Population-Based Techniques for the Multiple Objective Optimisation of Sandwich Materials and Structures

A thesis submitted for the degree of Doctor of Philosophy at Newcastle University

C. W. Hudson

Stephenson Building
School of Mechanical and Systems Engineering
Newcastle University

January 2010

To my parents

and my two brothers

Work like you don't need the money.

Love like you've never been hurt.

And dance like nobody's watching.

Satchel Paige

Acknowledgements

I would like to thank the following for their assistance in making this thesis possible:

- Firstly, my supervisor, Dr Joe Carruthers. His support, commitment, and guidance have provided me with much motivation throughout the project's duration.

- To NewRail, The School of Mechanical and Systems Engineering and Newcastle University for the resources and funding they provided.

- Dr Sandy Anderson and James Hoy whose willingness to discuss ideas openly has provided me with good direction.

- My colleagues Conor and Gaetano who offered their advice and made the experience thoroughly enjoyable.

- To the Royal Academy of Engineering for providing two International Travel Grants which allowed the work to be presented at two major conferences.

- Finally, I would like to thank the people who assisted in the proof reading of this manuscript; Joe Carruthers, Conor O'Neill, William Hudson, Linda Hudson, Neil Hudson, James Hudson, Paul Chubbock, James Hoy and Thomas Mitchell.

Craig Hudson

Abstract

Sandwich materials, consisting of two thin, stiff facings separated by a low density core, can be used to produce structures that are both light and flexurally rigid. Such assemblies are attractive for applications in transport and construction. However, their optimisation is rarely straightforward. Not only is this due to the complex equations that govern their mechanics, but also because multiple design variables and objectives are often present.

The work in this thesis identifies population-based optimisation techniques as a novel solution to this challenge. Three of these techniques have been developed in MATLAB specifically for this purpose and are based on particle swarm optimisation (*sandwichPSO*), ant colony optimisation (*sandwichACO*), and simulated annealing (*sandwichSA*).

To assess their suitability, a benchmark problem considered the application of these techniques to a multiple objective sandwich beam optimisation. Optimised for stiffness mass and cost, a selection of 16 materials for both facing and core were available. Several constraints were also present. The *sandwichACO* technique demonstrated superior ability as it was able to obtain all optimal solutions in most cases. However, the *sandwichPSO* and *sandwichSA* techniques struggled to identify local optimum solutions for the multi-ply, fibre-reinforced polymer sandwich facing laminates.

A further case study then applied *sandwichACO* to the optimisation of a sandwich plate for a rail vehicle floor panel. In addition to the benchmark, the problem was extended to include 40 materials. Also, the material and thickness of the top face was allowed to be different to the bottom. Furthermore, orthotropic fibre-reinforced facing constructions were included, as well as a localised load constraint. A broad range of optimal solutions were identified for the applied minimum mass and cost objectives. Sandwich constructions provided a significant (approximately 40%) saving in both mass and cost compared to the existing plywood design. More significant mass saving designs were also identified (of over 40%), but with a cost premium.

Overall, population-based techniques have demonstrated successful application to the design of sandwich materials and structures.

Nomenclature

Latin symbol	Description	Unit
B	Bending moment	Nm
C	Total cost	€
D	Flexural rigidity for a beam	Nm^2
D_c	Flexural rigidity per unit cost	Nm^2/€
D_m	Flexural rigidity per unit mass	Nm^2/kg
D_x	Flexural rigidity per unit width	Nm
E	Young's modulus	N/m^2 (Pa)
G	Shear modulus	N/m^2 (Pa)
L	Length of sandwich	m
M	Total mass	kg
$\hat{M}$	Mass index	-
N	Total number of variables in a set **x**	-
Q	Shear stress	N/m^2 (Pa)
S	Standard deviation	-
T	Temperature parameter for simulated annealing	-
U	Total number of objectives in a set **f**	-
W	Weighting parameter	-
$\hat{P}$	Load index	-
b	Width of sandwich	m
c	Cost per unit mass	€/kg
c_s	Cost per unit length	€/m
c_1, c_2	Motion influencing parameters for particle swarm optimisation	-
d	Distance between centrelines of two facings	m
e	Parameter governing the error ratio	-
f	An objective in a set **f**	-
f'	Amalgamated objective function	-
$\hat{f}$	Maximum objective value	-
g	Step size parameter for steepest decent method	-
h	Total through-thickness of sandwich	m

i	An iteration	-
k	Probability of moving to an available variable x for ant colony optimisation	-
l	Span	m
m_s	Mass per unit length	kg/m
p	Acceptance probability for simulated annealing	-
q	Distributed load	N/m^2
r	A random number	-
t	Thickness	m
v	Velocity parameter for particle swarm optimisation	-
w	Inertial weight parameter for particle swarm optimisation	-
x	A variable in a set **x**	-
d	Gradient vector for steepest decent method	-
f	A set of U objectives	-
x	A set of N variables	-

Greek symbol	**Description**	**Unit**
Δ	Pertaining to a range	-
Λ	Thermal conductance	W/K
Π	Product	-
Σ	Sum	-
Φ	A set of general solutions that may be both non-dominated and inferior	-
Ψ	A set of non-dominated solutions	-
Ω	A set of available variables x	-
α_1, α_2	Pheromone influencing parameters for ant colony optimisation	-
β	Sandwich plate coefficient	-
δ	Maximum deflection	m
ε	Constraint boundary	-
η	Visibility parameter for ant colony optimisation	-
θ	An angle	°

λ	Thermal conductivity	W/mK
μ	Wind factor for particle swarm optimisation	-
ρ	Density	kg/m^3
ρ	Evaporation parameter for ant colony optimisation	-
σ	Direct Stress	N/m^2 (Pa)
$\sigma_{wrinkling}$	Wrinkling Stress	N/m^2 (Pa)
τ	Pheromone parameter for ant colony optimisation	-
τ'	Intermediary pheromone for ant colony optimisation	-
υ	Fibre volume fraction	-
φ	A solution in a general set Φ	-
χ	Constriction factor for particle swarm optimisation	-
ψ	A non-dominated solution in a set Ψ	-
ω	Cooling factor for simulated annealing	-

Subscript	**Description**
x, y, z	Property refers to Cartesian direction
c	Property refers to the sandwich core
f	Property refers to the sandwich face
f1, f2	Refers to dissimilar faces
s	Referring to the timber support
area	Per unit area
max	Maximum value
min	Minimum value
total	Total value

Superscript	**Description**
current	Current value
new	New value
norm	Normalised value
global	Pertaining to a non-dominated set
personal	Pertaining to an individual search agent
popular	Pertaining to current interest from search agents

Contents

1 Introduction

1.1 Sandwich structures

1.1.1 The sandwich concept

A sandwich structure typically consists of three main parts as illustrated in Figure 1.1. Two relatively thin, stiff and strong facings are separated by a thicker, lower density core material. The layers are firmly bonded together so that when a load is applied to the structure, the forces are transferred between them.

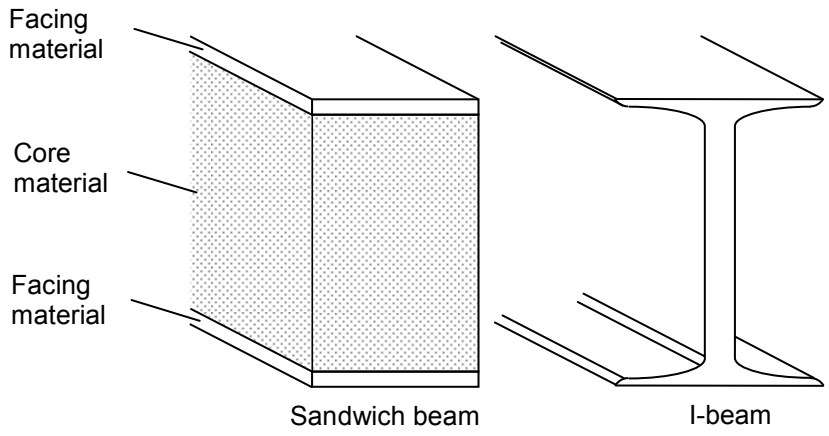

Figure 1.1. The structure of a sandwich, the principle of which is similar to an I-beam.

Structurally, the reason for using a sandwich is that the second moment of area can be dramatically improved without significant increase to the weight compared to a monolithic. The principle is similar to that of an I-beam where as much of the material as possible is situated furthest from the neutral axis. However, the difference with a sandwich is that the flexural stiffness is gained by employing the core, rather than the thin web of an I-beam, to keep the load bearing facings apart.

Such assemblies have a number of characteristics that make them attractive for applications in transport and construction. Their high mass specific stiffness and strength make them a good lightweight structure, leading to improved performance and / or lower life cycle costs. Sandwich materials also provide opportunities for design integration, i.e. the ability to combine different functionalities within a single material construction. For example, mechanical properties such as stiffness or strength can often be combined with thermal properties such as insulation. Also, because there are many facing-core material combinations available, the properties of a sandwich can be closely tailored to suit the application.

1.1.2 Sandwich applications

An early example of sandwiches being used on a large scale was in England with the construction of the World War II plane called the Mosquito in the early 1940's [1]. Originally conceived as a bomber, it used veneer faces and a balsa wood core (Figure 1.2). The lightweight construction allowed competitive speeds and distances to be attained in comparison to other aircraft of its day. Research into theoretical studies on sandwich construction followed World War II with several papers being published between 1945 and 1955, the theories of which can be found for instance in the work of Plantema [2] or Allen [3]. Since then, much development has been made in the aerospace industry using sandwich materials. To date, sandwich constructions can be found in many structural parts for commercial airliners such as stabilisers, flaps and doors [4]. Similarly, other instances where sandwiches have made significant developments can be found in the marine industry. A recent example here is that of The Mirabelle – the world's largest single mast sailing yacht which was constructed in 2004 using a glass fibre-reinforced sandwich structure (Figure 1.3). The motor sports industry is also a sector in which sandwiches have had a major impact [5]. The earliest example of an entirely composite sandwich chassis dates back to the mid 1960's. Made by McLaren, it utilised a balsa wood core bonded between two aluminium faces (Figure 1.4). Sandwich design in the rail industry occurred later. The Intercity 125 passenger train built in 1975 [6] used fibre glass facings and a polymer foam core for the construction of the driver's cab (Figure 1.5).

Figure 1.2. An early application of sandwich construction; the British designed de Havilland Mosquito, a World War II bomber plane [7].

Figure 1.3. The Mirabelle – the World's largest single mast yacht [8].

Figure 1.4. McLaren M2B, the first Formula 1 car to be raced that utilised a sandwich chassis [9].

Figure 1.5. The Intercity 125 passenger train utilised a sandwich design for the driver's cab [10].

1.1.3 Challenges of sandwich design

With the advantages that sandwiches can offer, it would at first seem unusual that they aren't more commonly used. The rail industry is one example that has not yet exploited the sandwich to its full advantage. Robinson et al [11] for instance have noted the benefits that could be obtained through replacing existing components of passenger trains with lightweight alternatives. Friedrich et al [12] recognise the improvements that could be achieved in the automotive sector with using lightweight alternatives for mass-produced passenger vehicles due to their advantages against conventional steel concepts. The construction industry is also keen to make the use of alternative materials more widespread due to their longevity and benefits with requiring less maintenance [13].

However, by far the biggest challenge with designing sandwich structures is managing the vast number of design variables so that good design solutions can be obtained faster and more reliably. To expand this point, sandwich materials are usually realised through an assembly of multiple parts and materials. For simple constructions, a designer has the challenge of selecting the most suitable facing and core materials and determining their optimum thicknesses to meet the needs of the application. But it is not uncommon to extend this by allowing different lengths and widths of the sandwich or facing materials with multi-ply orientated laminae for instance. In addition, there will often be conflicting objectives (e.g. mass versus cost) that will need to be suitably reconciled. Objectives are the functions that need either to be maximised or minimised. Several failure modes may also need to be considered to ensure the product is suitable for its application. With this in mind it is clear to see that the number of design options available is vast. Enumerating all of them by hand or by computer would not be feasible or realistic. Hence, a design strategy that can optimise multiple conflicting objectives and consider the many material and geometric options effectively would be very advantageous. This would not only speed up the process of obtaining a suitable design, but make the designer more confident that the selected sandwich construction is indeed the most appropriate.

1.2 Optimisation: a general overview

1.2.1 Scope of the optimisation techniques to be investigated

To solve the challenges highlighted with multiple objective sandwich design, a broad investigation to find the best optimisation technique for the purpose will be performed. However, it should be noted that marked developments have been made in optimisations research during the last 25 years with optimising complex, multiple objective problems such as those presented by sandwich design. Accordingly, this is where the majority of the effort will be concentrated.

1.2.2 Common working principles of optimisation techniques

While each technique has its own rules for conducting a search in its own right, the general process of performing an optimisation is largely similar. Consider a sandwich beam. Suppose a single objective (minimum mass) is required to be optimised subject to a certain minimum stiffness. Provided with a wide range of facing and core materials and thicknesses, which combination gives the lightest design yet still meets the stiffness requirement? Although fairly trivial, this is an example where optimisation techniques can be employed.

To describe the typical process, the procedure begins by selecting various materials and thicknesses as potential candidates. This is usually done at random. After that, the stiffness of the resulting sandwich designs is then calculated. This completes the first iteration. In subsequent iterations, further sets of materials and geometries are selected to produce more sandwich designs. This time however, the manner in which new sandwich designs are selected differs depending upon the optimisation technique implemented. That is because a history of the searching process now exists. For some, a relatively unsophisticated rule governs the outcome. But for other techniques, the historical information is utilised more *intelligently* to progress the search towards better designs. This can be extremely effective at increasing performance. In any case, better solutions are

identified by continually comparing new designs with existing sandwich constructions. The searching process is exhausted when no more better designs can be found.

1.2.3 Exploiting optimisation techniques for the multiple objective optimisation of sandwich materials and structures

While some examples of sandwich optimisation have been acknowledged [14-23], the opportunity to exploit several areas has been identified. Firstly, previous research conducted on sandwich optimisation lacks complexity in one respect or another. Particularly, this attributes to not considering multiple objectives, not considering the selection of both material and geometry, or being restricted with the general number of design options available. Secondly, little or no research has been conducted on several current state-of-the-art optimisation techniques for sandwich design. Thirdly, no examples offer comparison between many of these methods for sandwich design. Finally, from an optimisations research point of view, an analysis of the types of complexities presented by sandwich design has not been conducted. Hence, the opportunity to more openly investigate a range of techniques for the multiple objective optimisation of sandwich materials and structures is evident and forms the subject of this thesis.

1.3 Scope of the thesis

The investigation and development of optimisation techniques for the multiple objective design of sandwich materials and structures will be addressed in this thesis. The content is broken down into the following sections:

- General aspects regarding the content and setup of a multiple objective sandwich optimisation is given in Chapter 2.

- State-of-the-art techniques for optimisation are reviewed in Chapter 3.

- An investigation and development of the supporting features from which successful optimisation techniques can be built is conducted in Chapter 4.

- A detailed analysis and development of particle swarm optimisation (PSO), ant colony optimisation (ACO) and simulated annealing (SA) for sandwich optimisation is given in Chapters 5 - 7.

- A comparison of the developed optimisation techniques is made by implementing them on a benchmark sandwich beam problem in Chapter 8.

- The preferred optimisation technique will be used to optimise the design of a sandwich structure for a rail vehicle floor panel application in Chapter 9.

- Finally, conclusions and recommendations for further work are given in Chapter 10.

1.4 References

1. (1942) The de Havilland Mosquito. *Flight*, Volume 42.
2. Plantema, F.J. (1966) *Sandwich Construction: The bending and buckling of sandwich beams, plates and shells*. John Wiley & Sons Ltd, Chichester.
3. Allen, H.G. (1969) *Analysis and design of structural sandwich panels*. Pergamon Press, London.
4. Zenkert, D. (1995) *An introduction to sandwich construction*. EMAS Publishing, London.
5. Savage, G. (2009) Formula 1 composites engineering. *Engineering Failure Analysis* **17**, 92-115.
6. Nock, O.S. (1980) *Two miles a minute*. Book Club Associates, London.
7. (1942) De Havilland Mosquito. The De Havilland Aircraft Company of Canada Ltd advertisement
8. (2009) The Mirabella. *http://www.mirabellayachts.com/mirabella5/*.

9. Green, A. (2008) McLaren M2B. *http://www.flickr.com/photos/algreen/2492849328/sizes/m/*.
10. Turner, J. (1994) 53A models of Hull collection. *http://farm3.static.flickr.com/2520/3974053919_db44f484c7_o.jpg*
11. Robinson, M.R., Carruthers, J., Palacin, R. (2006) Lightweighting for mass transit applications. *JEC Composites 2006 (Journees Europeennes du Composites)*, 18-27.
12. Friedrich, H., Kopp, J., Stieg, J. (2003) Composites on the way to structural automotive applications. *Materials Science Forum*.
13. Mirmiran, A., Bank, L.C., Neale, K.W., Mottram, J.T., Ueda, T., Davalos, J.F. (2003) World survey of civil engineering programs on fiber reinforced polymer composites for construction. *Journal of Professional Issues in Engineering Education and Practice* **129**, 155-160.
14. Di Sciuva, M., Gherlone, M., Lomario, D. (2003) Multiconstrained optimization of laminated and sandwich plates using evolutionary algorithms and higher-order plate theories. *Composite Structures* **59**, 149-154.
15. Erdal, O., Sonmez, F.O. (2005) Optimum design of composite laminates for maximum buckling load capacity using simulated annealing. *Composite Structures* **71**, 45-52.
16. Aymerich, F., Serra, M. (2008) Optimization of laminate stacking sequence for maximum buckling load using the ant colony optimization (ACO) metaheuristic. *Composites Part A: Applied Science and Manufacturing* **39**, 262-272.
17. Suresh, S., Sujit, P.B., Rao, A.K. (2006) Particle swarm optimization approach for multi-objective composite box-beam design. *Composite Structures* **81**, 598-605.
18. Kathiravan, R., Ganguli, R. (2006) Strength design of composite beam using gradient and particle swarm optimization. *Composite Structures* **81**, 471-479.
19. Tan, X.H., Soh, A.K. (2007) Multi-objective optimization of the sandwich panels with prismatic cores using genetic algorithms. *International Journal of Solids and Structures* **44**, 5466-5480.
20. Bassetti, D., Brechet, Y., Heiberg, G., Lingorski, I., Pechambert, P. (1997) Genetic algorithm and performance indices applied to optimal design of sandwich structures. *Mechanics of Sandwich Structures*.
21. Gantovnik, V.B., Gurdal, Z., Watson, L.T. (2002) A genetic algorithm with memory for optimal design of laminated sandwich composite panels. *Composite Structures* **58**, 513-520.

22. Wang, T., Li, S., Nutt, S.R. (2009) Optimal design of acoustical sandwich panels with a genetic algorithm. *Applied Acoustics* **70**, 416-425.

23. Kovacs, G., Groenwold, A.A., Jarmai, K., Farkas, J. (2004) Analysis and optimum design of fibre-reinforced composite structures. *Structural and Multidisciplinary Optimization* **28**, 170-179.

2 Multiple objective optimisation: general aspects

In chapter 1, the general outline of a sandwich structure was given alongside the reason why its optimisation is not straightforward. The opportunity to investigate a range of optimisation techniques for dealing with the complexities involved was identified. However, before any optimisation techniques are reviewed for this purpose, some of the more general aspects that make up a multiple objective sandwich optimisation are first explained. This will give a good understanding of the basic problem setup and the likely complexities that may emerge.

2.1 Variables, objectives and constraints

A sandwich design to be optimised must have at least one *variable*, x. Variables are parameters that can be altered by a designer or, as in the work described in this thesis, by an optimisation algorithm. Variables can be discrete (e.g. the choice of sandwich core material) or continuous (e.g. the sandwich core thickness). The envelope that is created when exploring different variable values is known as the *variable space*. Often there will be more than one variable in a sandwich design optimisation. A complete set of such variables, $\mathbf{x}$, may be considered as a vector that consists of a set of N variables such that $\mathbf{x} = (x_1, x_2, \ldots, x_N)$.

The aim of an optimisation will normally be to maximise or minimise one or more *objective* functions, $f(\mathbf{x})$. Examples of objectives might be to minimise the overall mass and/or cost of a sandwich construction. A set of U objective functions may be expressed as $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_U(\mathbf{x}))$. As with the variables, the values of the objectives map out an equivalent *objective space*. Objective values cannot be modified directly. Rather they are controlled by the variable values. Hence a mapping process exists between the variable and objective search spaces (Figure 2.1). The method by which variables are selected and evaluated in order to find the best set of objective values is the responsibility of the optimisation algorithm. Depending upon the context, both variable and objective space may be referred to more generally as either the *design space* or *search space*.

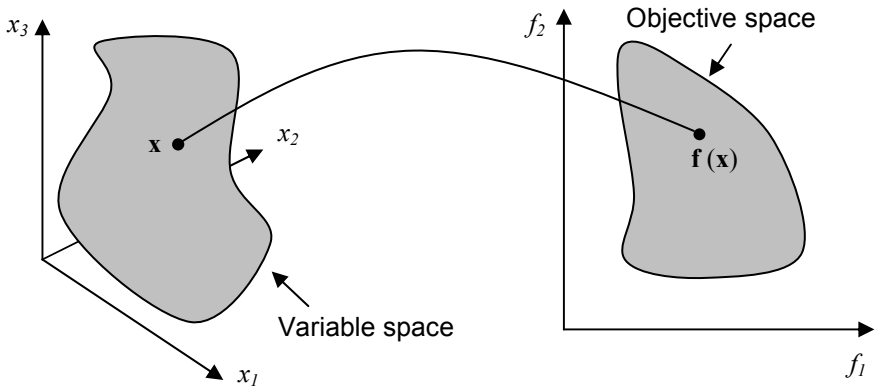

Figure 2.1. A representative mapping process between a solution in the variable space (left) to its equivalent point in the objective space (right). Here, there are three variables x_1, x_2 and x_3, and two objective functions f_1 and f_2.

Next there are the *constraints*. While other constraint classifications exist [1-3], in this thesis two types of constraint are considered. Firstly, there are those constraints that are applied directly to the variable space. Examples would be restricting the range of permissible facing thicknesses, or specifying a particular sandwich beam length. Such constraints reduce directly the overall size of the variable space and have been termed *direct* constraints. The other types of constraint are those that are dependent on a given set of variable values, e.g. the maximum permissible deflection of a sandwich beam, or the onset of a particular failure mode. These are called *dependent* constraints and depend upon the variable values. As such they may be expressed as a function of the variables. These act to divide the variable and objective spaces into feasible and infeasible regions. This introduces complexity for any optimisation algorithm as it must be capable of locating and

navigating all the feasible regions of the search space without becoming lost, trapped or overwhelmed by infeasible areas. While other constraint classifications exist [1-3], the terminology used here focuses attention towards the implications that the constraint has on the search space.

2.2 Implications of multiple objectives and Pareto optimality

It is common in sandwich design for there to be more than one objective. This will usually mean that there will be no single optimal solution. Instead a series of solutions exist that each contain an element of optimality. By way of explanation, consider an ordinary mono-material beam of fixed dimensions. Suppose that there was a requirement to optimise the mass of this beam subject to a certain minimum stiffness. If the beam material was the only variable, the optimisation would be trivial. The material with the lowest density that still met the required stiffness would be selected. Similarly, if the sole objective was to minimise the cost of the beam, the optimal material would be the cheapest option. However, if the objective was instead to optimise both the mass *and* the cost of the beam subject to a certain minimum stiffness, the situation becomes less clear. This is because it is unlikely that the material that produces the lightest solution would also provide the cheapest solution. Instead, when both objectives are considered, a trade-off boundary between mass and cost is formed. The result is a set of solutions which, when all objectives are considered, show some degree of optimal quality. The solutions in this set are not dominated by any other and are referred to as the *true Pareto-optimal set* [2]. Additionally, this is also the definition for *Pareto-optimality*.

2.3 Quantifiable requirements for optimisation

Among the many different features that constitute an optimisation problem, it may be possible to split them up into one of two categories according to the type of information each part provides. Firstly, there is the information which is quantifiable and clear-cut. Included in this category are parameters such as material properties, geometries, strengths, costs, masses etc. This kind of information can be easily entered into a computer database. It is far more straightforward to measure the excellence of a sandwich design using these aspects since their values are fixed and they are difficult to misinterpret.

However, other requirements exist that are not so easy to assess. These are for instance specific to the manufacturing process, material supplier or geographical location. They often revolve around in-house needs specific to a company and form the non-technical, qualitative, experience-driven decisions that must be made. As such, this kind of information is classed as being of a higher-order [2]. If these aspects were included in the optimisation process, they would first need to be quantified in some way so they could be recognised by a computer program. However, devising countable measures for these higher-level factors requires caution. If not represented correctly, some solutions may be underestimated. Ultimately, this could lead to the wrong type of design being classed as *optimal*. This can occur especially if the nature of problem is not well understood. In the current approach, multiple objectives led to the production of a non-dominated set of optimal solutions. The decision-maker must then select a suitable design from the available set. Hence, while it is no doubt possible to develop interpretations of the higher-order information, it seems far more logical, and reasonable that this type of information be negotiated by the decision-maker themselves, after the optimisation has been conducted, rather than during process itself.

2.4 The ideal optimal set

In the case for sandwich design at least, obtaining a Pareto-optimal set of solutions is the ideal approach to multiple objective problem solving. This is because it allows not just one, but a selection of good alternatives to be presented to a designer. This is very advantageous because it gives the designer freedom to select one of the options based on any special considerations or in-house requirements. To take full benefit from the optimisation, two goals have been formally defined by Deb [2] and should be considered when obtaining a Pareto-optimal set. This gives a necessary focal point around which to develop optimisation techniques later down the line. The goals are;

1. To find a set of solutions as close as possible to the Pareto optimal front
2. To find a set of solutions as diverse as possible.

The first goal is perhaps more obvious. Solutions closer to the Pareto-optimal front are more desirable than those further from it. On the other hand, the second goal is entirely specific to multiple objective optimisation. This requires the solutions to be well distributed along the Pareto-optimal front. A diverse set is one which has a broad and even range of solutions over the trade-off between objectives. As such, this gives the best overview of the design alternatives available and is the favoured approach for the problems here.

2.5 Complexities with negotiating the design space

It has been observed that multiple variables, objectives and constraints will be present in sandwich optimisation. This fact alone makes obtaining optimal solutions a challenging task. However, further to this argument, Deb [4] has identified several features that may cause difficulties for multiple objective handling algorithms to arrive at the ideal optimal set. With obtaining the set itself, multimodality, deception, isolated optima and collateral

noise have been identified as issues which may cause problems. In addition, difficulty with maintaining a diverse non-dominated set may arise if the Pareto-optimal front is convex, non-convex, discontinuous or non-uniformly distributed. While great depth has been avoided, each of these issues will be described to give a general appreciation of the likely scenarios which may arise when optimising sandwiches. In each of the cases presented, the graphs consider objective minimising functions.

2.5.1 Multimodality

Multimodality in an optimisation problem occurs particularly when a very large number of near-optimal solutions (or multiple peaks) are present in the problem. This can cause the optimisation algorithm to get stuck at sub-optimal non-dominated fronts rather than converging to global optimal solutions. Figure 2.2 shows a representative multimodal problem.

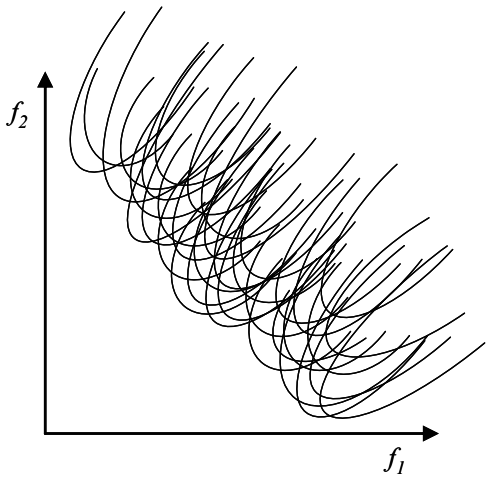

Figure 2.2. A representative example of multimodality.

2.5.2 Deception

Deception occurs when an algorithm is drawn to a non-dominated set that is local to a particular area of the entire solution space. In some cases, this may not even be truly Pareto-optimal (i.e. sub-optimal). Particularly, if a large proportion of solutions in the search space lead to the deceptive front, this can heavily influence the search.

Consequently, it can be difficult to explore sparse, uncharted regions significantly further away where true Pareto-optimal solutions may lie. Figure 2.3 shows a representative example of deception in which most search agents are drawn to a sub-optimal local region.

