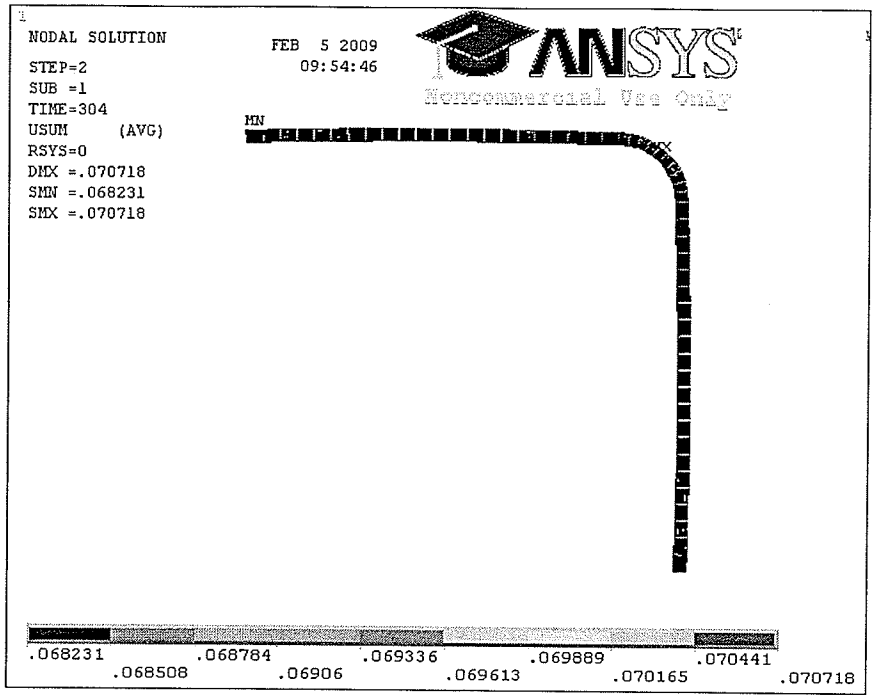


Figure 5.15 Shape of the GQA optimized part after manufacturing.

