This is to certify that the
thesis entitled

DISCRETE CROSS-SECTION LAYOUT OPTIMIZATION OF UNIFORM CELLULAR PANELS FOR BIOFIBER-POLYMER COMPOSITES

presented by

Aradhana Sharma

has been accepted towards fulfillment
of the requirements for the

M.S. degree in Civil Engineering

Major Professor's Signature

12/02/05

Date

DISCRETE CROSS-SECTION LAYOUT OPTIMIZATION OF UNIFORM CELLULAR PANELS FOR BIOFIBER-POLYMER COMPOSITES

By

Aradhana Sharma

A THESIS

Submitted to
Michigan State University
In partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Civil and Environmental Engineering

2005

ABSTRACT

DISCRETE CROSS-SECTION LAYOUT OPTIMIZATION OF UNIFORM CELLULAR PANELS FOR BIOFIBER-POLYMER COMPOSITES

By

Aradhana Sharma

Increased emphasis on sustainability and eco-efficiency of material resources for all applications, including structural applications, has motivated exploration of natural fiber composites, or biocomposites. Yet, due to their lower stiffness and strength properties, biocomposites are unable to compete with existing construction materials for load-bearing applications. However, as seen in nature, engineered or optimized designs can lead to improved performance through efficient material and structural configurations.

This thesis presents two discrete optimization approaches to obtain optimum material distribution in the cross-sectional layouts of continuous biocomposite cellular panels. The first approach uses a gradient-based sizing optimization method and the second one uses a genetic algorithm to select coupons with finite size features from a pre-defined library. Unlike conventional topology optimization the presented methods enabled optimum hybrid designs. The designs were evaluated through laboratory scale component testing.

The optimization results led to layouts that are easier to manufacture than those that can be obtained by traditional topology optimization methods. As expected, the manufactured optimum designs demonstrated improved performance. The material layout optimization methods incorporating finite-size features presented in this study are thus thought to be viable for optimizing material distribution in cross-sections of biocomposite cellular panels and have led to designs feasible for manufacturing.

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Rigoberto Burgueño for his excellent guidance, patience, time and continuous encouragement throughout this work. It has been a great learning experience to work with him. I would not have been to complete this work successfully without his vision and support.

I would also like to thank my master's committee members, Dr. Ronald Harichandran and Dr. Ghassan Abu Lebdeh for their time and valuable guidance.

I also wish to thank Dr. Amar Mohanty and Dr. Majusri Misra for their support for manufacturing biocomposite cellular beams.

I wish to thank Siavosh Ravanbaksh for his help in conducting my experiments at the Civil Infrastructure Laboratory.

I am grateful to my friends and colleagues Christina Isaac, Mahmoodul Haq and Golrokh Nossoni for their help and constant support throughout this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$\alpha^{(k)}$	=	Step size
αL	=	Fraction of span length
αP	=	Fraction of applied load
d	=	Diameter of hole in the coupon
δ	=	Displacement under applied load in a beam
$d^{(k)}$	=	Search direction
Δx	=	Small change in current design
f_1	=	Objective function moment of inertia
f_2	=	Objective function moment of inertia per unit volume
f_3	=	Vector of objective functions f_1 and f_2
f_4	=	Vector of objective functions moment of inertia about *x* axis and moment of inertia about *y* axis
f_5	=	Vector of objective functions moment of inertia per unit volume about *x*-axis and moment of inertia per unit volume about *y*-axis
g(x)	=	Constraint function
k	=	Iteration number
n_2	=	Modular ratio of top hybrid layer in a coupon (with respect to base coupon elastic modulus)
n_3	=	Modular ratio of bottom hybrid layer in a coupon (with respect to base coupon elastic modulus)
p_c	=	Probability of cross-over
ρ^*	=	Density of cellular material

ρ_{base} = Relative density of base cellular design

r_i = Radius of circular holes in a coupon

$\rho_{optimal}$ = Relative density of optimized cellular design

ρ_{rel} = Density of cellular material divided by density of solid from which cells are made of

ρ_s = Density of solid material from which cells are made of

w_i = Weighting coefficient

x = Design variable

y_i = Distance of the centroid of the hole from the base of the coupon

z_i = Distance of a layer of holes in a plate from a reference axis

φ_i = Fitness value of ith design

E = Elastic modulus

$\mathbf{F(x)}$ = Vector objective function

$F(x)$ = Scalar objective function

K = Parameter space

K = Initial stiffness of a cellular beam

I = Moment of inertia

I_x = Moment of inertia about *x*-axis

I_y = Moment of inertia about *y*-axis

L = Cellular beam span

P = Load applied on the cellular beam

R_i = Fraction of fitness value of a design with respect to total fitness of all designs

V	=	Volume of cellular beam
Ω	=	Feasible region
Λ	=	Objective function space

1. INTRODUCTION

1.1 General

Fiber Reinforced Polymer (FRP) composites have emerged as attractive candidate materials for civil infrastructure applications because of their light weight, high strength, non-corrosive and non-magnetic characteristics compared to conventional construction materials such as concrete and steel [24]. In spite of these attractive features their use has been limited due to various factors like higher initial material costs, their use in non-efficient structural forms and their environmental impact [18]. This chapter presents an overview of this thesis, specific objectives and is concluded with a brief outline of the thesis.

1.2 Overview

With increasing emphasis on sustainability and eco-efficiency in the selection of materials [24] for all applications, including civil infrastructure applications, there is more demand to use bio-based products as alternatives to conventional as well as new materials like FRP. Natural-fiber-reinforced polymer composites, or biocomposites, seem to be a cost-effective and environmentally friendly option to low to medium grade synthetic FRP composites [24]. In spite of this appeal, their use has been limited to non-primary or non-load-bearing applications due to their lower strength and stiffness. Biocomposite materials with specific properties equivalent to entry-level structural materials are feasible [24] for example specific modulus (modulus per unit density) of a hemp and unsaturated polyester resin (UPE) composite with 25% fiber volume fraction is about 5.07×10^6 N-mm/g while specific modulus of a glass/UPE composite with 29% fiber

volume fraction is about 5.74 x10^6 N-mm/g [28]. However, this performance level is still unable to compete with existing construction materials.

The performance of a structural component, however, depends not only on its material but also its sectional and structural properties. Nature is abundant with examples where low individual properties of simple materials are enhanced manifold through hybrid and hierarchical designs that place such materials in specific locations for highest structural efficiency [8][9][26][29]. Among nature's most common efficient structures are hierarchical cellular sandwich structures [9][12][18][29]. These structures consist of a complex arrangement of cells of varied sizes arranged across the section such that dense regions are integrally connected to regions with lower density region or core (Figure 1.1).

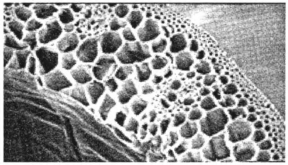


Figure 1.1 Cellular structure of a plant stem [18]

Varied cell sizes in specific locations along the cross-section lead to *hierarchy* in their arrangement. Smaller cell sizes appear towards regions of high stress in the cross-section, implying the importance of more material in those regions, while larger cell sizes in the cross-section indicate removal of material where it is not needed. Thus, the different cell sizes are arranged in the order of their importance. The feasibility of using biocomposites for load bearing components has been positively evaluated by prior work on laboratory scale (13mm x 102mm x 305mm) cellular plates made from industrial hemp fibers and

unsaturated polyester resin (UPE) using circular cells of varying diameter. Analytical and experimental results showed that the cellular bio-panels can be as efficient as commercial fiberglass panels [3][5][28].

Nature's complex level of material and structural design is an outcome of millions of years of evolution [28][31]. With the aid of high speed computational tools for exploration of complex shapes through mathematical optimization, engineers have been able to successfully simulate nature's ingenious evolutionary processes to a great extent. Structural optimization methods, in particular topology optimization methods, can be employed to remove or redistribute material in a rational iterative manner from within given structural domains subject to load and boundary conditions [9]. The optimal forms thus obtained represent a contour diagram of material density from which the optimal material distribution must be discerned either automatically, using image processing algorithms, or intuitively, using engineering judgment, or by a combination of both. A typical topology optimization solution for a cantilever subjected to a tip load is shown in Figure 1.2.

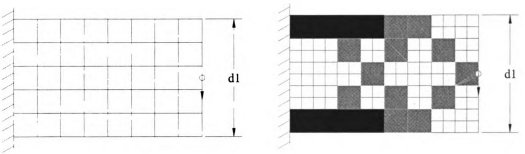
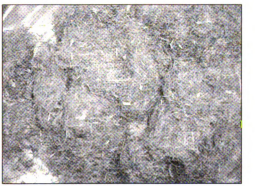


Figure 1.2 Typical topology optimization solution for a cantilever subjected to point load

It can be seen that the topology optimization solution has material distribution with varying material density indicated by different colored cells which brings hierarchy in the

material distribution. Thus, topology optimization can be used to obtain hierarchical designs, as in nature, to improve the low strength and stiffness properties of biocomposite components. However, in spite of its power and efficiency, typical results from topology optimization feature complex geometries that can be very difficult to manufacture using biocomposite materials as explained next.

Polymer biocomposites consist of natural fibers embedded in a natural or synthetic matrix. Natural fibers are typically short and intertwined, or clumped, (Figure 1.3(a) and (b)). Thus, manufacturing methods like hand-layup, compression molding or vacuum assisted resin transfer method (VARTM) are commonly used to manufacture biocomposites using natural fibers. The nature of biocomposite materials thus poses restrictions to the geometries and topologies that are feasible to manufacture with them. The results from conventional optimization methods such as topology optimization methods are thus difficult to use to obtain hierarchical forms for biocomposite components.



(a) Industrial hemp fibers

(b) Kenaf fibers

Figure 1.3 Natural fibers

Innovative topology optimization methods have been recently proposed that use introduction of finite-size features/perforations, some of which may be comparable to the

size of the structure, by using multiresolution analysis for material distribution [7]. Such approaches are robust and sophisticated yet they are not ideally suited for the problem at hand. Finite-size topology optimization routines are based on strain-energy minimization formulations. Thus, this applies to the longitudinal response of a loaded element, i.e. region B in Figure 1.4. In addition, the classical formulation of topology optimization is not directly suitable for multiobjective optimization problems or hybrid materials, which require re-formulation of the algorithm. Thus, in order to obtain cross-sectional layout made up of regions with stiffer material layers along the longitudinal direction (shown by area "A" in Figure 1.4), subjected to uniform stress distribution, as in the objective of this study a different optimization approach was needed.

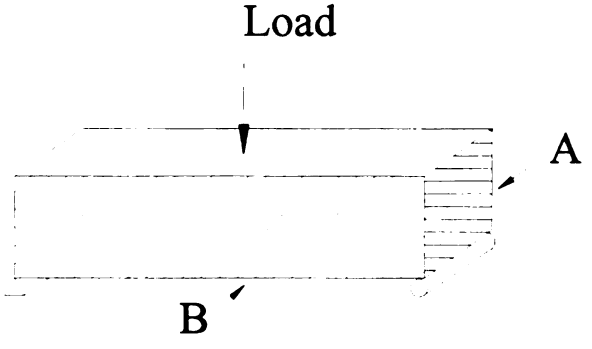


Figure 1.4 Transverse and longitudinal cross-sections in a beam considered for material layout optimization

Considering the manufacturing limitations to components made from biocomposites, the level of complexity of present layout optimization methods using finite-size features, and their inability to have hybrid material distribution, two approaches are presented in this thesis aimed at optimizing the sectional properties of biocomposite panels through hybrid and hierarchical cellular geometries. Both approaches rely on creating cellular designs by

introducing finite size perforations in a panel's cross-section to optimize the distribution of material. The first approach treats the problem as a sizing optimization problem and achieves hierarchical cellular geometries along beam/panel cross-sections through varied perforation sizes. This approach uses a gradient based method for optimization. The second approach treats the problem as a stochastic search for material distribution, or layout optimization problem, by using a pre-defined library of hybrid and non-hybrid coupons with finite-size features. The optimum layout for the beam/panel cross-section is then obtained by selecting coupons from this library using a directed random search with genetic algorithms.

The presented approaches for optimal cellular cross-sectional designs are not limited to biocomposite materials and can be used for other materials. Thus, the cellular designs based on the proposed approaches aim to maximize material and structural performance by improving sectional properties with designs that have finite size features in their cross-sections and which are thus easier to manufacture.

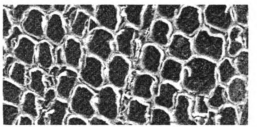
1.3 Biomimetics

Nature is abundant with structures in which design and function have been optimized over millions of years of continuous evolution. Engineers are faced with similar design constraints and objectives as nature. Thus, engineers have always been inspired by biological structures and have borrowed ideas from nature. This inspiration led to an altogether dedicated discipline for the study of same called 'Biomimetics.' In words of J.F.V. Vincent [29] biomimetics is "the abstraction of good design from Nature."

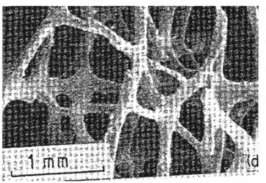
Though most of the materials used in nature are not of high performance their success lies in their strategic arrangement. The most efficient structural forms commonly seen in

nature are cellular sandwich structures [8][9], which consist of a dense skin integrally connected to a low density cellular core, as shown in Figure 1.4(a) [12][18][29]. Cellular materials are made of interconnected network of solid struts or plates, which can either have a general form of two-dimensional honeycombs (Figure 1.4(b)) or three-dimensional foams (Figure 1.4(c)). By integrally connecting the low density core and face sheets delamination between the core and face sheets, which can lead to premature failure of sandwich structures, can be avoided.

(a) A section through a bird's wing (Thompson 1961)



(b) Wood 2-D cellular structure



(c) Cancellous bone 3-D structure (Gibson and Ashby 1988)

Figure 1.5 Sandwich structures and cellular structures in nature

Thus, through hybridization of materials having lower stiffness with ones having higher stiffness and arranging material in hierarchical cellular structures, Nature tries to improve

the performance of its "structures." There is also a multiplicity of structures in nature that represent different levels of aggregation of the load bearing materials. These *hierarchical* organizations are present in all biological composites [29][30]. Thus, hybridization and hierarchical arrangement of simple materials can lead to efficient load-bearing structures [4].

1.4 Objective and scope

The objective of this research is to develop optimized designs having finite size features for beam/panel cross-sections with the aim of attaining improved performance for biocomposite beams/panels as load-bearing structural components. The optimization approaches developed to achieve this objective make use of introducing finite size features in beam/panel cross-sections and distributing material in hybrid and hierarchical arrangements. The designs thus obtained are easier to manufacture using biocomposite materials. The optimized designs were also evaluated experimentally by manufacturing and testing laboratory-scale beams. The beam components were tested to evaluate their flexural performance as continuous beams to be used in one-way panel-like systems for load-bearing structural applications. The improvement gained from the strategic material and structural layout of biocomposite beams/panels in the optimal designs was determined by comparing the specific stiffness of an optimized design to one with a base cellular architecture. The organization and contents of the thesis is as follows:

Chapter 2 presents a background on structural optimization and mathematical background of various optimization methods. It also provides details for the specific optimization methods used in this study and their computational implementation.

Chapter 3 discusses the two optimization approaches used in this research work, formulation of the optimization problems to achieve the specific research objectives, and key results obtained from them. Limitations observed in the two optimization approaches are also discussed.

Chapter 4 describes the manufacturing and testing of cellular designs obtained from the optimization approaches developed in this study for evaluation of the manufacturing feasibility and improvements from the optimization process.

Chapter 5 summarizes the key conclusions of this study and briefly discusses recommendations for future work in the same direction.

2. STRUCTURAL OPTIMIZATION BACKGROUND

2.1 General

Engineering design is a complex and iterative process which often aims at finding "the best possible" or "optimum" solution that fulfills all design criteria within a given set of conditions. In general, 'best' implies cost-effective, efficient, reliable and durable systems. This conventional process becomes more and more complex as the design domain, number of variables involved in the design problem and number of constraints or conditions to be satisfied by the design, increases. This calls for a more systematic and efficient approach than the conventional iterative design process of trial and error. Optimization is one such mathematical tool that aims to solve the problems involved in the conventional design process. This chapter contains an introduction to structural optimization and discusses the general mathematical background of the various optimization methods used in this research, namely gradient based methods and genetic algorithms.

2.2 Structural optimization

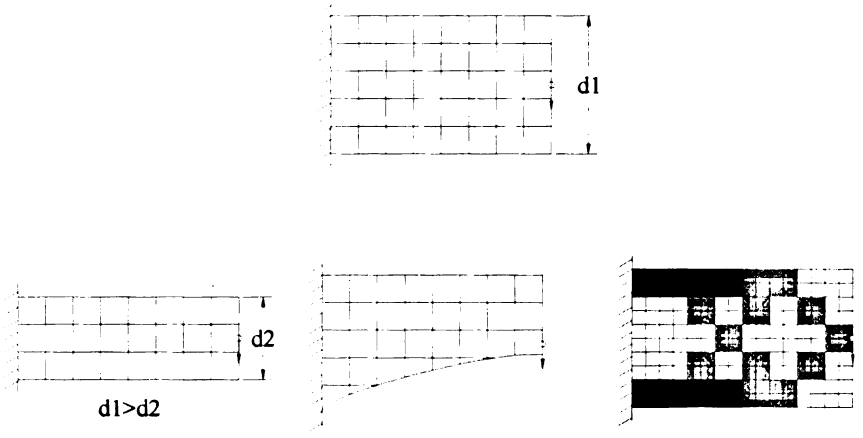
The structural design process consists of several stages. The first stage can be considered as the conceptual stage where the structural system and form are chosen. The second step is the preliminary design stage, where the shape of the structure and the defining geometry are decided. The final stage is the detailed design stage. The conventional design process goes through a number of iterations before a satisfactory detailed design is obtained. Use of optimization techniques at different stages of the structural design can

increase the efficiency and speed of the overall process and tremendously improve the results [13].

Structural optimization can be classified into several broad categories. First in its level of complexity is *sizing* optimization. It involves, for example, finding the optimum cross-sectional properties of a truss or a frame or thickness of a plate structure (Figure 2.1(a)). In these problems the design domain is fixed and doesn't change during the optimization process. Sizing optimization can be considered as the implementation of optimization at the detailed design stage.

A more complex type of optimization is called *shape* optimization, which aims at finding the optimal boundaries of two and three dimensional structural components. In this problem the domain is not fixed but the topology is. Coordinates of some key points in the boundary of the structure are usually considered as design variables (Figure 2.1(b)). For example, in the case of truss structures shape optimization considers two sets of design variables. First are geometric design variables, which control the position of the truss joints and second are the topological design variables which decide which nodes are connected by elements or members. This can be viewed as implementation of optimization techniques at the preliminary design stage.

Sizing and shape optimization may not lead to optimal results as they do not begin with an optimal topology. *Topology* optimization tries to overcome this shortcoming of sizing and shape optimization (Figure 2.1(c)). Typically, topology optimization of two and three dimensional continua aims at determining material distribution features such as holes and their location. Topology optimization essentially consists of the gradual removal of small portions of low stressed material that are being used inefficiently [13].



(a) Sizing optimization (b) Shape optimization (c) Topology optimization

Figure 2.1 Types of structural optimization

An ideal optimization approach is to simultaneously optimize the geometry (size and shape) and topology of a structure. This is sometimes called layout optimization and can be considered as optimization at the conceptual stage [13]. Various mathematical optimization techniques have been developed to solve the structural optimization problems just described at different design stages.

Structural topology optimization can be seen as a procedure for optimizing the rational arrangement of the available material in the design space and eliminating material that is not efficient. Many methods have been proposed for topology optimization [13], among them are: (i) the ground structure approach, (ii) the homogenization method, (iii) the bubble method and (iv) the fully stressed design technique. All these methods have been shown to be robust in performance and can yield solutions with fine material resolution and varied material density. However, practical designs with fine material resolution in complicated patterns are difficult to manufacture. An alternative approach has been

proposed to overcome this problem by introducing multiple finite-size heterogeneities in the structure for obtaining optimum layouts [7]. This approach uses a wavelet-based decomposition of material distribution and a multiresolution analysis. To represent the finite-sized perforations including perforations with dimensions comparable to the size of the structure a numerical homogenization method is used in this approach to characterize the "effective behavior" of perforated coupons of finite size. The method is again very robust and sophisticated and successfully solves typical structural topology optimization problems.

The structural optimization methods mentioned in section 2.2 are based on different techniques of mathematical optimization. The following section discusses in detail some of the mathematical optimization methods and their use in structural optimization.

2.3 Optimization Design Process

Success of an optimum design problem depends greatly on its mathematical formulation. Formulation of an optimum design problem involves translating a verbal description of the problem into a well-defined mathematical statement. The formulation process begins by identifying a set of variables to describe the system called *design variables*. Once numerical values are assigned to the design variables a design for the system can be obtained. All systems are designed to perform within a given set of constraints which include limitations on resources, material limits, acceptable response of the system, member sizes, etc. If a design satisfies all of its constraints, it is referred to as a *feasible design or system.* A criterion is needed to judge whether or not a given design is better than another. This criterion is called the *objective function, or cost function,* and is a

function of the design variables. A typical design process using optimization as a tool to achieve the desired design goal is summarized in the flow chart shown in Figure 2.2 [1].