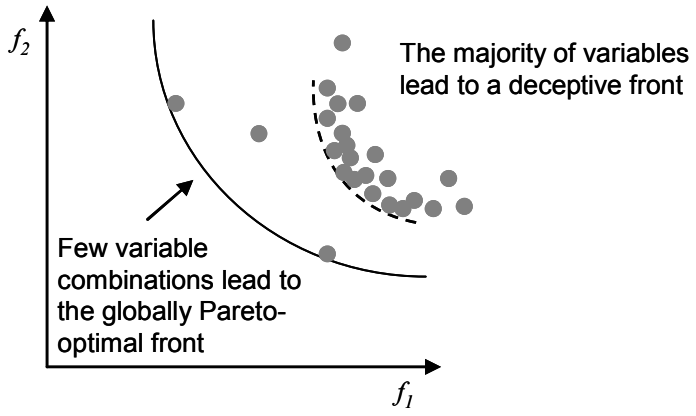

Figure 2.3. A representation of a deceptive front.

2.5.3 Isolated points

Some problems exist where an optimum is surrounded by a fairly flat search space. That is to say that the objective value of surrounding solutions is commonly poorer. No useful information may be acquired as to the optimums whereabouts, even if a trial solution searches close-by (Figure 2.4). As such, it is difficult for any optimisation process to obtain these points and in many cases only an exhaustive search would guarantee their identification.

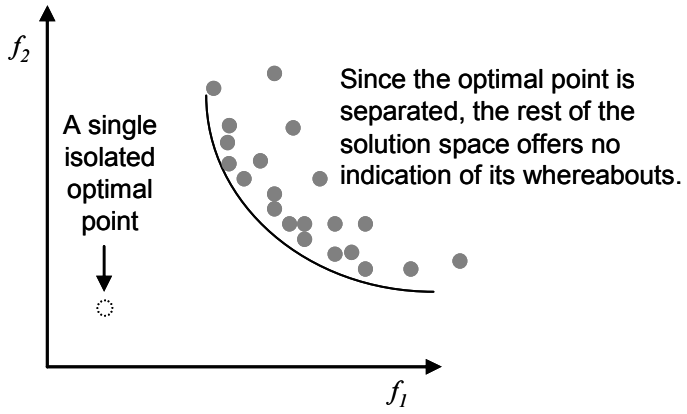

Figure 2.4. A representation of an isolated optimum point.

2.5.4 Collateral noise

Collateral noise is a feature that exists in the Pareto-optimal front when its overall trend contains an element of distortion. It is characteristic of a *rugged* landscape with frequent fluctuation in objective value (Figure 2.5). However, this aspect is less associated with optimisation of static sandwich design and is more common with dynamic problems where the optimal solutions change continually with time.

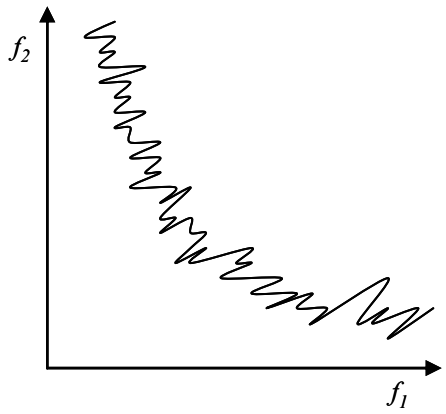

Figure 2.5. A representation of collateral noise affecting the Pareto-optimal front.

2.5.5 Convex and non-convex Pareto-optimal fronts

Cooper and Steinberg [5] state that the geometric shape of the design space is crucial with respect to the difficulty encountered when solving an optimisation problem, especially when it is constrained. Particularly, they relate this to the convex and non-convex characteristic shapes of either the variable or objective space. While convex Pareto-optimal fronts are not without their complications, they are in one sense, are easier to deal with. A space is convex if for every pair of points within it, every point on the straight line segment that joins them is also within the search space (Figure 2.6). On the other hand, for a non-convex instance, a straight line segment will exist that ventures outside the space. Hence, any space that is hollow or has a dent in it, for example, a crescent shape, is non-convex. This aspect has significant practical consequence because some objective handling methods are entirely unable to detect non-convex parts of the Pareto-optimal front. In general, Deb [2] points out that it is difficult to know in advance of solving a problem whether it is non-convex. Given the constrained nature of sandwich optimisation, and as

with the other complexities, it is reasonable to anticipate either scenario and guard against it.

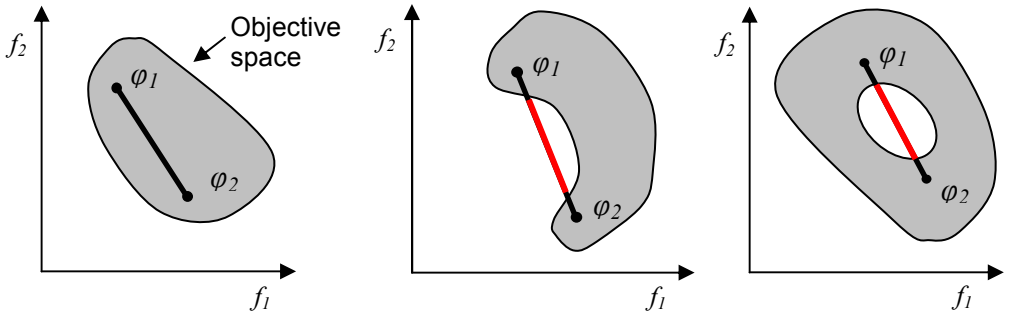

Figure 2.6. Example of a convex (left) and a non-convex (centre & right) objective space shown for a pair of points φ_1 & φ_2 [5, 6].

2.5.6 Discontinuous Pareto-optimal fronts

Discontinuous Pareto-optimal fronts are those which do not have a continuous flow from one point to the next (Figure 2.7). Usually, once a section of the Pareto-optimal is found, it can be easy for an optimisation technique to traverse along it and uncover more. However, if discontinuities occur, the optimiser must instead be able to "jump" to the other Pareto-optimal regions.

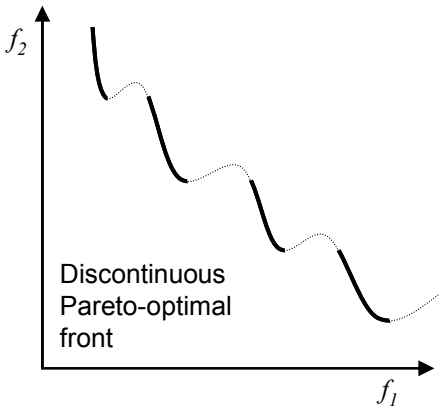

Figure 2.7. The dotted line shows the outline of a representative objective space. The solid line represents the resulting discontinuous Pareto-optimal front.

2.5.7 Non-uniformly distributed Pareto-optimal sets

A non-uniform spread of solutions over the Pareto-optimal front can occur if the objective function in question is either nonlinear, or is a function of more than one variable. This can cause difficulties since the aim (for the optimisation techniques in this thesis at least) is to produce, in contrast to how the solutions are actually distributed, an even spread within the non-dominated set over the entire trade-off surface. Also, the challenge with many nonlinear functions is that they cannot be solved analytically. Often, the only way the shape of the function can be determined is to use an approximation e.g. Taylor series, or to numerically work out the function value using each variable in turn.

2.5.8 Anticipated complexities with sandwich design

Having explained the aspects which may cause difficulty for an optimisation, some speculation as to which of these are present in sandwich design will be made.

Firstly, it is probable that multimodality will be an issue here. The number of near optimal solutions would increase say depending upon the number of facing-core material options available. If a choice of facing thicknesses was also permitted say, this would create a range of optimal solutions for each combination.

A second aspect which may be present is that of deception. If a significant proportion of facing-core material combinations within a certain range of thicknesses led to roughly the same optimal value, search agents significantly gravitate towards them. It would then be tricky for any optimal solutions that differed notably from this majority to be found.

Isolated points too may cause difficulty. This may occur as a result of the anticipated constrained nature of sandwich design (e.g. strength or thermal). Constraining the design space creates infeasible regions. This may cause feasible areas to become isolated and hence make them more difficult to obtain.

Regarding the convex nature of the search space, In general, Deb [2] points out that it is difficult to know in advance whether a problem is non-convex. Given the constrained

nature of sandwich optimisation, and as with the other complexities, it is reasonable to anticipate either scenario and guard against it.

Discontinuous Pareto-optimal fronts are also a likely scenario. This is more obvious with the simple inclusion of discrete variables e.g. facing material, core material, fixed facing thicknesses or ply angle of laminated fibre-reinforced facings.

Finally, given the nonlinear nature of sandwich mechanics, a non-uniform spread of solutions in the design space will almost certainly be present.

2.6 Combinatorial optimisation problems

An area of optimisation which has been extensively studied, particularly with population-based techniques, is that of combinatorial optimisation problems. These are problems where the set of feasible solutions is discrete or can be reduced to a discrete one, and the goal is to find the best possible solution. To illustrate this, several classic examples of such problems will now be described. These will be referred to at various points later in the thesis. Classically, they are all single objective problems. However, in particular instances, they have been extended to include multiple objectives.

2.6.1.1 The travelling salesman problem

In general, the basic formulation of the travelling salesman problem involves a number of different towns (or nodes) which all need to be visited in the shortest possible distance. This is a single objective (distance minimising) problem, which requires the first and last towns (nodes) visited to be the same.

2.6.1.2 Job shop scheduling

Usually in job shop scheduling, a number of jobs (that take varying lengths of time) need to be scheduled on a set of identical machines. The aim is then to work out which job should be allocated to each machine to minimise the total time to complete them all.

2.6.1.3 Vehicle routing

Commonly, the vehicle routing problem seeks to service a number of customers with a given fleet of vehicles. Often, the problem is to deliver goods to the customers from a single central depot. The aim is to minimise the total cost of distributing the goods.

2.6.1.4 Knapsack problem

The classic knapsack problem normally involves a given set of items, each with a weight and a value. The objective is to fill a knapsack so that the total weight is less than a given limit but the value is as large as possible.

2.7 Conclusions

The general aspects which formulate a multiple objective sandwich optimisation have been explained. It has been recognised that multiple variables, objectives and constraints will be present and an appreciation of special considerations when handling multiple objectives has been given. Explanations of some common combinatorial optimisation problems have also been given. In addition, several different factors likely to crop-up have been described which are known to cause difficulty with finding optimal solutions. For sandwich design, these were anticipated to be multimodality, deception and isolated points. Difficulties with maintaining a well spread an even non-dominated set were also anticipated. Particularly, the convexity of the solution space and the inclusion of discontinuous, non-uniformly distributed Pareto-optimal fronts may be of a concern.

Now that these complexities have been addressed, a critical analysis of the literature may be conducted which focuses more directly at finding suitable techniques for sandwich optimisation.

2.8 References

1. Ashby, M.F. (2005) *Materials selection in mechanical design*. Elsevier Butterworth-Heinemann, Italy.
2. Deb, K. (2001) *Multi-objective optimization using evolutionary algorithms*. John Wiley and Sons Ltd, Chichester.
3. Surry, P.D., Radcliffe, N.J., Boyd, I.D. (1995) A multi-objective approach to constrained optimization of gas supply networks. *AISB-95 Workshop on Evolutionary Computing*, 166-180.
4. Deb, K. (1999) Multi-objective genetic algorithms: problem difficulties and construction of test problems. *Evolutionary Computation* **7**, 205-230.
5. Cooper, L., Steinberg, D. (1970) *Introduction to methods of optimization*. W. B. Saunders Company, London.
6. Kunzl, P.H., Tzschach, G.H., Zehnder, C.A. (1968) *Numerical Methods of Mathematical Optimization*. Academic Press Inc., London.

3 Optimisation for sandwich design: a state-of-the-art review

In the previous chapter some of the more general aspects that formulate a multiple objective sandwich optimisation problem were explained. While it was shown that multiple variables, objectives and constraints are present in the optimisation process, it was acknowledged that the inclusion of multiple objectives meant that special treatment was required. This led to a definition of Pareto-optimality being given which demonstrated that not just one, but a trade-off of multiple optimal solutions can exist. In addition, several features known to cause difficulty with finding optimal solutions were identified. A review of optimisation techniques will now be given in relation to their suitability to sandwich design. This will make it clear which techniques should be carried forward and developed specifically for the sandwich purpose.

3.1 Some terminology

Throughout this thesis, several terms are used interchangeably to refer to aspects of the same nature. Depending upon the context, it may be more appropriate in particular instances to use one over another. One of these regards the entire optimisation process from start to finish. This may also be called a simulation or a run.

The optimisation method (for example particle swarm optimisation or ant colony optimisation) used to perform a simulation may also be called a technique, an algorithm or a process.

The elements of the algorithm which carry out the search procedure itself may also be referred to as search agents, trial solutions, birds, ants, atoms or particles.

Likewise, a number of different terms are also used to describe the variables (e.g. facing thickness, core material etc.) that exist in the optimisation. These may be referred to as nodes, towns, positions, variables, points, locations, or trails.

3.2 Sandwich optimisation: general classification

Previous works on the optimisation of sandwich structures have approached the subject from a number of different perspectives. Among the many that exist, it may be possible to split them up into a loose hierarchy. At the top, the most general categorisation is that of analytical and numerical methods. The numerical methods themselves may be broken down into two classes: single point techniques and population-based techniques. Since it is the population-based methods where most of the efforts in this thesis have been focused upon, this category makes up the majority of the content in this chapter. Justification for this is given through critical analysis of each other method available.

3.3 Analytical and numerical optimisation methods

Analytical methods mainly require the user to carry out the optimisation manually. Commonly this requires a systematic procedure to be followed in order to arrive at a particular optimal design. Recent examples for solving sandwich design problems via

analytical methods have proven to be successful. Steeves and Fleck [1] followed the approach of Gibson and Ashby [2] to produce failure mechanism maps for sandwich beams. However the study was restricted to a single objective problem (minimum mass in three-point bend) for which a characteristic mass index was minimised for a given load, material and geometry. Figure 3.1 shows a typical example of such a failure map. In this case, the dashed lines indicate non-dimensional load ($\hat{P}$) and mass ($\hat{M}$) indices of constant value for different facing-to-core thickness ratios (t_f/t_c). The predicted failure modes for a given set of values are superimposed and can be acquired directly.

More recently, Pflug and Verpoest [3] extended the well known Ashby [4] material selection chart method for sandwich problems. Figure 3.2 shows a typical Ashby-type material selection chart for selecting the facing and core materials based on a performance index that combines the Young's Modulus and density. However, whilst such Ashby-based methods have been used to accommodate multiple objectives and even identify Pareto-optimal solution sets [5], their general approach, alongside other analytical techniques, is somewhat contrary to the direction taken in this thesis. This is because they generally rely on narrowing down an exhaustive set of material options so that a decision can be made between a manageable few. For this thesis however, the aim is to keep the range of material combinations deliberately large to allow any potentially new or non-obvious solutions to be discovered.

In contrast to analytical approaches, numerical optimisation methods are largely automated procedures executed via computer simulation. They are governed by a set of transition rules, which when implemented iteratively, enable better solutions to be obtained. Due to this, they can more freely explore the nature of the equations that govern the problem. If managed correctly, tremendous benefits may be obtained through the employment of computer processing power to solve vast quantities of data. Due to these advantages, the rest of this review will be concerned only with numerical methods.

Numerical methods can be categorised into two types: single-point and population-based techniques. Although it is not absolute, the categorisation has been formulated to make a clear distinction between optimisation techniques that work on an individual point by point basis, using comparatively less intelligent rules to conduct the next move, and those which utilise a population of search agents at every step in the process.

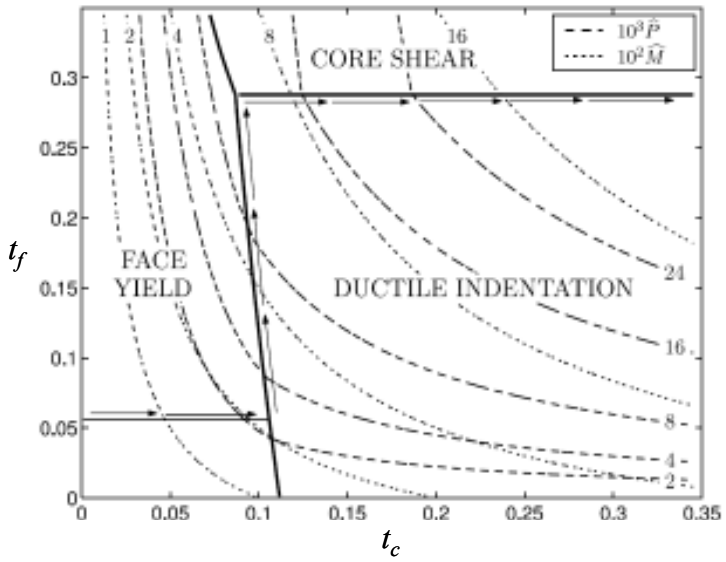

Figure 3.1. Typical failure mechanism map predicting the failure of a sandwich beam for various facing-to-core thickness ratios. Figure taken from Steeves and Fleck [1].

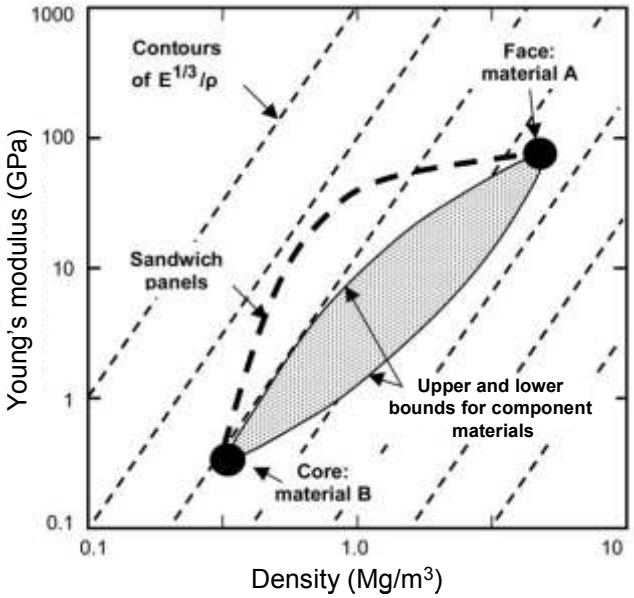

Figure 3.2. A typical Ashby-type material selection chart comparing the performance of sandwich panels with their component materials based upon the Young's modulus and density. Figure taken from Ashby and Brechet [6].

3.4 Single point techniques

The classification of the techniques described here all have the common aspect of operating on a single point-by-point basis. Also, in many cases, while it is possible for these techniques to handle multiple objectives, they were originally developed as single objective optimisers. Also, due to their long established development compared to population-based techniques, a number of authors have also termed these single point methods as traditional or classical [7-9].

Before any explanation of these techniques is given, a process called normalisation of objectives is first explained. That is because some of them rely on this process to work effectively. The general aim is to scale the objectives of a problem to ensure they are of a similar order of magnitude.

3.4.1 Normalisation of objectives

Normalisation is the process of scaling each objective in a multiple objective problem so that, between the ranges of their values, they more or less have the same order of magnitude. This allows the objectives to be directly comparable. For example, for a given set of solutions, if cost and mass objectives are compared, the cost may vary from €1 - €1000 whereas the mass may only differ between 0.01kg – 0.1kg. Clearly, the ranges of these values differ significantly. So, by dividing each objective value by its range, this brings the retrospective orders of magnitude suitably in line with each other. An equation to normalise an objective, u, may be written:

$$f_u^{norm} = \frac{f_u}{\Delta f_u} \tag{3.1}$$

Where f_u^{norm} is the normalised value and Δf_u represents the known range of the objectives up until that point in the simulation.

3.4.2 Gradient-based techniques

Gradient-based methods use the gradient of the objective function to optimise the problem. Using gradient information can be a rapid approach to finding optimal solutions. However, this is generally only the case when the objective functions to be solved are fairly simple. For instance, these methods do not perform well if an objective function has many local optima [8, 9]. This is because they often terminate once the gradient of the objective function is very close to zero. Also, the gradient of the function actually has to be obtainable. Given the complex governing equations of sandwich design, it is likely that these difficulties may arise. In addition, most, if not all of the techniques here were originally conceived as single objective optimisers. If multiple objective criteria are desired, an amalgamation of objectives or some kind of work-around to visualise the problem as a single objective case may be required. Also, depending upon the complexity or size of the problem, several stages in the process may be needed. Examples of gradient based methods include Newton-Raphson method [10], steepest descent method [11], Fletcher-Powell method [11] and the Davidon method [10]. To give a better understanding of these techniques, the basic application of the steepest descent method will be described [12].

This method firstly requires the initial variables of the problem to be selected by the user. For a single objective problem, the partial derivatives of the objective function are calculated for each variable. This gives the gradient of the objective function in each of the relative variable directions. For a given iteration, *i*, this *gradient vector* then points in the direction which gives the greatest rate of increase in objective function value. This vector may then be normalised to ensure that it is of unit length, **d**, then multiplied by a fixed step size, *g*, to acquire the next point. If the objective function is to be minimised, this is subtracted from the current position, $\mathbf{x}_i$, to determine the new point, $\mathbf{x}_{i+1}$. The equations may be written as:

$$\mathbf{d}_i = \frac{\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_N}}{\left[\sum_{n=1}^{N}\left(\frac{\partial f}{\partial x_n}\right)^2\right]^{1/2}} \tag{3.2}$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i - g_i \mathbf{d}_i \tag{3.3}$$

Figure 3.3 shows a representative example of the method. Solutions φ_1 to φ_5 are plotted as a result of the first 5 iterations of a problem with two variables (x_1 and x_2). There is a single minimising objective, the contours of which have been superimposed. Notice that after the 5th iteration, solution φ_5 is much closer to the minimum than solution φ_1.

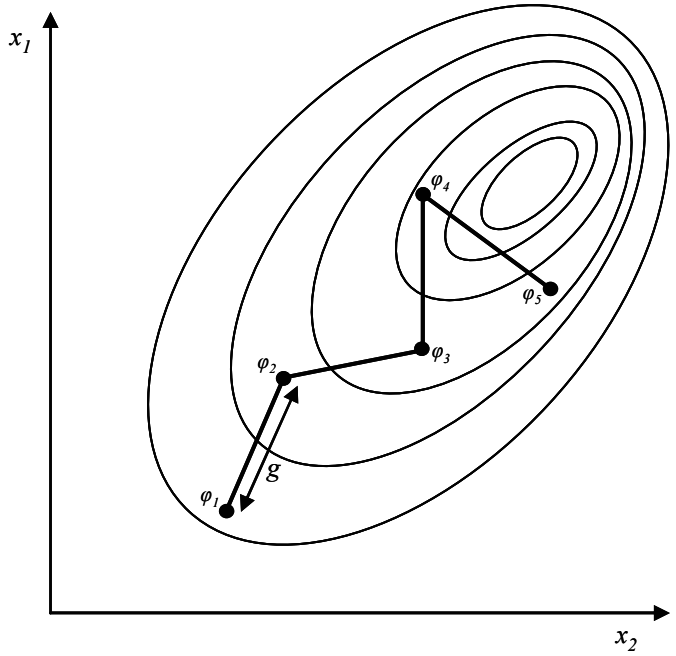

Figure 3.3. A representation of the steps involved in the steepest decent method. The variable space is plotted with contours corresponding to a single objective function with a single minimum.

3.4.3 Direct search techniques

The second half of this categorisation is concerned with *direct search* methods. Several definitions of direct search appear [8, 10, 12]. What is common to all is they state that only evaluation of the objective function(s) is needed and, in contrast to gradient-based methods, they do no require evaluation of derivatives. Under any of these definitions, strictly, it would be expected that population-based optimisation be included [13]. However, in order to consider population-based methods in their own right, a difference has been drawn to allow only direct search methods, which use a single point, to be

discussed here. A typical example of a single point direct search method is the simplex method by Spendley et al [14] and is briefly described.

In the simplex method, a single objective function is evaluated at $N + 1$ equally distant points in the space of N independent variables. For ease, suppose there are two variables ($N = 2$) and a single objective is to be minimised. In this case, three initial points ($\varphi_{1\text{-}3}$) would form the vertices of an equilateral triangle. Figure 3.4 shows the variable space on which representative objective function contours have been plotted.

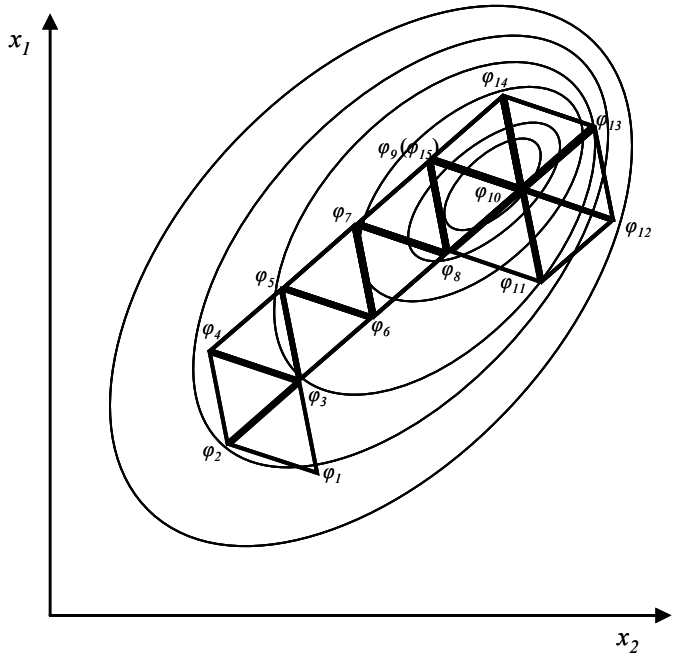

Figure 3.4. A representation of the steps involved in the simplex method. The variable space is plotted with contours corresponding to a single objective function with a single minimum.

The basic iterative procedure is then as follows:

1) Evaluate and compare the objective value at each of the three points. The point with the largest value is noted and a reflection about the centroid of the other two points is performed.
2) Evaluate the new objective function values of the new point and revert to step (1)

If the new point happens to be of greatest function value, then the procedure would merely oscillate between the last two points. To prevent this, a rule is introduced:

3) If the most recently introduced point is of greatest value, select the next largest point.

In Figure 3.4 rule (3) has been implemented for the points φ_8, φ_{10} and φ_{11}, point φ_8 being rejected instead of point φ_{11}. Owing to the fact that point φ_{10} is close to the minimum, the search revolves around this point. This is a characteristic of using equilateral triangles. When it occurs, to obtain a better approximation of the minimum, the distance between the points (size of the triangle) must be reduced and entire process repeated.

Several improvements on this procedure were later proposed; namely the Nelder-mead method [15] and the complex method of Box [16, 17]. Other techniques belonging to this category include Fibonacci search [11], random search [12], Powell's method [18], Hooke-Jeeves search [19], and Rosenbrock's method [20]. However, while this covers only a small fraction these methods, the point to note is that they are all not very suitable for the type of sandwich design considered here. Several reasons for this exist [21]. Primarily, they were originally conceived as optimisation techniques for problems with single objectives. If multiple objective criteria are desired, as with the gradient based methods, a way of picturing the problem as a single objective case is required. Also, despite being relatively simple to implement, their rules on which to make the next move are relatively primitive. So, if many local optima exist in the search space, they are susceptible to becoming trapped. This means that the success of the technique is more heavily dependent on the initial starting point [8, 12]. Furthermore, an intimate knowledge of the problem is often needed to ensue the method will work effectively which can be a time consuming process. Finally, some require the initial starting point to be feasible which can be problematic if the problem is constrained.

An example of sandwich optimisation using such less intelligent direct search methods has been found. Markis et al [22] investigated the single objective maximum transmission loss for providing acoustically-damped sandwich panels. Three variables were considered: core thickness, density and facing thickness. Three facing materials were also considered, but due to the complexity this created, this optimisation was executed separately. In addition, an upper limit on the mass was also applied. While this example shows that these techniques are not entirely unusable, a more intimate knowledge of the problem was required to ensure the selected method was suitable. Furthermore, even with this fairly

restricted search space, multiple stages in the optimisation process were required as opposed to a single run.

3.5 Population-based techniques

Population-based optimisation techniques employ a population of search agents or trial solutions at every step during an optimisation process. The principle of utilizing a group of search agents working towards common objectives is better than a sole agent acting independently. Not only that, but of the methods reviewed here, they can *intelligently* select potentially good solutions by building upon the current success of the procedure. Given the multiple variable, objective, and constrained nature of sandwich design optimisation, population-based methods appear to lend themselves as excellent candidates. While not all population-based methods have been reviewed in detail here, several have shown significant success in the areas they have been applied. Of those that hold potential, critical investigation has been carried out for their suitability for sandwich optimisation.

In relation to the wider field, a number of different terms have been used to describe the methods detailed in this section. These terms include heuristic, meta-heuristic, probabilistic, stochastic, evolutionary, and population-based. Each of these carries meaning in its own right, yet several authors use different terms to refer to the same method. While an element of overlap no doubt exists, in this thesis, the optimisation techniques in this section have solely been referred to as population-based. This, it is felt, conveys a simplistic and obvious meaning to the reader, and arguably avoids the use of an extended vocabulary.