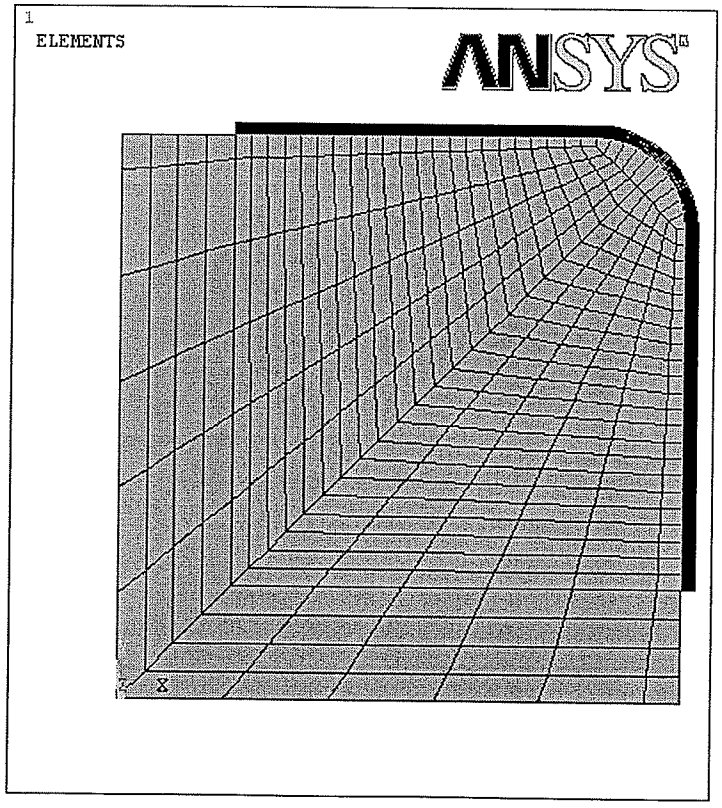


Figure 5.16 Optimized tool / part shape using Non-Linear Programming

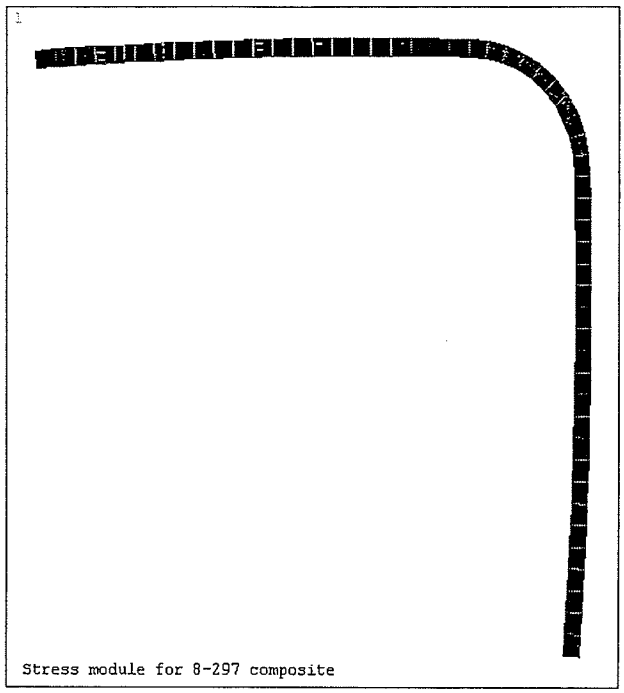


Figure 5.17 Shape of the Non-Linear Programming optimized part after manufacturing