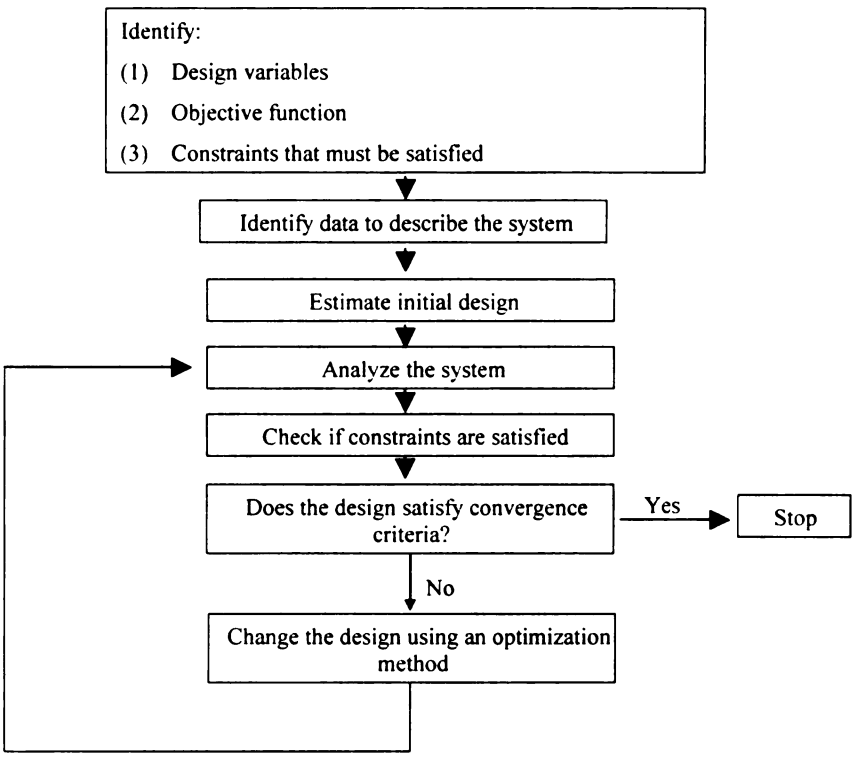


Figure 2.2 Optimum design process (adapted from [1])

2.4 Design variables, Objective functions and Constraints

2.4.1 Design variables

The notion of improving or optimizing a structure implicitly presupposes some freedom to change the structure. The potential for change is typically expressed in terms of ranges of permissible changes to a group of parameters. In optimization terminology such parameters are usually called *design variables*. Design variables may be *continuous* or *discrete*. Continuous design variables have a range of variation

and can take any value in that range. Discrete design variables can take only isolated values, typically from a list within a given range of permissible values. Once numerical values are assigned to the design variables, a design for the system can be obtained.

2.4.2 Objective function

There can be many possible or feasible designs for a system. As some of them might be better than the others, a criterion is required to constitute a basis of selection of one of the several alternatives. Such a criterion is called *an objective function* or *a cost function.* The objective function is a scalar function of the design variables and its least (or greatest) value is sought in an optimization process.

In a general optimization formulation, rarely does a single objective with several hard constraints adequately represent the problem being faced. More often there is a vector of objectives $F(x) = \{F_1(x), F_2(x), ...F_m(x)\}$ that must be satisfied in some way. The relative importance of these objectives is not generally known until the system's best capabilities are determined and trade-offs between the objectives are fully understood. As the number of objectives increases, trade-offs are likely to become complex and less easily quantified. There is no general and reliable method for solving multiobjective problems. However, several treatments for such situations are possible. For example, a composite cost function for the problem can be defined as a weighted sum of all the cost functions. A second way is to select the most important criterion as the objective function and treat the rest as constraints [1].

2.4.3 Constraints

A *constraint* is a restriction to be satisfied in order for the design to be feasible. It may take the form of a limitation imposed directly on a variable or group of variables (explicit constraint), or it may represent a limitation on quantities whose dependence on the design variables cannot be stated directly (implicit constraint) [1].

Many constraint functions include only first-order terms of the design variables. These are called linear constraints. Most general problems, though, have non-linear constraint functions. Design problems may have equality as well as inequality constraints. Constraint functions are generally designated as follows:

$$g_i(x) = 0 \qquad i=1, 2...m, \text{ for m such constraints} \tag{2.1.1}$$

$$g_i(x) \leq 0 \qquad i=1, 2.....n, \text{ for n such constraints.} \tag{2.1.2}$$

2.4.4 Mathematical statement of optimization problem

In algebraic terms, an optimization problem can be stated as [1]:

Minimize $F(x)$

$$\text{Subjected to } g_i(x) = 0 \quad i=1, 2...m, \text{ for m such constraints} \tag{2.2.1}$$

$$g_i(x) \leq 0 \qquad i=1, 2.....n, \text{ for n such constraints} \tag{2.2.2}$$

and explicit bounds on design variables

$$x_{il} \leq x_i \leq x_{iu} \quad i=1 \text{ to n} \tag{2.2.3}$$

where x_{il} and x_{iu} are respectively the smallest and largest allowed values for the ith design variable x_i.

Optimization methods in general can be classified into two broad categories:

(a) Optimality criteria methods or indirect methods

(b) Search methods or direct methods.

Each of these methods can be employed to solve both unconstrained as well as constrained optimization problems and are described further in the following section.

2.4.4.1 Optimality criteria methods or indirect methods

Optimality criteria are the conditions a function must satisfy at its minimum point. Optimization techniques seeking solutions to optimality conditions are referred to as indirect methods [1]. As the number of design variables and constraints becomes large, functions for the design problem (cost and constraint) become non-linear or the cost and/or constraint functions are implicit in the design variables. This is commonly seen in practical design optimization problems, so use of optimality criteria or indirect methods becomes difficult due to the large number of equations involved in the solution. This situation requires systematic numerical approaches for the optimum design of engineering systems. An alternative to indirect methods are direct methods, which are described in the next section.

2.4.4.2 Direct methods for unconstrained optimization

Direct methods seek an optimum solution by starting with an estimate of the initial design and improve it until optimality conditions are satisfied [1]. Many numerical methods based on search or direct methods are available to address the problems posed by optimality criteria methods, or indirect methods, for solving non-linear optimization problems. The key advantage of these numerical methods is that they are easy to

implement computationally. All unconstrained numerical optimization methods are based on the iterative formulae given in Equations 2.3.1 and 2.3.2, and schematically shown in Figure 2.3.

$$\text{Vector form} \quad \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \Delta\boldsymbol{x}^{(k)} \quad k = 0, 1, 2, \tag{2.3.1}$$

$$\text{Component form} \quad x_i^{(k+1)} = x_i^{(k)} + \Delta x_i^{(k)} \quad k = 0,1,2, \tag{2.3.2}$$

In these equations, k represents the iteration number, i denotes the design variable number, $\boldsymbol{x}^{(k)}$ is any starting design and $\Delta\boldsymbol{x}^{(k)}$ represents a small change in the current design. The iterative scheme noted above is continued until optimality conditions are satisfied or an acceptable design is obtained (Figure 2.3).

The problem of obtaining the change in design $\Delta\boldsymbol{x}$ is usually decomposed into two parts: the finding direction and the step size determination, expressed as

$$\Delta x^{(k)} = \alpha_k d^{(k)} \tag{2.3.3}$$

where $d^{(k)}$ is a desirable search direction of movement in the design space and $\alpha^{(k)}$ is a positive scalar called the step size in that direction. For an optimization problem with several variables, the finding direction problem is solved first. A step size is then determined by searching for the function minimum along a given direction in the design space.

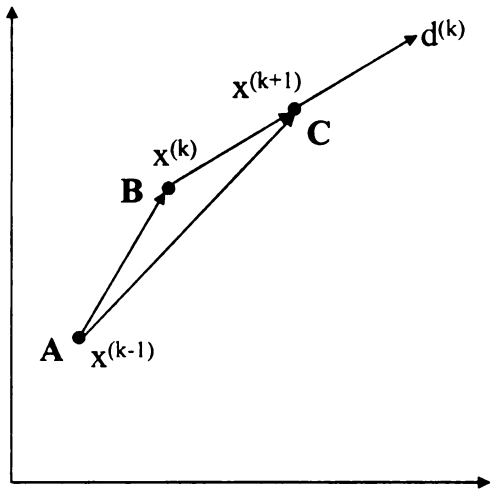


Figure 2.3 Conceptual diagram for iterative steps of an optimization method

Various numerical methods based on above concept have been developed to solve unconstrained optimization problems, e.g., the steepest descent method, Newton methods, and Quasi-Newton methods [1]. All these methods determine the search direction in different ways. For example, the steepest descent method uses a negative gradient vector to represent the direction of steepest descent for the cost function. For a function $f(\mathbf{x})$ of n variables $x_1, x_2, \ldots\ldots x_n$ the vector of partial derivatives of $f(\mathbf{x})$ with respect to variables $x_1, x_2, \ldots\ldots x_n$ at a given point x^* is termed as the gradient vector. Thus, the steepest descent method begins with an initial estimate for the minimum design and computes the direction of steepest descent at that point. If the direction is non-zero after moving along it, as long as possible to reduce the cost function, the new design point is thus obtained. Once at the new design point the direction of steepest descent is computed again and the entire process is repeated. This method uses only first order derivative information in the representation of the cost function at a point to determine the direction of travel. If second-order derivate information is used a better rate of

convergence can be expected. Newton's method uses a second order Taylor's series expansion of the function about the current design point. From it, the Hessian matrix is obtained by differentiating each component of the gradient vector with respect to each x_i. The Hessian matrix thus defines a quadratic rate of convergence for Newton's method. However, this approach suffers from certain drawbacks. For most engineering problems calculation of the Hessian matrix is expensive due to the large number of calculations involved. Like the steepest descent method, Newton's method does not possess a learning process, that is, each iteration starts with a new set of design variables without using information from previous iterations. Also, if the Hessian of the function becomes singular at any point then the method runs into difficulties. Quasi-Newton methods try to overcome these difficulties by using computation of not only the first derivatives but they use information from previous iterations [1]. This leads to higher rates of convergence towards the minimum. Several other methods are also available based on same iterative philosophy.

2.4.4.3 Direct methods for constrained optimization

The unconstrained optimization methods discussed above can also be used to solve constrained optimization problems by transforming the constrained problem to an unconstrained one by means of a composite function. The composite function is defined using the cost and constraint functions. It also contains certain parameters called penalty parameters that penalize the composite function for the violation of constraints. The larger the violation, the larger is the penalty. Once the composite function is defined for a set of penalty parameters, it is optimized using any of the unconstrained optimization techniques. The penalty parameters are then adjusted based on given conditions and the

composite function is redefined and optimized. The process is continued until there is no significant improvement in the estimate of the optimum point.

Constrained optimization problems may also be solved directly by so called "primal methods." Conceptually, algorithms for constrained and unconstrained optimization problems are based on the same iterative philosophy. However, for the former, constraints must be considered while determining the search direction as well as step size for the constrained problem.

All algorithms need a design estimate to initiate the iterative process. The starting design can be feasible or infeasible. Most of the direct search algorithms comprise of following basic steps [1]:

1. Linearize the cost and constraint functions about the current design estimate.
2. Define a search direction determination subproblem using the linearized cost and constraint functions.
3. Solve the subproblem that gives a search direction in the design space.
4. Calculate the step size required to minimize an appropriate descent function in the search direction.

Linearization of the cost and constraint functions means that at each iteration the design change is computed by solving an approximate "subproblem" obtained by writing linear Taylor's series expansions of the cost and constraint functions.

Based on the above philosophy many numerical methods or algorithms have been developed to solve constrained optimization problems. Some examples are the sequential linear programming method (SLP), the sequential quadratic programming method (SQP), the constrained steepest descent method (CSD), and constrained quasi-Newton methods

(CQN), [1]. SLP uses linear programming methods to obtain the design change. SQP iteratively solves a quadratic programming subproblem and the step size is determined by minimizing a descent function along the search direction. The quadratic programming (QP) subproblem has a quadratic cost function and linear constraints. The solution of the QP is used to form the search direction. The constrained steepest descent methods modify the search direction so that it satisfies all the linearized constraints. Several other methods, like the method of feasible directions, the gradient projection method, and the generalized reduced gradient method, are also used to solve constrained optimization problems.

2.4.5 Multiobjective optimization

In any general optimization formulation, rarely does a single objective with several hard constraints adequately represent the problem being faced. More often there is a vector of objectives $\boldsymbol{F}(x) = \{F_1(x), F_2(x), ...F_m(x)\}$ that must be collectively satisfied in some way. The relative importance of these objectives is not generally known until the system's best capabilities are determined and trade-offs between the objectives are fully understood. As the number of objectives increases, trade-offs are likely to become complex and less easily quantified. There is much reliance on intuition of the designer and his or her ability to express preferences throughout the optimization cycle. Thus, requirements for a multiobjective design strategy are to enable a natural problem formulation to be expressed, yet be able to solve the problem and enter preferences into a numerically tractable and realistic design problem.

Multiobjective optimization is concerned with the minimization of a vector of objectives $\boldsymbol{F}(x)$ that may be the subject of a number of constraints or bounds.

$$\text{Minimize } \boldsymbol{F}(x) = \{F_1(x), F_2(x), \ldots\ldots F_m(x)\} \tag{2.3.4}$$

$$x \in \kappa^n$$

$$g_i(x) = 0 \qquad i = 1,\ldots, m_e \tag{2.3.5}$$

$$g_i(x) \le 0 \qquad i = m_e + 1,\ldots\ldots, m \tag{2.3.6}$$

$$x_l \le x \le x_u \tag{2.3.7}$$

The key point to be noted here is that, because $\boldsymbol{F}(x)$ is a vector, if any of the components of $\boldsymbol{F}(x)$ are competing there is no unique solution to this problem. Instead, the concept of noninferiority (also called Pareto optimality) must be used to characterize the objectives. A noninferior solution is one in which an improvement in one objective requires a degradation of another. To define this concept more precisely, consider a feasible region Ω in the parameter space $x \in \kappa^n$ that satisfies all the constraints, i.e.

$$\Omega = \left\{ x \in \kappa^n \right\} \tag{2.3.8}$$

Subject to

$$g_i(x) = 0 \qquad i = 1,\ldots, m_e \tag{2.3.9}$$

$$g_i(x) \le 0 \qquad i = m_e + 1,\ldots\ldots, m \tag{2.3.10}$$

$$x_l \le x \le x_u \tag{2.3.11}$$

This allows defining a corresponding feasible region for the objective function space Λ.

$\Lambda = \left\{ y \in \kappa^n \right\}$ where $y = \boldsymbol{F}(x)$ subject to $x \in \Omega$

The performance vector, $\boldsymbol{F}(x)$ maps the parameter space into the objective function space as is represented for a two-dimensional case in Figure 2.4.

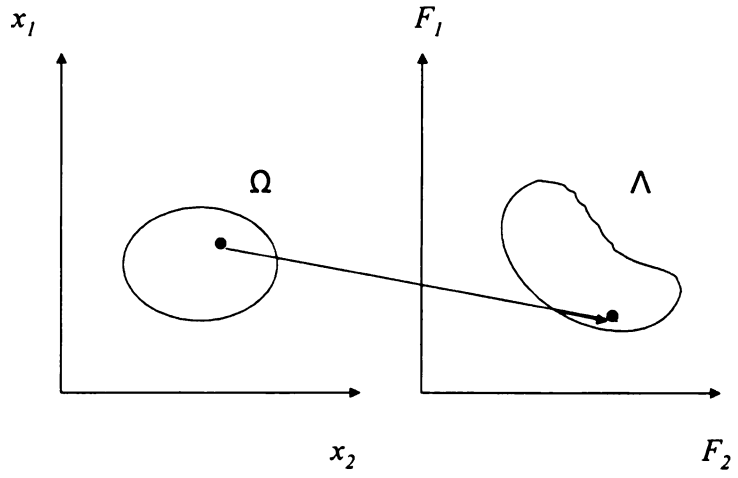


Figure 2.4 Mapping from parameter space into objective function space

A noninferior solution point can now be defined as a point, for some neighborhood of x^*, such that there does not exist a Δx such that $(x^* + \Delta x) \in \Omega$ and

$$F_i(x^* + \Delta x) \leq F_i(x) \qquad i = 1,...,m \tag{2.3.8}$$

$$F_j(x^* + \Delta x) \leq F_j(x) \qquad \text{for some j} \tag{2.3.9}$$

If A and B are two such points under consideration then they are noninferior solution points because an improvement in one objective, F_1, requires a degradation in the other objective, F_2, i.e.:

$$F_{1B} < F_{1A}, F_{2B} > F_{2A}.$$

Since any point in that is *not* a noninferior point represents a point in which improvement can be attained for all of the objectives, it is clear that such a point is of no value. Multiobjective optimization is, therefore, concerned with the generation and selection of

noninferior solution points. The techniques for multiobjective optimization are wide and varied [1][14]. An overview of one of them is given next.

2.4.5.1 Weighted Sum Strategy

The weighted sum strategy converts the multiobjective problem of minimizing the objective function vector into a scalar problem by constructing a weighted sum of all the objectives.

$$\text{Minimize } f(x) = \sum_{i=1}^{m} w_i . F_i(x) \qquad (2.3.10)$$

$$x \in \Omega$$

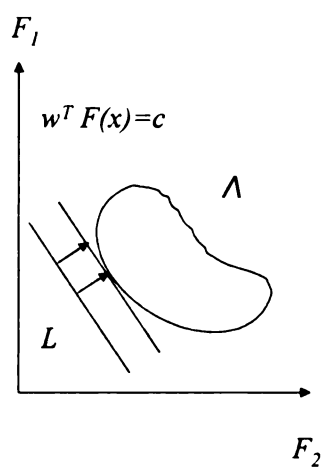


Figure 2.5 Graphical representation of the weighted sum method

The problem can then be optimized using a standard unconstrained optimization algorithm. The problem here is in defining weighting coefficients to each of the objectives. The weighting coefficients do not necessarily correspond directly to the relative importance of the objectives or allow trade-offs between the objectives to be expressed. Further, the noninferior solution boundary may be nonconcurrent so that certain solutions are not accessible. This can be illustrated graphically. Consider the two-objective case in Figure 2.5. In the objective function space F_1F_2 a line L is drawn. The

minimization of Eq. 2.3.10 can be interpreted as finding the value of c for which L just touches the boundary of Λ as it proceeds outwards from the origin. Selection of weights, therefore, defines the slope of L, which in turn leads to the solution point where L touches the boundary of Λ. It is important to note that a convexity problem arises when the lower boundary of is nonconvex. In this case the set of noninferior solutions between A and B is not available.

Both indirect and direct methods for scalar as well as multiobjective optimization suffer from certain limitations. First, both these methods are local in scope, that is, the optima they seek are best in a neighborhood of the current point. So, if the function is not smooth or if it is highly non-linear, these methods might give only a local minimum and not a global minimum. Second, calculus-based methods depend upon the existence of derivatives. Even if numerical approximations of derivatives are obtained, they are not suitable for discontinuous and multi-modal functions. Thus, these methods are insufficiently robust in rugged domains. Exhaustive search or enumerative schemes have also been considered for such problems, in which the search algorithm looks at objective function values at every point in space, one at a time. But for very large spaces the exhaustive search methods turn out to be inefficient and computationally expensive. Random search algorithms address to some extent the shortcomings posed by exhaustive search methods; however, undirected random search can become computationally expensive and inefficient, especially if the design domain is extremely large. The answer to most of the above mentioned problems probably lies in a directed random search like that provided by *genetic algorithms*.

2.4.6 Genetic algorithms

A genetic algorithm is a directed random search that tries to find the global optimum in complex, multidimensional and multi-modal search spaces. It emulates natural evolution wherein the operators used are inspired by the evolution process. Genetic algorithms use random choices as a tool to guide a highly exploitative search through coding of a parameter space [6]. There are two basic mechanisms that link a genetic algorithm to the problem that it is solving. The first is a way of *encoding* solutions to the problem on chromosomes and the second one is an *evaluation function* that returns a measurement of the merit of any chromosome in the context of the problem.

An overview of a conceptual genetic algorithm is given as follows:

1. Initialize a population of chromosomes.
2. Evaluate each chromosome in the population.
3. Create new chromosomes by mating current chromosomes; apply mutation and recombination as parent chromosomes mate.
4. Delete the members of the population to make room for the new chromosomes.
5. Evaluate the new chromosomes and insert them into the population.
6. If a termination condition is met, stop and return the best chromosome; if not go to 3.

The individuals of the population, i.e. the parameters to be optimized, are represented in a string called "chromosome," as genetic operators are suitable for this type of representation. The string can be represented in terms of binary numbers or real numbers (integers); the former representation is more commonly used. These strings are equivalent

to chromosomes in natural evolution, thus named chromosomes from here on. Each chromosome represents a possible solution, i.e., values from a variable set. To create new individuals two parents are required, which are then acted upon by *genetic operators* (namely *crossover* operators) to yield "off springs" or "better solutions." The crossover operator creates an information exchange between the parent chromosomes. The "offsprings" obtained by crossover possess features from both parents but sometimes crossover can also result in offsprings that are completely different from their parents. Thus, it is largely responsible for increasing diversity in a given population. There exists a fitness value for every individual, which indicates the merit of an individual in the evolution process. Parents are randomly chosen from the population (by probabilistic selection methods or operators) and individuals with high fitness value have higher probability of getting selected as parents than individuals with low fitness value. The genetic operation of *mutation* is randomly applied to the individuals. This causes a random modification of a local part of the chromosomes, which may cause the new chromosomes to be different from their parents. Since individuals may pass away, as their lifetime depends upon their fitness value, the mutation operator mainly tries to provide supplementary diversity in a population to avoid loss of diversity. This diversity allows the exploration of larger regions of the search space.

If all goes well throughout the process of simulated evolution an initial population of chromosomes will improve as parents are replaced by better and better children. The best individual in the population can thus be a highly-evolved solution to the problem. Finally, when the termination condition is met the best individual is returned as the optimum solution. The termination criterion can be a specified number of generations or other

stopping conditions such as the number of successive generations for which no modification occurs in the problem.

2.4.6.1 Computer implementation of a genetic algorithm

Genetic algorithms process populations of strings. So, the primary data structure for a simple genetic algorithm is a string population. One of the ways to implement populations is to construct it as an array of individuals where each individual contains the phenotype (the decoded parameter or parameters), the genotype (the artificial chromosome or bit string), and the fitness (objective function) value along with other auxiliary information. This structure is shown in Figure 2.6[10].

INDIVIDUAL NUMBER	INDIVIDUALS			
	STRING	x	f(x)	OTHER
1	01111	15	225	
2	01001	9	81	
n	00111	7	49	

Figure 2.6 Schematic of a string population in a genetic algorithm (adapted from [10])

Genetic operators may be applied to an entire population at each generation and as a result they act on two non-overlapping populations. The offsprings are created from

members of the *old population* (data structure) and are placed in the *new population.* The *old population* is then reset or regarded as the *new population.* A single overlapping population can also be maintained by carefully keeping record of who replaces whom in successive populations [6][10][27].