3.5.1 Genetic algorithm (GA)

The first practical application of genetic algorithms (GA) was conducted by Schaffer [23] in 1984 with a technique called the Vector Evaluated Genetic Algorithm (VEGA). However, it was the work of Goldberg [24] in 1989 which sparked the development of

several more widely used techniques [7]. Although many variations have been developed, the basic principles are common. For clarity of the terminology, evolutionary strategies, evolutionary algorithms and evolutionary programming also appear in the literature [25]. However, at least for the purposes of this thesis, they may be regarded alongside GAs as having similar working principles. In any case, the general process is based on mimicking the principles of biological genetics. A group of candidate solutions, or strings, initially populate the search space at random. Each solution has their fitness evaluated. A sample of the best are then placed into a *gene pool*. In the gene pool, crossover takes place. Crossover represents reproduction of the species. It involves swapping elements of two strings with one another. This creates a hybrid which is hopefully better than any previous solution. After that, a mutation operator is employed. This makes small random changes to the strings and adds diversity to the population by enabling some strings to search otherwise uncharted areas.

3.5.2 Particle swarm optimisation (PSO)

Particle swarm optimisation (PSO) was first proposed by Kennedy and Eberhart [26]. It aims to mimic the social behaviour of flocking birds. A flock of birds (particles) with common objectives (e.g. the best food source or roosting site) is more likely to find good locations (optimum solutions) than a sole agent acting independently. Each bird in the flock is guided by three types of information: the best solution that each individual bird finds, a solution known globally to the whole flock, and the previous motion made by the bird. These three factors are added to the bird's current position to establish its next move.

3.5.3 Ant colony optimisation (ACO)

Ant colony optimisation (ACO) was first implemented by Dorigo et al [27]. Similarly to PSO, it employs a group of information-sharing search agents tasked with finding good objective values. However, the mechanisms of movement and information sharing are quite different to those of PSO. ACO is based on the analogy of ants leaving their nest in search of food. As an ant traverses the variable space, it leaves behind a pheromone trail that increases the likelihood that other ants, in subsequent iterations, will follow the same

path. After each iteration, an evaporation mechanism is used on all pheromone levels to discourage poorer solutions from being followed. Importantly, once all the ants have completed their journey (iteration) to the food source, they are returned to the nest ready for the next iteration. As such, the ants have no memory of where they and their colleagues have been. They are solely influenced by the residual pheromone levels.

3.5.4 Simulated annealing (SA)

Simulated annealing (SA) was developed independently by Kirkpatrick et al [28] and by Cerny [29]. It is inspired by the manner in which a molten metal cools during annealing. By controlling the rate at which a metal cools, the atoms are allowed to reach a state of minimum energy, and hence find optimal solutions. Each atom in the optimisation process moves randomly and independently of the others. The degree of permitted movement of an atom is dependent on the *temperature* at any given iteration. A higher temperature implies a higher atom energy and therefore a greater range of permitted movement. The temperature reduces over the course of the simulation at a rate specified by the *cooling schedule* which governs the convergence of the algorithm.

3.5.5 Tabu search (TS)

The Tabu search (TS) was originally developed by Glover [30] to be used as a local search method in conjunction with a global optimiser. From an initial random starting point, a list of possible moves which could provide the next iteration is produced. All the moves are evaluated, the best move is selected and the search moves on. Recently visited solutions are termed “tabu” and are not allowed to be revisited until a certain number of iterations has elapsed. This prevents the algorithm from cycling (becoming trapped in local minima) and encourages movement to uncharted areas. If a new move, labelled as tabu, is found to be of high quality, then an “aspiration criteria” or exception to the rule gives the solution an opportunity still to be visited. But this is only if this criterion is met. So if the revisited solution is indeed optimal, this allows the point not to be unnecessarily avoided.

3.5.6 Simulated biological growth (SBG)

Simulated biological growth (SBG) was developed by Mattheck and Burkhard [31, 32] and is a process that mimics the way trees optimise their growth by keeping the skin surface stress constant. To this extent, it shares a common element with many population-based techniques in that it replicates natural phenomenon. Hence that is why it has been categorised here. However, it must be noted that it does not utilise a population of search agents and nor does not operate under the same searching principles as the other population-based techniques. Instead, the process is more systematic. It works by progressively adding or removing material from an existing design in order to find the optimal shape.

3.6 Previous research conducted on population-based techniques for sandwich design

Several examples of population-based methods applied to composite components have been noted [9, 33-36]. However, few exist which actually deal with optimisation of sandwich structures. Of those that do, mainly genetic algorithms have been utilised. Furthermore, only one example has been found which considers multiple objectives. This was conducted by Tan et al [37]. They performed a multiple objective optimisation of a sandwich plate for minimum weight and maximum heat transfer. However, only geometrical aspects of the sandwich were optimised and the core and facing materials were restricted to aluminium. This meant the problem was rather limited in terms of potential options available.

Other cases of genetic algorithms applied to sandwich optimisation exist but only consider single objective optimisation. An early example of which was conducted by Bassetti et al [38] who optimised an insulating sandwich panel for a truck. This, they claim, proved the feasibility of using software to perform both material and geometry selection of sandwich structures. The single minimum mass objective was able to integrate different stiffness or

strength criteria. However, the optimisation did not consider the material composition or lay-up of the laminated composite sandwich facings, which would otherwise have significantly increased the magnitude of the search space. Gantovnik et al [39] optimised the geometry of a sandwich panel with a minimum mass objective. Fibre-reinforced facings were considered for which optimal stacking sequence and number of plies were investigated. However, no material selection was conducted. Wang et al [40] on the other hand did conduct a material selection of both facing and core material with a minimum mass objective. However, material thickness was the only geometric variable. The only instance where simulated annealing has been applied to a sandwich optimisation was performed by Di Sciuva [9]. In addition, optimisation of a sandwich-like structure was attempted by Kovacs et al [41] using particle swarm optimisation. However, in both cases, angle orientation of the laminated facings was the only variable optimised by the two techniques.

3.7 Comparison of existing population-based techniques

In this section, several previous works involving population-based techniques are compared to establish their potential suitability for sandwich design.

Coello Coello [42] compared a PSO with two well known GAs (SPEA and NSGA-II [43, 44]) and another GA which they developed called microGA. They showed that for a convex problem, the PSO was not only better at producing solutions *at* the Pareto-optimal front, but also produced them closer to it. In addition, a problem with a discontinuous Pareto-optimal front was considered in which the PSO was again superior at finding solutions that lay on the Pareto-optimal front. Two multimodal functions were also tested [45]. The PSO was able to arrive closer to the Pareto-optimal front than the GA techniques and also obtained a significantly wider distribution over the entire trade-off surface. Two out of three GAs in this case produced non-dominated solutions that were poorly

distributed and largely clustered together. The authors also comment that the PSO was computationally very fast in comparison to the other GAs.

The work of Garcia-Martinez et al [46] tested a range of different ACO techniques against the SPEA [44] and NSGA-II [43] genetic algorithms. Several multiple objective travelling salesman problems [47] were used to give an indication of how adaptable the algorithms were to different problem scenarios. Four metrics described by Zitzler et al [48] were used to give a quantitative evaluation of the performance of each technique. From the visual analysis it was found that the non-dominated solutions sets found by the majority of the ACO techniques dominated those found by any of the GA techniques. In the vast majority of problems tested, the GAs produced solutions further away from the Pareto-optimal front than the ACO techniques. The NSGA-II is considered to be one of the state-of-the-art multiple objective GAs for continuous optimisation. However, its relatively poor performance shows that it is not as well-suited as the ACO in the cases presented

Comparing SA with GA, Di Sciuva et al [9] showed that the SA produced results in good agreement with the GA technique used. However, because the computational effort required by the SA was significantly less, it was chosen as the preferred method. In addition, the SA was able to produce a family of optimal stacking sequences for a sandwich plate problem as appose to a single configuration given by a gradient based method. A recent study by Zheng et al [49] compared the performance of GA, PSO and ACO for minimising the production of nitrogen oxides from a coal-fired utility boiler. The results showed that ACO was found to perform the best out of the three techniques used. However, PSO performed less well with a marked susceptibility to becoming trapped in local minima rather than fully searching the entire variable space. Dong et al [50] compared the performance of a PSO against a GA on six test functions. The vast majority of problems tested showed PSO generated superior solutions. In particular, the PSO showed superior quality when tested on a multimodal problem, a non-convex linear problem and an exponential problem. Elsewhere, it has also been pointed out that when presented with multiple optima, a phenomenon known as genetic drift can cause a population of solutions to converge to only one optima and give poor sampling of the solution set [51]. This was first observed in GAs by Goldberg and Segrest [52]. Chen et al [53] noted that GAs usually suffer from premature convergence in solving deceptive problems because most search agents become trapped into local minima due to the lack of

diversity. Furthermore, simple GAs have been noted to have premature convergence and below par performance on multimodal problems [54].

To make comment about the TS, Machado et al [55] noted that the method has not been applied to many areas of engineering design and so developed a TS for optimisation on a multimodal function with continuous variables. In this instance, they demonstrated that TS can be a better technique than SA. However, they point out that the use of a Tabu list can cause the algorithm to become trapped in local minima if continuous variables are present. Youssef et al [56] compared the performance of SA, TS a GA on a multi-criteria floor planning problem of very large integrated circuits. The TS gave better results in terms of solution quality because it spent significantly less time re-visiting the same area. The GA required the most effort to implement and tune the parameters to suit the problem.

Turning attention now to hybrid methods, Smaili and Diab [57] have recently shown success with using ACO in combination with a gradient-based method. The single objective was to find the optimal linkage lengths of a four bar mechanism so that the motion generated by the mechanism was as close as possible to the desired trajectory. In addition, several constraints imposed upon the motion and geometry were also integrated in to the objective function. Since many local optima were present in the design space, the ACO was regarded as an effective means to provide an initial global search. The gradient-based method was then responsible for refining the end solution. The method was shown to be a rapid approach to solving the problem and competitive against another hybrid technique which utilised a tabu search as the global search algorithm [58]. However, while gradient based methods have no doubt shown success in case specific instances, they still carry the more general disadvantages mentioned earlier with converging to local optima and requiring the gradient of the objective function to be obtainable. Elsewhere, Praveen et al [59] combined a direct search (Nelder-mead) method with a PSO algorithm. The basic idea was to split up the global search algorithm (PSO) into several clusters. The direct search method was then used separately within each cluster to improve the local search performance. However, despite the authors remarking on the success of the performance on some well established test functions [42, 48] they also note that the local search method is more suited to problems with relatively smooth trade-off boundaries with little collateral noise. In a separate case, Jeon and Kim [60] used the TS in connection with an the SA algorithm. Here, the problem was to produce large scale wiring networks

connections. Due to its good convergence property, the SA acted as the main search algorithm to search the global search space. The TS was then used afterwards to explore the local region and hone in on particular optimal solutions. Although some technical boundaries were met when combining the two, the method was shown to be successful.

3.8 Critical analysis of population-based optimisation techniques

After presenting several comparative examples, multiple cases have shown that the GA is out-performed by various other population-based techniques. In particular, the GA has shown susceptibility to becoming trapped in local optima. Consequently, it is possibly not the best technique to adopt for sandwich design. Also, owing to it widespread use, the GA has been the subject of more development than any of the other techniques. Hence, the GA is arguably at its most optimal and offers little room for further improvement. If the further potential of each technique was analysed on this fact, greater advances could be anticipated from those techniques that are more recent. This means greater mileage may be found spending energies on more up-and-coming areas of optimisation; areas where a bigger impact can be made. Due to this, it has been decided that the GA will not be developed for sandwich optimisation here. However, the GA will not be avoided all together. Some important features first developed using GAs will be utilised for the later development of the algorithms in this thesis. These will be shown in the next chapter.

Due to its relatively recent development, it seems clear that the potential of the PSO has not yet been fully realised. In applications where it has already been tried out, it has proven to be robust and efficient. In addition, few examples of its application to engineering design exist. Hence, good cause is given to exploit the technique for this purpose.

The ACO was developed at a similar time to the PSO and both are fairly modern in comparison to either SA or the GA. However, unlike PSO, several considerably different

variations have already been developed for case specific problems. Due to its expanding success in many different optimisation applications, ACO is seen as a favourable technique to implement for sandwich optimisation. Indeed, due to the increased use of the technique in recent years, this opinion is also expressed by Dorigo and Blum [61] who state that "the field of ACO is flourishing".

From the evidence provided, SA has shown to be a competitive global optimisation technique. In addition, despite it being developed around a similar time to the GA, it has seen comparatively limited use. Hence, it seems natural to take advantage of this aspect and develop the technique further for sandwich optimisation.

Although it has been used as a global optimiser [58], the role of the TS algorithm seems more appropriate as a supporting (local search) technique in combination with another to form a hybrid. With regard to this and other hybrid methods, the instances they have been applied to have shown successful application. However, their main advantage is simply that they have a faster convergence than other techniques [62]. They do not necessarily show any notable searching ability. So, while single point methods still carry the more inherent disadvantages mentioned earlier, the need for implementing a hybrid will only be necessary if none of the other techniques, on their own, presents any useful application to sandwich optimisation.

Regarding the SBG technique, while it has shown some successful application, its relevance to the type of sandwich design required here is somewhat distanced. This is because unlike the other methods, SBG is only concerned with optimising geometric size and shape [63, 64]. As such, it is not very adaptable for the sandwich purpose. So no attempt to implement this technique will be made.

3.9 Conclusions

In this chapter, a wide range of optimisation techniques have been investigated for their suitability for sandwich design. In addition, many methods that are currently used for the

task have been established. While a number of different perspectives to current approaches have been taken, there appears to be significantly more potential available with numerical methods over analytical techniques. Due to this, a broad range of numerical optimisation techniques have been investigated. Of those described, population-based methods are particularly well suited as they are the most capable of dealing with many of the complexities mentioned in Chapter 2. In addition, even when faced with multiple parameters, little knowledge of the problem needs to be known for multiple non-dominated solutions to be found. From those described, three techniques have been identified as the most promising in terms of benefit that could be obtained. These are particle swarm optimisation (PSO), ant colony optimisation (ACO) and simulated annealing (SA). The next chapter will see a detailed investigation of each of these techniques. While a significant proportion of literature comments on their successful extension to multiple objective problems, and several cases appear where they have been used to optimise laminated composites, it is clear that few applications of these techniques to sandwich design exist. Furthermore, none exist which consider the optimisation of both sandwich materials and structures to the extent considered in this thesis. Therefore, good cause is given to pursue each of the three techniques and develop them specifically for this purpose.

3.10 References

1. Steeves, C.A., Fleck, N. A. (2004) Material selection in sandwich beam construction. *Scripta Materialia* **50**, 1335-1339.
2. Gibson, L.J., Ashby, M.F. (1988) *Cellular solids: structure and properties*. Pergamon Press, Oxford.
3. Pflug, J., Verpoest, I. (2006) Sandwich materials selection charts. *Journal of Sandwich Structures and Materials* **8**.
4. Ashby, M.F. (2005) *Materials selection in mechanical design*. Elsevier Butterworth-Heinemann, Italy.
5. Ashby, M.F. (2000) Multi-objective optimization in material design and selection. *Acta Materialia* **48**, 359-369.

6. Ashby, M.F., Brechet, Y.J.M. (2003) Designing hybrid materials. *Acta Materialia* **51**, 5801-5821.
7. Deb, K. (2001) *Multi-objective optimization using evolutionary algorithms*. John Wiley and Sons Ltd, Chichester.
8. Saravanan, R. (2006) *Manufacturing optimization through intelligent techniques*. CRC Press, London.
9. Di Sciuva, M., Gherlone, M., Lomario, D. (2003) Multiconstrained optimization of laminated and sandwich plates using evolutionary algorithms and higher-order plate theories. *Composite Structures* **59**, 149-154.
10. Dixon, L.C.W. (1972) *Nonlinear Optimisation*. The English Universities Press Ltd, London.
11. Cooper, L., Steinberg, D. (1970) *Introduction to methods of optimization*. W. B. Saunders Company, London.
12. Box, M.J., Davies, D., Swann, W.H. (1969) *Non-Linear Optimization Techniques*. Oliver and Boyd Ltd, Edinburgh.
13. Haataja, J. (1999) Using genetic algorithms for optimization: Technology transfer in action. *Evolutionary Algorithms in Engineering and Computer Science* (K. Miettinen, P. Neittaanmaki, M. M. Makela and J. Periaux, eds), John Wiley and Sons Ltd, Chichester.
14. Spendley, W., Hext, G.R., Himsworth, F.R. (1962) Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics* **4**, 441-461.
15. Nelder, J.A., Mead, R. (1965) A simplex method for function minimization. *The Computer Journal* **7**, 308-313.
16. Box, M.J. (1965) A new method of constrained optimization and a comparison with other methods. *The Computer Journal* **8**, 42-52.
17. Kunzl, P.H., Tzschach, G.H., Zehnder, C.A. (1968) *Numerical methods of mathematical optimization*. Academic Press Inc., London.
18. Powell, M.J.D. (1965) A method for minimizing a sum of squares of non-linear functions without calculating derivatives. *The Computer Journal* **7**, 303-307.
19. Hooke, R., Jeeves, T.A. (1961) Direct search solution of numerical and statistical problems. *Journal of the Association for Computing Machinery* **8**, 212-229.
20. Rosenbrock, H.H. (1960) An automatic method for finding the greatest or least value of a function. *Computer Journal* **3**, 175-184.

21. Srinivas, N., Deb, K. (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* **2**, 221-248.
22. Makris, S.E., Dym, C.L., Smith, J.M. (1986) Transmission loss optimization in acoustic sandwich panels. *The Journal of the Acoustical Society of America* **79**, 1833-1843.
23. Schaffer, J.D. (1984) *Some experiments in machine learning using vector evaluated genetic algorithms (artificial intelligence, optimization, adaptation, pattern recognition)*. Vanderbilt University, Nashville.
24. Goldberg, D.E. (1989) *Genetic algorithms for search, optimization, and machine Learning*. Addison-Wesley, Wokingham, England.
25. De Jong, K. (1999) Evolutionary Computation: Recent Developements and Open Issues. In *Evolutionary Algorithms in Engineering and Computer Science* (K. Miettinen, P. Neittaanmaki, M. M. Makela and J. Periaux, eds). John Wiley and Sons Ltd, Chichester, 231-258.
26. Kennedy, J., Eberhart, R. (1995) Particle swarm optimization. *IEEE International Conference on Neural Networks - Conference Proceedings*.
27. Dorigo, M., Maniezzo, V., Colorni, A. (1996) Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **26**, 29-41.
28. Kirkpatrick, S., Gelatt Jr., C. D., Vecchi, P. M., (1983) Optimization by simulated annealing. *Science* **220**, 498-516.
29. Cerny, V. (1985) Thermodynamical approach to the traveling salesman problem - an efficient simulation algorithm. *Journal of Optimization Theory and Applications* **45**, 41-51.
30. Glover, F. (1989) Tabu search - part II. *ORSA Journal on Computing* **1**, 4-32.
31. Mattheck, C., Burkhard, S. (1990) A new method of structural shape optimization based on biological growth. *Inernational. Journal of Fatigue* **12**, 185-190.
32. Mattheck, C. (2006) Teacher tree: The evolution of notch shape optimization from complex to simple. *Engineering Fracture Mechanics* **73**, 1732-1742.
33. Erdal, O., Sonmez, F.O. (2005) Optimum design of composite laminates for maximum buckling load capacity using simulated annealing. *Composite Structures* **71**, 45-52.

34. Aymerich, F., Serra, M. (2008) Optimization of laminate stacking sequence for maximum buckling load using the ant colony optimization (ACO) metaheuristic. *Composites Part A: Applied Science and Manufacturing* **39**, 262-272.
35. Suresh, S., Sujit, P.B., Rao, A.K. (2006) Particle swarm optimization approach for multi-objective composite box-beam design. *Composite Structures* **81**, 598-605.
36. Kathiravan, R., Ganguli, R. (2006) Strength design of composite beam using gradient and particle swarm optimization. *Composite Structures* **81**, 471-479.
37. Tan, X.H., Soh, A.K. (2007) Multi-objective optimization of the sandwich panels with prismatic cores using genetic algorithms. *International Journal of Solids and Structures* **44**, 5466-5480.
38. Bassetti, D., Brechet, Y., Heiberg, G., Lingorski, I., Pechambert, P. (1997) Genetic algorithm and performance indices applied to optimal design of sandwich structures. *Mechanics of Sandwich Structures*.
39. Gantovnik, V.B., Gurdal, Z., Watson, L.T. (2002) A genetic algorithm with memory for optimal design of laminated sandwich composite panels. *Composite Structures* **58**, 513-520.
40. Wang, T., Li, S., Nutt, S.R. (2009) Optimal design of acoustical sandwich panels with a genetic algorithm. *Applied Acoustics* **70**, 416-425.
41. Kovacs, G., Groenwold, A.A., Jarmai, K., Farkas, J. (2004) Analysis and optimum design of fibre-reinforced composite structures. *Structural and Multidisciplinary Optimization* **28**, 170-179.
42. Coello Coello, C.A., Pulido, G.T., Lechuga, M.S. (2004) Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation* **8**, 256-279.
43. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**, 182-197.
44. Knowles, J.D., Corne, D.W. (2000) Approximating the nondominated front using the Pareto Archived Evolution Strategy. *Evolutionary Computation* **8**, 149-172.
45. Deb, K. (1999) Multi-objective genetic algorithms: problem difficulties and construction of test problems. *Evolutionary computation* **7**, 205-230.
46. Garcia-Martinez, C., Cordon, O., Herrera, F. (2007) A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research* **180**, 116-148.

47. Jaszkiewicz, A. (2002) Genetic local search for multi-objective combinatorial optimization. *European Journal of Operational Research* **137**, 50-71.
48. Zitzler, E., Deb, K., Thiele, L. (2000) Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation* **8**, 173-195.
49. Zheng, L.-G., Zhou, H., Cen, K.-F., Wang, C.-L. (2009) A comparative study of optimization algorithms for low NOx combustion modification at a coal-fired utility boiler. *Expert Systems with Applications* **36**, 2780-2793.
50. Dong, Y., Tang, J., Xu, B., Wang, D. (2005) An application of swarm optimization to nonlinear programming. *Computers & Mathematics with Applications* **49**, 1655-1668.
51. Fonseca, C.M., Fleming, P.J. (1995) An overview of evolutionary algorithms in multi-objective optimization. *Evolutionary Computation* **3**, 1-16.
52. Goldberg, D.E., Segrest, P. (1987) Finite Markov chain analysis of genetic algorithms. *Proceedings of the Second International Conference on Genetic Algorithms*, 1-8.
53. Chen, Y., Hu, J., Hirasawa, K., Yu, S. (2008) Solving deceptive problems using a genetic algorithm with reserve selection. *IEEE Congress on Evolutionary Computation CEC 2008*.
54. Goldberg, D.E., Richardson, J. (1987) Genetic algorithms with sharing for multimodal function optimization. *Proceedings of the Second International Conference on Genetic Algorithms*, 41-49.
55. Machado, J.M., Shiyou, Y., Ho, S.L., Peihong, N. (2001) A common tabu search algorithm for the global optimization of engineering problems. *Computer Methods in Applied Mechanics and Engineering* **190**, 3501-3510.
56. Youssef, H., Sait, S.M., Adiche, H. (2001) Evolutionary algorithms, simulated annealing and tabu search: a comparative study. *Engineering Applications of Artificial Intelligence* **14**, 167-181.
57. Smaili, A., Diab, N. (2007) Optimum synthesis of hybrid-task mechanisms using ant-gradient search method. *Mechanism and Machine Theory* **42**, 115-130.
58. Ahmad, A.S., Nadim, A.D., Naji, A.A. (2005) Optimum synthesis of mechanisms using tabu-gradient search algorithm. *Journal of Mechanical Design* **127**, 917-923.
59. Praveen, K., Sanjoy, D., Stephen, M.W. (2007) Multi-objective hybrid PSO using (mu)-fuzzy dominance. *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation* ACM. London, England.

60. Jeon, Y.-J., Kim, J.-C. (2004) Application of simulated annealing and tabu search for loss minimization in distribution systems. *International Journal of Electrical Power & Energy Systems* **26**, 9-18.

61. Dorigo, M., Blum, C. (2005) Ant colony optimization theory: A survey. *Theoretical Computer Science* **344**, 243-278.

62. Dulikravich, G.S., Martin, B.H., Dennis, B.H., Foster, N.F. (1999) Multidisciplinary hybrid constrained GA optimization. *Evolutionary Algorithms in Engineering and Computer Science* (K. Miettinen, P. Neittaanmaki, M. M. Makela and J. Periaux, eds). John Wiley and Sons Ltd, Chichester, 233-258.

63. Steven, G., Querin, O., Xie, M. (2000) Evolutionary structural optimisation (ESO) for combined topology and size optimisation of discrete structures. *Computer Methods in Applied Mechanics and Engineering* **188**, 743-754.

64. Das, R., Jones, R., Peng, D. (2006) Optimisation of damage tolerant structures using a 3D biological algorithm. *Engineering Failure Analysis* **13**, 362-379.

4 Implementing a successful algorithm

In the previous chapter, a wide range of optimisation techniques were reviewed for their suitability for sandwich design. Population-based techniques were identified as offering the best potential. In particular, it was decided that particle swarm optimisation (PSO), ant colony optimisation (ACO) and simulated annealing (SA) will be utilised. They are effective at finding Pareto-optimal solutions to multi-dimensional problems even when the design space is not well understood. Analytical methods were found to be unsuitable because they generally rely on narrowing down an exhaustive set of material options so that a decision can be made between a manageable few. Single point numerical methods were disregarded as their success is heavily problem dependent. Also, they commonly require an intimate knowledge of the problem, especially for multi-dimensional problems that have many local optima.

Each of the population-based techniques will now be developed specifically for sandwich design. However, before the specifics of each technique are investigated, the aspects which accompany them first need to be addressed. This involves the detail of all the supporting features. For instance, determining what method should be used to handle multiple objectives, how the constraints are negotiated, and other factors to ensure optimal solutions are of the best quality. The best option in each case will be selected, and in some cases developed further.

4.1 Handling multiple objectives

It is now understood that problems with multiple objectives are more complex than their single objective counterparts. Greater care in the approach to handling them is needed. Several different ways this can be done exist. Some of the more common will be described here. The best approach to advocate for sandwich design will then be highlighted. But before this is done, an appreciation of some classifications that have been proposed for multiple objective handling will be outlined.

4.1.1 Multiple objective handling classifications

Due to the many methods of handling multiple objectives that exist, a number of attempts to classify the various types have been made [1, 2]. What is common to these is that deciphering suitable characteristics on which to classify has, to an extent, been problematic. This is because there is no characteristic which completely distinguishes between them. Often, a method will belong to more than one category.

Probably the most recognised categorisation is that of Miettinen [1] and is the only categorisation explained here. This is based on the way *preference* is managed throughout the process. Preference refers to a decision-maker's opinion concerning anticipated points in the objective space. Any influence the decision-maker has before, during or after an optimisation process is a form of preference. However, while this categorisation is largely based on the entire optimisation process, to an extent, it also separates out the different ways of obtaining optimal solutions. It is this second aspect that is drawn upon here. The objective handling methods are broken down into four groups:

- No-preference
- A priori
- A posteriori
- Interactive methods

No-preference methods do not take the opinions of the decision maker into consideration. The problem is solved using some relatively simple method. The solution obtained is presented to the decision maker who may accept it or reject it. These methods are only acceptable when no special requirement about the chosen optimum is needed as little control over the optimal solution can be given.

For a priori methods, the decision maker must specify their preferences before the simulation is carried out. This causes difficulty if it is not known beforehand what solutions are possible or how realistic the expectations are.

In contrast, a posteriori methods primarily lead to the development of a Pareto-optimal set. This allows the decision-maker to select a design from a list of preferred alternatives. However, the downside is that the process can be computationally expensive and the obtained optimal set may contain too many alternatives to choose from.

Interactive methods allow the decision-maker to correct their preferences and selections during the simulation process. This means that little information needs to be known about the problem to obtain satisfactory results. In addition, only part of the Pareto-optimal set needs to be generated as the decision-maker can specify preferences during the simulation to direct the search. However, despite this, problems arise with knowing what kind of data should be used to interact with the decision-maker at each step in the process. This requires a detailed knowledge of the problem.

Explaining this categorisation of multiple objective handling gives sufficient appreciation of the wider field. With this in mind, some of the more common methods of collecting optimal solution for multiple objective problems can now be described.

4.1.2 Weighted sum method

Probably the most common example of multiple objective handling is the *weighted sum* method [3]. Each objective, u, of the problem is aggregated or combined together to form a single overall objective, f'. A general equation for the weighted sum method may be written as:

$$f' = \sum_{u=1}^{U} W_u f_u \tag{4.1}$$

Often, the objectives are *normalised* in advance (see section 3.4.1) to ensure they are of similar scale. Weighting factors, W, are applied to each of the objectives to reflect the relative importance of each. This creates a preferential search direction which forces the search to favour the solutions with good objective value in relation to their weights. If, for instance, there were two objectives of equal weighting priority, Figure 4.1a shows how the optimum point is selected. The dotted lines 1, 2 and 3 represent the contour of the combined objective function. The gradient of the contour depends on the relative weighting of the objectives. The effect of lowering the contour line from 1 to 2 is, in essence, jumping from solutions of a higher value in f' to a lower one.