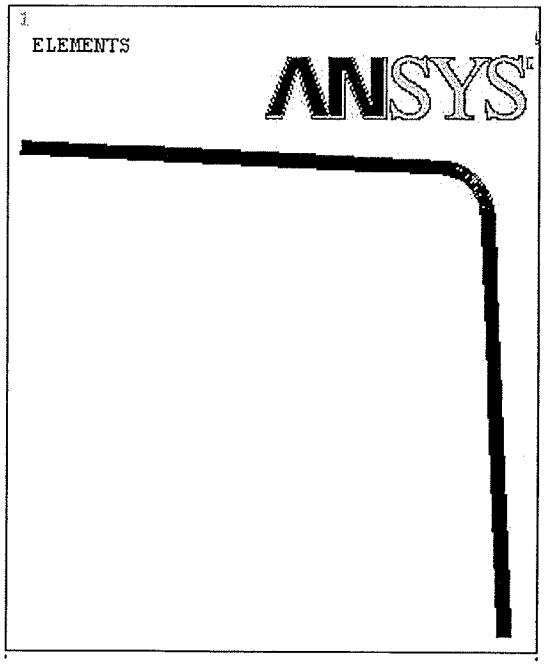


Figure 5.18 GQA optimized tool/part shape

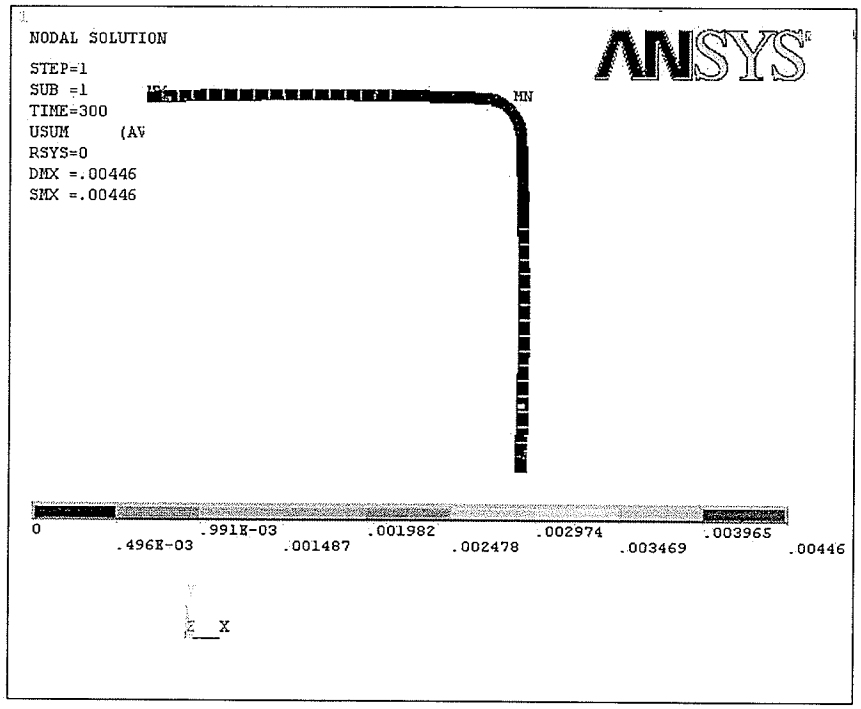


Figure 5.19 Final shape of the GQA optimized part after manufacturing

Since the frictional data used in this analysis did not correspond to the tool – material pair used in this study, additional optimization simulations were completed without any tool in contact with the part (i.e. the part could freely slide on the tool). The optimal tool shape obtained using GQA is shown in Figure 5.18. The variables corresponding to this shape, tabulated in Table 5-6, are $\gamma = 2.5°$ and $r = 0.010$ m. Warpage after the autoclave process (without tool) was $\varphi = 0.056°$ (degree) as shown in Figure 5.19. The results are similar to that obtained for the simulation of the part processed on a tool. The negligible warpage obtained through this optimization for a part processed without tool demonstrates again the potential of NSGQA for multi-objective optimization of complex tool shapes for composite manufacturing

The optimized tool shape obtained using non-linear programming functions of MATLAB and the same design variables and constraints used in GQA, is shown in Figure 5.20. The variables corresponding to this optimal shape, as tabulated in Table 5-6, are $\gamma = 0°$ and $r = 0.022$ m. The predicted part shape after autoclave processing is shown on Figure 5.21 and the warpage was 2.6869°. The warpage of the part optimized by GQA is much lower (almost negligible) when compared to warpage of the part optimized by non-linear programming. This comparison demonstrates again the ability of GQA to find a better optimal than Nonlinear Programming.

Tool shape optimization is examined on warpage minimization of composite part using GQA and compared with Nonlinear Programming results. Optimization results show the advantageous of using GQA for shape optimization and applicability of optimizing 3-D complex autoclave cured composite part.

Table 5-6 Without tool part interaction optimization.

	Warpage (°)	Optimal radius (m)	Optimal design γ (°)
GQA	φ = 0.056°	r = 0.010 (m)	γ = 2.5°
Matlab-NLP	φ = 2.6869 °	r = 0.0222 (m)	γ = 0°