2.4.6.2 Initial population

Most genetic algorithms work with a fixed-size population so that the size of population of designs determines the size of population in all future generations as well. Choosing the population size is often a matter of trial and error. In general, the optimal size of a population increases with problem size. The initial population is typically generated at random. Once a population is generated, a series of genetic operators are applied that produce a new generation and replace the initial population with a new one. The first of these operators is selection [6][10][27].

2.4.6.3 Selection and fitness

The selection process in genetic algorithms mimics biology in giving more fit designs a higher chance to breed and pass on their genes to future generations. Genetic algorithms, therefore, need to define fitness and use it in a procedure that selects pairs of parent designs that will be used to create child designs for future generations. For unconstrained problems, the fitness of a design can be the problem objective function (for maximization problems) or a constant minus the objective (for minimization problems). For constrained problems the fitness must also consider constraint violations or constraint margins. Once the fitness of all the designs in a population are defined, a common procedure for parent-

selection is to simulate a biased roulette wheel, with each design being assigned to a sector of the roulette wheel with an area proportional to its fitness. That is, with n_s designs having fitness values of ϕ_i, $i = 1,.....,n_s$, the ith design gets a fraction R_i of the wheel, where

$$R_i = \frac{\phi_i}{\sum_{j=1}^{n_s} \phi_j} \tag{2.3.17}$$

Defining fitness on the basis of the numerical value of the objective function carries the disadvantage that towards the end of the optimization, where the differences between competing designs becomes small, the selection pressure in favor of better designs becomes small, and thus progress slows down. To overcome this difficulty many other selection procedures such as stochastic remainder sampling, tournament selection, stochastic universal sampling and linear ranking have been proposed [6][10][27].

2.4.6.4 Crossover

Once pairs of parents are selected, the mating of a pair also involves a random process called crossover. In a genetic algorithm, crossover recombines the genetic material of two parent chromosomes to make two children. The simplest crossover operator is the one-point crossover. One point crossover occurs when parts of two parent chromosomes are swapped at a randomly selected point to create two children. One of the important features of one-point crossover is that it can produce children that are radically different from their parents. Two examples of one-point crossover are shown in Figure 2.7. Another important feature of one-point crossover is that it will not introduce differences

for a bit in a position where both parents have the same value, as shown in second example in Figure 2.7.

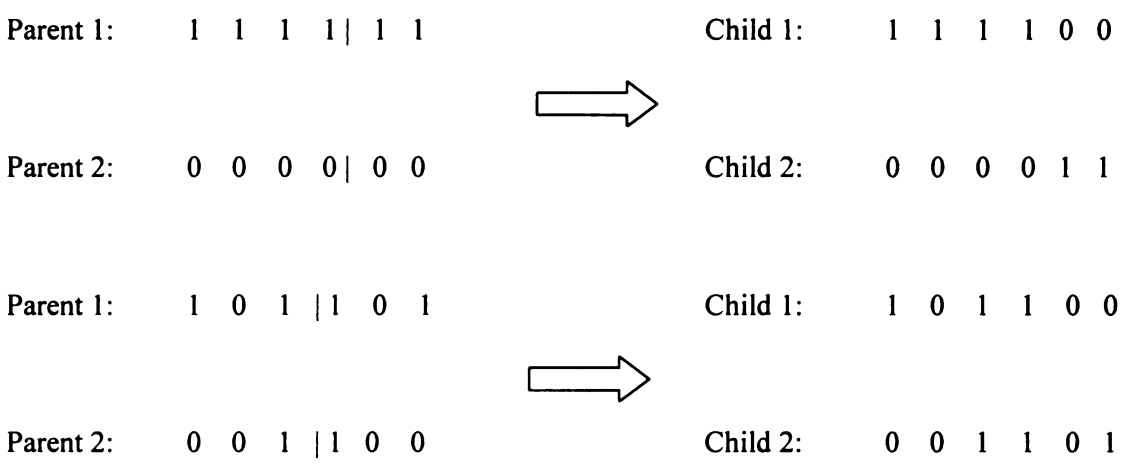


Figure 2.7 Two examples of one-point crossover (adapted from [6])

Crossover is usually implemented with some probability p_c. If crossover is not implemented, then one of the parents is cloned into the next generation. Implementation of crossover begins with the generation of a single random number r uniformly distributed between 0 and 1. If the number is less than p_c crossover will be performed by generating another random number for the cutoff point. The decisions on which parent to clone or which child design to select are not important, as the parents were selected at random. The processes of selection and crossover are repeated until there are n_s child designs. Multiple point crossovers in which information between the two parents is swapped among the string segments are also possible, but, because of the string mixing, the crossover becomes a more random process and might degrade the performance of the algorithm. However, a two-point crossover is an exception and choosing the end of the

string randomly as the second crossover point improves the performance of the algorithm. When integer or real variables are represented by binary strings, crossover may generate child designs that do not bear any resemblance to the parent designs. Although such possibilities reduce the effectiveness of the crossover operator for combining existing traits from parent designs into child designs, it contributes to the exploration of new design alternatives [6][10][27]. The process of exploration of alternative designs is usually handled by the mutation operator and it is preferable to let the crossover operator take care of recombination and let the mutation operator handle the exploration of new traits as described in next section.

2.4.6.5 Mutation

Mutation performs the important task of preventing premature loss of important genetic information by introducing an occasional random alteration to a string. Inferior designs may have some good traits that can get lost in the gene pool when these designs are not selected as parents. Additionally, mutation is needed when an integer or real coding is used because in most cases there is a low probability that all the possible genes are represented in the initial population. Mutation is implemented by randomly changing the value of a digit in the string with small probability (Figure 2.8). When real design variables are coded as binary numbers, there are situations where small changes in the design can not be achieved by mutations. This occasionally results in slower progress for the algorithm. To counteract this problem another type of binary coding called *gray coding* is sometimes used, which doesn't have abrupt changes in digits caused by small changes in the value of the number. Alternatively, another mutation operator called *local mutation* can be applied to the original variables rather than the coded ones. This local

mutation operator is also useful when actual values rather than binary numbers are used for the variables in the string [6][10][27].

Old Chromosome				Random Numbers				New Bit	New Chromosome			
1	0	1	0	0.80	0.102	0.266	0.373	-	1	0	1	0
1	1	0	0	0.12	0.096	**0.005**	0.840	0	1	1	1	0
0	0	1	0	0.760	0.473	0.894	**0.001**	1	1	0	1	**1**

Figure 2.8 Examples of bit mutation (adapted from [6])

2.4.6.6 Knowledge based evolution

In the course of evolutionary optimization solutions are often generated with low phenotypic (physical) fitness even though the corresponding genotype may be close to an optimum. Without additional information about the local fitness landscape such genetic *near misses* would be overlooked under strong selection. Such issues are usually addressed by hybrid genetic algorithms, which combine a global search using genetic algorithms and a local search using individual learning algorithms. Hybrid algorithms can either exploit active learning, using Lamarckian inheritance [6][10], or passive learning through the Baldwinian effect [6][10].

Under Lamarckian algorithms [6][10], performance gains from individual learning are mapped back into the genotype used for the production of the next generation. This is analogous to Lamarckian inheritance in evolutionary theory – whereby characteristics acquired during the parent's lifetime are passed on to their offspring. Although Lamarckian inheritance is rejected as a biological mechanism under the modern

synthesis, the algorithms based on the approach are able to perform mapping from the learned phenotype to the genotype [6]. It is sometimes referred to as an active mechanism and it results in change in the genetic makeup or genotype of the individual in the course of evolution.

The Baldwinian evolution model [6] suggests that individual learning can alter the course of evolution. According to this theory, evolution pressure favors individuals having learning capabilities. Individuals able to learn are less dependent on their genetically encoded traits. The frequency of the genes responsible for learning thus increases in subsequent generations. Individuals able to learn a certain adaptation (allowing learning individuals to survive preferentially) are more likely to produce offsprings having this learning capability. This implicit learning model is referred to as *Baldwinian effect.*

The Baldwinian effect highlights an indirect or passive mechanism allowing individual learning to influence the rate of evolutionary progress. By increasing survival chances and genetic diversity, individual learning capabilities increase the probability that the population evolves genetically encoded traits that better fit a challenging environment. It is referred to as a passive effect as learning capabilities enhance the chances of survival of an individual but do not change the genetic makeup of the individual.

3. DISCRETE LAYOUT OPTIMIZATION OF CELLULAR STRUCTURES

3.1 General

The manufacturing limitations of structural forms from biocomposite materials with hybrid material layouts make the implementation of conventional topology optimization solutions difficult. Thus, two approaches were developed for optimizing the sectional properties of biocomposite panels with hybrid material designs and hierarchical geometries. Both approaches consider finite size perforations in the cross-section of biocomposite panels to optimize the distribution of material. This chapter firstly presents the formulation of these optimization problems by identifying design variables and the objective functions. It is followed by the description and implementation of the respective optimization algorithms. The chapter is concluded by a discussion of the limitations observed in the two approaches.

3.2 Sizing optimization of cellular cross-sections using gradient search

The sizing optimization problem aimed at maximizing the cross-sectional stiffness and specific cross-sectional stiffness of panels subjected to one-way bending in a continuous panel system is shown in Figure 3.1. Considering a representative strip from the panel as shown in Figure 3.1, the indicated cross-section was optimized for maximum stiffness and specific stiffness (stiffness per unit volume of material). Solving an optimization problem involves transcribing a verbal description of the problem into a mathematical statement. Hence, sizing optimization was started by identifying design variables, a scalar

objective function, a vector of objective functions (called multiple objective functions) and corresponding constraints, which are presented in the following sections.

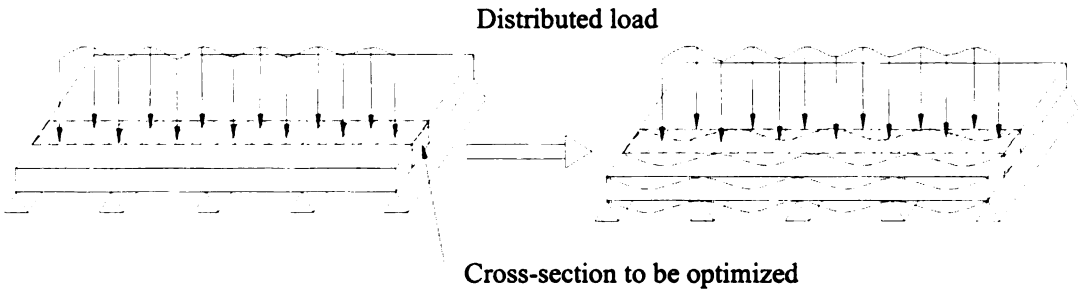


Figure 3.1 Continuous panel system subjected to distributed load in one-way bending

3.2.1 Design variables

In order to implement the sizing optimization problem a numerical problem was considered wherein a plate with cross-sectional dimensions of 100 mm x 80 mm was considered (Figure 3.2).

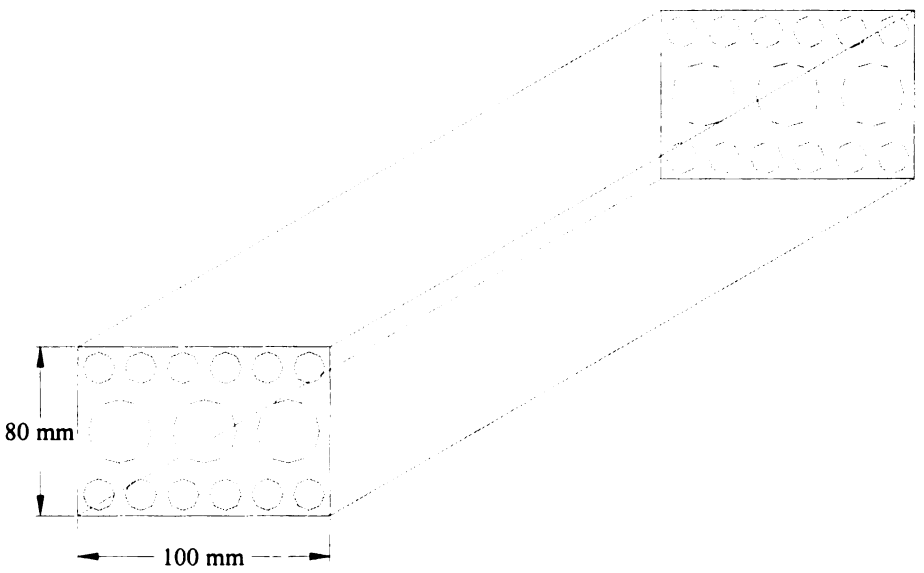


Figure 3.2 Cross-sectional view of a hierarchical cellular biocomposite panel

The cellular plate was assumed to be discretized into distinct layers where each layer was perforated with circular holes such that same size holes were considered to be present across a given layer. A linear bending-induced strain distribution was assumed across the cross section. Thus, for simplicity each layer in the plate was considered to be made up of a given number of repeating units or coupons. The repeating unit consisted of a circular hole in the center of a square coupon where the hole diameter was a function of the coupon size as shown in Figure 3.3.

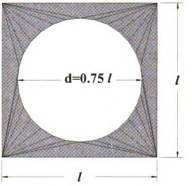


Figure 3.3 A characteristic repeating coupon

Circular holes were considered as a specific case but the approach is not limited to circular perforations and can be extended to other geometric shapes. A single column of these characteristic coupons was considered to be repeated throughout the width of the cross-section since the component was assumed to be under one-way bending. The coupon column can be repeated along the cross-section in two possible ways as shown in Figure 3.4(b) and Figure 3.4(c). The interpretation shown in Figure 3.4(b) has been followed throughout the sizing optimization results.

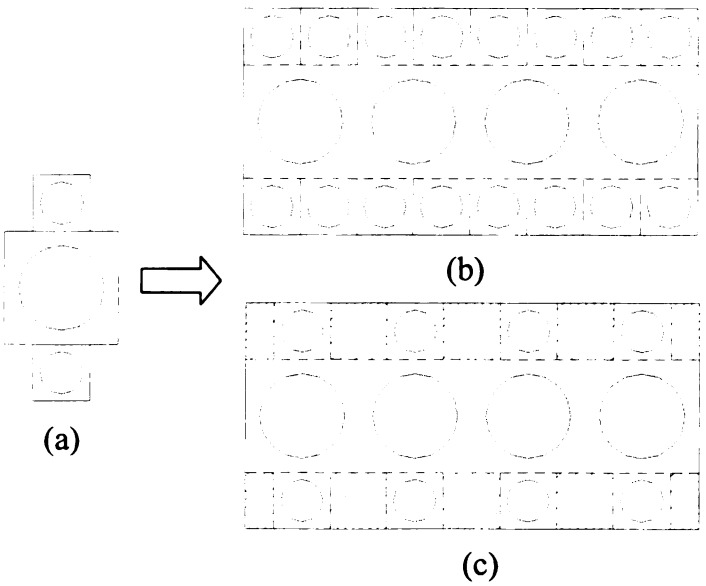


Figure 3.4 Interpretation of a three layer repeating column layout in a three layer plate

Variation of layer thickness changes the hole diameters as shown in Figure 3.4 for a three-layer plate. Therefore, distances of different layers from a reference axis were chosen as design variables (see Figure 3.5, Figure 3.6 and Figure 3.7). Three different panels, each consisting of three, five and seven layers were considered for the numerical example. The approach is not limited to the chosen numbers of layers and can clearly be extended to different numbers of layers. The number of layers chosen for each of the three panels was odd since symmetry about the midplane was intuitively expected following that the material was assumed to be linear elastic. Consideration of different number of layers was to add more resolution in the sizing optimization problem, thus allowing the presence of greater hierarchy of perforations in the panels. The design variables for three, five and seven layer panels are shown in Figure 3.5, Figure 3.6 and Figure 3.7, respectively.

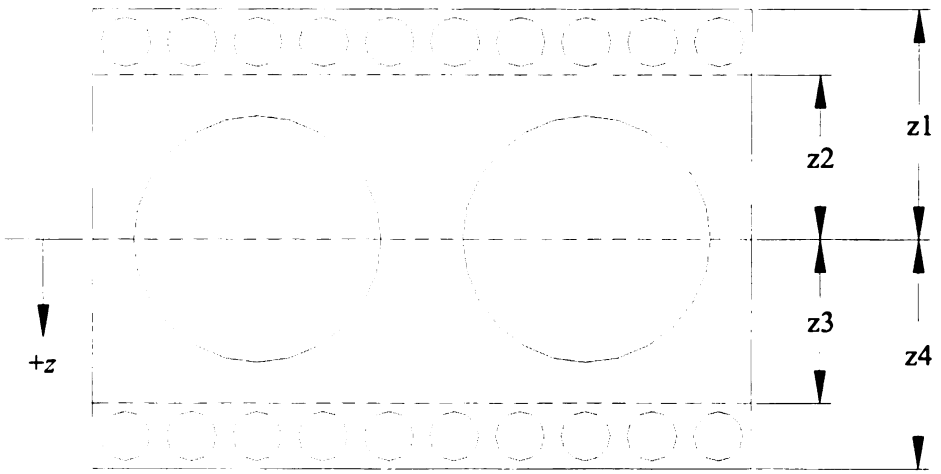


Figure 3.5 Design variables for the three-layer panel

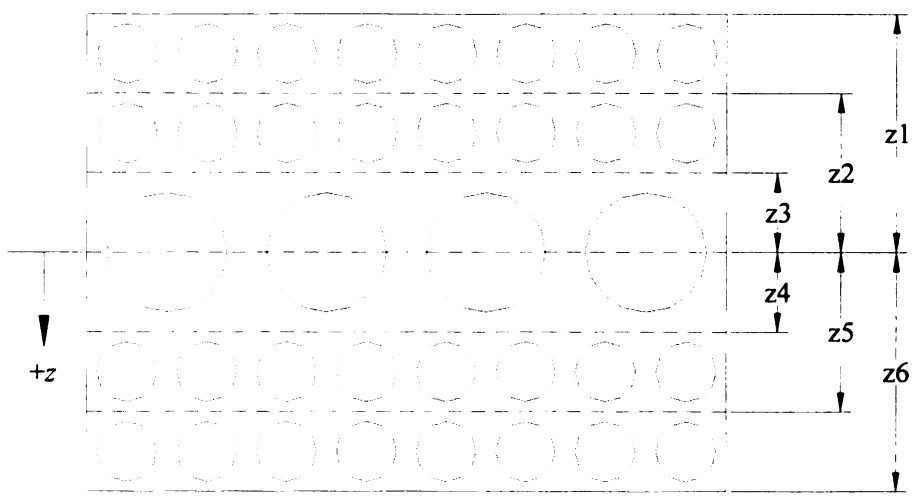


Figure 3.6 Design variables for the five-layer panel

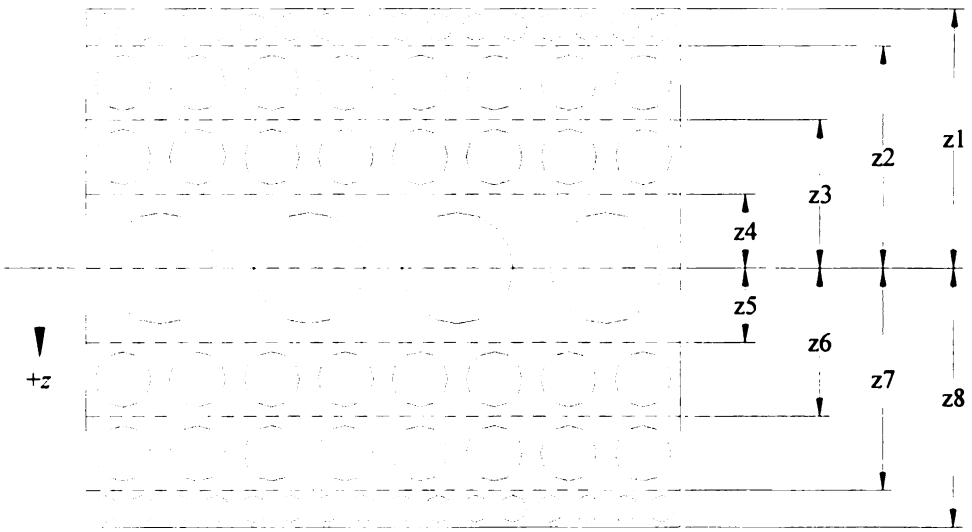


Figure 3.7 Design variables for the seven-layer panel

3.2.2 Objective function

The objective functions used for the scalar optimization problem were the section moment of inertia (Equation 3.1) and the section moment of inertia per unit volume (Equation 3.2). These objective functions were used to quantify the sectional stiffness and sectional specific stiffness (stiffness per unit weight).

$$f_1 = I \text{ and} \tag{3.1}$$

$$f_2 = \frac{I}{V} \tag{3.2}$$

For the multiobjective optimization problem a linear combination of moment of inertia and cross-sectional volume with weighting coefficients was used (Equation 3.3). The multiobjective formulation was considered to understand the trade-off in maximizing moment of inertia while simultaneously minimizing weight. The multiobjective function is given by:

$$f_3 = w_1 I + \frac{w_2}{V} \tag{3.3}$$

where w_1, w_2 = weighting coefficients and $w_1 + w_2 = 1$.

3.2.3 Constraints

The limits to which the movement of layers was confined were selected as the problem constraints. For the numerical problem discussed in Section 3.2.1, for the three layer plate two sets of constraints were studied as given below:

(I) -4 cm $< z_2 < 0$ and $0 < z_3 < 4$ cm

(II) -3 cm $< z_2 < 1$ and $1 < z_3 < 3$ cm.

Similarly, for the five layer plate the constraint sets were:

(III) -4 cm $< z_2 < 0$, $0 < z_3 < 4$ cm, $0 < z_4 < 4$ cm and $0 < z_5 < 4$ cm

(IV) -3 cm $< z_2 < 1$, $1 < z_3 < 3$ cm, $1 < z_4 < 3$ cm and $1 < z_5 < 3$ cm.

And for the seven layer plate the constraint sets were:

(V) -4 cm $< z_2 < 0$, $0 < z_3 < 4$ cm, $0 < z_4 < 4$ cm, $0 < z_5 < 4$ cm, $0 < z_6 < 4$ cm and $0 < z_7 < 4$ cm

(VI) -3 cm $< z_2 < 1$, $1 < z_3 < 3$ cm, $1 < z_4 < 3$ cm, $1 < z_5 < 3$ cm, $1 < z_6 < 3$ cm and $1 < z_7 < 3$ cm.