The more obvious disadvantage with many of these methods is due to the formation of the single optimising function. This means that only one optimal solution can be obtained as opposed to a non-dominated set of solutions (a priori). If a non-dominated set of solutions are required, (a posteriori) many runs need to be performed whilst systematically altering the weights to find the trade-off boundary. Or, the weights may be altered by the user after each iteration (interactive). In either of the latter two cases, the strategy may also be regarded by Fonseca and Fleming [2] as an aggregated method, and a population-based non-Pareto approach. However, not only can either of these operations be time consuming, the weighted sum method is unable to identify non-convex Pareto-optimal fronts. This is shown in Figure 4.1b. By altering the objective weight values gradually so that the gradient of the contour moves from 4 to 5, or 6 to 5, any solution in the non-convex region cannot be detected. This is because before the line forms a tangent with any point between solutions φ_2 - φ_3, it also becomes a tangent at another better (with smaller f') point (either φ_1, or φ_4) in the objective space. Since it is, in general, difficult to know whether the resulting objective space is non-convex, the weighted sum method must be applied cautiously.

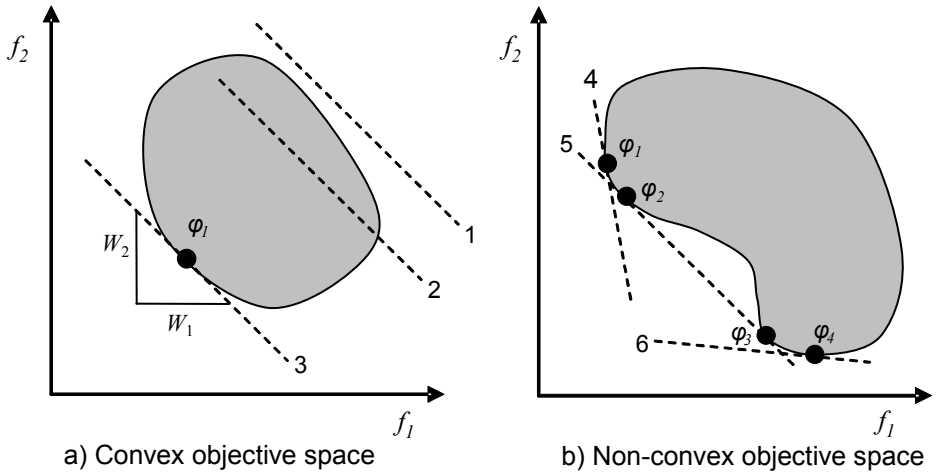

Figure 4.1. Shows how the weighted sum method generates an objective contour from which the best solution is obtained for convex (a) and non-convex (b) objective spaces.

4.1.3 ε-constraint method

Another way to approach multiple objective handling using a single criterion is via the ε-constraint method. The basic idea is that all objectives except one are turned into constraints. The boundaries of the newly formed constraints are defined by the user who has a predefined idea about the nature of the problem (a priori). The lone objective is then optimised with regard to all constrained objectives. This approach was introduced by Haimes et al [4] to alleviate some of the difficulties with the weighted sum method in dealing with non-convex objective spaces. However, as with the weighted sum method, an element of pre-defined knowledge of the problem is required in order to form accurate constraint boundaries. In addition, if a non-dominated set is required (a posteriori), multiple optimisation runs need to be performed using different constraint limits on the objectives. Figure 4.2 shows an objective minimising case where f_1 is minimised and f_2 has an associated constraint value. Values $\varepsilon_1 - \varepsilon_3$ show different constraint values for f_2. Everything above the dashed line is infeasible. Allowing several constraint values between $\varepsilon_1 - \varepsilon_3$ allows the non-convex part of the graph to be uncovered. Consider the constraint is at ε_2 say (in the non-convex region). Constraining the objective space here means only part of the Pareto-optimal front is feasible. This allows everything up to point φ_2 to be identified.

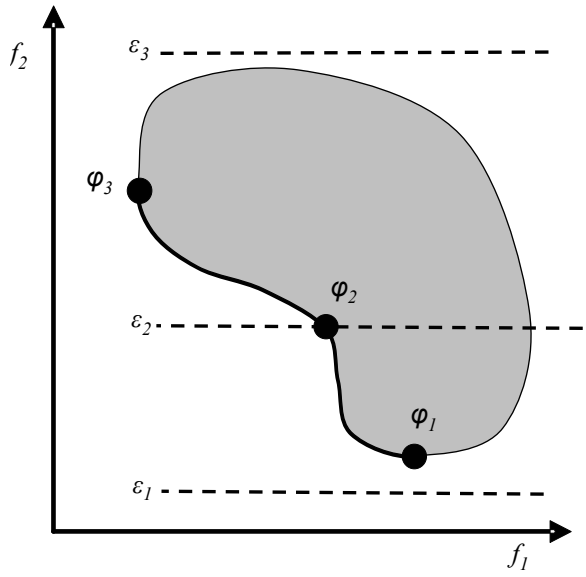

Figure 4.2. Shows how the ε-constraint method can be used to obtain non-convex Pareto-optimal fronts.

4.1.4 Global criterion

In this method, an infeasible reference or target point is first selected. The optimal solution is then classed as the closest feasible solution to the target value. This is done by obtaining the objective differences between the target and each trial solution. The sum of this is then minimised. Commonly, all objectives are equally important. This technique is only acceptable when the user does not have any special expectations of the chosen solution (no-preference). This is because unless the topography of the search space is well understood, the closest point to the target value cannot be known beforehand. Consequently, the selected optimal solution may not best suit the decision-maker. Also, if the target point is pessimistic and better solutions exist, these will not be selected.

4.1.5 Goal programming

Goal programming was first introduced by Charnes et al [5] in 1955. For the general technique, the decision-maker must specify aspiration levels (or goals) for each of the objective functions (a priori). Ideally, the aspiration levels are selected so that they are achievable, but not all simultaneously. Commonly, it is the sum of the deviations from each goal which is minimised. In this sense, goal programming is similar to the global

criterion method. However, each goal value is considered separately, rather than closing-in on a single point. This instead forms a target region (Figure 4.3). If a non-dominated set of solutions is required, a weighted sum similar to section 4.1.2 (called weighted goal programming) may be used, but carries similar disadvantages [3]. But for this case, the objective deviations are minimised instead of the objectives themselves. In Figure 4.3 equal weighted priority of objectives would result in solution φ_1 being optimal, solutions φ_2 and φ_3 are optimal when absolute priority is given to objective f_1 and f_2 respectively.

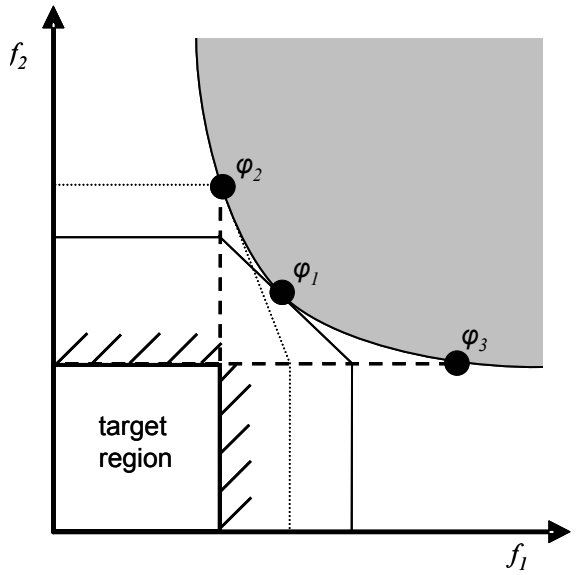

Figure 4.3. The target region that is formed when each objective goal is set using goal programming. Multiple non-dominated points (e.g. φ_1 - φ_3) are found by altering weighted priority of deviations.

4.1.6 Lexicographic ordering

With Lexicographic ordering, the objectives have to be first arranged in order of absolute importance. This means that a more important objective is infinitely more important than a less important objective. A given set of solutions are initially ranked based on the most important objective. If more that one solution is optimal at this stage, the best are then ranked using the next most important objective. This is repeated until only one solution remains or all objectives have been considered. Not only does this method require the user to place absolute priority of one objective over another (a priori), but it is largely used to obtain a single optimal solution as opposed to a non-dominated set.

4.1.7 The concept of domination

In chapter 2, it was introduced that multiple conflicting objectives lead to the generation of an optimal set of solutions as opposed to a single optimum. Here, it will be discussed how this can be used more explicitly as an objective handling method. To distinguish it as such, it will be referred to in this thesis as the concept of domination. Unlike the other methods described so far, it was developed more specifically for handling multiple objectives. Instead of rating the importance of each objective to focus the problem towards a single optimum, it considers all objectives separately (and equally) in their own right. It operates by making comparisons between all generated solutions and maintaining a record of the "best" found. Comparing any two solutions leads to three possible outcomes. If at least one objective function is better, but none are worse, the solution is *superior* to its comparator. It dominates. On the other hand, if at least one is worse, but none are better, the solution is *inferior*. The third instance is entirely specific to multiple objectives. This is when the solutions are *non-dominated*. A non-dominated solution, ψ, in a set, Ψ, is one which, when compared to the others, shows superior quality in at least one objective function, or is no worse in value across all objective functions. Neither solution dominates.

When the comparisons are performed, superior solutions are always favoured. However, once the limit of the trade-off boundary is reached, solutions on the trade-off will only be non-dominated to their peers. Hence, it is these solutions that are obtained and lead to the development of a non-dominated set.

This is illustrated in Figure 4.4. Solutions $\varphi_1 - \varphi_4$ lay on the trade-off boundary and represent a non-dominated set in which two objectives (f_1 and f_2) are to be minimised. If for instance solution φ_3 is compared to solution φ_6, clearly, both objectives of solution φ_3 are better. Therefore, solution φ_3 dominates solution φ_6. If φ_3 was compared with φ_5, φ_3 is better with respect to objective f_1, but equal to f_2. So in this case φ_3 is superior. However, if φ_1 is compared with φ_2, while φ_1 is better in f_1, φ_2 is better in f_2. Neither solution dominated the other. Hence, both solutions appear in the non-dominated set.

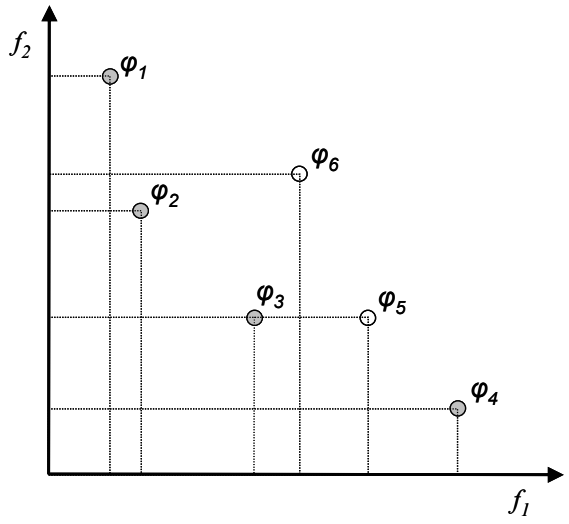

Figure 4.4. Shows how a non-dominated set of solutions is depicted for an objective minimising problem.

4.1.8 The chosen objective handling method

Common methods for obtaining optimal solutions to multiple objective problems have been described. Interestingly, most of these were originally developed for single objective optimisation. But for one, this is not the case. This method is the concept of domination and is specific to multiple objectives. It inherently operates on the basis of finding a non-dominated set of solutions as opposed to a single optimum, and it is able to identify the entire trade-off surface in a single optimisation run. Furthermore, it is easy to apply even if the search space is not well understood. The concept of domination also allows many disadvantages that occur with other methods to be avoided. For instance, other methods require weighting parameters to be set by the user which can be difficult. Also, some are not appropriate if the Pareto-optimal front is non-convex, noisy or discontinuous [6].

In addition, many of the other methods originally needed to be modified to accommodate multiple objectives. Historically, they were born out of a lack of suitable optimisation processes for the task. They offered a way around the problem, or mediocre fix. This fact is highlighted by Deb [3] who states that:

> "The majority of these methods avoid the complexities involved in a true multi-objective optimisation problem and transform multiple objectives into a single objective function."

Due to the argument presented, an objective handling method based on the concept of domination will be used for the algorithms in this thesis.

An important point to note is that whilst the various optimisation algorithms discussed here attempt to identify the non-dominated (i.e. best) set of solutions, it cannot be absolutely known whether the set that they generate does indeed match the true Pareto-optimal set to the problem. In many cases, this can only guaranteed if a complete exhaustive search of the design space is conducted. Due to this, the best *known* set of non-dominated solutions that can be obtained will be accepted as the true Pareto-optimal set.

4.2 Obtaining a non-dominated set

It is now clear that an objective handling process based on acquiring a non-dominated set via the concept of domination will be followed. However, several alternative methods of acquiring the non-dominated set exist. In this section, an overview of an early attempt to acquire a non-dominated set via the concept of domination will be given. Thereafter, two recent techniques will be described. One of which will be adopted for this thesis.

4.2.1 Origins of the concept of domination

The early advances with the concept of domination stemmed from Goldberg [7]. At the time, this involved the development of a revolutionary non-dominated sorting procedure. It worked by firstly obtaining the non-dominated solutions, Ψ, from a general set, Φ, using the definition of Pareto-optimality given earlier (section 2.2). Once obtained, solutions in this set were given a grade of 1, removed from the set, and placed in a separate repository. For the solutions left over, the non-domination check was carried out again. This second set of non-dominated solutions were given a grade of 2 and placed in another repository. The process was repeated until there were either no more solutions to sort, or the number of required repositories had been accounted for. In summary, the process involved

progressively flagging and removing solutions of subsequent non-dominated layers from the population.

Despite this description, no indication of how to implement the procedure was given. Hence, researchers in the field were left to come up with their own implementations [3, 8]. Among the strategies created, the relative merits differ. Some approaches are able to better represent the Pareto-optimal front depending upon its shape. Also, the method used has significant implications over the computational efficiency of the computer program.

In the following section, the non-dominated sorting procedure of Deb et al [8] is described. Afterwards, a procedure described by Fonseca and Fleming [9] will be outlined. The most suitable method will then be highlighted.

4.2.2 Deb et al's non-dominated sorting procedure

To perform the non-domination check as described by Deb et al [8], each trial solution, φ_a, from the general set, Φ, must be compared with every other solution in that set, φ_b. When the comparisons are made, two entries are sought for each trial solution. Firstly, the number of solutions that dominate the trial is obtained. This is termed the *dom count*. The second entry is a matrix containing all the solutions that the trial dominates. Here, this has been termed the *inferior set*.

All solutions in the first non-dominated front have a *dom count* of zero; no solutions dominate them. These are removed from the set and stored in a separate repository (rep_1). Once complete, each solution in the repository then has their *inferior set* consulted. Solutions contained in these inferior sets have their *dom count* reduced by one (*dom count* – 1). If by doing this, a *dom count* is reduced to zero, then the solution is stored in another repository (rep_2). These belong to the second non-dominated front. The process is repeated until all subsequent fronts are identified. Figure 4.5 shows the grades given to a representative set of solutions if the procedure were applied. A pseudo-code for the process is given in Figure 4.6.

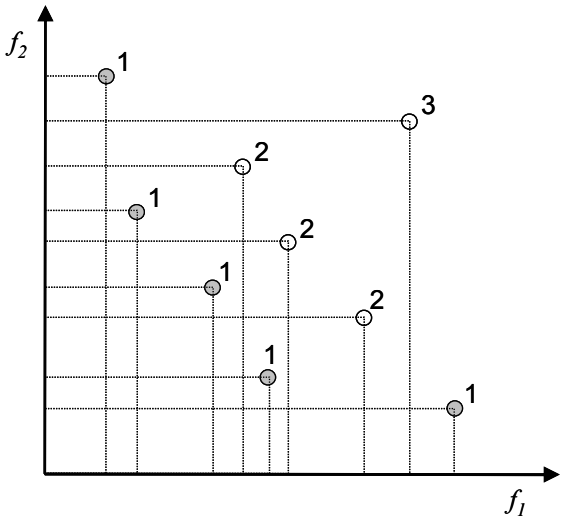

Figure 4.5. An objective minimising problem. Solutions are given grades based on which non-dominated front they appear in.

```
for each solution, φ_a in set Φ
dom count_a = 0
inferior set_a = [ ]
front = 1
rep_front = [ ]

   for each solution, φ_b in set Φ
      if φ_b dominates φ_a
      dom count_a = dom count_a + 1
      elseif φ_a dominates φ_b
      inferior set_a = inferior set_a + φ_b
      end
   end

   if dom count_a = 0
   rep_front = rep_front + φ_a
  end
end

**to find subsequent optimal fronts**

while number of fronts is not reached
   rep_temp = [ ]
   for each solution, φ_a, in rep_front
      for each solution, φ_b, in inferior set_a
      dom count_b = dom count_b − 1
         if dom count_b = 0
          rep_temp = rep_temp + φ_b
         end
      end
   end
   front = front + 1
   rep_front = rep_temp
end
```

Figure 4.6. Pseudo-code for the adopted non-dominated sort procedure [8].

4.2.3 Fonseca and Fleming's Pareto ranking

An alternative non-dominated sorting procedure for categorising solutions into optimal ranks was suggested by Fonseca and Fleming [9]. However, the difference was that instead of producing several non-dominated sets, each solution was simply given a rank according to the number of solutions that dominated it (Figure 4.7).

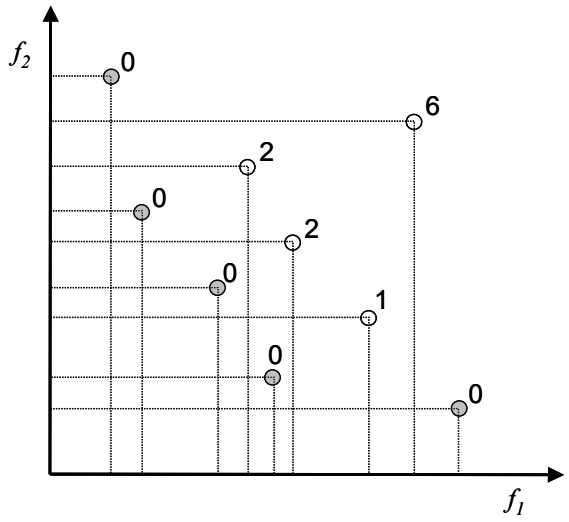

Figure 4.7. An objective minimising problem. This shows the Pareto rank of each solution in terms of how many solutions dominate it.

A disadvantage of this method is that if convex Pareto-optimal fronts are present (Figure 4.8), intermediate solutions (white dot) dominate a greater region of the objective space than those at the extremes (black dot). While this has no direct impact on the solutions collected, the algorithms may show a bias towards intermediate solutions [10]. Alternatively, it may also be said that Pareto ranking is blind to the convexity or non-convexity of the trade off surface [2].

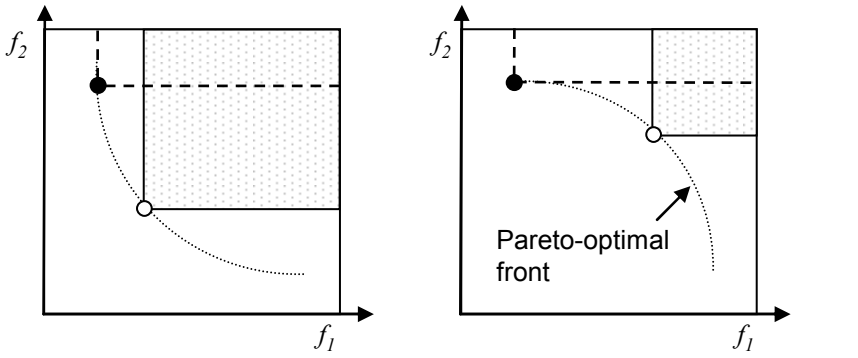

Figure 4.8. For convex Pareto-optimal fronts (a), intermediate solutions (white) dominate larger areas of the objective space than those at the extremes (black). This does not occur in the non-convex case (b).

4.2.4 The chosen procedure for obtaining a non-dominated set

For this thesis, the non-dominated sorting procedure developed by Deb et al [8] will be employed to perform the task as it is fast, efficient, and parameterless. Fonseca and Fleming's Pareto ranking procedure was not selected due to the biasing it shows when convex Pareto-optimal fronts are present. It should be noted that while multiple non-dominated grades may be obtained using Deb et al's [8] approach, only the first non-dominated grade is actually required for the problems detailed in later chapters.

4.3 Diversity preservation

It was mentioned in section 2.4 that the second aim in obtaining the ideal optimal set was to find a set of solutions as diverse as possible. The method used to achieve this forms the topic of discussion here. The need to promote diversity when population-based methods acquire a non-dominated set has been acknowledged [3, 11]. Without it, the collected non-dominated set would likely bunch-up, be unevenly distributed and unlikely to be spread across much of the trade-off boundary. Figure 4.9 gives a graphical representation of these possible alternatives. However, it must be noted that the graph is only a qualitative representation and in many cases it may not be possible to achieve a completely even

distribution (i.e. as shown in Figure 4.9b) by virtue that a solution may simply not exist at any preconceived point in the objective space.

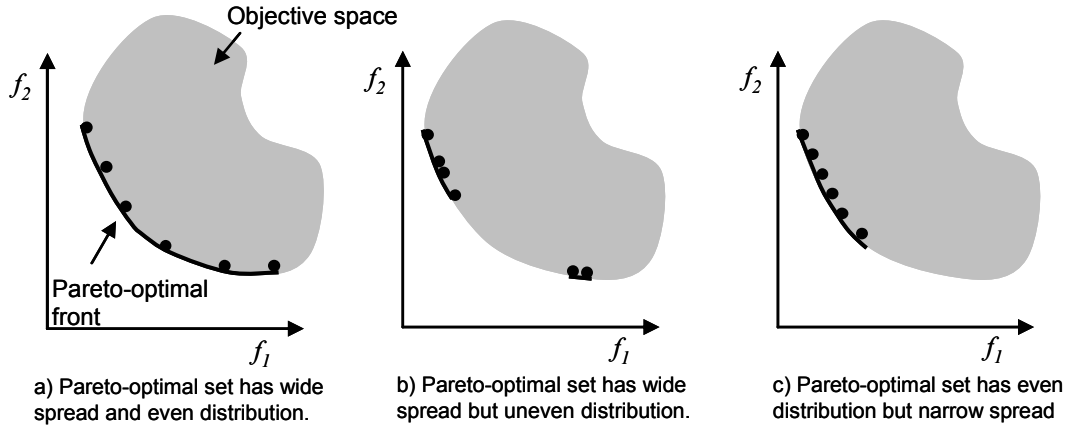

Figure 4.9. Three graphs which demonstrate the extreme cases of how the non-dominated set could develop. The black dots represent particular solutions. Case (a) is the ideal. Cases (b) and (c) are less preferable.

In addition, for a given problem, a great number of non-dominated solutions may be available. While it is beneficial to have a set of optimal solutions to choose from, it is clearly undesirable if there are far too many to consider. This may occur if continuous variables are present, e.g. if any value of facing thickness, or beam length was permitted. An easy way to manage this problem is to cap the size of the non-dominated set to a manageable number. This not only allows effective analysis, but also keeps the computational effort to a reasonable level.

Early involvements of diversity were concerned with how search agents hunted-out new solutions *during* the simulation, rather than with maintaining diversity in a non-dominated set. Most of this research was done on genetic algorithms [3]. A process known as *niching* was used to describe "any method which emphasises solutions corresponding to poorly represented regions in the population." Probably the most well known method of preserving diversity via niching is through the use of a fitness sharing parameter. Fitness sharing was introduced by Goldberg and Richardson [11]. The basic procedure works by directly reducing the fitness (objective value) of a solution in relation to its proximity to the rest of the population. Solutions in comparatively crowded areas have their fitness reduced more than those in less crowded regions. A solution is degraded by dividing each objective function by a *niche count*. The reduced fitness value or *shared fitness* is then

used for comparison to obtain *optimal* solutions. The equation to calculate the shared fitness for each objective, u, of a solution, φ, may be written.

$$shared\ fitness\ = \frac{f_u}{niche\,count} \tag{4.2}$$

Several formulations of the niche count exist but generally it is a measure of how close the rest of the population is to a solution. Individuals that are comparatively more crowded have higher niche counts. Solutions in more densely populated areas are continually replaced with less crowded solutions. The system works to prevent the search agents bunching together.

Since the original technique, Horn et al [12] state that several variations have been implemented to improve its general performance. However, despite notable improvements, specification of a problem dependent sharing parameter is still required. This can affect performance significantly. In addition, because early developments of these operators were conducted on genetic algorithms, their developments have been largely specific to GAs. Not for the more general optimisation technique. Further examples of diversity preservation are available [2, 13, 14]. However, recently, two other methods of maintaining a diverse non-dominated set have been noted as offering good potential. These are based on an adaptive grid approach by Knowles and Corne [15], and a crowding distance operator by Deb et al [8]. They are both discussed below.

4.3.1 Knowles and Corne's adaptive grid approach

Basically, an external repository or archive collects the current best set of non-dominated solutions found during the searching process [15]. The archive itself has a fixed size, and once full, a mechanism to promote diversity within the repository is engaged. This selects the most diverse and ensures a well distributed set of non-dominated solutions is collected.

The adaptive grid works by dividing up the known search space into a number of user defined regions equal in size. These regions may also be termed *hypercubes* due the multi-dimensional nature of a problem. With a full repository, the number of non-dominated

solutions in each region is then kept as low a possible. This is conducted by substituting solutions in crowded areas for those contained in sparse regions. This causes spreading out, and even spacing of solutions in the non-dominated set.

4.3.2 Deb et al's crowding distance operator

Initially, the non-dominated solution set is given an allowable limit defined by the decision-maker [8]. When this limit is exceeded, the crowding distance operator is initiated. The now oversized non-dominated set is sorted in ascending order of magnitude for each objective function value in turn. Each solution lying on the boundary (i.e. the maximum and minimum values) is assigned an infinite *crowding* distance. For all intermediate solutions, the crowding distance needs to be calculated. But before this is done, each objective function is first *normalised* (see section 3.4.1) and the absolute value is taken. Using these values, the crowding distance for each solution is calculated as the Euclidean distance between neighbouring solutions (Figure 4.10). Problems with two objectives have two adjacent solutions. For problems with more objectives, this increases.

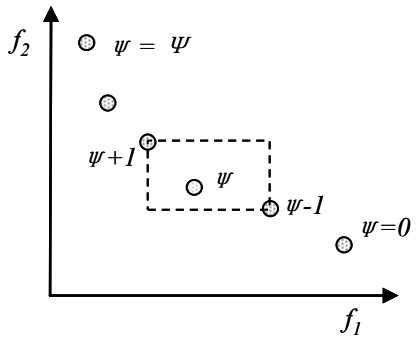

Figure 4.10. Crowding distance operator is calculated for a non-dominated solution, ψ, using its two neighbours.

After all members of the non-dominated set are assigned a crowding distance, solutions are then compared. Larger crowding distances represent less crowded solutions. These are favoured. The non-dominated set is reduced to its limiting value by discarding the most crowded solutions.

4.3.3 The chosen approach to preserving diversity

While successes of both approaches are evident [16, 17], the decision has been made to use the crowding distance operator of Deb et al [8]. This is because the approach is parameterless. So unlike Knowles and Corne's adaptive grid [15], the user escapes the need to specify any values. In addition, it has also been suggested to give better performance [16].

4.4 Constraint handling

When considering the type of constraints that sandwich design may impose, only the dependent type of constraint (section 2.1) needs special consideration here. This is because they split the search space up into feasible and infeasible regions. The amount of infeasibility present in the objective space governs the amount of consideration needed for handling solutions which lie in this region. For instance, if few constraints are present and the majority of the search space is feasible, an optimiser is unlikely to have difficulty finding feasible solutions. In which case, any infeasible solutions may simply be ignored. However, what is more likely in sandwich design is that several constraints will be present. Not only that, but due to the complex, multi-dimensional nature of the problem, finding feasible solutions will be a much more challenging task. Therefore, a careful and thought-out approach is needed.

Different categorisations of constraint handling processes exist. While Michalewicz et al [18] provide a categorisation more specific to genetic algorithms. More recently, Coello Coello [19] provides a general, but detailed survey of constraint handling methods. Several favourable methods have been considered here and will now be explained with the relative strengths of each approach being noted.

4.4.1 Ignoring infeasible solutions

A common and simple way to deal with the problem is simply to ignore any constraint violating solution [20]. This is a quick and easy approach and is most effective for problems which have a large proportion of feasible search space. This is commonly the case when few constraints are present. If a problem becomes more heavily constrained, this generally reduces the size of the feasible region and the chances of finding it. This makes it more difficult for the algorithm to find any optimal solutions, especially if the Pareto-optimal front is discontinuous or contains areas that are non-convex.