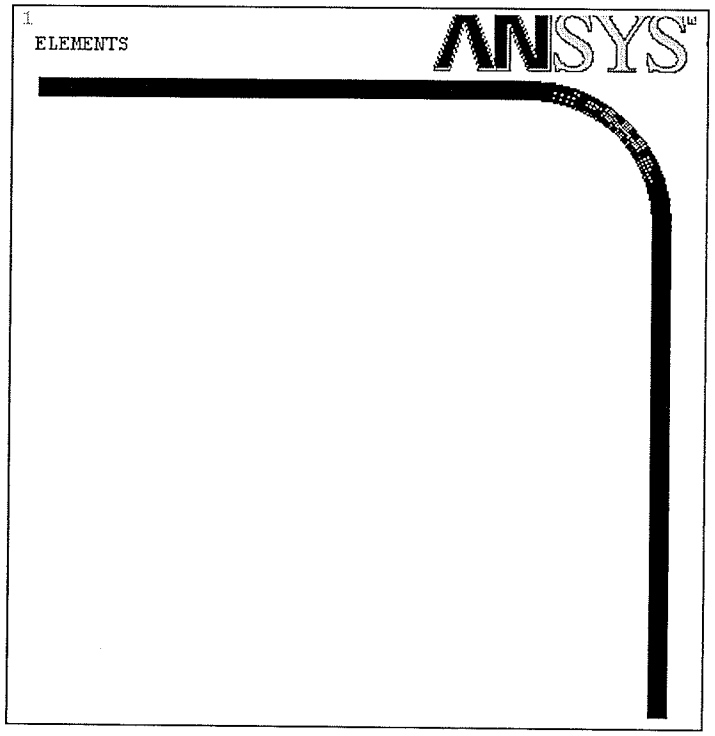


Figure 5.20 Tool / Part shape optimized using non-linear programming

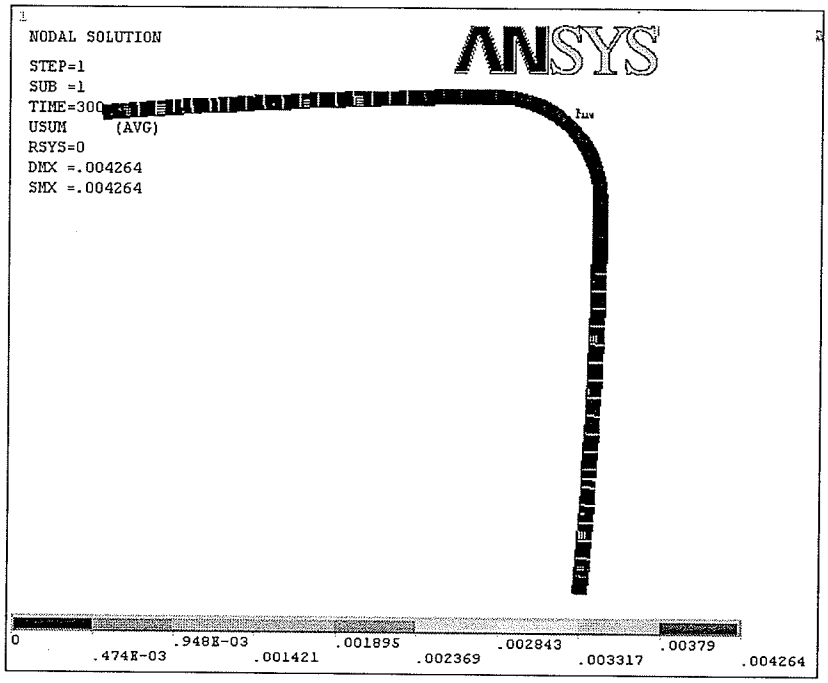


Figure 5.21 Final shape of the Nonlinear Programming optimized part after manufacturing.

CHAPTER 6

CONCLUSION & RECOMMENDATION

This thesis presents a new multi-objective optimization (MOO) method, which is inspired from the idea of non-dominated sorting genetic algorithm (NSGA) and genetic quantum algorithm (GQA). The developed method is called non-dominated sorting genetic quantum algorithm (NSGQA). The developed method is tested with benchmark problems collected from literature, which have characteristics representing various aspects of a MOO problem. Test results show that NSGQA has overall better performance on most benchmark problems than current state-of-the-art MOO methods, when the total number of function evaluations is limited. The NSGQA is also successfully applied to a 47-bar truss tower design problem, which entails 44 design variables, 44 design variable constraints, and 2 inequality constraint. The integration of GQA with MOO, and the systematic comparison with other MOO methods on benchmark problems and its application in engineering design, should be of general interests to researchers on MOO and practitioners using MOO methods in design. The developed NSGQA was applied to the problem of 2-D tool shape optimization to minimize process-induced warpage in autoclave cured composites. It was integrated with an ANSYS-based 2-D process model (which was developed by significantly augmenting an existing process model) to successfully demonstrate that the tool shape (i.e. initial shape of the un-cured part) could be optimized to minimize warpage in L-shaped composite part considering tool part interaction. GQA resulted in a better optimal solution when compared to nonlinear programming.

The results confirm the potential of NSGQA in multi-objective shape optimization of complex 3-D tools to reduce process-induced warpage in a complex autoclave cured composite structures.

REFERENCE

[1] Hague D.S, Rozendaal H.L., Woodward F.A., "Application of Multivariable Search Techniques to Optimal Aerodynamic Shaping Problems," Journal of the Astronautical Sciences, Vol. 15, (1968), pp. 283-296.

[2] Holland, John H, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, (1975).

[3] Kirkpatrik S., Gelatt C. D., Vecchi M. P., "Optimization by Simulated Annealing". 1982 IBM Research Report RC 9355.

[4] Kennedy J., Eberhart R., "Particle Swarm Optimization," in Proceedings of IEEE International Conference Neural Network, Madison, WI, USA ,Vol. 4, (1995), pp. 1942–1948.

[5] Colorni A, Dorigo M.,Maniezzo V ,"Distributed Optimization by Ant Colonies," actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, December (1991), pp.134-142.

[6] Horn J., Nafpliotis N., Goldberg D.E., "A Niched Pareto Genetic Algorithm for Multiobjective Optimization", Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Orlando, USA, , Vol. 1, June (1994), pp. 82-87.

[7] Laumanns M., Rudolph G., Schwefel H., "A Spatial Predator-Prey Approach to Multi-Objective Optimization: A Preliminary Study", Proceedings of 5th International Conference on Problem Solving from Nature - PPSN V, Amsterdam, Netherlands, Vol. 5, September (1998), pp. 241-249.

[8] Knowles J., Corne D., "The Pareto Srchived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimization," Proceedings of the 1999 Congress on

Evolutionary Computation. Washington, DC, USA, Vol.1, July (1999), pp. 98–105.