Since the overall plate thickness was kept constant for all cases, for the 3 layer plate $z_1 = z_4 = 4$ cm, for the 5 layer plate $z_1 = z_6 = 4$ cm and for the 7 layer plate $z_1 = z_8 = 4$ cm.

3.2.4 Implementation of sizing optimization problem in MATLAB

The sizing optimization problem was solved using the optimization toolbox in MATLAB [16] employing both scalar and multiobjective approaches for constrained optimization. The function "*fmincon*" was used to solve both the scalar and multiobjective optimization problems. MATLAB's "fmincon" function uses a sequential quadratic programming algorithm [1] to carry out constrained optimization (see Section 2.3.3). This function requires the constraints and the objective function to be passed as its arguments, which were written separately in script files and passed to it as functions. Figure 3.8 shows schematically the input arguments and output options, for the "fmincon" function. The output options can be decided beforehand and can include information like the optimum value of the design variables, or their value at the optimum solution, value of the objective function at the optimum solution, value of the gradient and Hessian [1] (see Section 2.3.2) at the optimum solution and as well as at each iteration. If the objective function is highly non-linear and/or analytical calculation of the gradient is not possible, "fmincon" can be made to use numerical gradients (calculated by the finite difference method) by turning the Hessian (see Section 2.3.2) option "off."

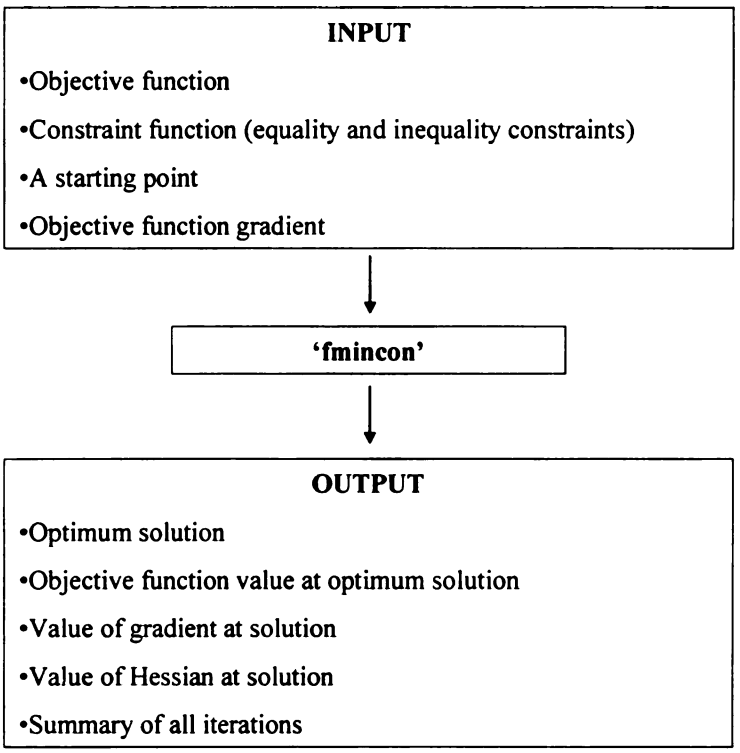


Figure 3.8 Constrained optimization using MATLAB's "fmincon" function

3.2.5 Scalar sizing optimization for maximization of stiffness

The results obtained for the scalar sizing optimization for maximization of sectional stiffness of three, five and seven layer plates subjected to constraint sets described in Section 3.2.3 are presented in Tables 3.1 through Table 3.3. In case of three, five and seven layer panels subjected to constraint sets I, III and V respectively, all the layers merged together. Thus, only a single layer of perforations was seen in the middle of the panel, leading to removal of maximum material from near the neutral axis and placement of maximum material towards top and bottom of the panels. Results for constraint sets I, III and V for the three, five and seven layer plates, respectively, show that in absence of

any restriction on movement of different layers the optimization process tries to remove most material from near the neutral axis and places it more towards the top and bottom, as seen in Figure 3.9. The layout for three layer panel subjected to constraint set II showed three distinct layers due to the lower bound on the design variables (Figure 3.10). The layouts obtained for five and seven layer plates with constraint sets IV and VI respectively were identical (Figure 3.11) and again showed material removal from the neutral axis and material placement towards top and bottom of the panel. For the seven layer panel two layers were merged and the layout had only five layers due to the imposed constraints and led to material placement towards the top and bottom of the panel. Thus, it was clearly seen that whenever some restriction was imposed on the movement of layers as seen in case of constraint sets II, IV and VI for three, five and seven layer panels, respectively, smaller sized perforations were placed on the top and bottom layers thus leading to removal of more material from the neutral axis. In the absence of such restrictions layers with smaller perforations at the top and bottom sides of the panel were avoided and the solutions tended towards a single large void layer at the section neutral axis.

Table 3.1 Results for scalar maximization of stiffness for a three-layer plate

Constraint set	z_2 mm	z_3 mm	Moment of Inertia (f_1) mm^4
I	-40	40	5.3×10^4
II	30	30	1.93×10^4

Table 3.2 Results for scalar maximization of stiffness for a five-layer plate

Constraint set	z_2 mm	z_3 mm	z_4 mm	z_5 mm	Moment of Inertia (f_1) mm^4
III	0	-40	40	0	5.3×10^4
IV	-30	-10	10	30	1.93×10^4

Table 3.3 Results for scalar maximization of stiffness for a seven-layer plate

Constraint set	z_2 mm	z_3 mm	z_4 mm	z_5 mm	z_6 mm	z_7 mm	Moment of Inertia (f_1) mm^4
V	-40	0	-40	40	0	40	5.3×10^4
VI	-10	-30	-30	30	30	10	1.93×10^4

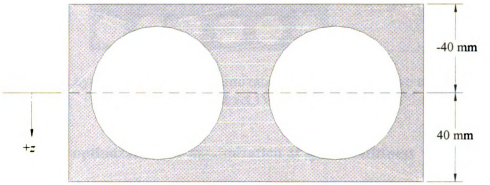


Figure 3.9 Layout obtained for maximization of stiffness for three, five and seven layer plates for constraint sets I, III and V respectively

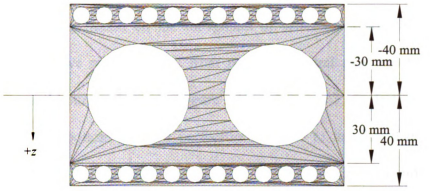


Figure 3.10 Layout obtained for maximization of stiffness for three layer plate for constraint set II

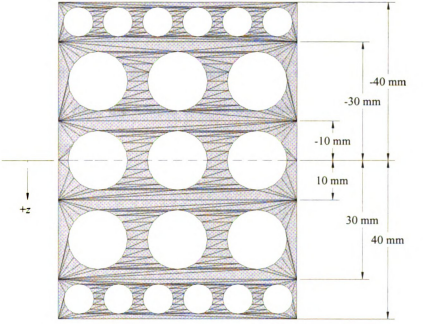


Figure 3.11 Layout obtained for maximization of stiffness for five and seven layer plates for constraint set IV and VI respectively

3.2.6 Scalar sizing optimization for maximization of specific stiffness

The results obtained for the scalar sizing optimization for maximization of specific sectional stiffness of three, five and seven layer plates subjected to constraint sets described in Section 3.1.3 are presented in Tables 3.4 through Table 3.6. The results for

constraint sets I, III and V for three, five and seven layer plates respectively show that, in the absence of any restriction on the movement of different layers, the optimization process tries to remove most material from near the neutral axis and places it more towards the top and bottom as seen in Figure 3.12 and Figure 3.13. In these figures, the presence of top and bottom layer of perforations is due to the fact that optimization algorithm tries to achieve a compromise solution between maximum moment of inertia (or stiffness) and minimum volume. However, when some restriction is imposed on the movement of layers, as seen in case of constraint sets II, IV and VI for the three, five and seven layer plates, respectively (Figure 3.14, Figure 3.15 and Figure 3.16), smaller sized perforation are seen near neutral axis as those layers were not allowed to coalesce together or move further away due to the lower bound on their movement, leading to more material removal from the neutral axis as was seen in case of constraint sets I, III and V mentioned above.

Table 3.4 Results for scalar maximization of effective stiffness for a three-layer plate

Constraint set	z_2 mm	z_3 mm	Moment of Inertia per unit volume (f_2) mm
I	-40	40	90
II	-10	10	70

Table 3.5 Results for scalar maximization of effective stiffness for a five-layer plate

Constraint set	z_2 mm	z_3 mm	z_4 mm	z_5 mm	Moment of Inertia per unit volume (f_2) mm
III	-25	-40	40	25	90
IV	-14	-10	10	14	60

Table 3.6 Results for scalar maximization of effective stiffness for a seven-layer plate

Constraint set	z_2 mm	z_3 mm	z_4 mm	z_5 mm	z_6 mm	z_7 mm	Moment of Inertia per unit volume (f_2) mm
V	-40	-25	-40	40	25	40	90
VI	-15	-13	-10	10	13	15	60

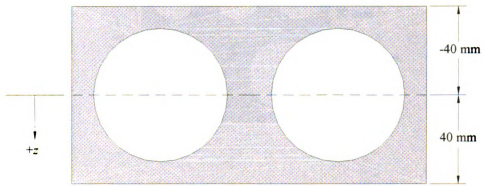


Figure 3.12 Layout obtained for maximization of effective stiffness for a three layer plate using constraint set I

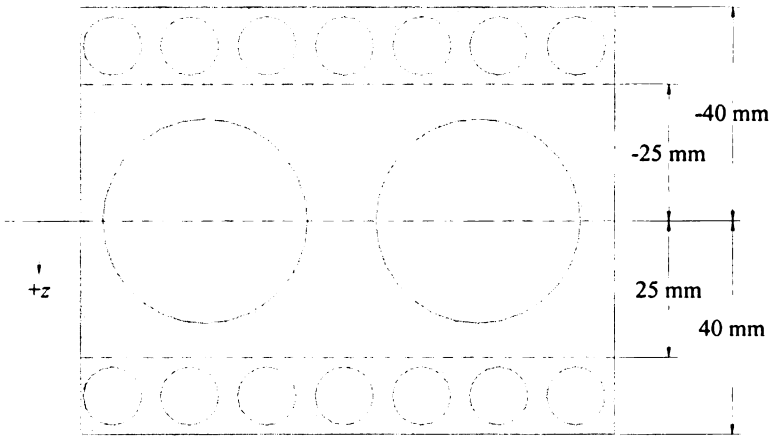


Figure 3.13 Layout obtained for maximization of effective stiffness for five and seven layer plates using constraint set III and V respectively

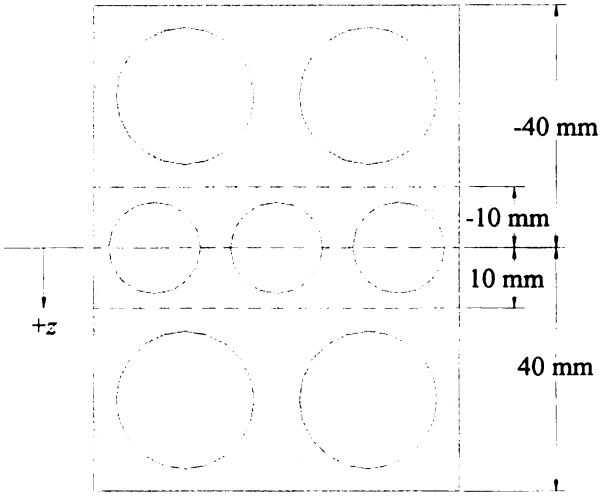


Figure 3.14 Layout obtained for maximization of effective stiffness for a three layer plate using constraint set II

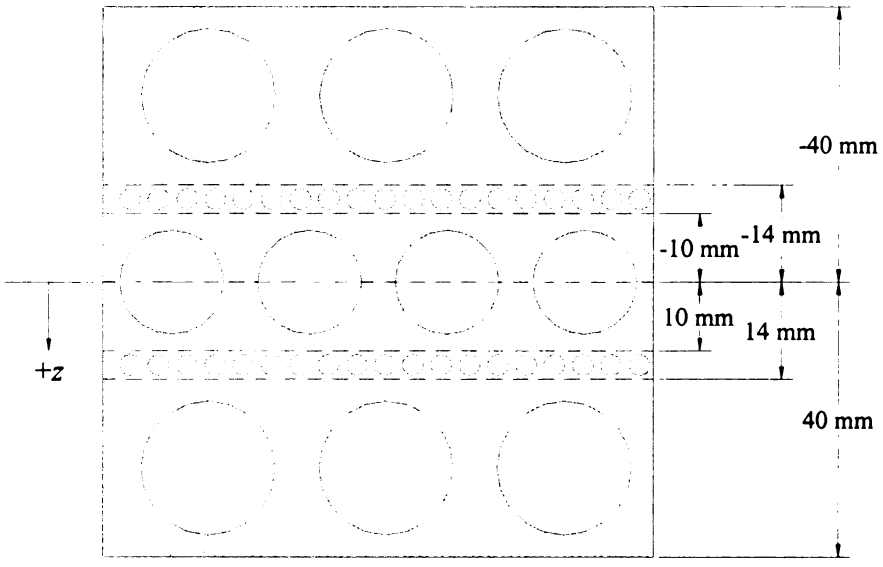


Figure 3.15 Layout obtained for maximization of effective stiffness for a five layer plate using constraint set IV

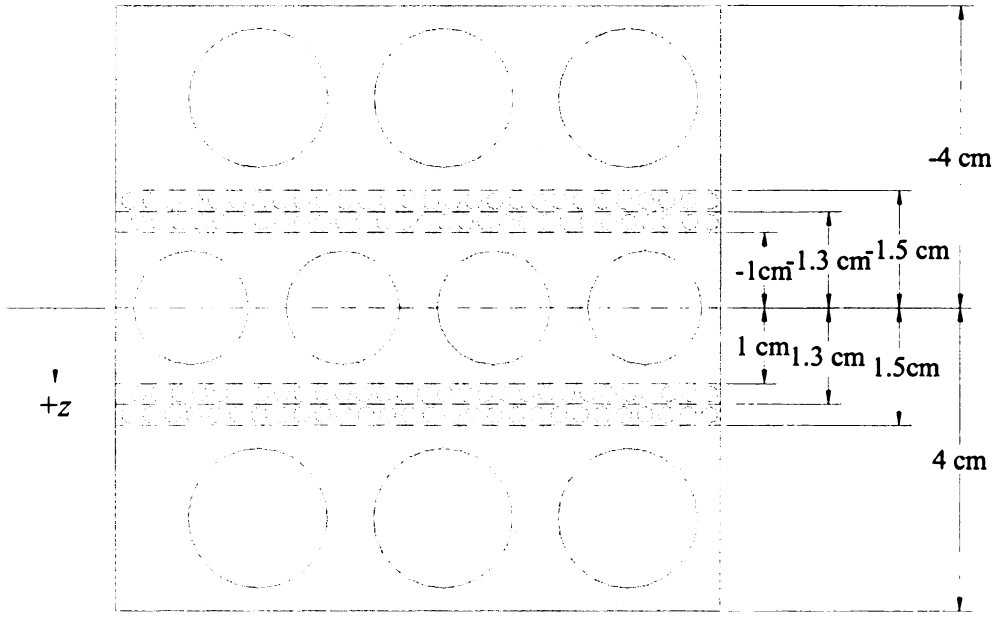


Figure 3.16 Layout obtained for maximization of effective stiffness for a seven layer plate using constraint set VI

3.2.7 Multiobjective sizing optimization using weighting coefficients method

The Pareto optimum sets for multiobjective optimization for maximization of stiffness and minimization of volume using the weighting coefficient method are presented in Tables 3.7 through 3.9 and described graphically in Figures 3.17 through 3.19. Only constraint sets I, II and V were considered for three, five and seven layer plates respectively. For the three layer plate not much significant trade-offs could be observed between the objective functions probably due to lower resolution, or too few layers. This implies that improvement in one function didn't lead to significant degradation of the other due to low resolution. However, with increase in the number of layers, as seen in case of five and seven layer panels, improvement in one function value led to significant degradation in the value of other and vice-versa. Thus, the increased resolution with different possible perforation sizes led to improvements and degradations in the two objective function values with different weighting coefficients.

Table 3.7 Pareto optimum set multiobjective optimization for a three layer plate

Weighting coefficient w	Maximization of moment of inertia objective function mm^4	Minimization of volume objective function mm^3
0.9	-1.14×10^6	1.7×10^4
0.8	-1.14×10^6	1.7×10^4
0.7	-1.13×10^6	1.6×10^4
0.6	-1.05×10^6	1.4×10^4
0.5	-1.05×10^6	1.4×10^4
0.4	-1.05×10^6	1.4×10^4
0.3	-1.04×10^6	1.4×10^4
0.2	-5.6×10^5	1.2×10^4
0.1	-5.1×10^5	1.2×10^4

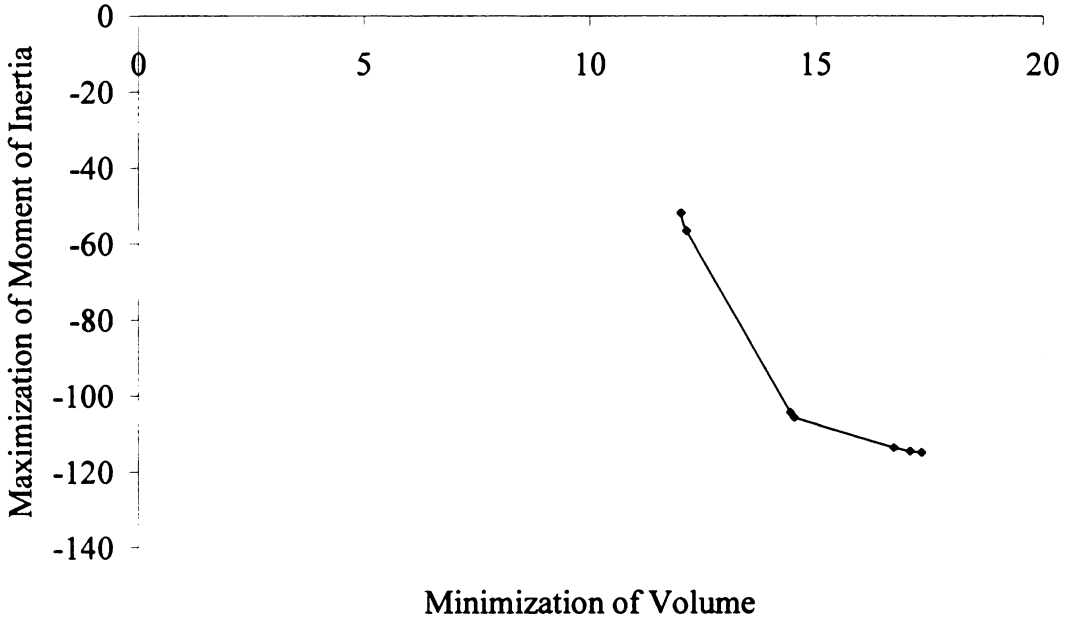


Figure 3.17 Pareto optimum set for multiobjective optimization of moment of Inertia and volume for a three layer plate

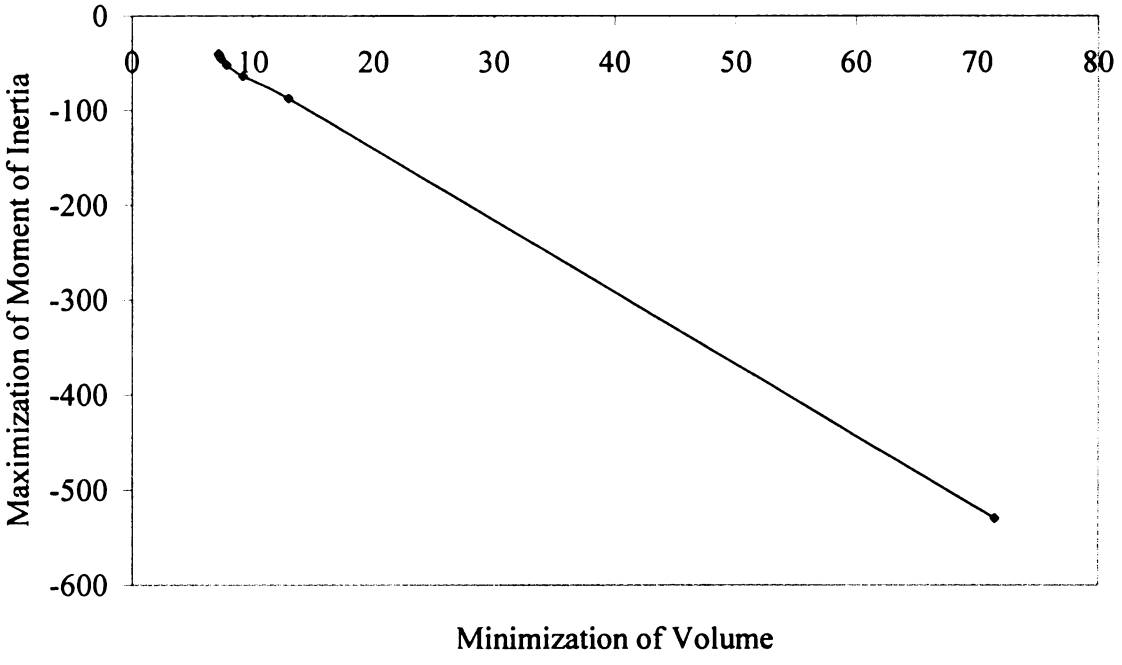


Figure 3.18 Pareto optimum set for multiobjective optimization of moment of Inertia and volume for a five layer plate

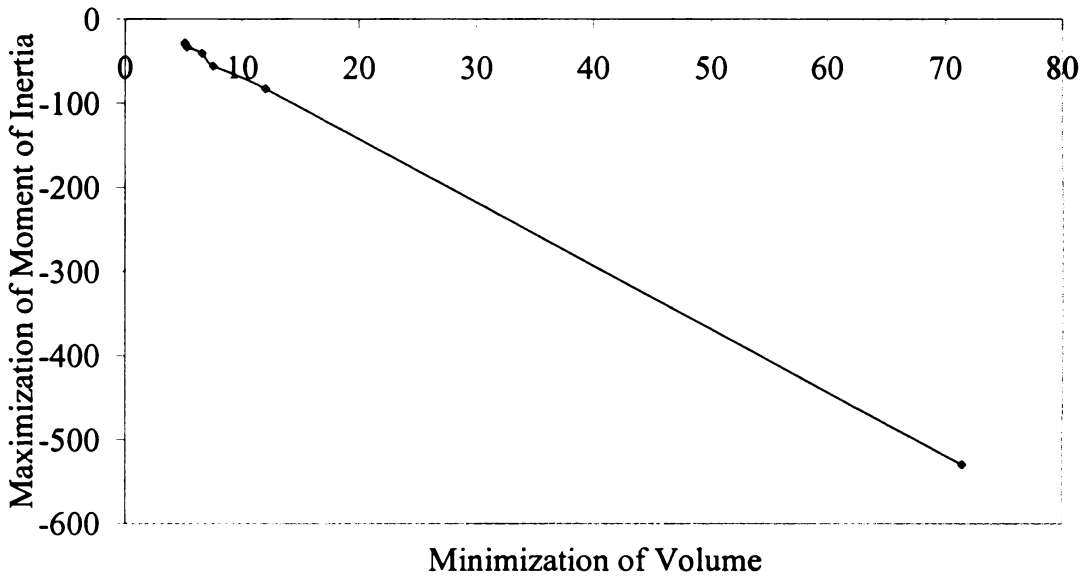


Figure 3.19 Pareto optimum set for multiobjective optimization of moment of Inertia and volume for a seven layer plate

3.2.8 Limitations of sizing optimization using gradient based methods

When using gradient-based optimization techniques convergence depends on a good initial guess or starting point for the problem. Thus, finding a global optimum largely depend upon the initial guess or the optimization routine might just return a local optimum value. Even if a good starting point has been provided, gradient optimization methods can't guarantee global optimum as the search often gets "stuck" in a local optima. These problems were observed in the gradient-based optimization approach used for the sizing optimization problem. This led to reformulation of the optimization problem as a material layout optimization problem using genetic algorithms as described in the next section.