4.4.2 Penalty function approach

This is the most common approach to handling constraints and was originally proposed by Courant in the 1940s [21]. Several variants of the penalty function approach exist [22]. However, they generally devalue the quality of the objective values by penalising solutions which violate constraints. This makes previously optimal solutions less favourable by superficially shifting the position of the optimal region. This is useful for more heavily constrained problems. It offers a means of assessing the performance of infeasible designs. So it is able to guide the search towards feasible solutions when none are known. However, the extent to which the optimal region is shifted depends largely on a penalty parameter. This is a user defined parameter that controls the amount of penalty incurred for each constraint which is violated. While a number of strategies and statistical means have been developed to obtain an effective value for any given problem, penalties create inherent difficulties. This is pointed out by Surry et al [23] who state that:

> "there is a wide-spread perception that penalty function methods are a rather blunt instrument for handling general constraints, exhibiting great sensitivity to the values of their many free parameters, and feeding rather too little information back to the algorithm to allow it to handle the constraints satisfactorily."

4.4.3 Non-dominated sorting of constraint violations

Ray et al [24] point out that handling constraints via a non-dominated sorting approach is a relatively new concept due to its origins from multiple objective optimisation. The basic method calculates the amount an infeasible solution violates each constraint and uses this to obtain a non-dominated set. This is then used to guide the search forward.

Surry et al [23] consider this kind of approach for optimising a gas supply network. The constraint violation of each solution was calculated and non-dominated sets were produced and sorted into ranks using the Pareto ranking technique described by Fonseca and Fleming [9] (see section 4.2.3). Solutions with less constraint violation were favoured in future iterations.

A more elaborate method of handling constraints that also used a non-dominated check of the constraint violations was investigated by Ray et al [24]. The process, initially setup for incorporation with a GA, revolved around three separate non-dominated rankings being produced. The first rank used objective value to carry out the non-dominated sorting procedure in the normal way. The second used constraint violation to obtain a non-dominated rank. Solutions with the smallest violations were sought and feasible solutions had zero constraint violation. The third non-dominated rank was performed using the objective and constraint values combined. From these three ranks, solutions were selectively chosen from different ranks to mate with each other in the crossover operator. This allowed infeasible solutions to be still used. However, new solutions were pressurised towards weeding-out the most infeasible and crowded solutions. A noteworthy benefit to this method is that no parameters need to be specified. So it can be used even when little or no information is known about the problem. This is in contrast to the method of Surry et al [23] who, despite remarking on their insensitivity, does require several parameters to be prescribed in their approach.

Another advantage of Ray et al's [24] method of handling constraints is that even when no feasible solutions have been found, the search can still be directed towards the feasible areas. Also, even if no feasible solutions exist, a non-dominated set of the most suitable solutions can still be presented to the user. Furthermore, infeasible solutions do not need to be artificially modified, as with other penalty approaches.

Due to the advantages mentioned, a constraint handling approach that uses a non-dominated set of solutions, based on constraint violation, will be developed here. However, due to their original incorporation with GAs, an element of modification will need to be made to make them transferable to other techniques. Hence, a novel constraint handling method developed specifically for the purpose here is detailed in the next section.

4.4.4 The developed constraint handling approach

To reiterate from the previous section, the developed method uses a non-domination check of the constraint violations [23, 24]. However, unlike Ray et al [24], this procedure is only engaged if no feasible solutions are found. Once a feasible solution exists, the constraint handling approach is no longer used. Also, only the first (constraint violating) non-dominated front is required. The process is described below.

During the first iteration, each particle in the population is randomly assigned a solution. At this stage, the direct constraints ensure that the algorithm selects only solutions which are physically possible. Once all solutions have been selected, the dependent constraints are then calculated. Any which do not satisfy all constraints are flagged as infeasible. If no feasible solutions exist, the constraint handling procedure is induced until a feasible solution is acquired. Each constraint that violates the given limit has the extent of the violation calculated. This is simply the difference between the obtained value and its limit. Solutions are then filtered to find the least infeasible non-dominated set using the procedure described in section 4.2.2. Progress of the algorithm is then made using this non-dominated set. Once a feasible solution does exist, only then do the objectives of the problem need to be calculated. Constraint violation is no longer used after this point and instead the search progresses using objective values in the normal way. A pseudo-code for the method is outlined in Figure 4.11.

```
while no feasible solution exists
   for each solution, φ, in a set Φ
   calculate constraints
   end

   If no feasible solution exists
   use constraint violation to direct the search
  else there is a feasible solution

      for each solution, φ, in a set Φ
      calculate objectives
      end

      use objective value to direct search
   end
end
```

Figure 4.11. Pseudo-code for the developed constraint handling method.

Although relatively simple, this is an effective way of dealing with heavily constrained situations. It can be directly integrated with the existing non-dominated sorting procedure (using objective value), and importantly, does not require any parameters to be defined.

4.5 Proposed structure for implementation

At this point, all aspects subsidiary to the main optimisation technique have now been dealt with. Besides being tailored for sandwich design, implementing a common procedure will allow a more direct comparison to be made between the optimisation techniques in the next chapter. The proposed basic algorithm structure is presented in Figure 4.12.

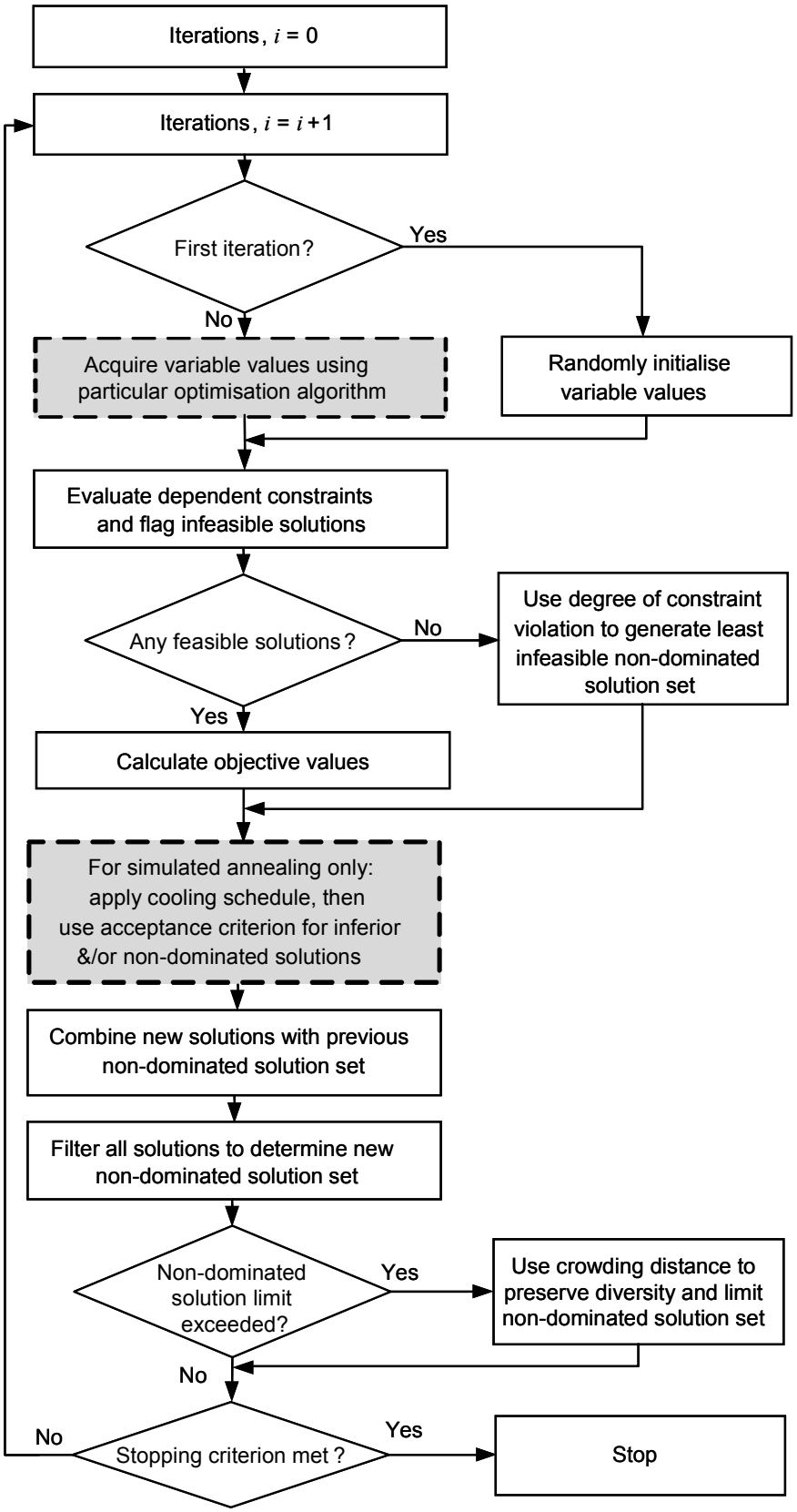

Figure 4.12. A flowchart showing the general structure surrounding each of the algorithms to be implemented.

4.6 Conclusions

To support the development of optimisation techniques for sandwich design, this chapter has addressed the key surrounding features that accompany the process. They form a common platform from which each of the algorithms (PSO, ACO and SA) can be built. In some cases, aspects from previous authors satisfy the requirements and have been utilised directly. This included the collection of a non-dominated set of solutions via the concept of domination, and a crowding distance operator to maintain a well-spread and even set of collected solutions. Both of these are provided by Deb et al [8]. However, a third aspect has led to the development of a novel approach to negotiate dependent constraints. It is a simple parameterless alternative that can direct the search towards feasible regions, even when no feasible solutions are known. Moving on from this, the following three chapters will see a detailed investigation of the optimisation techniques themselves. In each case, a process will be developed that is geared towards the needs of sandwich optimisation.

4.7 References

1. Miettinen, K. (1999) *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, Boston.
2. Fonseca, C.M., Fleming, P.J. (1995) An overview of evolutionary algorithms in multi-objective optimization. *Evolutionary Computation* **3**, 1-16.
3. Deb, K. (2001) *Multi-objective optimization using evolutionary algorithms*. John Wiley and Sons Ltd, Chichester.
4. Haimes, Y.V., Lasdon, L.S., Wismer, D.A. (1971) On a bicriterion formation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man and Cybernetics* **1**, 296-7.
5. Charnes, A., Cooper, W., Ferguson, R. (1955) Optimal estimation of executive compensation by linear programming. *Management Science* **1**, 138-151.

6. Garcia-Martinez, C., Cordon, O., Herrera, F. (2007) A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research* **180**, 116-148.
7. Goldberg, D.E. (1989) *Genetic Algorithms for search, Optimization, and Machine Learning*. Addison-Wesley, Wokingham, England.
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**, 182-197.
9. Fonseca, C.M., Fleming, P.J. (1993) Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. *Proceedings of the Fifth International Conference on Genetic Algorithms*, 416-423.
10. Deb, K. (1999) Multi-objective genetic algorithms: problem difficulties and construction of test problems. *Evolutionary computation* **7**, 205-230.
11. Goldberg, D.E., Richardson, J. (1987) Genetic algorithms with sharing for multimodal function optimization. *Proceedings of the Second International Conference on Genetic Algorithms*, 41-49.
12. Horn, J., Nafpliotis, N., Goldberg, D.E. (1994) Niched Pareto genetic algorithm for multiobjective optimization. *IEEE Conference on Evolutionary Computation - Proceedings*.
13. Singh, G., Deb, K. (2006) Comparison of multi-modal optimization algorithms based on evolutionary algorithms. *Genetic and Evolutionary Computation Conference GECCO*.
14. Sareni, B., Krahenbuhl, L. (1998) Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation* **2**, 97-106.
15. Knowles, J.D., Corne, D.W. (2000) Approximating the nondominated front using the Pareto Archived Evolution Strategy. *Evolutionary computation* **8**, 149-172.
16. Coello Coello, C.A., Pulido, G.T., Lechuga, M.S. (2004) Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation* **8**, 256-279.
17. Reddy, M.J., Kumar, D.N. (2007) An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design. *Engineering Optimization* **39**, 49-68.

18. Michalewicz, Z., Deb, K., Schmidt, M., Stidsen, T. (2000) Test-case generator for nonlinear continuous parameter optimization techniques. *IEEE Transactions on Evolutionary Computation* **4**, 197-214.
19. Coello Coello, C.A. (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering* **191**, 1245-1287.
20. Coello Coello, C.A., Christiansen, A.D. (1999) Moses: A multiobjective optimization tool for engineering design. *Engineering Optimization* **31**, 337-368.
21. Courant, R. (1943) Variational Methods for the Solution of Problems of Equilibrium and Vibrations. *Bulletin of the American Mathematical Society* **49**, 1-23.
22. Smith, A.E., Coit, D.W. (1995) Penalty Functions. *Handbook of Evolutionary Computation* (T. Baeck, D. Fogel and Z. Michalewicz, eds), Oxford University Press, Pittsburgh.
23. Surry, P.D., Radcliffe, N.J., Boyd, I.D. (1995) A multi-objective approach to constrained optimization of gas supply networks. *AISB-95 Workshop on Evolutionary Computing*, 166-180.
24. Ray, T., Tai, K., Seow, K.C. (2001) Multiobjective design optimization by an evolutionary algorithm. *Engineering Optimization* **33**, 399 - 424.

5 Developing particle swarm optimisation (PSO) for sandwich design

In the previous chapter, the surrounding aspects that support each of the developed optimisation techniques were put into place. In particular, a method of collecting optimal solutions was adopted, as well as a mechanism to ensure a wide and even distribution of solutions was maintained. In addition, a novel approach to constraint handling was developed. This had the advantage of being simple to implement yet designed for heavily constrained problems.

In this chapter, the first of the three optimisation techniques to be developed for sandwich design is discussed. This is the particle swarm optimisation (PSO) technique. A detailed analysis is presented in relation to its application for this purpose. Once complete, the developed technique (*sandwichPSO*) will then be described.

5.1 The original PSO algorithm

PSO aims to mimic the social behaviour of flocking birds. A flock of birds (particles) with common objectives (e.g. the best food source or roosting site) is more likely to find good locations (optimum solutions) than a sole agent acting independently.

For the original PSO [1], each bird in the flock is guided by three types of information: the best solution that each individual bird finds, a solution known globally to the whole flock, and the previous motion made by the bird. These three factors are added to each variable of the particle's current position to establish its next move, x_{i+1}, in the variable space. The equations that governed this movement are:

$$x_{i+1} = x_i + v_{i+1} \tag{5.1}$$

where x is the value of each variable for a given particle position, i is the iteration number, and v is change in the particle's position given by:

$$v_{i+1} = v_i + c_1 r_1 \left(x^{personal} - x_i\right) + c_2 r_2 \left(x^{global} - x_i\right) \tag{5.2}$$

The first term on the right hand side of Equation (5.2) represents the influence of a given particle's previous motion (the so-called 'inertial' influence). The second term represents a given particle's knowledge about its own previous best solutions (the 'cognitive' influence). The third and final term represents information sharing with the rest of the swarm as to the global best solutions found so far by any member of the group (the 'social' influence). The parameters c_1 and c_2 are essentially weighting factors for the cognitive and social influences, whereas r_1 and r_2 are random numbers between 0 and 1. However, it wasn't until shortly afterwards when the more recognisable form of Equation (5.2) was developed by Shi and Eberhart [2] with the addition of the inertial weight parameter, w:

$$v_{i+1} = wv_i + c_1 r_1 \left(x^{personal} - x_i\right) + c_2 r_2 \left(x^{global} - x_i\right) \tag{5.3}$$

From a user point of view, the c_1 and c_2 parameters control the amount of preference given to either the personal or global information. Comparatively larger values of the cognitive parameter, c_1, imply that particles concentrate their search more locally. Larger values of the social parameter, c_2, imply the particles concentrate more heavily towards the global solutions common to all particles. The inertial term, w, was introduced to balance the effect of the global and local search parameters. Comparatively higher values of inertial weight imply a greater effect of the previous motion, v_i. This means the particles have a tendency to fly further than expected and concentrate more on exploring the entire solution

space. On the other hand, smaller values imply particles have less momentum. So they search more locally in the regions close-by.

As an additional note, an influence factor called *craziness* was also introduced but later removed from the algorithm as it was found not to make a difference to the searching capability. Simply, craziness was a factor that provided random changes to a particle's motion and provided an additional level of variation into the system.

5.2 Multiple objective PSO strategies

With regard to multiple objective problem solving, the original PSO was not used for this purpose. Currently however, this idea is not unfamiliar. The work of Coello Coello et al [3] and Reddy and Kumar [4] provide examples where recent developments in this direction have been made. Furthermore, not only do their objective handling approaches show large similarity, they themselves describe techniques which meet closely with the subsidiary approach considered in the earlier sections of this chapter. Due to this, both of these methodologies are outlined. Firstly, the method of Coello Coello et al [3] will be described. However, particular attention will be paid to the way multiple objectives are handled, rather than on the actual equations for selecting new moves. After that, the approach by Reddy and Kumar [4] will be described with an appreciation of the former technique. These current examples demonstrate notable advances with PSO for solving multiple objective problems from a general standpoint. As such, they also provide the interested reader with a source for investigating general multiple objective PSO techniques.

Coello Coello et al [3] applied a PSO algorithm to several multiple objective test functions. Comparison against three genetic algorithms (GAs) was conducted, two of which are better known in GA research [5, 6]. The third was developed by Coello Coello and Pulido [7] and termed micro-GA. The core mechanism of the PSO used in their study is analogous with the technique described in Equations (5.1) and (5.3) [2]. However, the c_1 and c_2 terms in this case were equal to 1.

To promote diversity in the non-dominated solution set, the adaptive grid approach [6] (section 4.3.1) was utilised in their study but a crowding distance operator [5] (section 4.3.2) was suggested as a way to improve the method in their further work. Due to the high speed of convergence, a mutation operator was added to prevent the swarm converging too early on local optima. Initially high, the probability of mutating a particle decreased rapidly with number of iterations. The amount of mutation allowed on each particle also decreased with the same relationship. Particularly in the early stages, this caused the particles to continually search new regions of the search space and therefore reduce the chance of early convergence. This, they stated, enabled the algorithm to exhibit more exploratory behaviour and search the full range of decision variables. The results of the study showed that their algorithm (termed MOPSO) was "the only algorithm from those adopted in the study that was able to cover the full Pareto front of all the functions used."

Reddy and Kumar [4] describe a PSO procedure which differs slightly with respect to the way in which the velocity term is prescribed. A user defined constriction factor, χ, was directly multiplied to the equation to restrict its magnitude, which, in their case was set to 0.9. A step time value, Δt, was also introduced to add variability to some factors. But since the value was made equal to 1, this had no overall effect of the governing equations:

$$x_{i+1} = x_i + \Delta t\, v_{i+1} \tag{5.4}$$

where

$$v_{i+1} = \chi \left[\omega v_i + c_1 r_1 \frac{\left(x^{personal} - x_i\right)}{\Delta t} + c_2 r_2 \frac{\left(x^{global} - x_i\right)}{\Delta t} \right] \tag{5.5}$$

Similarly to earlier work [3], an external repository of fixed size was used. However, instead of an adaptive grid to promote diversity, the crowding operator [5] was used. In addition to this, an *elitist-mutation* operator was included to increase the searching ability of new areas of the search space. It acted on a pre-defined number of particles where parts of their solutions were adjusted to suit the least crowded solutions in the non-dominated set. They state that this initially replaced any infeasible solutions with the least crowded solutions in the non-dominated set. In the later phase, it concentrated the search towards the sparsely populated areas of the non-dominated set. Interestingly, unlike Coello Coello

et al [3], this mutation operator utilises only known information, as opposed to making random (uninfluenced) changes. So while the authors remark that it "helps the exploration and exploitation of the search space for the feasible non-dominated solutions," it would seem that given the nature of their mutation operator, it is solely good in these regions. This is in Contrast to Coello Coello et al [3] whose mutation operator targets more global exploration.

5.3 PSO in composite design

With regard to the PSO technique for sandwich design, while several laminated composite stacking sequence problems have been attempted, none exist which deal with multiple objectives using the concept of domination. Cases that have been found to be the most similar to the needs of this thesis, in terms of industrial application, are discussed below.

Suresh et al [8] describe the optimisation of a laminated composite box-beam for a helicopter rotor blade in which the objective was to maximise the stiffness. Design variables included the dimensions of the box-beam and the ply orientation angles of the laminate. A 26 ply stack was considered, however, due to symmetry of the laminate and fixed constraints on the outer plies, only five ply angles were considered as variables where a range of discrete angles between 0° and 90° could be selected. Only a single amalgamated objective function was employed. Also, a fairly restricted search space in terms of the variables was used. Nevertheless, a comparison of results with PSO and a GA showed that PSO was always able to identify solutions that were closer to the target stiffness. Also, in a separate performance evaluation, PSO was found to require less computational effort.

Kathiravan and Ganguli [9] described a similar analysis in which the optimum ply angles were sought for a composite beam in order to maximise strength. They state that despite its straightforward implementation, "most composite optimisation works have not used PSO." The study they conducted compared PSO against a gradient-based optimisation technique. A number of different load cases were considered in which only symmetric lay-

ups were permitted. The angle range for each ply was set between -90° and 90°. For each load case the PSO algorithm identified material constructions that were at least as strong as, or stronger than, those identified by the gradient-based method. However, as with Suresh et al [8], this was only single objective optimisation which considered a search space restricted to just ply angle.

5.4 Observations from existing PSO techniques

One of the more interesting points to note with PSO is that the underlying mechanism of the technique itself has changed very little since it was first developed. Due to the large success of the technique, there has been no need to differ significantly from its original form. However, not only has the PSO proven to be robust in many instances, but it is inherently adaptable to multiple objective scenarios. Particularly, this is due to the transferability that each source of information (which guides the PSO) has from the single, to multi-objective case. For instance, each particle is directed by its own personal best solution. This remains the same regardless of how many objectives a problem has. Also, the global best information required for each particle can be easily obtained by simply using a solution from the non-dominated set. While this may seem trivial, considerable modifications need to be made to the other techniques (ant colony optimisation (ACO) and simulated annealing (SA)) to apply them here.

To make further observation, several efforts have been made to increase the searching capability of the algorithm with the use of an additional operator. This was first conducted by Kennedy and Eberhart [1] with their craziness operator. Later, Fourie and Groenwold [10] adopted this operator to add a layer of variation to the system by mimicking "random (temporary) departures of birds in the flock." However, their method only influenced the magnitude and direction of the velocity, instead of the entire motion. Further to this, the mutation operators [3, 4] mentioned earlier in section 5.2 also show similarity here. The overall effect being to increase the search capability of the algorithm for the purposes required. Hence, given the number of authors that have addressed this issue, it is a favourable aspect to include in the developed PSO.

5.5 The developed PSO algorithm (*sandwichPSO*)

Having conducted a detailed survey of PSO regarding its use for optimising sandwich materials and structures, it is now time to present the technique that has been developed for this purpose. This technique has been termed by the author as *sandwichPSO*.

As noted in the previous section, the PSO algorithm has changed very little since it was first introduced. Its considerable previous success and natural transferability to multiple objectives require that only minor adjustments have been made here to the initial underlying equations. Each variable in a particle's next move for *sandwichPSO* is given by:

$$x_{i+1} = x_i + v_{i+1} \tag{5.1}$$

$$v_{i+1} = wr_1 v_i + c_1 r_2 \left(x^{personal} - x_i\right) + c_2 r_3 \left(x^{global} - x_i\right) \tag{5.6}$$

Equation (5.1) is that of the original PSO. However, Equation (5.6) includes an extra factor. Instead of a random number being applied to the cognitive and social influence parameters, they are now applied to all three terms. So the equation now contains r_1, r_2 and r_3. This has been done to increase the searching ability by allowing the effect of the previous motion, v_i, to fluctuate more freely. These influencing factors are summarised in Figure 5.1. For the parameters w, c_1, c_2, μ, and the number of particles in the swarm, while recommendations elsewhere are honoured, they will nevertheless require tuning for particular case examples. The advantage of this is that the user is given some control over the searching nature of the particles.

To introduce an additional element of searching ability, a further parameter was included and has been termed the wind factor, μ. This was a novel aspect included in the development of this algorithm and achieved a similar effect to the mutation and craziness parameters mentioned in sections 5.1 and 5.2 [1, 3, 4, 10]. The wind factor gave each particle the chance of searching somewhere completely different. Somewhere it might not otherwise reach through normal motion. Under the bird analogy, one might consider it as a

strong, unexpected, random gust of wind that blows the particle off-course, away from its normal path. This was implemented as a defined probability, that, on each iteration, any given particle's position would be randomly reinitialised rather than following the normal scheme of motion. Including a wind factor added variability to the process and was applied as a two part operation. It allowed some instances where a particle could be blown off-course just slightly, as well as entirely. Its implications mean that even with small wind factors, a significant possibility of obtaining completely new solutions still remains. The pseudo-code for the wind factor operator is shown in Figure 5.2.

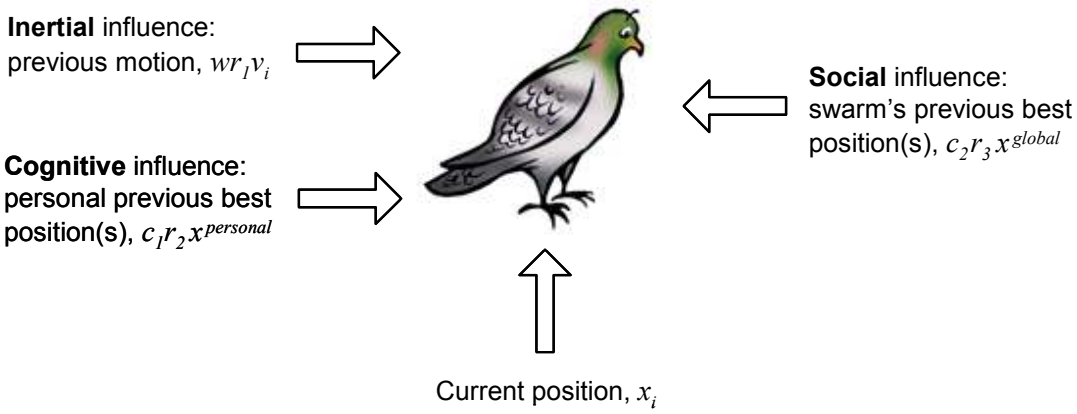

Figure 5.1. Factors influencing the motion for the *sandwichPSO* optimisation technique.

For each solution
  If μ > *rand* (apply wind)
    If μ > *rand* (randomise the entire particle)
      For all variables
      $x_i = x^{\min} + rand * \left(x^{\max} - x^{\min}\right)$
      end

    else (decide to randomise particular variables)
      For each variable
        If μ > *rand* (mutate variable)
        $x_i = x^{\min} + rand * \left(x^{\max} - x^{\min}\right)$
        end
      end
    end
  end
end

Figure 5.2. Pseudo-code for the wind operator.

A wind is applied to a particle if the value of the wind factor is larger than a random number (*rand*). So larger wind factors imply more moves are generated at random. Both take values between 0 – 1. The complete developed algorithm is shown as a flowchart in Figure 5.3 and relates back to the general procedure given in the previous chapter (Figure 4.12).

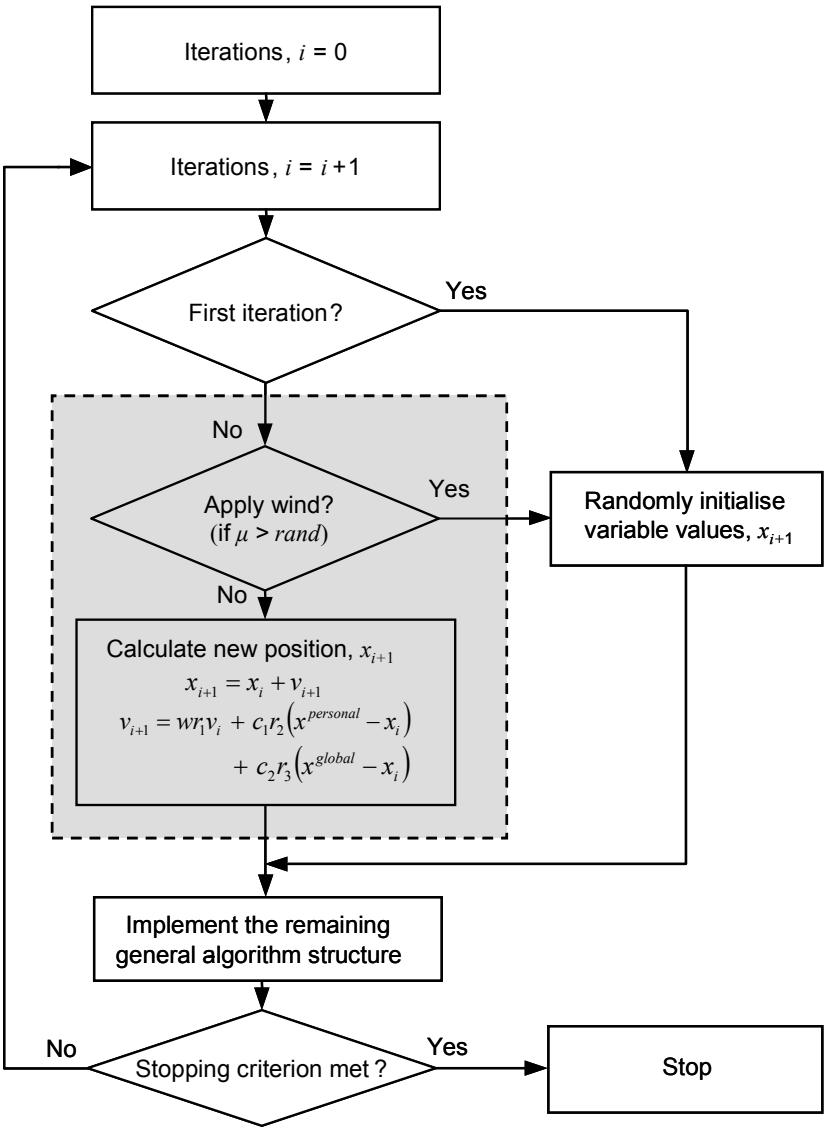

Figure 5.3. Flowchart of the proposed algorithm. Greyed areas mark parts specific to *sandwichPSO*.