[9] Zitzler E., Thiele L., "Multiobjective Evolutionary Algorithms a Comparative Case Study and the Strength Pareto Approach", IEEE Transactions on Evolutionary Computation, Vol.3, (1999), pp. 257-271.

[10] Deb M., "Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems," Evolutionary Computation, Massachusetts Institute of Technology, Vol. 7, (1999), pp.205-230.

[11] Zitzler E., Deb K., Thiele L., "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," Evolutionary Computation, Massachusetts Institute of Technology, Vol. 8, (2000), pp.173-195.

[12] Cui X., Li M., Fang T., "Study of Population Diversity of Multi-objective Evolutionary Algorithm Based on Immune and Entropy Principles," Proceedings of the Congress on -Evolutionary Computation, Coex, Seoul, Korea, Vol.2, May (2001), pp. 1316-1321.

[13] Zitzler E., "Evolutionary Algorithm for Multiobjective Optimization," Proceedings of the conference on Evolutionary methods for design optimization and control, CIMNE Barcelona, Spain, (2002), pp. 19-26.

[14] Zitzler E., Laumanns M., Thiele, L., "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization," Proceedings of the conference on Evolutionary methods for design optimization and control, CIMNE Barcelona, Spain, (2002), pp. 95-100.

[15] Deb K., Pratap A., Agarwal S., Meyarivan T., "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation, Vol. 6, (2002), pp.183-197.

[16] Antony W. I., Xiaodong L., "A Cooperative Co-evolutionary Multi-objective Algorithm Using Non-dominated Sorting," Proceedings of the GECCO, Springer-Verlag Berlin Heidelberg, Seattle, USA, Vol. 3102, June (2004), pp. 537–548.

[17] Keerativuttitumrong N., Chaiyaratana N., Varavithya V., "Multi-objective Co-operative Co-evolutionary Genetic Algorithm," Proceedings of the 7th International Conference on Parallel Problem Solving from Nature , Granada, Spain, , Vol. 2439, September (2002), pp. 288-297.

[18] Zeng S., Ding L., Chen Y., Kang L., "A New Multiobjective Evolutionary Algorithm: OMOEA," Congress on Evolutionary Computation, Canberra, ACT, Australia, Vol.2, December (2003), pp.898 - 905.

[19] Jin H., Wong M., "Adaptive Diversity Maintenance and Convergence Guarantee in Multi-objective Evolutionary Algorithms," Congress on Evolutionary Computation, Canberra, ACT, Australia, Vol.4, December (2003), pp. 2498 – 2505.

[20] Coello C.A.C., Becerra R., "Evolutionary Multi-objective Optimization Using a Cultural Algorithm," Proceedings of the 2003 IEEE Swarm Intelligence Symposium, Indianapolis, USA, April (2003), pp. 6 – 13.

[21] Sarker R., Abbass H.A., "Differential Evolution for Solving Multi-objective Optimization Problems", Asia-Pacific Journal of Operations Research, World Scientific, Vol. 21, (2004), pp. 225-240.

[22] Sato H., Aguirre H.E., Tanaka K., "Local Dominance Using Polar Coordinates to Enhance Multi-objective Evolutionary Algorithms," Congress on Evolutionary Computation, Vol.1, June (2004), pp. 188-195.

[23] Freschi F., Repetto M., "Multiobjective Optimization by a Modified Artificial Immune System Algorithm", Proceedings of the 4th International Conference on Artificial Immune Systems, Banff, Canada, Vol. 3627, August (2005), pp. 248-61.

[24] Tran K., "Elitist non-dominated sorting GA-II (NSGA-II) As a Parameter-Less Multi-objective Genetic Algorithm", Proceedings of the IEEE Southeast Conference, Ft. Lauderdale, United Kingdom, April (2005), pp. 359-67.

[25] Ang K., Chong G., Li Y., "Preliminary Statement on the Current Progress of Multi-objective Evolutionary Algorithm Performance Measurement", Proceedings of the 2002 Congress on Evolutionary Computation, Honolulu, USA, Vol.2, May (2002), pp. 1139-1144.

[26] Lu H., Yen G., "Rank-Density-Based Multiobjective Genetic Algorithm and Benchmark Test Function Study", IEEE Transactions on Evolutionary Computation, Vol.7, (2003), pp. 325-343.