3.3 Layout optimization of beam/panel cross-sections using genetic algorithms

The approach presented in this section is aimed at optimizing cross-sectional properties of beam/panel cross-sections by using genetic algorithms. In this approach the entire domain was discretized in a mesh, or grid, and then pre-defined material designs, or coupons, were selected from a library. The coupons in this library had finite size features and pre-defined geometric and material properties. A linear bending-induced strain distribution was assumed across the beam/panel cross-section. Thus, for evaluation of the objective function the problem was simplified to a repeating column unit made up of coupons. The entire approach is conceptually summarized in Figure 3.20. The formulation of the layout optimization problem using genetic algorithms is outlined in the following section.

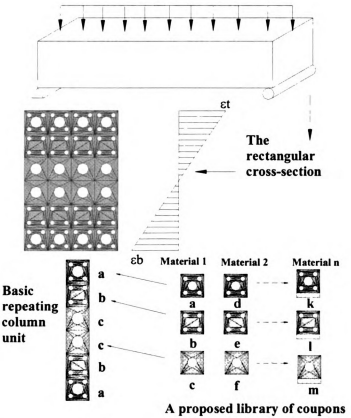


Figure 3.20 Overview of layout optimization using library of coupons with finite-size features

3.3.1 Design variables

Coupons in a pre-defined library were stored as an array of structures where each structure contained all the geometric and material properties of a coupon. A characteristic coupon with its finite-size features and geometric parameters is shown in Figure 3.21 where r_i denote the radii of different perforations or circular holes and y_i denote the distances of centroids of the perforations, or finite size features, from the base of the coupon. The library was defined based on the above nomenclature and included both non-hybrid and hybrid coupons (Figure 3.23). Various possible layouts, based on the typical coupon, were incorporated in the coupon library ranging from coupons with only a central hole to coupons with all possible holes present. For every non-symmetric coupon a corresponding mirror image was also included. The hybrid coupons were assumed to have a thin layer of comparatively stiffer material (i.e. having a modulus of elasticity five times that of base coupon material) at the top, the bottom or both top and bottom of the coupon. The moment of inertia of each coupon about its own centroidal axis was calculated assuming it to be a composite section with internal circular voids, and this value was then stored together with the other geometric and material parameters into the coupon structure array (Figure 3.22).

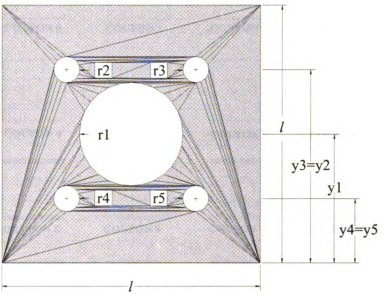


Figure 3.21 Layout of a general coupon with finite size features used for optimization

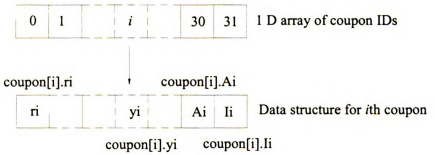


Figure 3.22 One-dimensional array of structures containing coupon data

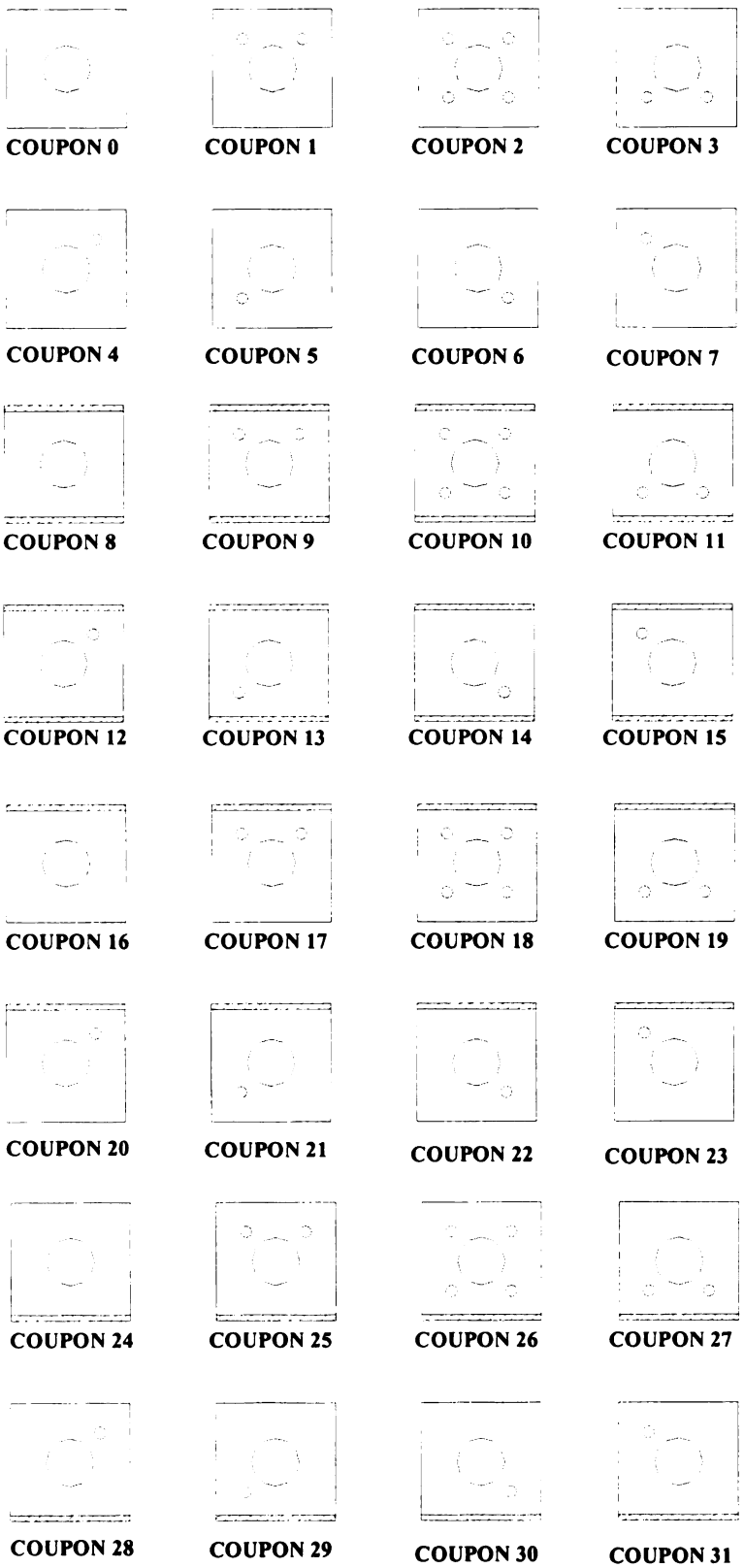


Figure 3.23 Library of hybrid and non-hybrid coupons

The identification numbers for the coupons (0 through 31 as shown in Figure 3.23) were used as the design variables. The variables were expressed as a "string of integers" or a "chromosome of integers." For evaluating the fitness of a particular layout of coupons the chromosome was first unpacked from its binary representation into integers and those integers were then used as the coupon identification numbers referencing the respective geometric and material properties. Thus, coupons IDs acted as the design variables for evaluation of the objective function.

3.3.2 Objective function

A multiobjective optimization was implemented to achieve the desired optimum layout of the beam/panel cross-section. Two sets of objective functions were considered for calculating the fitness of a given layout. The first one was a linear combination of moments of inertia along the x and y axes (quantifying sectional stiffness) and the second one was a linear combination of moments of inertia per unit volume along the x and y axes (quantifying specific sectional stiffness) as expressed in equations (3.4) and (3.5), respectively:

$$f_1 = w_1 I_x + w_2 I_y \tag{3.4}$$

$$f_2 = \left(\frac{w_1 I_x + w_2 I_y}{V} \right) \tag{3.5}$$

where for each case w_1, w_2 = weighting coefficients and $w_1 + w_2 = 1$.

Encoding and decoding of the design variables and objective function is described schematically in Figure 3.24.

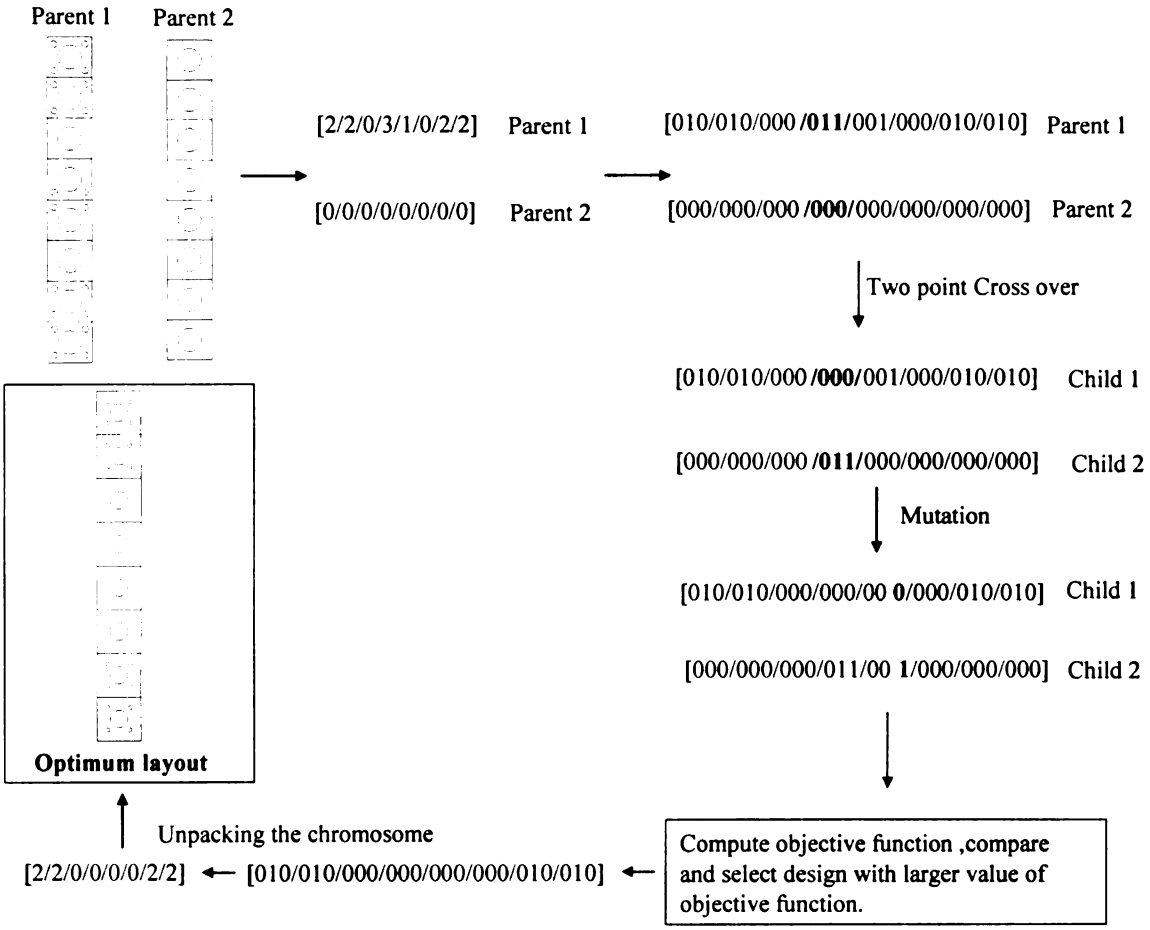


Figure 3.24 Determination of optimum layout for a given generation in GALOPPS

The moment of inertia of hybrid coupons was calculated by considering the entire coupon to be a composite section. The moment of inertia and volume calculations were thus based on a transformed section to the base coupon elastic modulus. Thus, the width of the top and bottom stiffer layers was increased by a factor of their modular ratio n_2 and n_3, respectively, as shown in Figure 3.25. Consequently, in the optimization of specific stiffness the use of hybrid coupons with stiffer material not only resulted in a higher stiffness value but it also led to higher coupon volume, thus penalizing the use of a stiffer (and most likely more expensive) material in hybrid coupons.

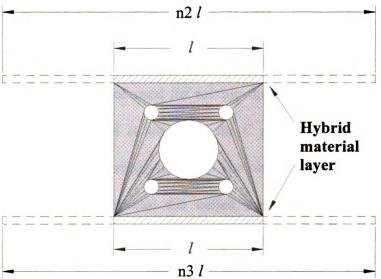


Figure 3.25 Transformed section dimensions of the hybrid coupon

In order to check the robustness of the proposed approach and to increase the size of design domain a square cross-section was also considered. The objective functions used were same as those outlined in Section 3.2.2 (Equations 3.4 and 3.5). However, in this case the design space was represented as a two-dimensional array instead of a one-dimensional array as shown in Figure 3.26.

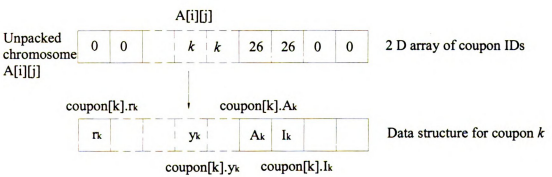


Figure 3.26 Two-dimensional array of structures containing coupon data

3.3.3 Implementation of layout optimization in GALOPPS

The genetic algorithm package GALOPPS ("Genetic Algorithm Optimized for Portability and Parallelism" System) [11] was used for solving the cross-sectional material layout optimization problem. GALOPPS is written in the C programming language and was compiled and linked using visualstudio.net [20]. The objective function to be optimized in GALOPPS is written in a special file called the *application file*. A standard application file template is available in this package and the user can define his/her own objective function within it. Input parameters can be provided through the command line or through an input file. Standard input file templates are also available with the package. Similarly, output can be obtained in a file and the desired output details can be specified. The basic organization of GALOPPS is outlined in Figure 3.27.

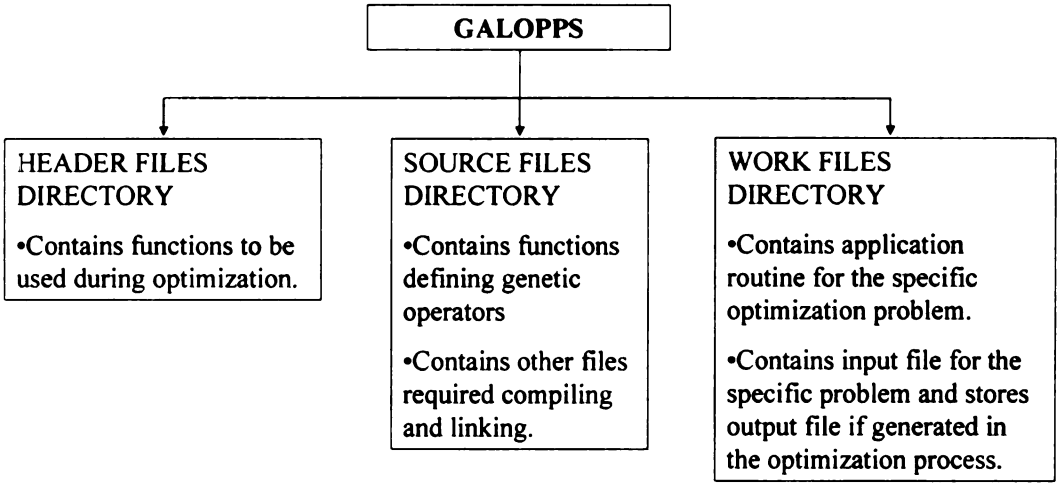


Figure 3.27 Organization of GALOPPS

As mentioned earlier, visualstudio.net was used to compile and link the above mentioned routines from GALOPPS. Project properties contain the information about input and output files and their location, e.g. their residing directory. Besides the application routine

a number of other source files are required to be added in the source files directory of the project. The source files required for any visualstudio project for GALOPPS are [8]:

checkhdr.c	checkrd.c	checkwt.c	ffscanf.c
filestat.c	generate.c	memory.c	random.c
report.c	statisti.c	utility.c	user_in.c

Finally, an application file containing the objective function and appropriate source files to specify the selection method, the crossover operator, the mutation operator, the inversion operator and the files named *mainone.c* and *startup.c* for a single population operator, need to be added to the source file directory of a project. After all these required files are added to the project, the application file is compiled and run to generate the desired output containing the solution.

3.3.3.1 Parameters used in GALOPPS for the genetic algorithm

A single population size of 200 was considered for all cases of the layout optimization problem. The number of maximum generations in a single run was selected as the termination condition. All of the optimization cases were solved as unconstrained optimization problems. A two-point cross-over was used as the recombination operator and bit-mutation as the type of mutation operator. For parent selection a stochastic universal sampling method was used [11] [10].

3.3.4 Multiobjective optimization of sectional stiffness for a square beam

Multiobjective optimization of $f = w_1 I_x + w_2 I_y$ was carried out using the library of coupons described in Figure 3.23. I_x is the moment of inertia of the cross-section about

the x-axis and I_y is the moment of inertia of the cross-section about y-axis. Three sets of values were used for the weighting coefficients w_1 and w_2 as shown in Table 3.10. The results are also summarized in Table 3.10. As expected, the results show that for maximization of sectional stiffness about the x-axis the coupons with hybrid material layer on the top and bottom were selected and placed such that the stiffer material was placed away from the section neutral axis and thus lead to a higher stiffness value (Figures 3.28 through 3.30). As there were no coupons in the library having stiffer layers on the top and bottom layers about y-axis the layout obtained for all three cases was same as shown in Table 3.10.

Table 3.10 Results for multiobjective optimization of sectional stiffness

w_1	w_2	CHROMOSOME	FITNESS VALUE (mm^4)
1	0	[8 8 8 8 24 24 24 24 16 16 16 16 8 8 8 8]	$8.5x\ 10^{10}$
0.5	0.5	[8 8 8 8 24 24 24 24 16 16 16 16 8 8 8 8]	$9.2x\ 10^{10}$
0	1	[8 8 8 8 24 24 24 24 16 16 16 16 8 8 8 8]	$9.8x\ 10^{10}$

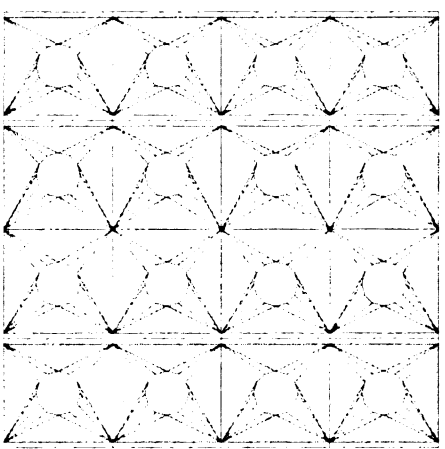


Figure 3.28 Layout for maximization of $f = I_x$

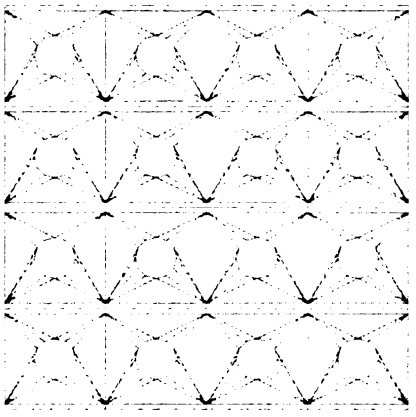


Figure 3.29 Layout for maximization of $f = \left(\frac{w_1 I_x + w_2 I_y}{2} \right)$

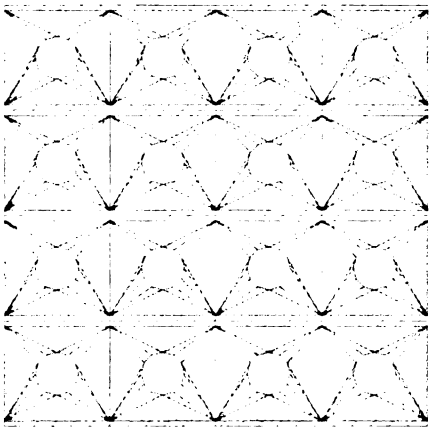


Figure 3.30 Layout for maximization of $f = I_y$

3.3.5 Multiobjective optimization of specific sectional stiffness for a square beam

Multiobjective optimization of $f = \left(\frac{w_1 I_x + w_2 I_y}{V} \right)$ was carried out using the library of coupons described in Figure 3.23 and the results are summarized in Table 3.11. Again I_x

is the moment of inertia of the cross-section about x-axis, I_y is the moment of inertia of the cross-section about y-axis and V is the volume of the component. The results for maximization of specific sectional stiffness demonstrated that coupons with hybrid material layer on top and bottom were selected to place more material away from the neutral axis. Conversely, coupons with more perforations, or less material, were placed closer to the neutral axis of the section to reduce the overall weight of the section, thus achieving a compromise solution for maximum stiffness per unit weight. For maximization of specific sectional stiffness about the *x*-axis, coupons with stiffer top-only and bottom-only layers were placed on the top and bottom of the section respectively. On the other hand, "plain" coupons were placed near the neutral axis leading to a maximum stiffness per unit volume (Figure 3.31). With weighting coefficient values of 0.5, coupons with more material were placed along the periphery of the cross-section while coupons with less material were placed close to the neutral axis to get maximum effective stiffness (Figure 3.32). In the case of maximization of specific sectional stiffness about the *y*-axis all of the selected coupons had top and bottom layers of stiff material as there were no coupons in the library with both top and bottom stiff layers about *y*-axis (Figure 3.33). In this case also coupons with maximum perforations were placed about *y*-axis to create a region with less material. Thus, the presence of all hybrid coupons in the maximization of effective stiffness about *y*-axis led to higher stiffness value, while the presence of coupons with maximum perforations towards the neutral axis led to minimization of volume. This arrangement thus led to a compromise solution for maximum specific stiffness about the y-axis.