5.6 Conclusions

In this chapter, a detailed analysis of PSO has been conducted with a view to further developing the technique for sandwich design. Although few examples exist where PSO has been applied to problems of a similar nature to sandwich optimisation, some have been found that optimise the stacking sequence of composite laminates. However, they only consider at best, single amalgamated objective functions and are primarily concerned with finding optimal stacking sequences. In addition, several examples of its application to multiple objective problems have also been noted. However, while marked successes for this purpose have been made, they are far less concerned with the optimisation of sandwich composite design. Taking all this into consideration, a PSO called *sandwichPSO* has been developed here which is able to deal with the multiple variable, objective and constrained nature involved with the optimisation of sandwich materials and structures. Hence, it is now ready to be deployed for a benchmark case study (Chapter 8). In addition to testing its performance, the benchmark will allow several algorithm parameters to be tuned to suit the particular problem. These are w, c_1, c_2, μ, and the number of particles in the swarm.

However, before this is done, the next two chapters consider the development of the ACO and SA techniques in a similar manner to the PSO here.

5.7 Publications

Hudson, C.W., Carruthers, J.J., Robinson, A.M. (2009) Application of particle swarm optimisation to sandwich material design. *Plastics, Rubber and Composites* **38**, 106-110.

5.8 References

1. Kennedy, J., Eberhart, R. (1995) Particle swarm optimization. *IEEE International Conference on Neural Networks - Conference Proceedings*.
2. Shi, Y., Eberhart, R. (1998) Modified particle swarm optimizer. *Proceedings of the IEEE Conference on Evolutionary Computation*.
3. Coello Coello, C.A., Pulido, G.T., Lechuga, M.S. (2004) Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation* **8**, 256-279.
4. Reddy, M.J., Kumar, D.N. (2007) An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design. *Engineering Optimization* **39**, 49-68.
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**, 182-197.
6. Knowles, J.D., Corne, D.W. (2000) Approximating the nondominated front using the Pareto Archived Evolution Strategy. *Evolutionary Computation* **8**, 149-172.
7. Coello, C.A., Pulido, G.T. (2001) Multiobjective optimization using a micro-genetic algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference GECCO*, 274-282.
8. Suresh, S., Sujit, P.B., Rao, A.K. (2006) Particle swarm optimization approach for multi-objective composite box-beam design. *Composite Structures* **81**, 598-605.
9. Kathiravan, R., Ganguli, R. (2006) Strength design of composite beam using gradient and particle swarm optimization. *Composite Structures* **81**, 471-479.
10. Fourie, P.C., Groenwold, A.A. (2002) The particle swarm optimization algorithm in size and shape optimization. *Structural and Multidisciplinary Optimization* **23**, 259-267.

6 Developing ant colony optimisation (ACO) for sandwich design

In the previous chapter, a detailed analysis of particle swarm optimisation (PSO) was conducted which paid particular attention to its application to sandwich optimisation. This led to the development of a technique called *sandwichPSO* for the purpose. Here, a similar process will now be conducted for ant colony optimisation (ACO).

6.1 The original Ant System (AS)

Similarly to PSO, ACO employs a group of information-sharing search agents tasked with finding good objective values. However, the ACO is based on the analogy of ants leaving their nest in search of food. Deposition of pheromone by the ants enables better solutions to be identified.

The original ACO implemented by Dorigo et al [1] was called the *Ant System* (AS) and was applied to a classical travelling salesman problem. Importantly, in this analogy, the distance travelled by the ants from the food source to the nest is the objective to be minimised, not the food source itself. The extent of this will be made clear later. However, in any given trip, to force the ants to make legal visits to all towns, transitions to previously

visited towns were disallowed. This was carried out using a *Tabu list* which remembered the past history of all moves for that iteration. Each new move would be tested against a *Tabu list* to ensure it was different, if not, it would be retaken. It is worth noting that despite the name, the author's remark that their *Tabu list* is not a hybridized implementation of the Tabu search algorithm by Glover [2].

For a given iteration, i, the probability, k, of an ant moving to the next available town (or variable), x_n, is based upon the amount of pheromone it occupies:

$$k_i = \frac{\left[\tau_i\right]^{\alpha_1} . \left[\eta\right]^{\alpha_2}}{\sum_{n=1}^{\Omega} \left[\tau_i^{x_n}\right]^{\alpha_1} . \left[\eta^{x_n}\right]^{\alpha_2}} \tag{6.1}$$

The η term is the so-called *visibility* and is inversely proportional to the distance to the next town. This implies closer towns have larger visibility values. Also, at any given point, not all towns may be accessible. The set of x towns that can be visited from the current location is represented by Ω. The existing pheromone on each town is τ_i. Once all ants have made their journeys, the new pheromone, τ_{i+1}, is updated ready for the subsequent iteration according to:

$$\tau_{i+1} = \rho\tau_i + \sum\left(\tau_i^{\,popular}\right) \tag{6.2}$$

This first involves evaporating the existing pheromone, τ_i, at a rate ρ. Extra pheromone, $\tau_i^{popular}$, is then added for every ant which visited in that iteration in accordance to the overall tour length (fitness) of the journey made. Pheromone from all ants visiting a particular town are summed together to give $\Sigma(\tau_i^{popular})$. This aspect enables the past history of the search to be carried-over and influence successive ant motions. The α_1 and α_2 terms are weighting factors to emphasise preference towards either the previous pheromone history, or to favour closer towns respectively.

As an extension to this work, the authors also proposed an additional parameter which increased the performance of the technique. They called this an *elitist strategy* which

added extra pheromone to the best trail found so far. The number of times it was applied was controlled by a user defined number of *elitist ants* in the colony.

6.2 The Ant Colony System (ACS)

As a successor of the Ant System, the *Ant Colony System* (ACS) was later proposed by Dorigo and Gambardella [3]. This targeted a difficulty that the AS had with handling larger solution spaces and has since formed the basis of a significant number of recent ACO developments [4-7]. It introduced three major modifications to the original Ant system. Firstly, an extra step was included to govern the movement of an ant from one variable to the next. A so-called *state-transition-rule* stated that if a random number between 0-1 was less than a user defined limit, then the probability of moving to a particular town was the same as in the original AS (Equation (6.1)). If not, the ant was forced to select the variable with the closest, most pheromone intense trail. This extra step allowed greater exploitation of the known good solutions if required. The second modification was with the application of a pheromone modifying parameter they called a *local updating rule*. Several different values for the local update, τ_i^{local}, were investigated but the general idea was to diminish the pheromone once an ant had visited, as opposed to increasing it. This was done by making the magnitude of the deposited pheromone sufficiently small. This kept the ants searching new areas and prevented them from converging to a common path. The overall effect could be modified in relation to a parameter ρ_1. As with the AS, it was applied repeatedly for every ant visiting each town in the current iteration. However, rather than waiting until all ants had completed their journeys, it was applied straight after each ant completed each tour and was only applied to towns that had actually been traversed.

$$\tau'_{i+1} = (1-\rho_1)\tau_i + \rho_1 \tau_i^{local} \tag{6.3}$$

The term τ'_{i+1} indicates the intermediary pheromone value since several pheromone modifications take place during any particular iteration. Also, the value of the local update

was set so that it corresponded to a lower pheromone limit to which no pheromone level was allowed to fall below [5].

Finally, in addition to the local update, a *global updating rule* was also applied. This increased the level of pheromone on the globally best trail in proportion to its path length (fitness), τ_i^{global}. In the case of the travelling salesman problem, this is the shortest known path. Furthermore, it was only applied to the towns belonging to the global best trail. However, unlike the local update, it was only performed once all ants had completed their journeys.

$$\tau_{i+1} = (1-\rho_2)\tau_i + \rho_2 \tau_i^{global} \tag{6.4}$$

6.3 Observations from early ACO techniques

The main working procedure of two early ACO techniques has been briefly described. However, several issues exist which hinder the direct use of ACO for sandwich design due to some incompatibilities. This is primarily due to its strong interlinked nature with the travelling salesman problem. It not only relates to the nature of the variables themselves but also presents difficulties regarding the extension of the technique to multiple objectives. These issues are outlined in more detail below.

As discussed, the ACO was originally designed to solve the travelling salesman problem. In particular, the objective of this problem is to minimise the total distance travelled to all towns. As each ant moves from town to town (each variable), the distance to every available town is used progressively in the optimising process. For instance, the effect of the optimised objective (distance travelled) can also be analysed directly, during the ant's journey. This aspect is apparent in the visibility term, η, defined earlier. For sandwich design however, it is not possible to know anything about the value of the objectives until the entire solution is complete. For instance, the cost of a sandwich beam cannot be known until all of the variables have been set, e.g. the core and facing materials, and the

dimensions etc. This means that the visibility term will either need to be revised if the parameter is to be meaningful, or ignored completely.

Another aspect of the ACO techniques described so far is that only discrete variables can be handled. This is because the pheromone must be deposited by the ants at a particular town, i.e. on a particular variable. This is unlike the PSO for instance where information of each influence factor is amalgamated together. Which, if the variable is continuous, may be accepted as it appears. If discrete, the nearest point to the obtained value is selected as the next move. For the ACO however, an easy fix to this problem would simply be to make any continuous variable discrete by dividing it up into a suitable number of discrete values.

In addition, the requirement to handling multiple objectives will also require consideration. Again, this is because the calculation of the pheromone includes the distance (objective value) of a completed ant pathway. Since this is a single metric, it would on first inspection require some sort of amalgamation of objectives if presented with a multiple objective scenario.

Finally, the way in which the Tabu list was used in the AS (to ensure only feasible solutions were found) would serve no purpose if the technique were applied to sandwich design. This is due to the selection process in sandwich design whereby only one value from each variable is required. For instance, only one core material, facing material, core thickness etc. is required. This is in contrast to the travelling salesman problem where a feasible order of *all* variables is selected. However, that is not to say that a Tabu search algorithm hybrid could be utilised in its more conventional sense, i.e. as by Glover [2] later down the line. This would temporarily restrict the selection of previously visited values, within each variable, thereby redirecting the search elsewhere.

6.4 Multiple objective ACO strategies

Several noteworthy developments have been made concerning the application of multiple objective problems using ACO. The work of Garcia-Martinez et al [5] presents a taxonomy of such multiple objective techniques and conducts a comparative study between them. In reviewing their work, it would appear that in order to deal with multiple objectives, different authors utilise either several ant colonies, pheromone trails (τ_i), or visibility terms (η). The idea being that each element focuses the search in some way towards each separate objective. In addition to this, while the concept of domination was used in most cases, lexicographic ordering was also considered by some. However, only those that obtained a non-dominated solution set were carried forward for experimental investigation. The results showed the majority of techniques were able to obtain a non-dominated set in close proximity to the Pareto-optimal front. From their results, a detailed description of the performance of all tested techniques was given. To offer a quick indication of the performance, a qualitative assessment of each technique is given in Table 6.1. Performance has been indicated on a scale of 1 – 5, larger values represent better ability in each aspect. However, it should be noted that this indication of performance has been conducted by the author of this thesis based on observations made from their work, not by the researchers themselves.

Table 6.1. Quantitative analysis performed by the author of this thesis from observing the results of the ACO techniques reviewed by Garcia-Martinez et al [5].

Algorithm and Author	Repeatability	Closeness to pareto front	Ability to reach extremes	Evenness of distribution
P-ACO: Doerner et al [6]	5	5	1	1
MONACO: Cardoso et al [7]	4	5	1	1
BicriterionAnt: Iredi et al [8]	4	5	3	4
BicriterionMC: Iredi et al [8]	3	5	1	2
UnsortBicriterion: Iredi et al [8]	3	5	5	5
MOAQ: Mariano and Morales [9]	2	3	2	5
MACS: Baran and Sheaerer [4]	4	5	5	5
COMPETants: Doerner et al [10]	1	1	2	5

In terms of overall performance, the ACO algorithm they called MACS or Multiple Ant Colony System (based on the ACS) developed by Baran and Schaerer [4] appeared to be one of the most competitive. This technique employed two separate ant colonies, one for

each objective and acquired a non-dominated set. Also, as in the original ACS, it utilised a single pheromone trail matrix, but applied several visibility terms which corresponded to different aspects of the problem. However, despite the differentiation they highlight with regard their taxonomy (i.e. in differentiating between techniques with one or several pheromone trails or visibility terms), Garcia-Martinez et al [5] conclude that the success of an ACO technique depends on the actual operational mode, or rather the characteristics of the specific method itself. This appears to be more intrinsic to the entire optimisation process. Hence, while large success has been achieved with extending ACO for multiple objectives, depicting more fundamental characteristics which lead to good optimisers is not so straightforward. This means that while Garcia-Martinez et al [5] provide a more general lead into the background of the topic, further development work for the application to multiple objective sandwich design will still need to be conducted.

6.5 ACO in engineering design

In recent years, a significant proportion of ACO research has been carried out on a comparatively narrow variety of optimisation problems, i.e. the travelling salesman problem, job shop scheduling and vehicle routing [11, 12]. However, few examples have been found where ACO has been attempted on cases more closely related to sandwich design. Of those that show similar elements, a brief description of the methodology will be given. This will be followed by some remarks about the technique in relation to its potential for further exploitation.

Abachizadeh and Tahani [13] examined the optimisation of a simply supported laminate plate. In terms of the proposed algorithm, the chosen approach largely followed that of the ACS [3] mentioned earlier. The problem was fairly restricted in that only two variables were considered. The first was angle orientation where only symmetric laminate lay-ups were considered. The angles were restricted to discrete values in the range of -90° and 90° with 15° increments. The second was a choice of two lamina materials; Glass/Epoxy or Graphite/Epoxy. To allow designs of equal thickness to be compared, the total thickness of the laminate was considered constant. This meant that the thickness of individual plies,

which was equal, was determined by the number of layers used. Although multiple objectives were considered; to maximise the fundamental frequency of the laminate and minimise the cost, these were actually aggregated into a single function. Hence, only a single optimum point was sought instead of a Pareto-optimal set of solutions. Results showed that in terms of objective value, the ACO was able to compete with and in some cases surpass those found by a genetic algorithm (GA) and a simulated annealing (SA) algorithm.

Particular interest should be given to the material selection here as it bares a common incompatibility with sandwich design. That is to say the variables of the problem share the same physical representation. As such, they offer a solution around the difficulties mentioned earlier with handling the visibility term η. Recall that visibility requires the *distance* between variables to be quantifiable. If the variables of material property and angle orientation are considered, it is clear to see that no relevant *distance* metric exists between the two. So in short, they bypassed the problem by simply ignoring the visibility term. This was justified through the earlier work by Dorigo and Gambardella [3] who showed that the effect of ignoring visibility only moderately deteriorated efficiency. So, a loss of efficiency was traded in order to make the algorithm far easier to implement. This meant that the probability of an ant transitioning to the next available node differed from Equation (6.1) and can be expressed as:

$$k_i = \frac{[\tau_i]}{\sum_{n=1}^{\Omega} [\tau_i^{x_n}]} \tag{6.5}$$

However, similarly to the ACS, this was only implemented if the value of a random number between 0-1 was less than a user defined limit. If not, the ant was forced to select the variable with the most pheromone intense trail.

In another example, optimisation of a laminated plate was conducted by Aymerich and Serra [14]. The employed ACO largely followed that of the original AS. However, optimisation of only a single amalgamated objective of the buckling and compressive failure load was conducted. In addition, the only variable was the stacking sequence which meant the search space was relatively small. To improve the computational efficiency, the

authors were able to restrict the number of available stacking sequence combinations by virtue that only balanced, symmetric laminates were permitted. Symmetry required only half of the laminate be optimised by the algorithm and the balanced condition was satisfied by confining available plies to pairs of $(0°_2)$, $(\pm 45°)$, and $(90°_2)$. Results showed that the ACO had good average performance and robustness when compared to two GAs and a tabu search (TS) algorithm [15-17]. Similarly to Abachizadeh and Tahani [13], due to the incompatibility of the variables, the visibility term in the pheromone update equation was chosen to be ignored. This meant that the transition probability of an ant moving to the next available node was purely the same as that described in Equation (6.5). At the end of each iteration, Aymerich and Serra [14] evaporated and updated the pheromone in a similar way to the AS described earlier. An elitist strategy was also used but differed somewhat to the original AS [18]. A set of elitist ants adding extra pheromone to the globally best trails was not used. Instead, pheromone was only added if a solution generated in the current iteration was either equal or superior to the best found so far. This gave the ants a strong incentive to search the region around the best solution, rather than exploring unvisited areas. So while this kind of approach was desired in their particular instance, they note that on a more general level, it may cause premature convergence to local minima. Another aspect which worked well in their case was that the algorithm used only a single ant. They stated that this gave the best balance between quality of solutions produced versus time allowed to run the simulation. However, they further acknowledge that the optimal number of ants to use is heavily problem dependent.

6.6 The developed ACO algorithm (*sandwichACO*)

The first point to note here is that the developed ACO technique does not operate under the normal ant colony analogy. Conventionally, the objective of the ants is to find the shortest *route* from the nest to the food source. However, this is no longer the case. For the *sandwichACO* technique described here, the ants are now tasked with finding the route which leads to the best *food source*. It was felt this was a logical change since the ants in

this case, unlike most, are not looking for short routes. Rather, the selection of variables leading to good sandwich designs is required. Now that this has been explained, the process itself may now be presented.

The probability that an ant will move to the next available town in a given iteration is:

$$k_i = \frac{[\tau_i]}{\sum_{n=1}^{\Omega} [\tau_i^{x_n}]} \tag{6.5}$$

This is in accordance with Abachizadeh and Tahani [13] and Aymerich and Serra [14] where the visibility term, η, has been ignored. This only leaves the pheromone update to provide the ants with a search direction, which is only conducted once all ants have completed their journeys. This was implemented as the meaning of the visibility term in the conventional ACO carries no physical meaning in sandwich design (see sections 6.3 and 6.5).

Pheromone is updated in three parts: evaporation of the existing pheromone ($\rho\tau_i$), addition of pheromone from all ants that visited that town in the current iteration ($\Sigma\tau_i^{popular}$), and addition of pheromone from all solutions contained in the current non-dominated set ($\Sigma\tau_i^{global}$):

$$\tau_{i+1} = \rho.\tau_i + \left(\sum \tau_i^{popular}\right)^{\alpha_1} + \left(\sum \tau_i^{global}\right)^{\alpha_2} \tag{6.6}$$

With other ACO techniques, commonly, addition of pheromone is carried out in proportion to the fitness of the route it corresponds to. However, each pheromone addition here is instead provided in discrete amounts (equal to 1). The addition of pheromone in discrete amounts is advantageous as it does not require knowledge of actual objective value. Using objective value (which is more common) would otherwise be difficult in a multiple objective case as the question of which objective should provide the pheromone addition, and to what amount, is avoided. In addition, regardless of any difference to the order of magnitude, it offers a way of assigning the same preference to all objectives and treats all solutions in the non-dominated set equally. As a result of this, and given the diversity

preserving aspect included within the algorithm, the use of a single ant colony, with a single pheromone storing system, is a logical decision.

Furthermore, a minimum residual pheromone of at least 1 unit was maintained on all trails throughout the simulation. This was to ensure that all routes had at least some chance of being visited. This helped to promoted diversity in the colony and hence the continual exploration of new search regions.

In total, four user definable parameters are present in the algorithm. These are the weighting factors α_1 and α_2, the pheromone evaporation rate, ρ, and number of ants and will need to be tuned to suit each problem the algorithm addresses. The advantage of this is that the user is given some control over the searching nature of the ants.

A diagram showing the influence factors of an ant's motion is displayed in Figure 6.1. A flowchart showing the detail of the *sandwichACO* algorithm process is given in Figure 6.2 in relation to the overall structure (Figure 4.12).

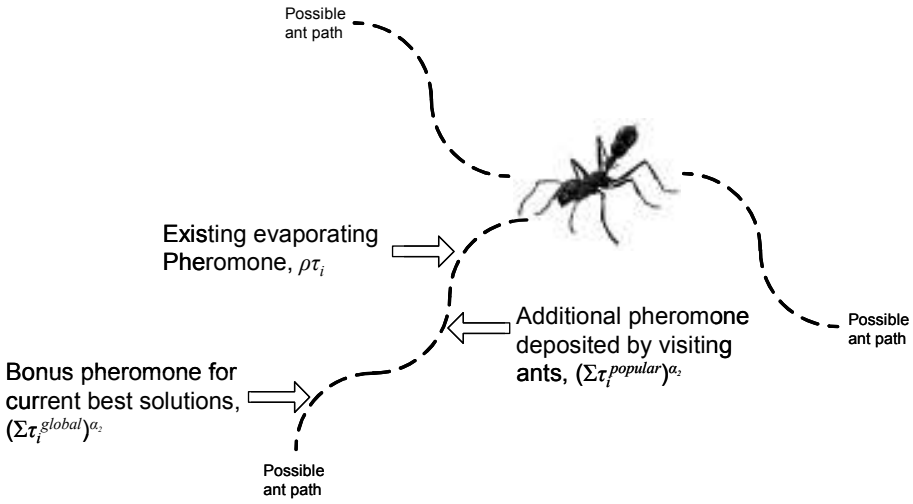

Figure 6.1. The factors which influence an ant's decision to choose a particular path for the *sandwichACO* technique.

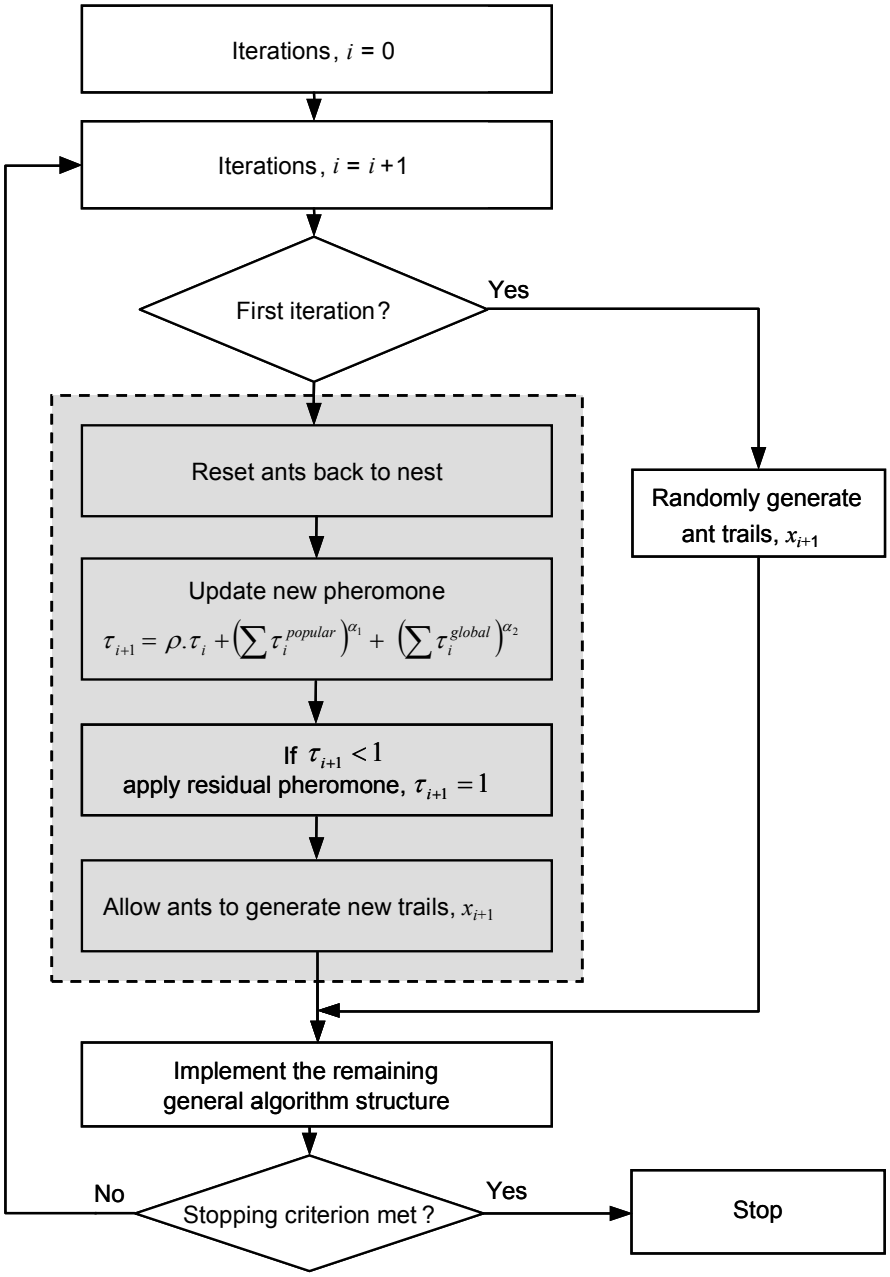

Figure 6.2. Flowchart of the proposed algorithm. Greyed areas mark parts specific to *sandwichACO*.

6.7 Conclusions

In this chapter, a detailed analysis of ACO has been conducted with a view to further developing the technique for sandwich design. Similarly to the PSO, surprisingly few

examples of ACO exist that show much relation to sandwich optimisation. Of those that do, some material selection has been attempted. However, at best, only single amalgamated objective optimisation has been considered in which stacking sequence of composite laminates has been the primary focus. Thus, the problems are fairly restricted. Furthermore, while several fairly different methods of extending ACO to multiple objectives have shown good application of the technique, the instances they discuss are largely unrelated to sandwich design.

Building upon these advances, an ACO technique developed for sandwich optimisation has been developed in this thesis which has been termed *sandwichACO*. Due to the nature of the original ACO algorithm, significant changes to the process needed to be made to make it applicable for this purpose. First of all, this required the analogy of the ACO algorithm to be changed. Instead of the ants searching for the *shortest route* to a particular food source, they now search for the routes that lead to the best *food source*. In addition, the visibility term from the governing ant motion was decidedly removed as it bore no physical representation with sandwich optimisation. Finally, to alleviate difficulties with prioritising and scaling objectives, pheromone was added in discrete units. This was instead of the more common approach where it is added in proportion to the progressive fitness of a solution.

As with PSO, several parameters of the *sandwichACO* algorithm will need to be tuned to suit the particular problem. These are α_1, α_2, ρ, and number of ants and will form part of a benchmark case study. However, before this can be carried out, the next chapter discusses the third and final optimisation technique to be investigated in this thesis. This is the simulated annealing (SA) technique and will be developed in a similar manner to previous two. After which, a benchmark case study (Chapter 8) will enable each algorithm to be tuned purposefully for sandwich design and also allow their suitability to be compared.

6.8 References

1. Dorigo, M., Maniezzo, V., Colorni, A. (1996) Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **26**, 29-41.
2. Glover, F. (1989) Tabu Search - Part II. *ORSA Journal on Computing* **1**, 4-32.
3. Dorigo, M., Gambardella, L.M. (1997) Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation* **1**, 53-66.
4. Baran, B., Schaerer, M. (2003) A multiobjective ant colony system for vehicle routing problem with time windows. *IASTED International Multi-Conference on Applied Informatics*.
5. Garcia-Martinez, C., Cordon, O., Herrera, F. (2007) A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research* **180**, 116-148.
6. Doerner, K., Gutjahr, W.J., Hartl, R.F., Strauss, C., Stummer, C. (2004) Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Annals of Operations Research* **131**, 79-99.
7. Cardoso, P., Jesus, M., Marquez, A. (2003) MONACO - Multi-objective network optimisation based on ACO. *Encuentros De Geometria Computacional*.
8. Iredi, S., Merkle, D., Middendorf, M. (2001) Bi-criterion optimization with multi colony ant algorithms. *Lecture Notes in Computer Science* **1993**, 359-372.
9. Mariano, C.E., Morales, E. (1999) MOAQ an ant-Q algorithm for multiple objective optimization problems. *Proceedings of the Genetic and Evolutionary Computation Conference* **1**, 894-901.
10. Doerner, K.F., Hartl, R.F., Reimann, M. (2003) Compet Ants for problem solving: The case of full truckload transportation. *Central European Journal of Operations Research* **11**, 115-141.
11. Talbi, E.G., Roux, O., Fonlupt, C., Robillard, D. (2001) Parallel ant colonies for the quadratic assignment problem. *Future Generation Computer Systems* **17**, 441-449.
12. Dorigo, M., Blum, C. (2005) Ant colony optimization theory: A survey. *Theoretical Computer Science* **344**, 243-278.