[27] Nebro A. J., Luna F., Alba E., Dorronsoro B, Durillo J.uan J., Beham A. "AbYSS: Adapting Scatter Search to Multiobjective Optimization." IEEE Transactions on Evolutionary Computation, Vol.12, (2007), pp.439-457.

[28] Durillo J.J., Nebro A.J, Luna F., Alba E., "Solving Three-Objective Optimization Problems Using a new Hybrid Cellular Genetic Algorithm". Proceedings of Parallel Problem Solving from Nature - PPSN X - 10th International Conference, (2008), pp. 661-670.

[29] Eskandari H., Geiger C.D, "A Fast Pareto Genetic Algorithm Approach for Solving Expensive Multiobjective Optimization Problems" Journal of Heuristics, Vol.14, June (2008), pp.203-241.

[30] Reyes M., Coello C.A., "Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and Epsilon-Dominance," In Coello, C., Hernandez,

A., Zitler, E., eds.: Third International Conference on Evolutionary MultiCriterion Optimization, EMO Springer. Vol. 3410, (2005).

[31] Zitzler E., Laumanns M., and Thiele L. "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization". In K. Giannakoglou et al., editors, EUROGEN 2001, International Center for Numerical Methods in Engineering (CIMNE), (2002), pp. 95–100.

[32] Coelho R. F., Multicriteria Optimization With Expert Rules for Mechanical Design, Dissertation, Universite´ Libre De Bruxelles Faculte´ Des Sciences Applique´ es, (2004).

[33] Reynolds D., McConnachie J., Bettess P., Christie W.C., Bull J.W., "Reverse Adaptivity – A New Evolutionary Tool for Structural Optimization," International Journal for Numerical Methods in Engineering, vol. 45, (1999), pp. 529-552.

[34] Sir George Cayley, Aeronautical and Miscellaneous Note-Book (ca. 1799-1826). W. Heiffer and Sons, Cambridge, England, (1933), pp. 69-71.

[35] Vanderplaats G. N., "Multidisciplinary Shape Optimization," General Motors Research Laboratories Symposia Series, (1986), pp. 263-281.

[36] Botkin M.E., "Three-Dimensional Shape Optimization Using Fully Automatic Mesh," AIAA Journal, Vol. 45, (1992), pp. 1932-1934.

[37] Gates A. A., Accorsi M. L., "Automatic Shape Optimization of Three-Dimensional Shell Structures with Large Shape Changes," Journal of Computers & structures, (1993), pp. 167-178.

[38] Thanedar P., King R., "Shape Optimization Using Adaptive High-Order Finite Elements," Journal Structural Optimization, Vol.6, (1993), pp. 189-193.

[39] Maute K., Ramm E., "General Shape Optimization - an Integrated Model for Topology and Shape Optimization," Proceedings of the 1995 1st World Congress

of Structural and Multidisciplinary Optimization, Goslar, Elsevier, Oxford, May (1995), pp.299-306.

[40] Wieghardt K., Hartmann D., Leimbach K.R., "Interactive Shape Optimization of Continuum Structures," Journal of Engineering Structures, Vol. 19, (1997), pp. 325-331.

[41] Lindby T., Santos J.L.T., "Shape Optimization of Three-Dimensional Shell Structures with the Shape Parameterization of a CAD system," Journal of Structural Optimization, Vol. 18, (1999), pp. 126-133.

[42] Han S.Y., "Shape Optimization for General Two-Dimensional Structures," Journal Acta Mechanica, Vol. 145, (2000), pp.117-125.

[43] Lin C.-Y., Chao L.-S., "Automated Image Interpretation for Integrated Topology and Shape Optimization," Journal of Structural and Multidisciplinary Optimization, Vol. 20, (2000), pp. 125-137.

[44] Ansola R., Canales J., Tarrago J. A., Rasmussen J., "An Integrated Approach for Shape and Topology Optimization of Shell Structures," Journal of Computers and Structures, Vol. 80, (2002), pp. 449-458.

[45] Zhou M., Pagaldipti N., Thomas H.L., Shyy Y.K., "An Integrated Approach to Topology, Sizing, and Shape Optimization," Journal of Structural and Multidisciplinary Optimization, Vol. 26, (2004), pp. 308-317.

[46] Samareh J. A., "Aerodynamic Shape Optimization Based on Free-Form Deformation," Collection of Technical Papers - 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Vol. 6, (2004), pp. 3672-3683.