Table 3.11 Results for multiobjective optimization of specific sectional stiffness

w_1	w_2	CHROMOSOME	FITNESS VALUE (mm)
1	0	[24 24 24 24 2 2 2 2 2 2 2 2 16 16 16 16]	3.6×10^3
0.5	0.5	[8 8 8 8 8 26 26 8 8 18 18 8 8 8 8 8]	3.7×10^3
0	1	[8 10 10 8 8 10 10 8 8 10 10 8 8 10 10 8]	3.9×10^3

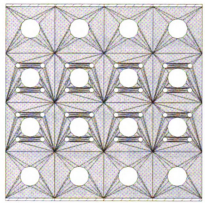


Figure 3.31 Layout for maximization of $f = \left(\frac{I_x}{V}\right)$

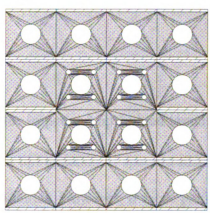


Figure 3.32 Layout for maximization of $f = \left(\frac{I_x + I_y}{2V}\right)$

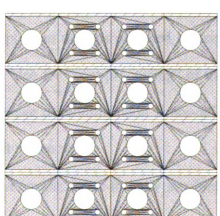


Figure 3.33 Layout for maximization of $f = \left(\frac{I_y}{V}\right)$

3.3.6 Limitations of layout optimization using genetic algorithms

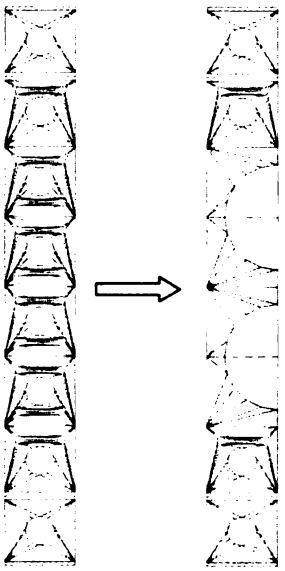
There were several limitations in approach just presented using a standard genetic algorithm. The approach works only for one constant scale of coupons and it is difficult to incorporate multiple scales of coupons in the problem. In other words, according to the present approach beam/panel cross-section can be discretized only into coupons of the same size and there is no option for different sized coupons. Also, as the domain of the problem increases sometimes there are chances of loss of diversity of solutions in the optimization process and good chromosomes might be lost leading to a solution that is not the global optimum.

3.4 A coalescence approach for improved optimized layout designs

In general, it was observed that optimization of specific stiffness resulted in the selection of coupons having more perforations. However, if more perforations could be associated with greater difficulty in manufacturing then a small improvement in the coupon stiffness might not be advantageous from a manufacturing point of view. Coalescence is an idea to obtain an improvement of optimum designs obtained by merging neighboring coupons that qualify to do so. It is an idea for "post-processing" of the optimum layout obtained. Coupons qualify for coalescence depending upon a pre-defined heuristics of merging. This idea can be implemented in two ways:

1. *One-dimensional coalescence*: Since initially a column of coupons is considered to be repeating itself throughout the panel cross-section, as only one-way bending is considered, two adjacent coupons within the repeating column units could be

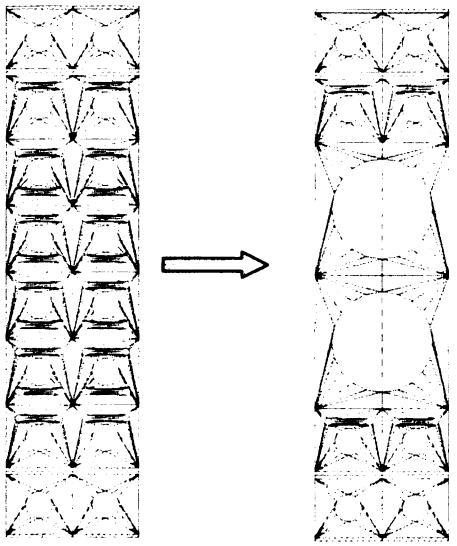
compared for coalescence and the scheme can be then extended to the remaining cross-section (Figure 3.34).



Before coalescence After coalescence

Figure 3.34 Coalescence in one dimension

2. *Two-dimensional coalescence*: Instead of extending the idea of a repeating column unit throughout the cross-section, coupons from two adjacent columns can be compared together such that if four adjacent coupons qualify for merging then they can be coalesced together (Figure 3.35).



Before coalescence After coalescence

Figure 3.35 Coalescence in two dimensions

3.4.1 Steps for coalescence approach

For the coalescence approach new coupons were added to the library of existing coupons. The new coupons consisted of a quarter of a circular hole as shown in Figure 3.36. Heuristics of coalescence was decided by considering coalescence of four neighboring coupons with ID=0 as shown in Figure 3.37. If four neighboring coupons had an ID equal to 0 they were coalesced and new layout was represented by replacing all four coupons by new coupons (coupons 32 to 35) as shown in Figure 3.37. If all four neighboring coupons did not have ID equal to 0 they did not qualify for coalescence and the next layout was checked. An overview of the two-dimensional coalescence algorithm is given in the flow chart shown in Figure 3.36.

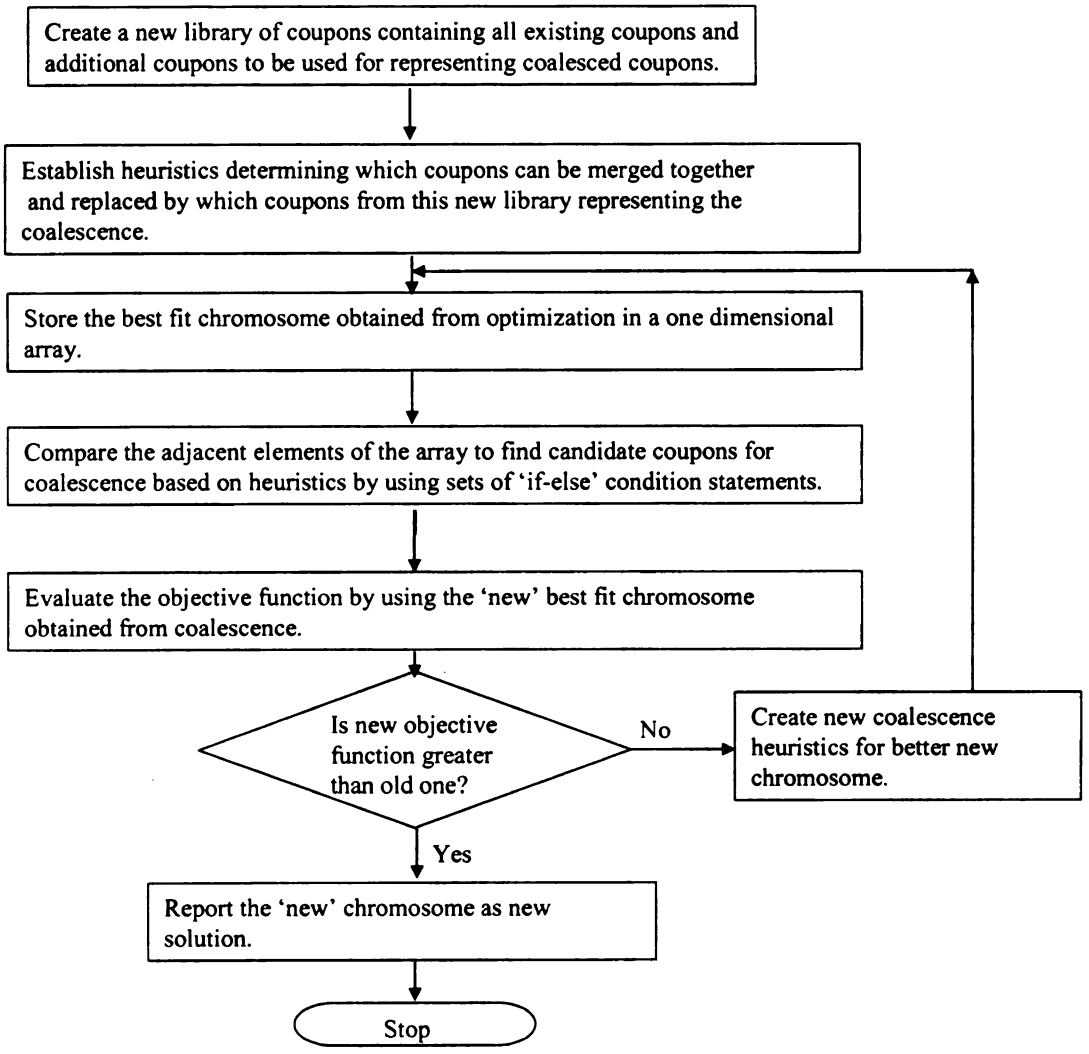


Figure 3.36 Overview of coalescence algorithm for one dimensional coalescence

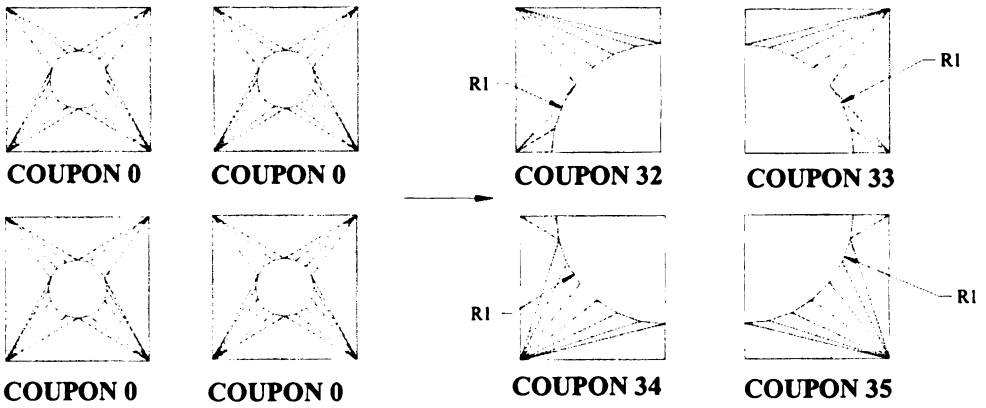


Figure 3.37 Coalescence scheme used in one and two dimensions coalescence

3.4.2 Two-dimensional coalescence for maximum effective stiffness

The coalescence approach when applied in one dimension to obtain optimum layouts for maximization of stiffness and effective stiffness didn't result in any improvement in the fitness value. However, better fitness values did result when applied in two dimensions (Figure 3.38) for maximization of stiffness as shown in Table 3.12.

Table 3.12 Results for coalescence approach in two dimensions for maximization of stiffness

Initial chromosome	Initial fitness value (mm^4)	Chromosome after coalescence	Fitness value after Coalescence (mm^4)
[8, 0, 0, 8 8, 0, 0, 8]	4.20×10^{10}	[8, 40, 41, 8 8, 42, 43, 8]	4.22×10^{10}

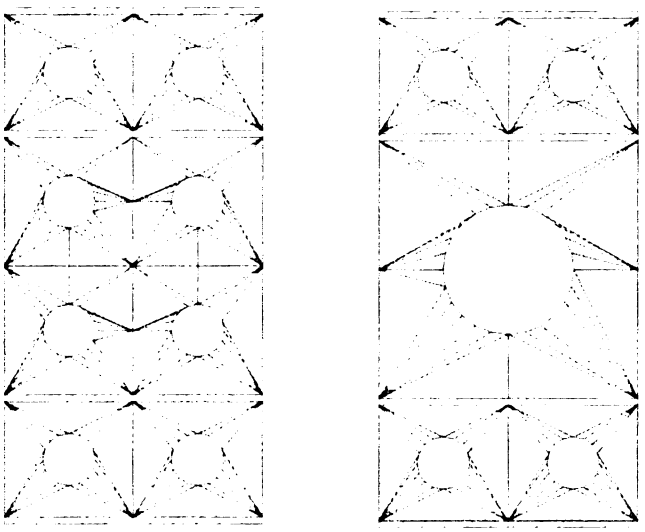


Figure 3.38 Layout for two-dimensional coalescence for maximization of stiffness

3.5 Integrated coalescence approach for Baldwinian inheritance

To solve the problem of loss of diversity among solutions with increase in the complexity of the optimization problem, and hence size of solution space, the idea of using knowledge based evolution was introduced. This consisted of utilizing knowledge gained from the coalescence of optimum layouts to influence the evolution process to search for layouts competing with improved fitness values obtained from coalesced designs. The idea was to explore the existence of designs with higher fitness values and passively influence the genetic algorithm to look for better designs that matched the new fitness value obtained through coalescence [6][14]. Figure 3.39 shows an integrated approach incorporating Baldwinian inheritance (a passive mechanism enhancing genetic diversity and survival chances of individuals through their learning capabilities) within the existing genetic algorithm module. The results obtained for layout optimization using the

integrated coalescence approach for stiffness and stiffness per unit volume are summarized in Section 3.6 next.

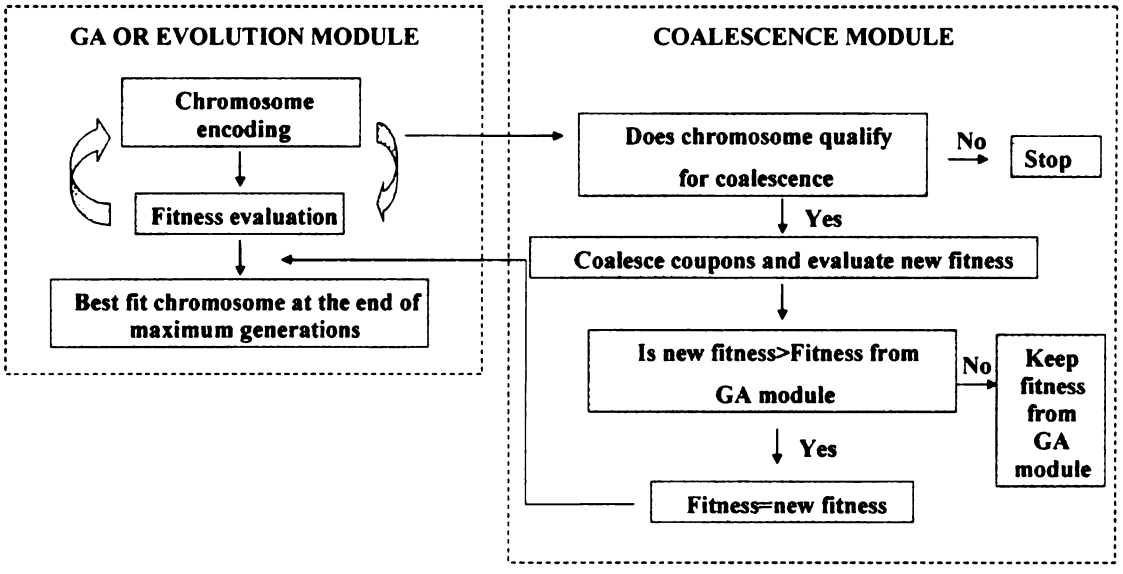


Figure 3.39 Integrated coalescence approach for Baldwinian inheritance

3.6 Multiobjective optimization using Baldwinian approach

The results for optimization of sectional stiffness using Baldwinian inheritance showed no change from the prior optimization results (Darwinian approach) without Baldwinian inheritance (Table 3.13). It was due to the fact that the problem was not very complex and thus the solution space was small. Consequently, a global optimum could be achieved even without Baldwinian inheritance. In the optimization of *specific* sectional stiffness, however, due to increase in the complexity of the optimization problem in the absence of Baldwinian inheritance the original optimization process could not converge to the global optimum. When external knowledge was used to force the genetic algorithm to look for better designs it resulted in the exploration of new designs (Table 3.14) with

higher fitness than before. The results for maximization of sectional stiffness and specific sectional stiffness are presented in Figure 3.40 through Figure 3.43.

Table 3.13 Results for multiobjective optimization of sectional stiffness using Baldwinian Approach

w_1	w_2	Chromosome from Darwinian approach	Fitness from Darwinian approach (mm^4)	Chromosome from Baldwinian approach	Fitness from Baldwinian approach (mm^4)
1	0	[8 8 24 24 16 16 8 8]	4.3×10^{10}	[8 8 24 24 16 16 8 8]	4.3×10^{10}
0	1	[8 8 8 8 8 8 8 8]	1.7×10^{11}	[8 8 8 8 8 8 8 8]	1.7×10^{11}

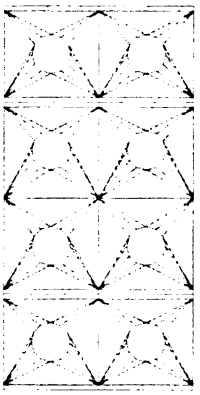


Figure 3.40 Layout for maximization of $f = I_x$ using Darwinian and Baldwinian approaches

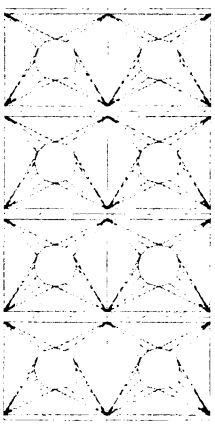
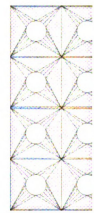


Figure 3.41 Layout for maximization of $f = I_y$ using Darwinian and Baldwinian approaches

Table 3.14 Results for multiobjective optimization of specific sectional stiffness using Baldwinian Approach

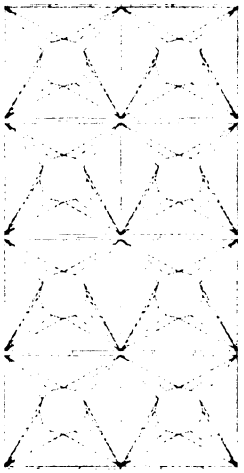
w_1	w_2	Chromosome from Darwinian approach	Fitness from GA (mm)	Chromosome from Baldwinian approach	Fitness from integrated coalescence (mm)
1	0	[24 24 2 2 2 2 16 16]	3.6×10^3	[24 24 8 8 8 8 16 16]	3.7×10^3
0	1	[8 8 8 8 8 8 8 8]	1.3×10^3	[10 10 10 10 10 10 10 10]	1.4×10^3

(a)



(b)

Figure 3.42 Layout for maximization of $f = \left(\frac{I_x}{V}\right)$ obtained from (a) Darwinian and (b) Baldwinian approaches respectively



(a) (b)

Figure 3.43 Layout for maximization of $f = \left(\frac{I_y}{V}\right)$ obtained from (a) Darwinian and (b) Baldwinian approaches respectively

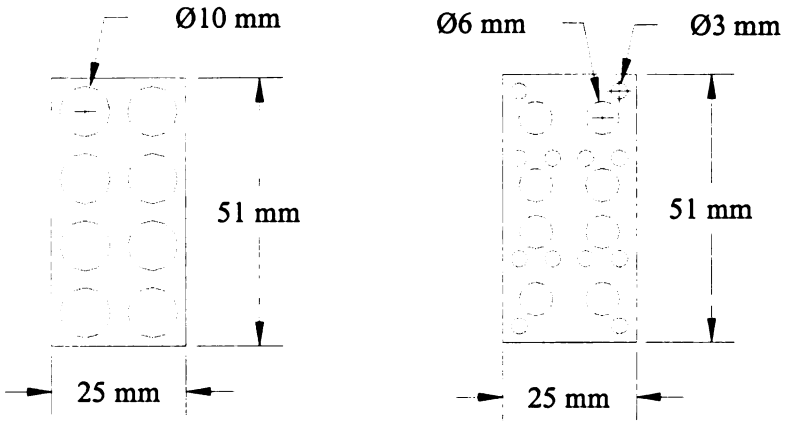
4. EXPERIMENTAL EVALUATION

4.1 General

The results obtained by using finite size features for optimization of beam/panel cross-sections enabled designs with optimum material distribution that could potentially be manufactured at laboratory-scale using biocomposite materials. To evaluate the results of the computational simulation an experimental study was conducted that was aimed at comparing the flexural performance of hybrid and hierarchical designs with respect to a base cellular design. The experimental program comprised of manufacturing and testing beams with cross-sectional layouts based on optimization studies. The structural layouts investigated for evaluation of the optimization approach are described in the following sections along with the manufacturing and testing details. The chapter concludes with presentation of the results and inferences of the experimental evaluation program.

4.2 Structural layouts

Two types of structural layouts were investigated in the experimental evaluation program namely a base cellular design and an optimized hierarchical cellular design. The cellular base design consisted of equal-sized cells repeated at equal distance, thus making the cells periodic. The optimized design consisted of two different cell sizes distributed in the cross-section in a manner similar to the analytical solutions obtained with the discrete layout optimization approach. The two designs are schematically shown in Figure 4.1(a) and (b).



(a) Base Design (b) Optimized Design

Figure 4.1 Cross-sectional layouts of base and optimal design used in experimental evaluation

Mechanical properties of cellular structures are greatly determined by their relative density ρ_{rel}, i.e., the density of the cellular material divided by the density of the solid from which the cells are made of ($\rho_{rel} = \rho^*/\rho_s$). The relative density of the base design was 0.53 while relative density of the optimum design was 0.7.