13. Abachizadeh, M., Tahani, M. (2008) An ant colony optimization approach to multi-objective optimal design of symmetric hybrid laminates for maximum fundamental frequency and minimum cost. *Structural and Multidisciplinary Optimization*, 1-10.
14. Aymerich, F., Serra, M. (2008) Optimization of laminate stacking sequence for maximum buckling load using the ant colony optimization (ACO) metaheuristic. *Composites Part A: Applied Science and Manufacturing* **39**, 262-272.
15. Le Riche, R., Haftka, R.T. (1993) Optimization of laminate stacking sequence for buckling load maximization by generic algorithm. *AIAA Journal* **31**, 951-956.
16. Kogiso, N., Watson, L.T., Gurdal, Z., Haftka, R.T. (1994) Genetic algorithms with local improvement for composite laminate design. *Structucal Optimisation* **31**, 951-95.
17. Pai, N., Kaw, A., Weng, M. (2003) Optimization of laminate stacking sequence for failure load maximization using Tabu search. *Composites Part B: Engineering* **34**, 405-413.
18. White, T., Kaegi, S., Oda, T. (2003) Revisiting elitism in Ant Colony Optimization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2723**, 122-133.

7 Developing simulated annealing (SA) for sandwich design

The previous two chapters conducted a detailed investigation of the particle swarm optimisation (PSO) and ant colony optimisation (ACO) algorithms. In both cases, a process was developed that is able to deal with the needs of sandwich optimisation. These have been termed *sandwichPSO* and *sandwichACO*. Simulated annealing (SA) is the third and final optimisation technique to be addressed in such a manner and forms the topic of this chapter.

7.1 The original SA technique

The original SA technique was developed independently by Kirkpatrick et al [1] and by Cerny [2]. Initially, as with the PSO and ACO techniques, the atoms (or search agents) are randomly positioned throughout the search space. Thereafter, for each new proposed random move of an atom, a decision is taken as to whether to accept it or reject it. If the new move provides a better solution, it is always accepted. If the new move provides an inferior solution, an *acceptance criterion*, *p*, is used. This is based on the Boltzmann factor which was first used as an acceptance probability, *p*, by Metropolis et al [3]:

$$p = \exp\left(-\Delta f / T\right) \tag{7.1}$$

Acceptance is granted if $p > rand$, where *rand* is a random number between 0 and 1. The likelihood of an inferior solution being accepted decreases with decreasing temperature, T, and increases with smaller objective differences, Δf, between the existing and proposed new solution. Overall, under this mechanism, atoms move towards better solutions. With controlled cooling, the closer an atom is to the true Pareto-optimal front, the more likely it is to explore the region in the near vicinity and thereby find better solutions.

For the original SA techniques, the user is left to find appropriate values of temperature for the cooling schedule. While the paper by Cerny [2] more explicitly states the cooling schedule used, it would appear that in both instances finding appropriate values of temperature is problem dependent. Obtaining suitable values may be done experimentally via a trial and error method, but as a rule-of-thumb, using values in relation to the magnitude of the objectives being optimised is appropriate. Temperature reductions occurred in a number of discrete stages, as opposed to continuously throughout the simulation. But importantly, it was stated that sufficient time at each temperature should be given to allow the particles to reach a steady state (provide no more better solutions).

7.2 Observations from the early SA technique

With regard to the application of SA to multiple objective sandwich design, only one significant issue strikes the author as requiring close attention. This is with the acceptance criterion, p, which governs the acceptance of new solutions. In a single objective case, the objective difference, Δf, is governed only by one objective. So a clear relation to the fitness of a solution and the acceptance value is exists. However, in a multiple objective case, the value of Δf depends on multiple entries. Hence, a method for solving this problem will need to be implemented. Several ways this could be done have been attempted by previous authors and it is pointed out by Kubotani and Yoshimura [4] that the performance of the SA depends significantly on selecting the correct method.

Another factor which would require consideration is with the cooling schedule. Among others, Youssef et al [5] noted that this can have a major impact on the performance and must be carefully crafted for the particular problem instance. However, this is not surprising given the number of tuneable parameters involved e.g. initial and final temperature, the number of temperature reductions, and the amount by which to reduce at each stage.

From the author's point of view, despite the SA being regarded as the less intelligent technique of the three, this feature may in actual fact play to its advantage. Because each solution acts independently to the rest, the SA may be more resilient to any strong net trends that develop in the solution data as the simulation progresses. Section 2.5 highlighted several features that make it difficult for an optimiser to find and maintain a diverse Pareto-optimal front. If any of these features were present, other algorithms, more heavily engaged with information-sharing, may become easily focused upon them, and hence less able to fully explore the entire solution space.

7.3 Types of cooling schedule

Elsewhere, several cooling schedules have been proposed [6-10]. However, the most common approach [4, 5, 11] is to cool by multiplying the current temperature by a fixed cooling factor, ω, after a set number of iterations:

$$T^{new} = \omega\, T^{current} \tag{7.2}$$

An alternative cooling schedule has been described by Suppapitnarm et al [9]. It allowed the disadvantage of scaling objectives to be bypassed. This was done by letting process to adapt to each problem. It is carried out by first of all considering a separate temperature and cooling factor for each objective:

$$T_u^{new} = \omega_u T_u^{current} \tag{7.3}$$

The cooling factor itself forms the adaptive part of the schedule. It is based upon the standard deviation of each objective, S_u, of the current non-dominated solutions found. It is formulated as:

$$\omega_u = \max\left(0.5,\ \exp\left[-\frac{0.7T_u}{S_u}\right]\right) \tag{7.4}$$

This method differs from conventional cooling schedules as the value of the cooling factor is continually updated depending on the magnitude of the known objective values as the simulation progresses. Its advantage over the fixed cooling schedule is that if the early searching process is less successful, the temperature can be reduced quicker thereby forcing the search agents closer towards the Pareto-optimal front thereby saving time by not exploring poor areas.

7.4 Acceptance criteria for multiple objective SA

One of the main concerns with applying SA to multiple objective scenarios has been highlighted with how the acceptance criterion can be adapted for this purpose. When a new solution (with multiple objectives) is created, its acceptance in favour of the current solution leads to one of three instances:

a) The new solution is **superior** - in this case the new solution is unconditionally accepted.

b) The new solution is **non-dominated** - acceptance can either be unconditional, or reliant upon the acceptance criterion depending upon the method used.

c) The new solution is **inferior** - the acceptance criterion always determines acceptance.

Figure 7.1 shows these possible outcomes graphically. The first case is trivial as the new move is always accepted. However, for the latter two cases, several different approaches to negotiate these instances are available.

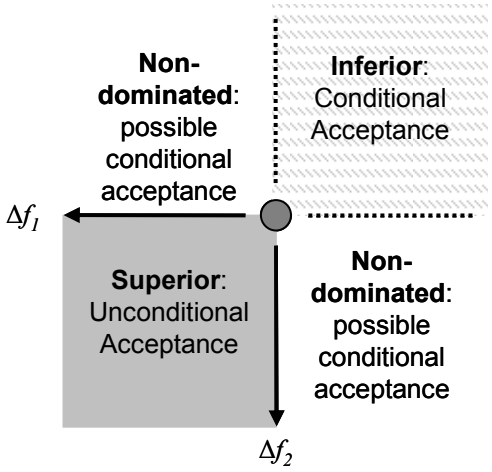

Figure 7.1. An objective minimising problem. The possible outcomes that could result when a new solution is compared to the existing one are shown.

Some of the more common methods for these multiple objective cases will be described in the sections to follow. They are the weighted sum, weighted product, strong and weak rule. In relation to Figure 7.1, the treatment that each of these methods gives to a new solution is represented in Figure 7.2. It is clear to see that each provides a different degree of acceptance towards new moves. A smaller proportion of unconditional acceptance (i.e. for the strong and product rule) concentrates the atoms more towards local exploration of the near vicinity. On the other hand, a larger proportion of unconditional acceptance (i.e. sum and weak rule) allows the search agents to reach more areas of the search space and perform better global exploration.

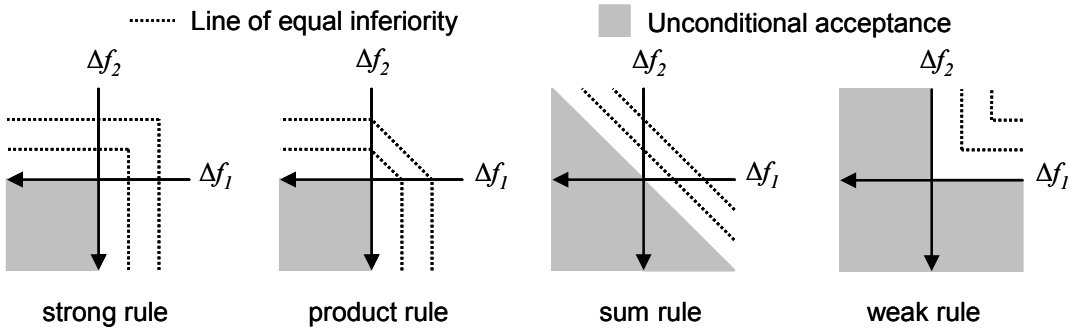

Figure 7.2. The different ways a new solution may be treated for the four acceptance criteria: strong, product, sum and weak rule [4, 12].

7.4.1 Weighted sum or scalar linear rule

A common approach for deriving the acceptance criterion is using the *weighted sum* or *scalar linear rule* approach [4, 6, 12]. The objective value difference, Δf_u, for each objective function is multiplied by a weighting parameter, W_u, and divided by the temperature, T_u. These U weighted objective value differences are then summed and processed according to Equation (7.5).

$$p = \exp\left(-\sum_{u=1}^{U}\frac{W_u \Delta f_u}{T_u}\right) \tag{7.5}$$

A multiple objective knapsack problem using this criterion has been attempted by Ulungu et al [12]. Although continual adjustment of the weighting parameters was required, a good set of non-dominated solutions was produced apart from at the extremities of each *efficient frontier* (the optimal solution or solutions which correspond to a particular weighting of objectives).

7.4.2 Weighted product rule

The weighted product considers the multiplication of each objective component. In relation to the weighted sum rule, the treatment of non-dominated solutions is very

different (see Figure 7.2) and is more concerned with further exploiting the known good solutions. Acceptance of a new solution is calculated as follows:

$$p = \prod_{u=1}^{U} \exp\left(-\frac{\Delta f_u}{T_u} \right)^{W_u} \tag{7.6}$$

This method was combined with the adaptive cooling schedule procedure used by Suppapitnarm et al [9] described earlier. In the conclusions of their study, the proposed SA algorithm was noted to be comparable against a genetic algorithm (GA) in terms of objective value and ability to reach the extremes of the Pareto-front. In addition, it was also remarked as being easier to implement.

7.4.3 The strong and weak rule

With regard to extending SA for multiple objectives, the strong and weak rule would on first inspection appear to be more suited to the handling of non-dominated solutions. Unlike the sum and product rules, they do not amalgamate objective values into one single acceptance criterion. Instead, they calculate an acceptance criterion for each objective separately. Then by applying a simple set of rules, one of the acceptances values is decidedly used. This avoids the need for aggregating. The two methods are described below [4, 6].

Firstly, the *strong (Čebišev) rule* requires that an acceptance criterion be calculated for a solution that is either non-dominated or inferior to the current. The equation may be expressed as:

$$p = \min\left(\exp\left\{ -\frac{W_u \Delta f_u}{T} \right\} \right) \tag{7.7}$$

Once done, simply the minimum value of these is then taken as the acceptance probability.

Similarly to the strong rule, the weak rule also calculates an acceptance criterion for each objective individually. However, in this case, non-dominated solutions are always accepted. Only inferior moves are subjected to the acceptance probability. The value used is simply the largest of these and may be expressed as:

$$p = \max\left(\exp\left\{ -\frac{W_u \Delta f_u}{T} \right\} \right) \tag{7.8}$$

In terms of the observed overall effect, the strong rule accepts fewer solutions unconditionally. So not only is it more directed towards the Pareto-optimal front, but also to regions of the solution space with good value in *all* objectives. The weak rule, on the other hand, proportionally accepts more solutions unconditionally. This makes it freer to explore the entire search space and also more able to investigate the extremes. This effect was supported by Kubotani et al. [4] who developed a parameterized acceptance probability to allow a single parameter to govern the proportion of unconditionally accepted solutions. A multiple objective knapsack problem and a travelling salesman problem were investigated. From the conclusions, the weak acceptance probability tended to give poorer results than the strong rule. However, with more objectives (> 5), the solution quality of the strong rule deteriorated. The reason being that with more objectives, solution variation is greater. So proportionally, significantly fewer solutions with good performance in all objectives existed. Consequently, in the final stages of the simulation, obtaining solutions which improved all objectives became difficult.

7.5 SA in engineering design

With regard to SA for composite design, relatively few examples have been found. Of the cases that share elements of commonality [11, 13-15], only single objective optimisation has been performed. In addition, as with the majority of the PSO and ACO examples, the search space of the problem is relatively confined.

Ananda Rao et al [11] investigated the optimum design of a multilayer composite plate. The single objective was to maximise its fundamental frequency. Although the ply thickness was variable, because the overall dimensions of the plate were fixed, the number of plies used was dependent upon the individual thickness. Both symmetric and anti-symmetric lay-ups were permitted as well as angles of between -90° and 90° for each ply. For the symmetric laminates, to improve efficiency, only half of the plies needed to be considered in the optimisation. Results showed that the technique was a computationally efficient approach to the design of stiff fibre-reinforced plates. The authors remark that the method could be extended to problems with different materials and more complex geometry.

Deng et al [14] applied SA to an optimal stacking sequence problem for a laminated plate subjected to a uniaxial load. The single objective was to minimise a stress component largely responsible for causing delamination in the edge of the material. The only variable was the angle orientation in which symmetric lay-ups were considered with angles of 0°, 90°, and ±45°. Again, as in the earlier case, only half the plies needed to be considered. Results showed that the algorithm was able to achieve optimal solutions within an acceptable timeframe.

Single objective design of composite laminates for maximum buckling load capacity was examined by Erdal and Sonmez [13]. Only balanced, symmetric laminate lay-ups were considered and the only variable was angle orientation of the plies. The implemented SA algorithm used an adaptive cooling schedule by Ali et al [16]. The results showed that the algorithm was able to locate all of the optimal designs. Expected performance was also given when the design space was enlarged by increasing the number of possible fibre angles.

Di Sciuva et al [15] investigated the optimal design of a laminated plate and a sandwich plate using both SA and a GA. Several test problems were constructed in which the buckling load or number of plies was the single objective to be optimised. Constraint limits were placed on the mass, natural frequency, centre point displacement and buckling load although no more than two were implemented on any given problem. Angle orientation of the plies was always variable although in some cases the optimal number of

plies was additionally sought. Although the problems were quite simplistic given the few parameters, the SA technique was able to produce solutions in good agreement with a GA.

7.6 The developed SA algorithm (*sandwichSA*)

For the PSO and ACO algorithms, a single optimisation technique was developed. However, for the developed SA method, several different alternatives will instead be presented. Yet despite this, due to their interchangeable nature within the entire algorithm, they will collectively be identified as *sandwichSA*. Principally, this approach has been taken because several distinctly different aspects of the technique exist. With the PSO and ACO techniques, variable searching tendency was provided via the weighting factors w, c_1, c_2, α_1 and α_2. However, to a large degree, the shift in search preference with SA is instead achieved by using different acceptance criteria. While other parameterized methods of executing this are recognised [4], including a broad range of techniques ensures a thorough investigation is conducted.

Of the methods to be trialled, it has also been decided that two different cooling schedules be employed: a fixed and adaptive temperature. This will be combined with the four different acceptance criteria: weighted sum, weighted product, strong and weak rule. This gives a total of eight independent methods to carry forward and test for sandwich designs.

It has been pointed out by Suman and Kumar [7] that careful consideration is required to choose an optimal cooling schedule for a problem. In light of this, in the cases when a fixed temperature is utilised, parameters were in part selected in line with their suggestions along with those of Kubotani and Yoshimura [4]. However, they were additionally supported with trial-and-error experimental data in accordance with Cerny [2]. A value of 0.95 was used for the cooling factor, ω. Also, the initial temperature, T_u^0, should be large enough to initially accept all possible atom positions. Due to this, T_u^0 has been made equal in magnitude to the maximum identified value of Δf for each objective value during the first 5% of the total number of planned iterations:

$$\left|T_u^0\right| = \left|\Delta\hat{f}_u^{initial\ 5\%}\right| \tag{7.9}$$

For the second approach, the adaptive cooling schedule as described by Suppapitnarm et al. [9] is used. Here, the initial temperature is simply set as the same magnitude as S_u (the standard deviation of each objective from the current non-dominated set). For both cooling schedules, the temperature was updated (i.e. reduced) after completion of each 5% of the total number of iterations.

Finally, with regard to the acceptance criteria, each objective function was given an equal weighting. So $W_u = 1$ for all cases. This seemed a natural choice as a bias search towards any particular objective is not intended as all objectives are considered to be equally important. Figure 7.3 shows the factors that influence an atoms' decision to accept a particular move. A flowchart of the developed SA algorithm techniques is given in Figure 7.4 with relation to how they fit with the surrounding main structure mentioned earlier (Figure 4.12).

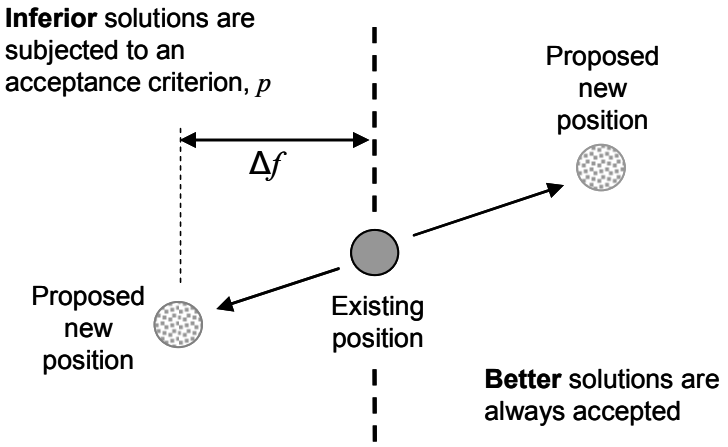

Figure 7.3. Shows the factors which influence an atoms' decision to accept a particular move.

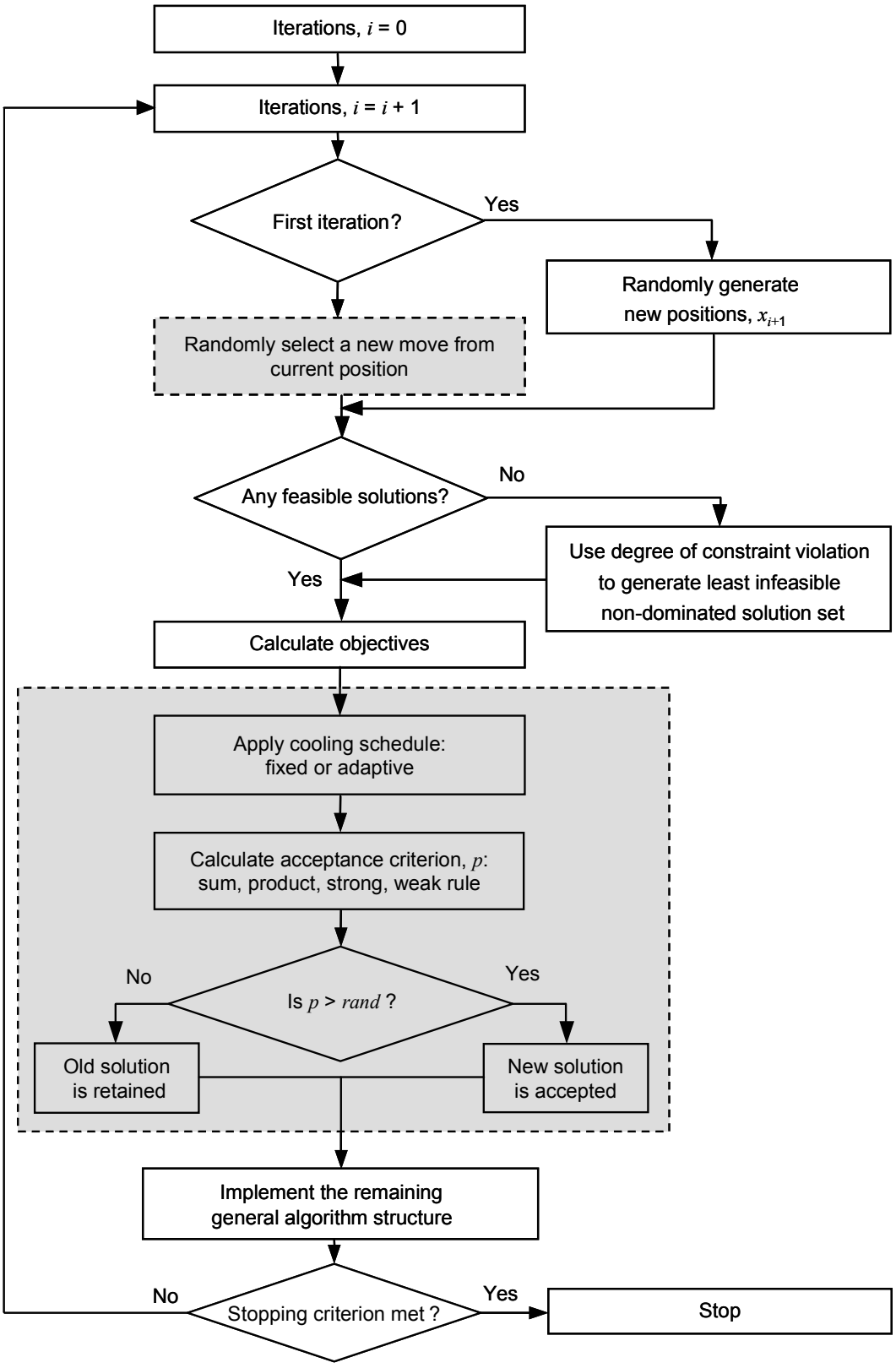

Figure 7.4. Flowchart of the proposed algorithm. Greyed areas mark parts specific to *sandwichSA*.

7.7 Conclusions

In this chapter, a detailed analysis of SA has been conducted with a view to further developing the technique for sandwich design. As with the PSO and ACO techniques, the examples that have the most in common with the type of sandwich optimisation required here only deal with single objective function problems that optimise the stacking sequence of laminates. With regard to SA applied to multiple objective problems, four acceptance criteria have been identified as offering potential candidates for the task. This, alongside two cooling schedules, has led to the development of not just one, but a collection of eight separate SA techniques for sandwich design. These have been termed *sandwichSA*. One advantage of these techniques is that they are largely parameterless. Little problem specific tuning of the techniques is required. Even in the case of the fixed temperature schedule, suitable values have been given. The only alterable parameter is the number of atoms. This is in contrast to the earlier developed *sandwichPSO* and *sandwichACO*, where a single optimisation technique emerged.

Now that each of the optimisation techniques have been fully explored and developed for sandwich design, it is time to implement them on a benchmark case study. This will allow comparison of each of the technique to be made and allow the best to be identified and carried forward for further exploitation.

7.8 References

1. Kirkpatrick, S., Gelatt Jr., C. D., Vecchi, P. M., (1983) Optimization by simulated annealing. *Science* **220**, 498-516.
2. Cerny, V. (1985) Thermodynamical approach to the traveling salesman problem - an efficient simulation algorithm. *Journal of Optimization Theory and Applications* **45**, 41-51.

3. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E. (1953) Equation of state calculations by fast computing machines. *Journal of Chemical Physics* **21**, 1087-1092.
4. Kubotani, H., Yoshimura, K. (2003) Performance evaluation of acceptance probability functions for multi-objective SA. *Computers & Operations Research* **30**, 427-442.
5. Youssef, H., Sait, S.M., Adiche, H. (2001) Evolutionary algorithms, simulated annealing and tabu search: a comparative study. *Engineering Applications of Artificial Intelligence* **14**, 167-181.
6. Tekinalp, O., Karsli, G. (2007) A new multiobjective simulated annealing algorithm. *Journal of Global Optimization* **39**, 49-77.
7. Suman, B., Kumar, P. (2005) A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the Operational Research Society* **57**, 1143-1160.
8. Hajek, B. (1988) Cooling schedules for optimal annealing. *Mathematics of Operations Research* **13**, 311-329.
9. Suppapitnarm, A., Seffen, K.A., Parks, G.T., Clarkson, P.J. (2000) Simulated annealing algorithm for multiobjective optimization. *Engineering Optimization* **33**, 59-85.
10. Jeon, Y.-J., Kim, J.-C. (2004) Application of simulated annealing and tabu search for loss minimization in distribution systems. *International Journal of Electrical Power & Energy Systems* **26**, 9-18.
11. Ananda Rao, M., Ratnam, C., Srinivas, J., Premkumar, A. (2002) Optimum design of multilayer composite plates using simulated annealing. *Proceedings of the Institution of Mechanical Engineers Part L: Journal of Materials: Design and Applications* **216**, 193-197.
12. Ulungu, E.L., Teghem, J., Fortemps, P. H., Tuyttens, D. (1999) MOSA method: A tool for solving multiobjective combinatorial optimization problems. *Journal of Multi-Criteria Decision Analysis* **8**, 221-236.
13. Erdal, O., Sonmez, F.O. (2005) Optimum design of composite laminates for maximum buckling load capacity using simulated annealing. *Composite Structures* **71**, 45-52.

14. Deng, S., Pai, P.F., Lai, C.C., Wu, P.S. (2005) A solution to the stacking sequence of a composite laminate plate with constant thickness using simulated annealing algorithms. *International Journal of Advanced Manufacturing Technology* **26**, 499-504.
15. Di Sciuva, M., Gherlone, M., Lomario, D. (2003) Multiconstrained optimization of laminated and sandwich plates using evolutionary algorithms and higher-order plate theories. *Composite Structures* **59**, 149-154.
16. Ali, M.M., Torn, A., Viitanen, S. (2002) A direct search variant of the simulated annealing algorithm for optimization involving continuous variables. *Computers and Operations Research* **29**, 87-102.

8 Comparison of the developed sandwich optimisation algorithms

Three algorithms (particle swarm optimisation (PSO), ant colony optimisation (ACO) and simulated annealing (SA)) have been identified as offering excellent potential for sandwich optimisation. The previous three chapters saw the development of these techniques specifically for this purpose. These have been termed *sandwichPSO*, *sandwichACO* and *sandwichSA*. Now this has been done, it is time to put these techniques to the test. Here, a benchmark case study involving the optimisation of a sandwich beam is presented. This will allow comparison to be made and determine which is the most suitable to be carried forward for further experimentation.

8.1 The benchmark case study

In order to evaluate the three optimisation algorithms, a relatively straightforward sandwich problem was adopted as a case study (Figure 8.1). It consisted of a simply-supported sandwich beam with a fixed span, l, under a uniformly distributed load, q. The width, b, and the total thickness, h, of the sandwich were fixed at 50 mm. The length of the beam, L, coincided with the span and was fixed at 550 mm. The upper and lower facings of the sandwich were identical.

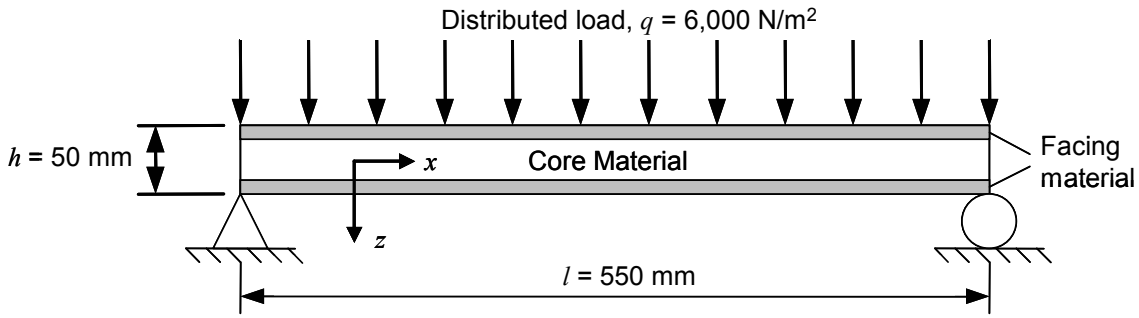

Figure 8.1. The benchmark sandwich problem used to evaluate the optimisation algorithms.

8.1.1 Design variables

The problem illustrated in Figure 8.1 contains three primary design variables: the sandwich facing material, the sandwich facing thickness, and the sandwich core material. These are the parameters for which optimal values were sought.