[47] Kegl M., "Parameterization Based Shape Optimization: Theory and Practical Implementation Aspects," Engineering Computations ,Swansea, Wales, Vol. 22, (2005), pp. 646-663,

[48] Tafreshi A., "Optimum Shape Design of Composite Structures Using the Boundary-Element Method," AIAA Journal, Vol. 43, (2005), pp. 1349-1359.

[49] El-Sayed M., Sun T., Berry J., "Shape Optimization with Computational Fluid Dynamics" Journal of Advances in Engineering Software, Vol. 36, (2005), pp. 607–613.

[50] Afonso S.M.B, Sienz J., Beiblidia F., "Structural Optimization Strategies for Simple and Integrally Stiffened Strategies Plates and Shells," International Journal for Computer- Aided Engineering and Software, Vol. 22, (2005), pp. 429 -452.

[51] Vazquez M., Dervieux A., Koobus B., "A Methodology for the Shape Optimization of Flexible Wings," Engineering Computations (Swansea, Wales), Vol. 23, (2006), pp. 344-367.

[52] ANSYS 11 Help Manual, SAS IP, Inc., 2007.

[53] Khorsand A.-R., Akbarzadeh-T. M.-R., "Quantum Gate Optimization in a Meta-Level Genetic Quantum Algorithm," Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Hawaii, USA, Vol.4, October (2005), pp. 3055-3062.

[54] Peter W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, November (1994), pp. 124–134.

[55] Matlab R2006a Help Manual, The MathWorks, Inc., 2006.

[56] Zeng X., Raghavan J., SAPC (Software for Autoclave Processing of Composites), Report Submitted to Boeing Canada Technology - Winnipeg Division, (2003).

[57] Kaushik V., Raghavan J., Zeng X., Hendrickson L. P., "Experimental Study of Tool-Part Interaction During Autoclave Processing of Thermoset Polymer Composite Structure for Aerospace", Proceedings of the International Conference on Mechanical Design and Production", Cairo, Egypt, Jan (2005).

[58] Hansen S. R., Vanderplaats G. N., "Approximation Method for Configuration Optimization of Trusses," AIAA Journal, Vol.28, (1990), pp. 161-168.

[59] Van Veldhuizen D.A., Lamont G.B. "Multiobjective Evolutionary Algorithm Research: A History and Analysis" Technical Report TR-98-03, Department of Electrical and Computer Engineering, Air Force Institute Technology, (1998).

[60] Zitzler E., Thiele L.. "Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study," Proceedings of the Conference on Parallel Problem Solving from Nature, Netherlands, Vol. 1498, September (1998), pp. 292–301.

[61] Durillo J. J., Nebro A. J., Francisco Luna, Bernabe Dorronsoro, Enrique Alba, "{jMetal}: A Java Framework for Developing Multi-Objective Optimization Metaheuristics,", Departamento de Lenguajesy Ciencias dela Computaci'on, University of Malaga, E.T.S.I. Informatica, Campus de Teatinos, ITI-2006-10, (2006).

[62] Kukkonen S., Lampinen J., "GDE3: the Third Evolution Step of Generalized Differential Evolution," IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, UK, Vol. 1, (2005), pp. 443 - 450.

[63] Nebro A.J., Durillo J.J., Coello Coello C.A., Luna F., Alba E. "A Study of Convergence Speed in Multi-Objective Metaheuristics," Parallel Problem Solving from Nature - PPSN X. Proceedings 10th International Conference. Dortmund. September (2008), pp. 763-772.

[64] Knowles J. D., Corne D.W., "The Pareto Archive Evolution Strategy: A new Baseline Algorithm for Pareto Multiobjective Optimization," in Proceeding of Evolutionary Computation Congress., Washington, DC, USA, Vol. 1, (1999), pp. 98–105.

[65] Zhou A., Jin Y., Zhang Q., Sendhoff B., Tsang E., "Combining Model-Based and Genetics-Based Offspring Generation for Multi-Objective Optimization Using a Convergence Criterion," IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, (2006), pp. 3234–3241.

[66] jmetal, http://mallba10.lcc.uma.es/wiki/index.php/Tools, March (2009).

[67] Koteshwara M. P., Parametric Study of Process-induced Warpage in Composite Laminates, M.Sc. Thesis, University of Manitoba, (2001).

[68] Thermal Equipment Corporation, "http://www.thermalequipment.com/", (2008).