4.3 Materials and constituents

The biocomposite system used for manufacturing the cellular and hybrid, hierarchical beams consisted of industrial hemp fibers (Flaxcraft, Inc., Cresskill, NJ) (Figure 4.3) and unsaturated polyester resin (UPE). The catalyst used was methyl ethyl ketone peroxide (MEKP). It was used in a proportion of 1% by weight of resin and cobalt napthalate (CN) was used as promoter in a proportion of 0.03% by weight of resin. The base design had a fiber weight fraction (ratio of fiber weight to total weight) of 23% while the optimal

design had a fiber weight fraction of 22%. For manufacturing purposes the beams were built with integral top and bottom mats of Hessian jute (IJIRA, Calcutta, India) as shown in Figure 4.2. Average properties of some hybrid material systems from prior experimental studies by our research group (Quagliata 2003) with biocomposite fibers are provided in Table 4.1.

Figure 4.2 Jute mat for top and bottom face sheets

Figure 4.3 Industrial hemp fibers

Table 4.1 Average properties of some material systems with natural fibers and UPE resin [28]

SAMPLE	MOE (N/mm^2)	Std. Dev. (N/mm^2)	Tensile Str. (N/mm^2)	Std. Dev. (N/mm^2)	Tensile Elong. (%)	Std. Dev. (%)
Green Hemp/UPE-25% wt.	5230	820	16.79	4.19	0.33	0.07
Woven jute/UPE-	4030	640	22	1.3	0.68	0.07
Green Hemp/UPE top & bottom jute-27% wt.	10,280	6120	28.02	4.5	0.39	0.24

4.4 Manufacturing

4.4.1 Automated manufacturing using VARTM

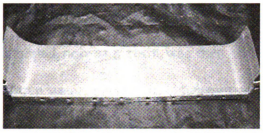
Manufacturing automation of the cellular biocomposite beams was evaluated by implementing a vacuum assisted resin transfer molding (VARTM) manufacturing setup. The method was selected as it has been found to be an effective process for manufacturing cellular structures using natural fibers [25]. VARTM is essentially a closed-mold process that can be used to manufacture fiber reinforced polymer composites

(FRP). The process consists of the placement of dry fibers inside a bagging film where resin is drawn into the sample by using a vacuum pump. The VARTM process can be effectively used for manufacturing of large FRP composite parts with high mechanical properties and complex shapes. Thus, it is ideal for making large structural components.

4.4.2 Material placement

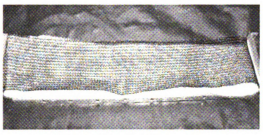
A mold with removable face plates was used for manufacturing all of the beams. The dimensions of the mold were 813 mm x 54 mm x 38 mm. The same mold was used for making both beams by using removable face plates for different cross-sectional cellular designs. The cellular sandwich base design was manufactured by using equal-size circular periodic holes in the longitudinal direction of the cross-section of the beam. The beam was provided with top and bottom face sheets of jute. The optimal design also consisted of circular cells in the longitudinal direction of the beam, arranged in varying sizes through the depth of beam. The dimensions of the test units for the above mentioned layouts and cell-sizes were decided considering the available circular metal rods which were available in standard sizes: in 3 mm (1/8") intervals. The effective cell sizes were then defined by the thickness of rubber tubes, also with diameters available in 3 mm (1/8") intervals, which were used to avoid bonding between the polymer and the metal rods. To manufacture the three designs at first the bottom of the mold was lined with a non-porous Teflon ply (Figure 4.4(a)) in order to release the beam from the mold after curing. A breather ply was placed on top of the Teflon ply to absorb excess resin from the sample (Figure 4.4(b)). A resin transfer media (high density polyethylene, green house shade cloth) was placed on top of breather ply to obtain a uniform flow of the resin through the sample (Figure 4.4(c)). A porous bleeder ply (polyester cloth) was used on

top of the resin transfer media to achieve a smooth surface after curing and absorb excess resin from the sample (Figure 4.4(d)). Finally a layer of porous peel ply was used to on top of the bleeder ply (Figure 4.4(e)) to allow excess resin to be squeezed out of the sample. After setting all of the above mentioned layers, a single layer of jute mat was placed in the mold such that it would wrap around the entire beam (Figure 4.4(f)). The bottom layer of short industrial hemp fibers was then placed on top of the jute mat and then the face plates were attached to the mold. After placing the face plates the circular rods (covered with rubber tubing) were inserted along the longitudinal beam cross-section. The fibers were then tightly placed in between the rod spacing. After placing the top-most layer of hemp fibers the top of the sample was wrapped completely with the jute mat and all the plies used in the bottom of the mold were mirrored on the top as well. The entire set up was then placed inside a vacuum bag and sealed with sealant tape around the mold. The ends of the rods and other sharp corners of the mold were covered with a layer of resin transfer media and breather cloth to prevent them from puncturing the vacuum bag when under pressure. Resin was infused using a vacuum pump through VARTM method as described in Section 4.4.3.

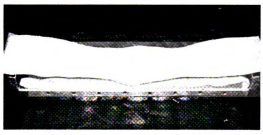


(a) Non-porous Teflon ply

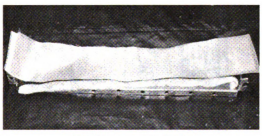
(b) Breather ply



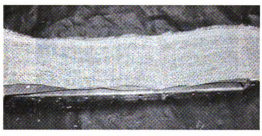
(c) Resin transfer media



(d) Porous bleeder ply



(e) Porous peel ply



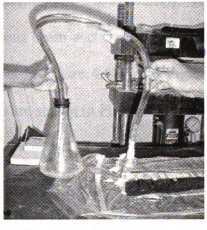
(f) Jute mat layer

Figure 4.4 Preparation of mold for material placement

4.4.3 VARTM: Setup and resin infusion process

The VARTM setup was prepared on a movable cart with a single vacuum pump on the bottom of the cart and a mold containing the sample on the top of the cart. Before the sample was sealed completely inside the vacuum bag using sealant tape, a vacuum port was placed on top of the sample and secured to the bag using additional sealant tape around the port. The vacuum port was covered with a small piece of resin transfer media and breather cloth to prevent the vacuum bag from blocking the port. The resin port was

connected to the other end of the sample, also from the top, and sealed again using tape. The cover of the sealant tape was then removed and the entire setup was then sealed. The unsaturated polyester resin was then added to the resin reservoir (Figure 4.5(a)) and a resin trap was connected to a pressure gauge (Figure 4.5(b)). A two-way inlet valve was used to control the flow of resin into the sample. The sample was compressed under vacuum (635 mm of mercury) before the resin was injected into the sample. Once the sample was compressed under pressure the inlet valve was opened and resin was drawn into the sample by the vacuum pressure. Once the sample was completely impregnated with the resin the valve was closed to let the sample absorb the already present resin and allow excess resin to be removed.



(a) Flow of resin from reservoir

(b) Excess resin collected in the resin trap

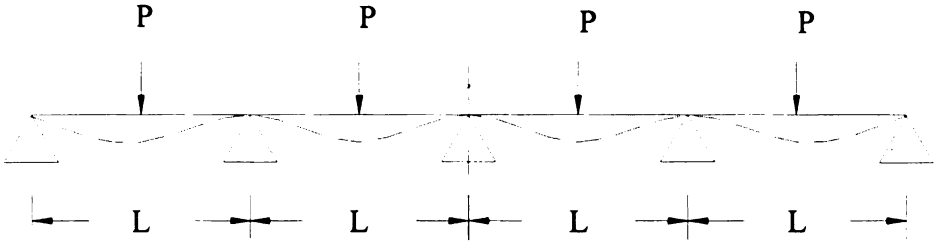
Figure 4.5 Sample with vacuum bag, resin reservoir and resin trap

The system was kept under vacuum pressure for approximately an hour to let the resin gel. After this, the vacuum pressure was removed and the ports were detached to prepare

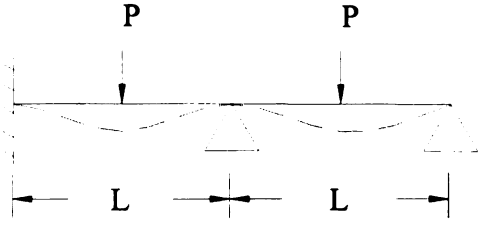
the sample for elevated temperature curing. The sample was oven cured inside the mold and with all bagging materials for 6 hours at $100^{\circ}C$. The curing temperature was kept at $100^{\circ}C$ to avoid melting the resin transfer media, which is made of high density polyethylene. Steel plates were placed on top of the sample during curing to keep the sample compressed and thus improve the density and quality of the sample.

4.5 Testing of beams

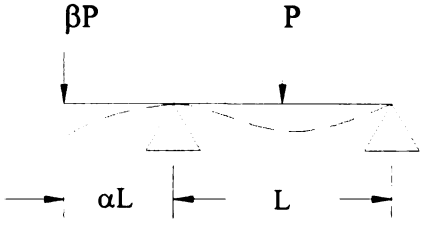
The biocomposite beams were optimized for a continuous panel type application, thus, their performance in flexure under both positive and negative moment demands was investigated. The panel system was assumed to be modeled as a continuous beam supported at equal intervals and loaded under concentrated loads at mid-spans of each panel length (see Figure 4.6(a)). To simplify the test set-up only a portion of the continuous beam was considered, after reducing the system due to symmetry conditions as shown in Figure 4.6(b) and Figure 4.6(c). The moment diagrams for the simplified systems are shown in Figure 4.7. In order to obtain the desired design moments as shown in Figure 4.7(b), the beams were tested as a simply supported beam with a cantilevered overhang as shown in Figure 4.7(c). The load levels at the mid-span and the cantilever tip were selected to match the desired design moments. The cantilever length αL (Figure 4.6(c)) was calculated as a fraction of the simply supported length L and α was equal to 0.19. The beam was loaded at the cantilever tip and at the middle of the simply supported span. The load at the cantilever tip βP (Figure 4.6(c)), was calculated as a fraction of the load at the middle of the simply supported span, P, where β was equal to 0.837.



(a) Continuous panel system with multiple supports

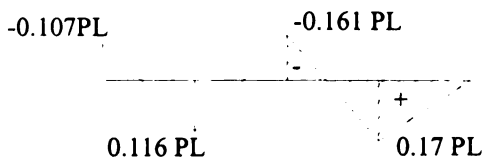


(b) Reduced two-span panel arrangement

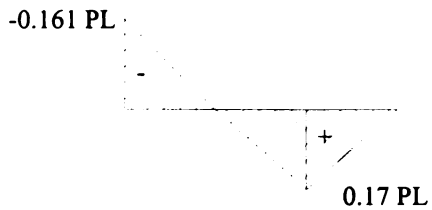


(c) Reduced two-span arrangement for test units

Figure 4.6 Simulation of a continuous panel system under symmetric loading and boundary conditions in a four-point bending test



(a) Bending moment diagram of symmetry-reduced continuous panel system



(b) Bending moment diagram for test unit

Figure 4.7 Bending moment diagram to be achieved for continuous beam test units

The continuous flexural test setup (see Figure 4.8 and Figure 4.9) was mounted on an MTS loading frame. Load was applied to the beams using the test frame loading ram and a loading fixture that applied two point loads at 457 mm apart. The total span was 787 mm. The samples were supported using a fixture from a steel I-beam and adjustable roller supports. All test units were loaded monotonically up to failure in displacement control at a deformation rate of 0.01mm/sec for the base design and a rate of 0.025mm/sec for the optimal design.

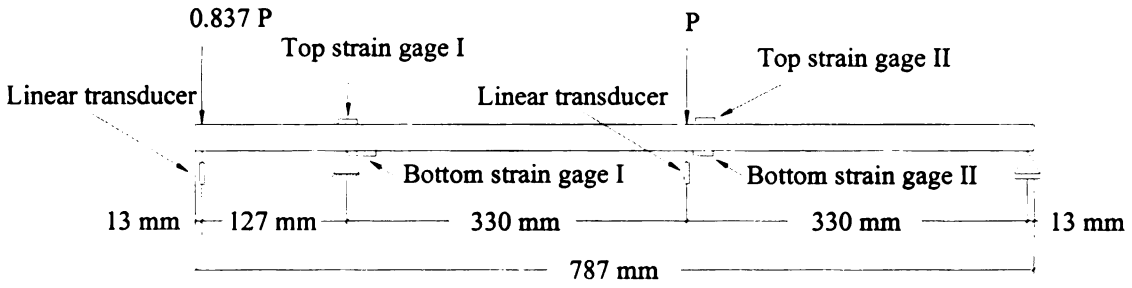


Figure 4.8 Cellular beam four-point bending test

For the four-point bending test the deflections of the beams at the center span and at the cantilever tip were measured using externally mounted linear transducers. Strains at the top tensile and compression fibers were measured using strain gages at the sections with maximum moments (mid-span under the load and at the intermediate support) as shown in Figure 4.8 and Figure 4.9. The strain gage readings, external displacements, and applied loads were simultaneously recorded with a digital data acquisition system. The results obtained from the four-point bending test and comparison of the flexural performance of the base and optimized designs are discussed in the next section.

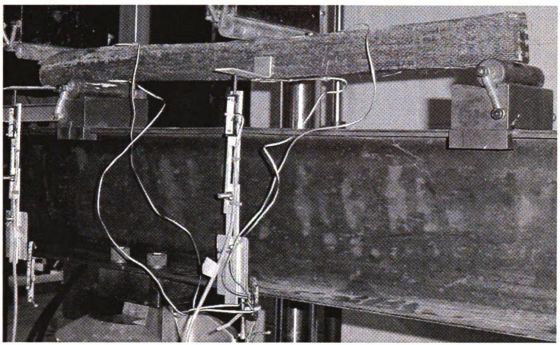


Figure 4.9 Overview of cellular beam four-point bending test

4.6 Observed Behavior and Test Results

4.6.1 Four-point bending

The load-deflection response from the four point bending test of both the base and optimal beams is shown in Figure 4.13 and Figure 4.14 respectively. The beams failed under combined flexural and shear demands near the intermediate support (near the cantilever end). This failure was most probably initiated by tensile cracking at the top of the beam as shown in Figure 4.10. Due to manufacturing limitations, namely poor resin injection, the end of the beam did not have adequate strength. Thus, significant crushing of the cantilever end was observed as seen in Figure 4.11.

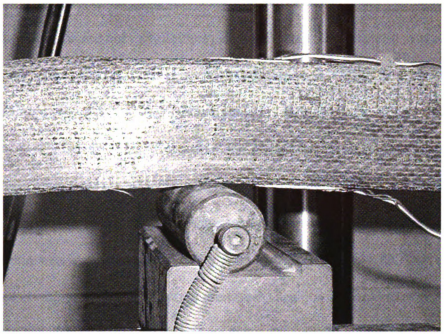


Figure 4.10 Combined flexural and shear failure observed in four-point bending test

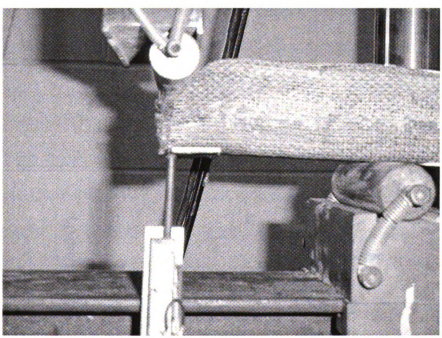


Figure 4.11 Crushing of cantilever tip in four-point bending test

Only certain void sizes could be attained for both base and optimal beams as the metal rods used for the cells are only available in standard sizes. Therefore, it was not possible to achieve the same relative density in both cellular base design and optimal design. Hence, to compare the performance of both designs, their response was normalized with respect to their relative densities. The actual and normalized load displacement responses and load strain responses are shown in Figures 4.13 through 4.18. In order to determine the initial stiffness from the load displacement response of the two beams the linear response region was identified qualitatively as indicated in Figure 4.12. Two points were then selected on the tangent drawn along the linear variation where one of the points was the origin. Initial stiffness was calculated as force per unit displacement using Equation 4.1 where P and δ were the load and displacement at the middle of the simply supported span as shown schematically in Figure 4.12

$$K = \frac{P}{\delta} \tag{4.1}$$

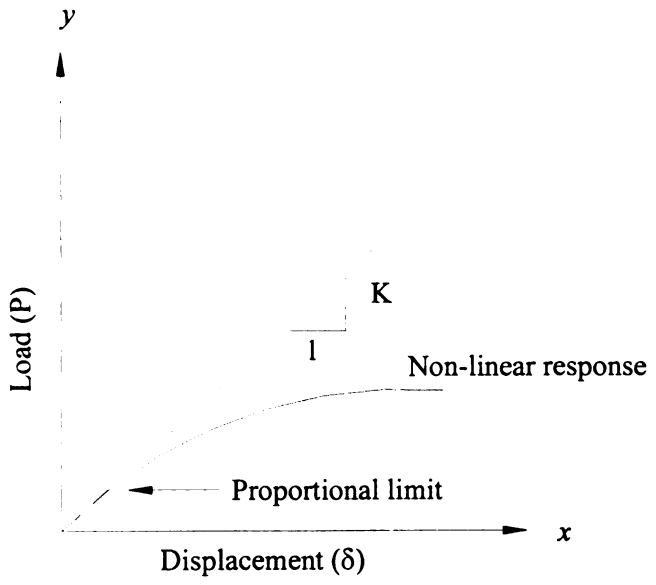


Figure 4.12 Measurement of initial stiffness in a non-linear response

The measured beam stiffness values were related to the beam sectional stiffness by assuming elastic material response and relating beam displacements to the flexural stiffness (*EI*) through simple beam theory. Theoretical values for *EI* were obtained by calculating the moment of inertia for the geometric layout of the beams and taking a value of *E* (5230 N/mm^2) from previous work (see Table 4.1) [28]. The measured and theoretical values are summarized in Table 4.2, where it can be seen that the optimal design beam was about 1.4 times stiffer than the base design as calculated from the normalized load-displacement response (Figure 4.14). The normalized flexural stiffness value for the base design was 21% higher than the theoretical value, while the normalized flexural stiffness value for the optimal design was 25% higher than the theoretical value. The difference in the observed and theoretical values can be attributed to various reasons such as manufacturing errors, like non-circular cells and non-uniform surface leading to non-uniform cross-section, or wrong value of modulus for the elasticity which was taken from previous work. The observed load-strain responses at the maximum negative moment section for the two beams are shown in Figure 4.17 and observed values are summarized in Table 4.3. The normalized failure strain and load response is shown in Figure 4.18. As mentioned earlier in this section, a combined flexure-shear failure was observed near the intermediate support (near cantilever end) in both the beams. Failure was characterized by the appearance of inclined cracks which started at the top of the section and proceeded towards the bottom. A drop in the load in the load-displacement response was also observed at this instance. Also no apparent signs of damage were observed near the maximum positive moment section. Thus, it was concluded that both the beams experienced a combined flexure-shear failure near the negative moment

section. The difference in the failure strain values in the negative moment section was most likely due to malfunctioning of the strain gage in the base cellular beam which was observed from the load-strain response. Otherwise the failure strain in the base cellular beam should have been similar to that of the optimal beam.

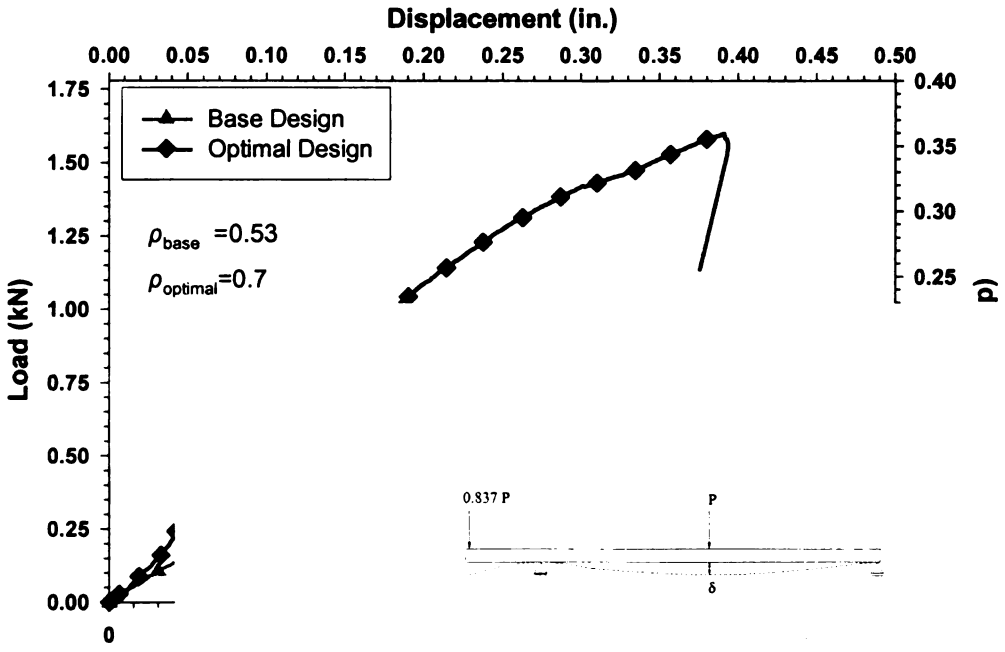


Figure 4.13 Force-displacement response at midspan in four-point bending

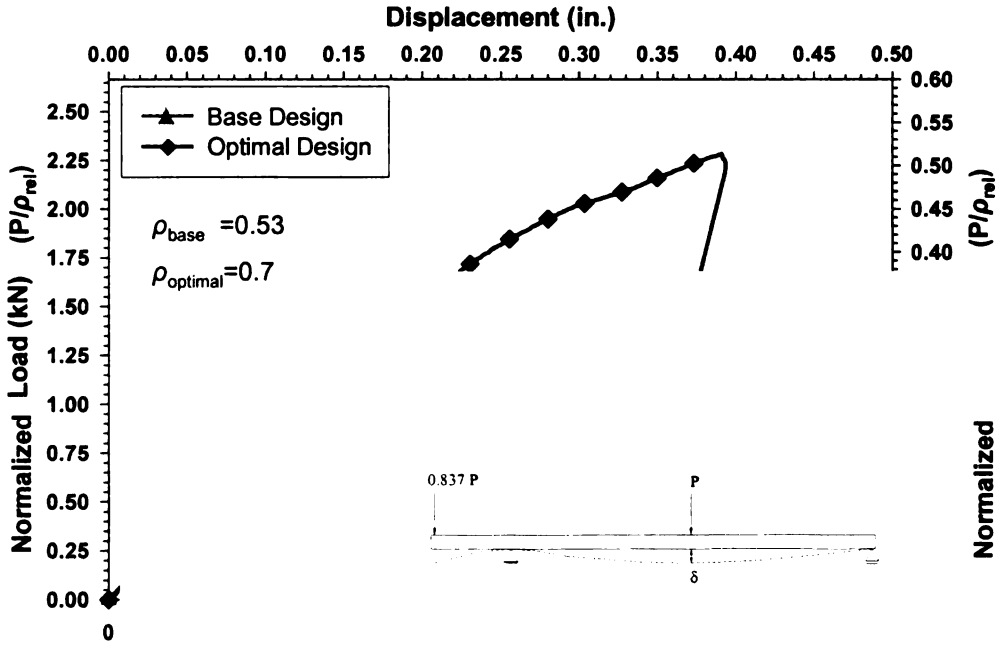


Figure 4.14 Normalized force-displacement response at midspan in four-point bending

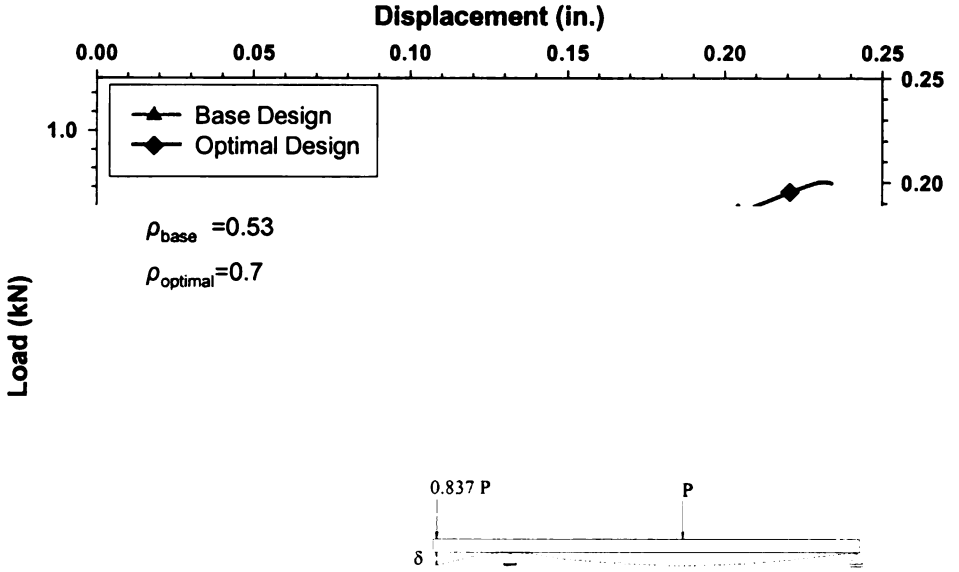


Figure 4.15 Force-displacement response at cantilever tip in four-point bending

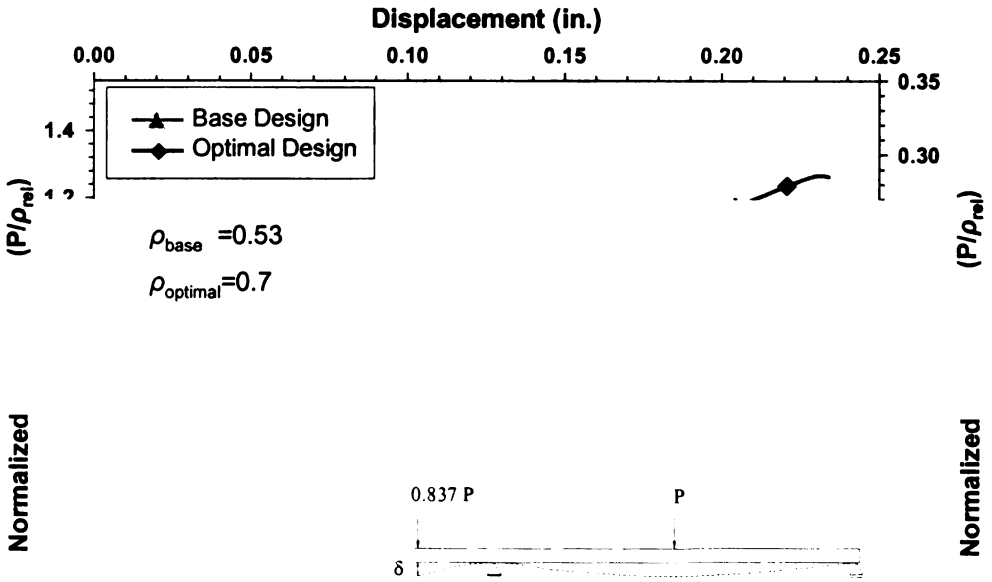


Figure 4.16 Normalized force-displacement response at cantilever tip in four-point bending

Table 4.2 Comparison of experimental and theoretical flexural stiffness from four-point bending test

Cellular design	Theoretical *EI* N-mm^4	Measured *EI* N-mm^4	Error (%)	Measured Normalized *EI* N-mm^4	Error (%)
Base	826	850	24%	1000	21%
Optimal	1130	1510	33%	1420	25%

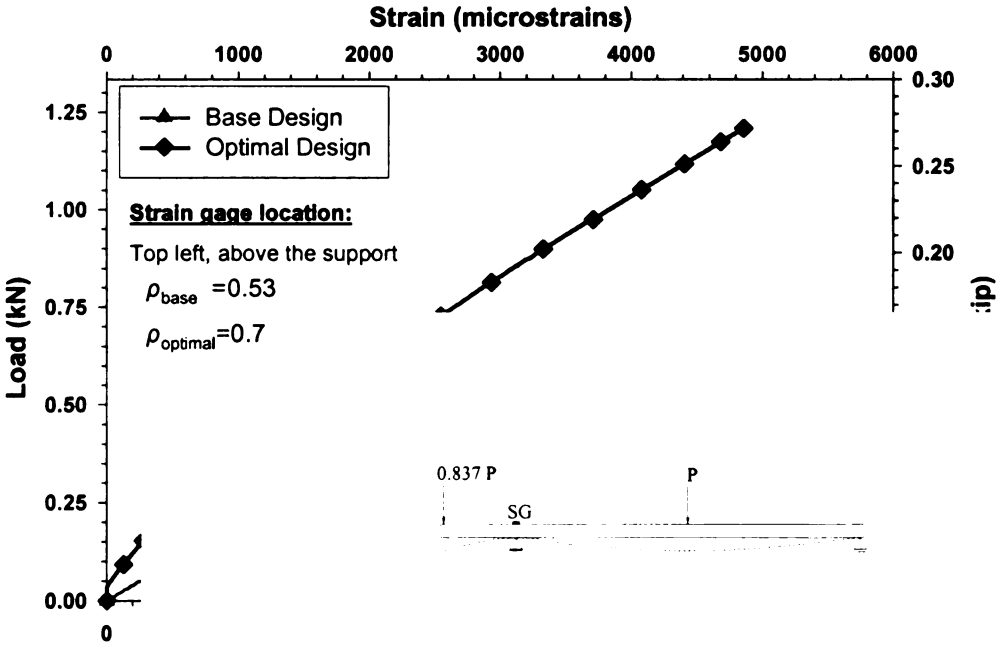


Figure 4.17 Load-strain response at extreme tension fiber on maximum negative moment section

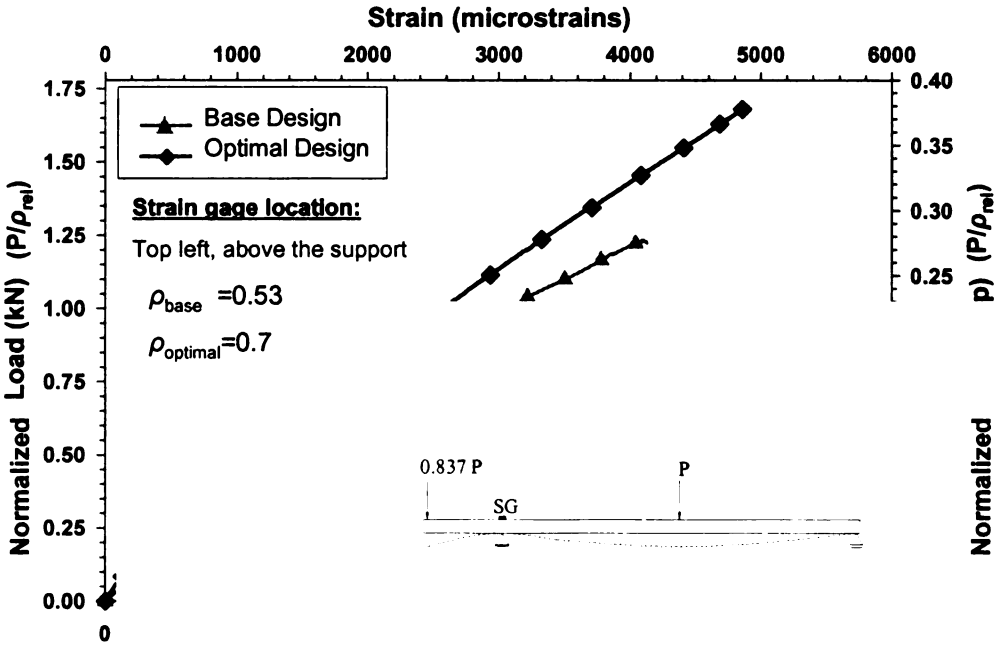


Figure 4.18 Normalized load-strain response at extreme tension fiber on maximum negative moment section

Table 4.3 Comparison of failure strain and load measured at extreme tension fiber in negative moment section

Cellular design	Failure strain (microstrains)	Measured load (N)	Normalized load (N)
Base	4100	623	1201
Optimal	4950	1245	1690

4.3.2 Three-point bending

After the end crushing of the cantilever tip started during the four-point bending test loading the beam all the way up to failure wouldn't have yielded any useful information. Thus, in order to obtain the response up to failure the simply supported span of both base and optimal beams was tested in three-point bending after removing the cantilever part. The beams had a span of 660 mm and the load was applied at the center of the span. The test set-up for three-point bending is shown in Figure 4.19. Both the beams were loaded monotonically up to failure in displacement control at a rate of 0.01 mm/sec for base design and at a rate of 0.025 mm/sec for optimized design. The deflection was measured under the centrally applied load using an externally mounted extensometer as shown in Figure 4.20.

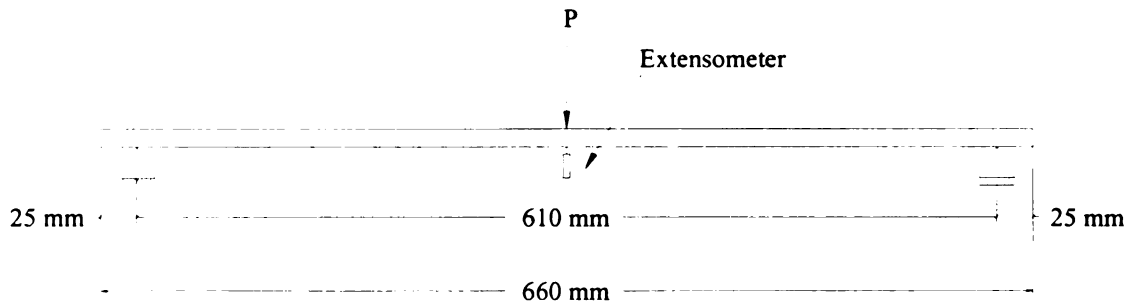


Figure 4.19 Cellular beam three-point bending test

The results of the three-point bending test of cellular beams show that both the beams, base and optimal, behaved non-linearly up to failure (Figure 4.21) after a brief initial linear response. The normalized load-displacement response (Figure 4.22) showed that

the optimal beam was about 1.12 times stiffer than the base design (Table 4.4). The beam experienced flexural failure under the central load (Figure 4.23).

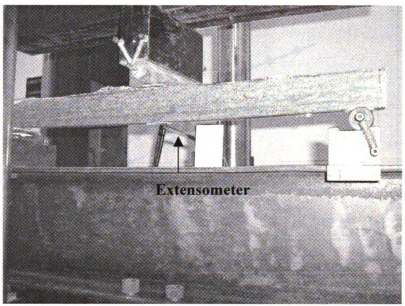


Figure 4.20 Cellular beam three-point bending test

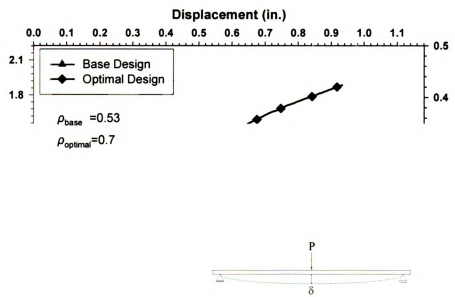


Figure 4.21 Load-displacement response at midspan for three-point bending

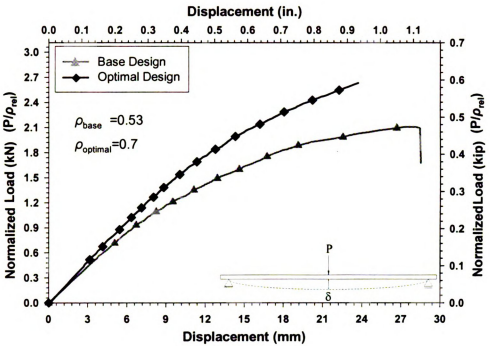


Figure 4.22 Normalized load-displacement response at midspan for three-point bending

Figure 4.23 Flexural failure in three-point bending

Table 4.4 Experimental stiffness from three-point bending test and theoretical stiffness for base and optimal designs

Cellular design	Theoretical *EI*	Measured *EI* N-mm^4	Error (%)	Measured Normalized *EI* N-mm^4	Error (%)
Base	826	353	57%	686	17%
Optimal	1130	6250	45%	7690	32%

The sectional stiffness (*EI*) values obtained from the normalized load displacement responses in four and three point bending for the base and optimal designs were not in close agreement as expected. This could be attributed to various reasons such as significant deformation in the cross-sections of the beams after the four-point bending test (due to local deformations from the test set-up) and manufacturing errors (like non-circular cells or non-uniform surface leading to non-uniform beam dimensions).

In spite of the discrepancies between the experimental and calculated values, the test results of both four-point bending and three point bending indicated that the optimal design had better flexural performance than the selected base design.

5. CONCLUSIONS

Results from the analytical studies on the sizing and discrete layout optimization approaches showed that both optimization methods are viable for optimizing material distribution in cross-sections of biocomposite panels. Both methods led to designs with finite-sized features and thus the designs were easier to manufacture as compared to most of the conventional topology optimization methods. These two methods were also able to handle multiple objectives and constraints. The discrete layout optimization method was also successful in using hybrid material in the cross-section, unlike conventional topology optimization methods. The experimental evaluation program also showed how an optimized cellular biocomposite beam had almost 1.4 times higher flexural stiffness than a base design through a four-point continuous bending test, and about 1.12 times higher flexural stiffness in a three-point bending test. The failure load of the optimized beam was 1.9 times higher than the base periodic cellular design. In particular, the findings in this study have led to the following conclusions regarding optimized hierarchical cellular biocomposite beams:

- The sizing and discrete layout optimization approaches using finite-size features led to designs that had higher stiffness and specific stiffness, were feasible to manufacture with natural fibers and could successfully handle multiple objectives, multiple constraints and hybrid material in the cross-section.
- The optimal hybrid and hierarchical cellular beam designs with finite-sized features obtained from discrete layout optimization were shown to have higher stiffness and thus demonstrated better flexural performance

than a periodic cellular beam when tested in continuous and simple bending.

Based on the findings and limitations observed in the optimization approaches presented in this thesis following recommendations are provided for future work:

- Different geometric shapes can be used as perforations in the cellular designs, besides circular perforations, using both sizing and discrete layout optimization methods.
- Other objective functions, such as strain energy should be explored using both the sizing optimization and the discrete layout optimization approaches. Non-uniform strain distribution in the cross-section should also be incorporated in the problem formulation. A finite-element module can be coupled with both optimization approaches for calculation of objective functions such as strain energy.
- An algorithm with multi-start capability needs to be used for the sizing optimization approach so that more initial starting points can be explored to enhance chances of reaching a global optimum.
- The layout optimization algorithm can be modified to incorporate different scales or coupon sizes in the original coupon library instead of relying on the coalescence approach alone. The coalescence concept can still be used to continuously merge repeating cells throughout the optimization process.
- If different scales can be successfully incorporated in the optimization process, as mentioned above, then further heuristics for coupon subdivision or splitting can also be investigated in addition to coalescence.

- More knowledge-based operators could be used in the layout optimization approach to preserve the diversity of the designs, thus enhancing the chances of finding a global optimum.

REFERENCES

1. Arora, Jasbir S., Introduction to Optimum Design, McGraw Hill International Editions, Singapore, 1989.

2. Ashby, M.F., "Materials and Shape," *Acta metal. mater*, Vol.39, No.6, pp. 1025-1039, 1991.

3. Burgueño, R., Quagliata, M.J., Mohanty, A.K., Mehta, G., Drzal, L.T., and Misra, M., "Load-Bearing Natural Fiber Composite Cellular Beams and Panels" *Composites Part A*, Vol. 35, No. 6, pp. 645-656, 2004.

4. Burgueño, R., Quagliata, M.J., Mohanty, A.K., Mehta, G., Drzal, L.T., and Misra, M., "Hierarchical Cellular Designs for Load Bearing Biocomposite Beams and Panels" *Materials Science & Engineering A*, Vol. 390, pp. 178-187, 2005.

5. Burgueño, R., Quagliata, M.J., Mohanty, A.K., Mehta, G., Drzal, L.T., and Misra, M., "Hybrid Biofiber-Based Composites for Structural Cellular Plates" *Composites Part A*, Vol. 36, No. 5, pp. 581-593, 2005.

6. Davis Lawrence, Handbook for genetic algorithms, Van Nostrand Reinhold, New York, 1991.

7. Diaz, A.R., and Chellappa S., and Bendsoe M.P., "Layout Optimization of Structures with finite-size features using multiresolution analysis," *Structural and multidisciplinary optimization*, Vol. 26 No. 1-2, p 77-91, 2004.

8. Gibson, L.J. and Ashby, M.F., Cellular Solids: Structures and Properties, 1st edition, Permagon Press, Oxford, 1998.

9. Gibson, L., The "Nature" of Engineering, Website, Massachusetts Institute of Technology, http://web.mit.edu/naturalmaterials/front.htm

10. Goldberg, D.E., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1999.

11. Goodman, E.D., The "Genetic ALgorithm Optimized for Portability and Parallelism" System (GALOPPS) Release 3.2, Michigan State University, East Lansing, MI, 1996.

12. Gunderson, S.L. and Thorp, K.E.G., "Natural Cellular Sandwich Structures for Innovative Design Concepts," Proceedings of the ASC 8th Annual Technical Conference, Cleveland, Ohio, 19-21 October 1993, 431-440.

13. Hassani, B. and Hinton, E, Homogenization and structural topology optimization-theory, software and practice, Springer, New York, 1999.

14. Hromkovic J., Algorithms for hard problems, Springer-Verlag, Berlin, 2003.

15. Huang, J.S., and Gibson L.J., "Microstructural Design of Cellular Materials: Honeycomb Beams and Plates," Acta metallurgical et Materialia, Vol. 43, No. 4, 1643-1650, 1995.

16. Jones R.M., Mechanics of Composite Materials. Taylor and Francis, 1999, Philadelphia, P.A.

17. Karbhari, V.M., and Seible, F., "Fiber Reinforced Composites-Advanced Materials for the Renewal of Civil Infrastructure," *Applied Composite Materials*, 7:95-124, 2000.

18. Lakes, R., "Materials with Structural Hierarchy," Nature, 361, 511-515, 1993.

19. The MathWorks Inc., Matlab 6.0, Natick, Massachusetts, September 2000.

20. Microsoft Corporation, Microsoft.NET Framework 1.1 (VisualStudio.NET), 2003.

21. Mishra, S., Mohanty, A.K., Drazal, L.T., Misra, M., Parija, S., Nayak, S.K., and Tripathy S.S., "Studies on Mechanical Performance of Biofiber/Glass Reinforced Polyester Hybrid Composites," *Composite Science and Technology*, 63, 1377-1385, 2003.

22. Mohanty, A.K., Misra, M. and Drazal, L.T., "Surface Modifications of Natural Fibers and Performance of the Resulting Biocomposites: An Overview," *Composites Interfaces*, 8, 313-343, 2001.

23. Mohanty, A.K., Misra, M. and Hinrichsen, G., "Biofibers, Biodegradable Polymers and Biocomposites: An Overview," *Macromolecular Materials and Engineering*, 276/277, 1-24, 2000.

24. Mohanty, A.K., Misra, M. and Drazal, L.T., "Sustainable Bio-Composites from Renewable Resources: Opportunities and Challenges in the Green

Materials World," Journal of Polymers and the Environment, Vol. 10, Nos. 1/2, 19-26, April 2002.

25. National Materials Advisory Board, Hierarchical Structures in Biology as a guide for New Materials Technology, Committee on Synthetic Hierarchical Structures, Commission on Engineering and Technical Systems, National Research Council, NMAB-464, National Academy Press, Washington, D.C., 1994.

26. Nogata, F., "Learning About Design Concepts From Natural Functionally Graded Materials," Composites and Functionally Graded Materials, ASME, MD, 80, Dallas, Texas, 11-18, 1997.

27. Pham D.T. and Karaboga D., Intelligent Optimization Techniques-Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks, Springer-Verlag, London, 2000.

28. Quagliata, M.J., "Development and Characterization of Biocomposite Cellular Beams and Plates for Load Bearing Components," Masters Thesis, Michigan State University, East Lansing, MI; 2003.

29. Vincent, J., Structural Biomaterials, Princeton University Press, Princeton, New Jersey, 1990.