For the facings, a range of material options were available including various aluminiums, steels, fibre-reinforced polymers and wood products. Furthermore, for the case of fibre-reinforced polymer facings, it was possible to select from a range of fibre and matrix materials, as well as specifying the fibre volume fraction, the number of plies in the laminate, and the orientation angle of each ply (0^o, $\pm 45^o$, or 90^o). Hence the use of fibre-reinforced polymer facings introduced five additional design variables to the problem. Similarly, a number of core material options were available including balsa wood and a variety of polymer foams. In total, there were 16 different core and facing materials to choose from (Table 8.1). This material database, when coupled with the fibre-reinforced polymer laminate design options, provided a very large number of potential sandwich material combinations.

A range of 0.25 – 5.00 mm was specified for the facing thickness. Different discrete thicknesses were permitted within this range for different materials to reflect real-world availability.

Table 8.1. A list of all materials used in the benchmark sandwich beam optimisation. Data taken from CES selector software [1].

All materials	Cost (€/kg)	Density (kg/m^3)	Young's Modulus (GPa)	Shear Modulus (GPa)	Tensile Strength (MPa)	Compressive Strength (MPa)	Shear strength (Gpa)	Poisson's Ratio	Thermal conductivity (W/mK)
Cores									
Phenolic foam	5.4	120	0.065	0.026	-	1.1	0.42	-	0.03
Polystyrene foam	1.6	50	0.028	0.009	-	0.9	0.55	-	0.04
Polymethacrylimide foam	46	75	0.088	0.027	-	1.4	1.1	-	0.03
Polyvinylchloride foam	11	30	0.03	0.014	-	0.3	0.43	-	0.023
Balsa (wood)	7.7	190	4.7	0.035	-	11	11	-	0.11
Non-reinforced facings									
Aluminum (5000 series)	1.1	2700	72	-	231	190	120	-	150
Steel (ultra high strength)	0.48	7900	210	-	1000	690	510	-	40
Fir (wood)	0.89	440	13	-	69	42	34	-	0.22
Hardboard	0.42	1100	9	-	51	56	23	-	0.33
Plywood	0.89	750	5.5	-	53	28	26	-	0.33
Reinforced fiber facings									
Carbon (high modulus)	26	1800	380	170	2400	3700	1200	0.11	140
Carbon (high strength)	17	1800	240	100	4700	5000	2300	0.11	140
E Glass	1.5	2600	79	33	2000	4500	1000	0.22	1.3
Matrices									
Epoxy	1.3	1300	2.4	0.86	67	140	34	0.4	0.19
Phenolic	1	1300	3.8	1.4	48	93	24	0.39	0.15
Polyester	1.1	1200	3.2	1.2	66	170	33	0.39	0.29

8.1.2 Design objectives

For the purposes of the case study there were two design objectives: to maximise the flexurally rigidity per unit mass of the beam, D_m, and to maximise the flexural rigidity per unit cost, D_c. The respective objective functions are given in Equations (8.1) and (8.2).

$$D_m = \frac{D}{M} \tag{8.1}$$

$$D_c = \frac{D}{C} \tag{8.2}$$

where the flexural rigidity of the sandwich beam, D, is given by [2]:

$$D = E_f \frac{bt_f^{\,3}}{6} + E_f \frac{bt_f d^2}{2} + E_c \frac{bt_c^{\,3}}{12} \tag{8.3}$$

in which E is Young's modulus, t is thickness, d is the distance between the centrelines of opposing facings (= $t_f + t_c$), and subscripts f and c pertain to the sandwich facings and core respectively.

The overall sandwich mass, M, in Equation (8.1) was calculated using:

$$M = Lb\left(2\rho_f t_f + \rho_c t_c\right) \tag{8.4}$$

where ρ is density.

The overall sandwich cost, C, in Equation (8.2) was calculated as:

$$C = Lb\left(2c_f \rho_f t_f + c_c \rho_c t_c\right) \tag{8.5}$$

where c is the cost per unit mass of a given material.

8.1.3 Design constraints

The *direct* constraints applied to the problem were as follows:

- Sandwich width, b = 50 mm.
- Overall sandwich thickness, h = 50 mm.
- Facing thickness range = 0.25 – 5.00 mm.
- Sandwich span, l = 550 mm.

The *dependent* constraints, (i.e. those constraints that were functions of the problem's variables) were:

- No failure of the sandwich by tensile or compressive facing failure, shear or compressive core failure, or wrinkling of the upper facing.
- Maximum allowable deflection of 2 mm was permitted for the sandwich under a uniformly distributed load of 6000 N/m^2.
- Maximum overall thermal conductivity of the sandwich, λ_{total} = 0.05 W/m.K.

The properties and performance of the sandwich materials and their constituents were estimated using analytical 'textbook' solutions. The fibre-reinforced polymer facing stiffness properties were estimated using classical laminate theory. This is well described in many standard texts (e.g. Gibson [3], Matthews & Rawlins [4]). To simplify the laminate equations, only balanced, symmetric, quasi-isotropic laminates were considered.

The mechanics of the sandwich beams were estimated using basic sandwich theory, as described, for example, by Allen [2] and Zenkert [5]. With respect to failure prediction, the tensile and compressive stresses in the faces, and the shear and compressive stresses in the core were compared against the respective material strengths.

For the facings, the maximum tensile and compressive stresses induced due to bending, σ, were calculated as:

$$\sigma = \frac{BE_f h}{2D} \tag{8.6}$$

Due to symmetry, stresses generated in the upper and lower facings are the same magnitude but opposite in sign. The maximum bending moment at the midpoint is given by B and calculated as:

$$B = \frac{qbl^2}{8} \tag{8.7}$$

For the non-reinforced faces, failure was predicted using the von Mises criterion. For the fibre-reinforced polymer facings, first ply failure was estimated using the Tsai-Hill criterion.

The maximum shear stress induced in the midplane of core, Q_c, is calculated as:

$$Q_c = \frac{qbl}{2D}\left(\frac{E_f t_f d}{2} + \frac{E_c t_c^2}{8}\right) \tag{8.8}$$

The stress to cause wrinkling in the upper face was also considered using the expression provided by Zenkert [5]:

$$\sigma_{wrinkling} = \frac{\sqrt[3]{E_f E_c G_c}}{2} \tag{8.9}$$

where G is the shear modulus.

The equation for the maximum midpoint deflection, δ, is given by Allen [2] and may be written as:

$$\delta = \frac{5qbl^4}{384D} + \frac{t_c q l^2}{8G_c d^2} \tag{8.10}$$

The overall through-thickness thermal conductivity of the sandwich, λ_{total}, was estimated using the expression provided by Ashby [6].

$$\lambda_{total} = \left(\frac{2t_f / h}{\lambda_f} + \frac{1 - 2t_f / h}{\lambda_c} \right)^{-1} \tag{8.11}$$

Whilst more complex and accurate methods of predicting the behaviour of a sandwich beam are available, their use would not have fundamentally altered the manner in which the optimisation was performed. It would just have required the substitution of one sandwich design algorithm for another within the optimisation process. For the purposes of this study, the textbook analytical solutions were considered sufficient for evaluating the three optimisation techniques.

8.2 Evaluation methodology: performance metrics

In order to provide a quantitative means of evaluating and comparing the PSO, ACO and SA optimisation algorithms, a number of metrics were adopted to benchmark their performance. These metrics were *error ratio*, *generational distance* and *spread*. They are described in turn below.

8.2.1 Error ratio

An optimisation algorithm's *error ratio*, as described by Van Veldhuizen and Lamont [7], is a measure of its ability to identify non-dominated solutions at the true Pareto-optimal front. The error ratio gives a quick indication of the proportion of solutions, ψ, in the

known non-dominated set, Ψ, that are not Pareto-optimal. So, larger values of error ratio imply comparatively worse algorithm performance. Error ratio has been used by previous researchers [8-10] to support the quantification of an optimisation algorithm's performance. The expression for error ratio is shown as Equation (8.12). If a given solution, ψ, is found to be included in the true Pareto-optimal set then $e_\psi = 0$. Otherwise, $e_\psi = 1$.

$$\text{Error ratio} = \frac{\sum_{\psi=1}^{\Psi} e_\psi}{\Psi} \tag{8.12}$$

8.2.2 Generational distance

One of the limitations of the error ratio metric is that it does not give an indication of how far from the true Pareto-optimal front a given solution is. However, the *generational distance* metric, which is again described by Van Veldhuizen and Lamont [7], does provide this information:

$$\text{Generational distance} = \frac{\sqrt{\sum_{\psi=1}^{\Psi} d_\psi{}^2}}{\Psi} \tag{8.13}$$

In Equation (8.13) d_ψ is the Euclidian distance (in objective space) between the ψ^{th} solution and the nearest member of the true Pareto-optimal set. Hence larger values of generational distance indicate that solutions are comparatively further away from the Pareto-optimal front. Metrics very similar to the one defined in Equation (8.13) have been used by a number of other researchers [10-14] to evaluate a range of optimisation algorithms.

8.2.3 Spread

The third metric is *spread*. Several definitions of spread have been noted. The expression proposed by Zitzler et al [14] has been used here. Spread monitors the breadth of a non-

dominated solution set based on the difference between its maximum and minimum objective values:

$$\text{Spread} = \sqrt{\sum_{u=1}^{U} \left| \max(\Psi_u) - \min(\Psi_u) \right|} \tag{8.14}$$

To summarise, the generational distance metric indicates how close the identified solutions are to the true Pareto-optimal set, whereas the error ratio simply gives an indication of the number of solutions that match the true Pareto-optimal set. Spread, does not require any knowledge of the true Pareto-optimal set. Instead it provides an indication of an algorithm's ability to seek extreme values.

8.3 Application of the optimisation algorithms to the sandwich case study

From the previous three chapters, it was apparent that each of the developed algorithms all had parameters that need to be tuned to a particular problem. These include the weighting factors associated with *sandwichPSO* and *sandwichACO*, and the various cooling schedules and acceptance criteria for the *sandwichSA*. Therefore, a systematic study was undertaken to evaluate the performance of the algorithms under a broad range of conditions.

For the *sandwichPSO* algorithm, five parameters can be adjusted. These are w (the inertial influence weighting factor), c_1 (the cognitive influence weighting factor), c_2 (the social influence weighting factor), μ (the probability of a position-randomising gust of wind), and the number of particles in the swarm. Based on the successful application of PSO in previous studies [12, 15, 16] the following default values were assumed: $c_1 = c_2 = 2$, $w = 0.01$, $\mu = 0.2$, and the number of particles = 20. These default values were then systematically adjusted according to the following schedule:

- The effect of the inertial influence weighting factor, w, was evaluated using values of 0.001, 0.01, 0.02, 0.04, 0.06, 0.08 and 0.1.

- The cognitive and social influence weighting factors were altered in tandem according to the relationship $c_1 + c_2 = 4$, where $c_1 = 1$, 2 and 3. This allowed the effect of adjusting the relative weighting between the cognitive and social parameters to be observed.

- The wind factor, μ, was altered from 0 – 1 in increments of 0.2.

- Simulations with the number of particles set at 1, 5, 10, 20, 50, 100, 1,000 and 10,000 were trialled.

For the *sandwichACO* algorithm, four parameters can be altered – the weighting factors α_1 and α_2, the pheromone evaporation rate, ρ, and number of ants. The default values of these parameters were taken as $\alpha_1 = \alpha_2 = 1$, $\rho = 0.1$, and the number of ants = 20. These default values were then systematically adjusted as follows:

- The α_1 and α_2 weighting factors were adjusted according to the relationship $\alpha_1 + \alpha_2 = 2$ where $\alpha_1 = 0$, 0.5, 1, 1.5 and 2. This variation was sufficient for observing the effects of an ant preference shift between following popular trails and following those that are known to lead to global best solutions.

- The evaporation rate, ρ, was altered from 0 – 0.8 in increments of 0.2.

- Ant colony populations of 1, 5, 10, 20, 50, 100, 1,000 and 10,000 were also trialled.

For the *sandwichSA* algorithm, the only parameter available for adjustment is the number of atoms. As with the other two optimisation techniques, populations of 1, 5, 10, 20, 50, 100, 1,000 and 10,000 were trialled. However, as described earlier, there are a number of different available options for both the cooling schedule (fixed and adaptive) and the acceptance criterion (weighted sum, weighted product, strong rule, and weak rule) and these were all investigated.

For each of the three optimisation algorithms, a given trial (simulation) was allowed to run for 5 minutes (on a standard Pentium 4, 3 GHz desktop PC), and each trial was repeated 10 times. From the results of each trial, the three performance metrics (error ratio, generational distance and spread) were calculated, and mean values across the 10 runs were taken.

8.4 Results and discussion

8.4.1 Estimation of the true Pareto-optimal set

In order to allow the error ratio and generational distance metrics to be calculated, the true Pareto-optimal set is required. However, for most optimisation problems of the type considered here, the true Pareto-optimal set is rarely known with absolute certainty. That is, after all, the reason for performing the optimisation in the first place. It was therefore necessary to obtain a good *estimate* of the Pareto-optimal set to support the calculations of the metrics.

In this study, the true Pareto-optimal set was estimated by pooling the results from all the simulations performed using all three algorithms. From this universal set of results, the non-dominated sub-set of solutions was identified, and this was regarded as the accepted true Pareto-optimal set. This approach has been taken elsewhere [13, 14]. The details of this true Pareto-optimal set are shown in Table 8.2. Figure 8.2 shows the optimal set in terms of a graph of sandwich flexural rigidity per unit cost, D_c, against sandwich flexural rigidity per unit mass, D_m.

Table 8.2. The obtained Pareto-optimal solutions acquired for the benchmark comparison optimisation.

Obtained Pareto-optimal solutions

Core	Facing	Fibre fraction, v	t_f (mm)	M (kg)	C (€)	D_m (kNm^2/kg)	D_c (kNm^2/£)
Polystyrene	HM carbon / phenolic	0.7	1.0 – 4.0	0.16 – 0.33	1.98 – 5.72	53 – 70	4
Polystyrene	HM carbon / polyester	0.7	2.0 – 3.0	0.24 – 0.33	3.85 – 5.73	66 – 70	4
Polystyrene	Steel	-	2.0 – 3.2	0.88 – 1.48	0.49 – 0.77	26	47 – 49
PVC	HM carbon / phenolic	0.7	2.0 – 3.0	0.22 – 0.31	4.16 – 6.02	73 – 75	4
PVC	HM carbon / polyester	0.7	3	0.31	6.02	75	4
PVC	Steel	-	1.5 – 3.0	0.68 – 1.35	0.72 – 1.02	26 – 27	25 – 35

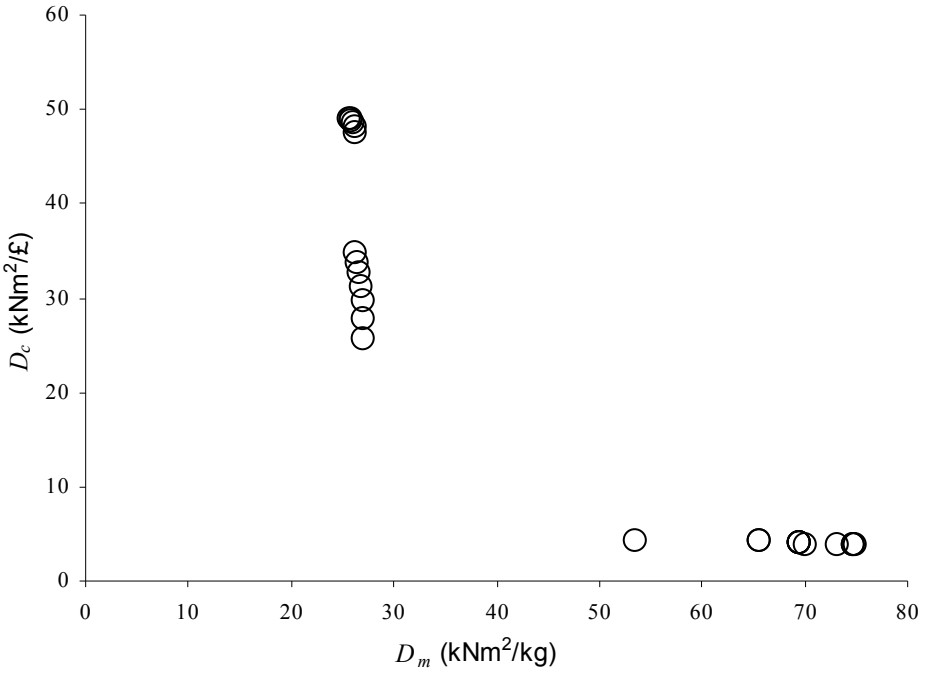

Figure 8.2. The accepted true Pareto-optimal set of solutions derived from the collated results of all simulations.

In Figure 8.2, it can be seen that the Pareto-optimal front is discontinuous and consists of two distinct regions: a lower cost / higher mass front towards the top-left ($D_c > 20$), and a lower mass / higher cost front towards the bottom-right ($D_c < 10$).

The first region, towards the top-left of Figure 8.2, corresponds to those sandwich constructions with steel facings of thicknesses between 1.50 – 3.25 mm and either a PVC or polystyrene foam core. For the vast majority of simulations, all three algorithms were able to identify these solutions with little difficulty.

The second, and more interesting, region towards the bottom-right of Figure 8.2 corresponds to sandwich constructions with fibre-reinforced polymer facings. The accepted true Pareto-optimal solutions at this point in the front employed carbon fibre-reinforced polyester or phenolic facings with a thickness of between 1 mm and 4 mm in conjunction with either a PVC or polystyrene foam core. It was this region that provided the biggest challenge for the three optimisation algorithms, and which best illustrates their relative strengths and weaknesses for sandwich design.

8.4.2 Identification of sandwich optimisation complexities

In Chapter 2, some complexities likely to cause difficulty with sandwich optimisation were speculated. Analysing Figure 8.2 shows that most of the anticipated issues are present. These will now be described.

One of the more obviously features of Figure 8.2 is that the Pareto-optimal front shows discontinuities. This is due to the discrete nature of the materials and geometries provided.

Multimodality has also shown to be present due to the many facing-core material combinations available. This is more noticeable in the top-left portion of Figure 8.2 where the optimal solutions plot-out two separate curved profiles. From Table 8.2, these correspond to steel / polystyrene and steel / PVC for a range of facing thicknesses.

In relation to the convexity of the problem, noting that the objectives are maximised, the Pareto-optimal front has an overall non-convex net trend. However, the individual curves that form each facing-core combination (observable in the top-left of Figure 8.2), on their own, they are themselves convex.

To an extent, deception also plays a part and is due to the non-reinforced facings. Several reasons for this exist. First note that a clear divide in objective value exists between them and the reinforced facings (Figure 8.2). Also observe that compared with the reinforced materials, there are significantly more that are non-reinforced. Furthermore, the reinforced region of the design space is far more heavily constrained due to the requirements with obtaining feasible stacking sequences. The combined affect of these factors draws the search towards sandwiches with non-reinforced faces. Hence, in this way, the non-reinforced facing materials act as deceptive local optima.

In addition, it may also be noted that due to the heavily constrained nature of obtaining feasible stacking sequences for reinforced faces, to a large extent, they exist as isolated points.

8.4.3 Performance of the *sandwichPSO* algorithm

Figure 8.3 shows the variation of the error ratio, generational distance and spread metrics for the particle swarm optimisation algorithm. It can be seen that *sandwichPSO* was relatively insensitive to changes in the various parameters, indicating that it is a robust technique.

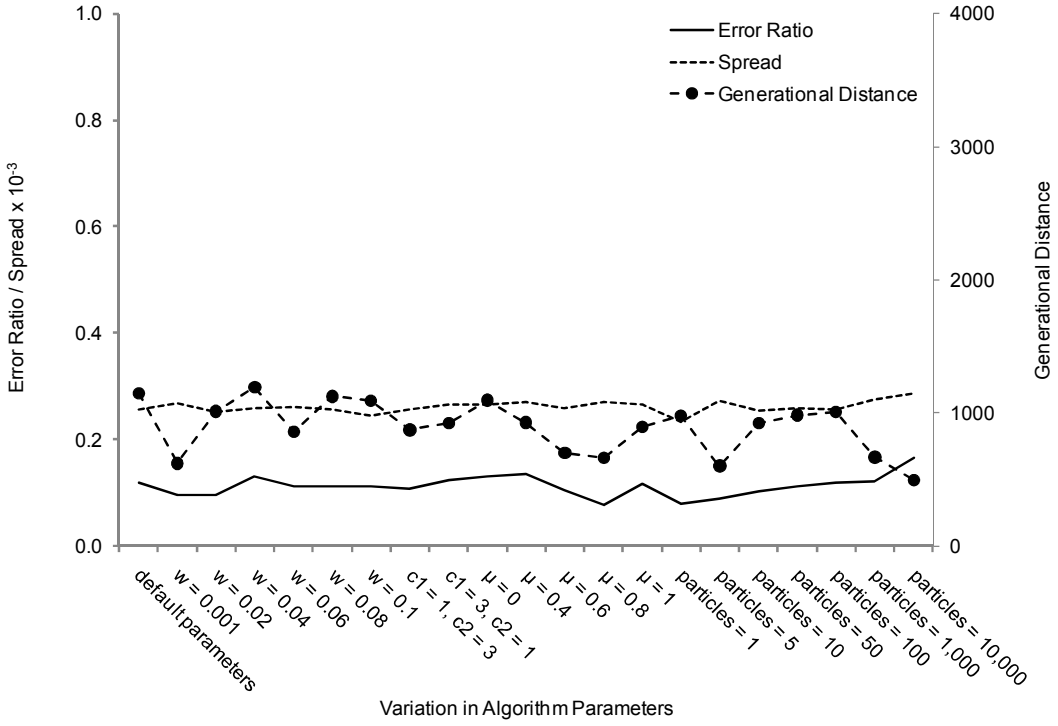

Figure 8.3. Benchmarked performance of the *sandwichPSO* algorithm.

8.4.4 Performance of the *sandwichACO* algorithm

Figure 8.4 shows the same performance data for the ant colony optimisation algorithm. It can be seen that, in comparison to *sandwichPSO*, *sandwichACO* exhibited a much wider variation in performance. Many of the parameter combinations exhibited very low values of error ratio and generational spread, indicating a very good ability to identify the true Pareto-optimal set. The best set of results was obtained with $\alpha_1 = \alpha_2 = 1$, $\rho = 0.2$ and 20 ants. For this particular combination, each of the 10 trials generated non-dominated sets that matched the accepted true Pareto-optimal set perfectly. This demonstrates excellent performance and repeatability.

However a few particular *sandwichACO* parameter combinations yielded very poor solution sets. These included those models in which popular paths were favoured over global best solution paths (i.e. $\alpha_1 > \alpha_2$), those with low levels of residual pheromone due to a high evaporation rate ($\rho = 0$), and those with a large number of ants (> 100). In these cases, both the error ratio and generational distance metrics were found to deteriorate markedly, although spread was much less sensitive. Whilst the drop-off in performance with disincentivised global best solution paths and a high evaporation rate seems intuitive, the same cannot be said for the drop-off with large ant numbers. In fact, the degradation in performance with large ant numbers was due to the limited 5 minute algorithm run time. With large numbers of ants there were insufficient iterations available for the algorithm to complete the search properly.

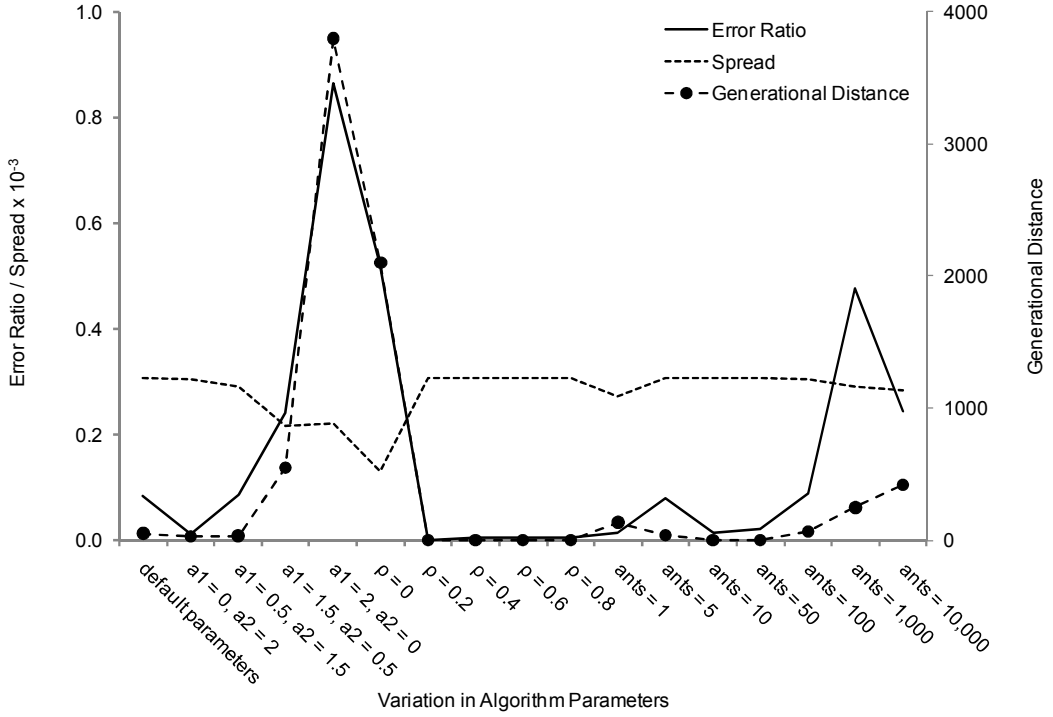

Figure 8.4. Benchmarked performance of the *sandwichACO* algorithm.

8.4.5 Performance of the *sandwichSA* algorithm

The performance of the *sandwichSA* algorithm was broadly comparable to *sandwichPSO*. Across all the *sandwichSA* formulations trialled (Figure 8.5), the quality of the solution sets was relatively insensitive to the parameters investigated. When comparing the two

different cooling schedules, the adaptive temperature reduction was found to provide marginally better solutions. In terms of acceptance criteria, the weighted product and weak rules performed better than the weighted sum and strong rules.

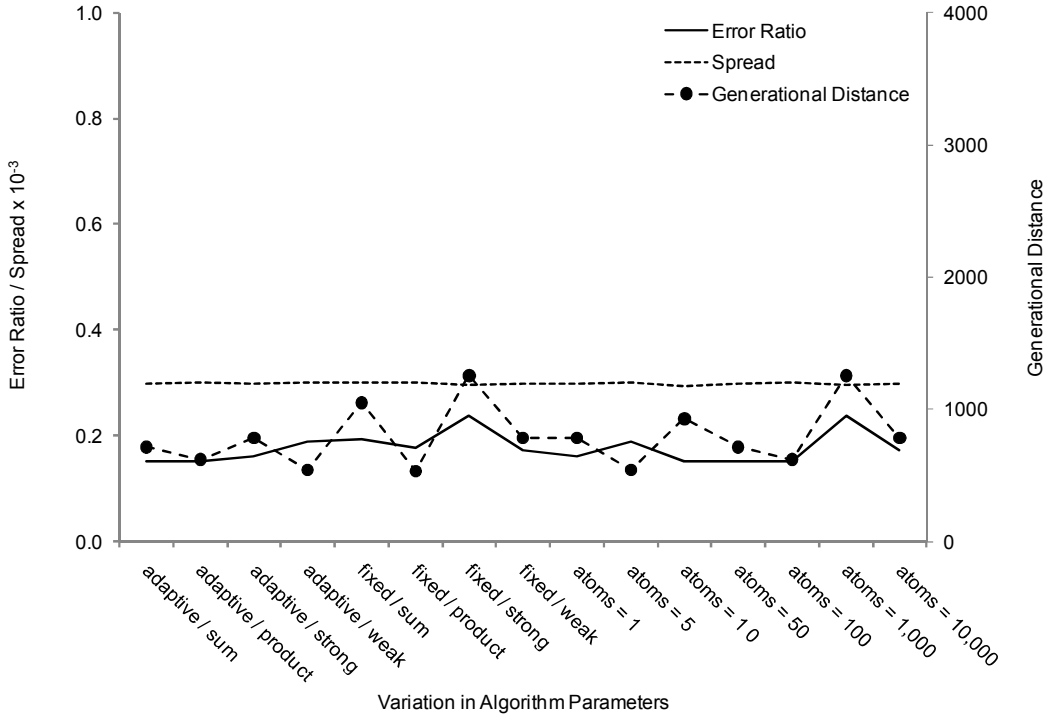

Figure 8.5. Benchmarked performance of the *sandwichSA* algorithm. The results reported for varying numbers of atoms are all based on adaptive / strong.

8.4.6 Comparative performance of the three optimisation algorithms

Figure 8.6 shows representative "good" non-dominated solutions produced by each of the three algorithms in the fibre-reinforced polymer facing region of the Pareto-optimal front. It is immediately clear that the *sandwichACO* algorithm has performed better than *sandwichPSO* and *sandwichSA*, both in terms of its ability to identify constructions with high stiffness per unit cost and high stiffness per unit mass, and in terms of the consistency of its results.

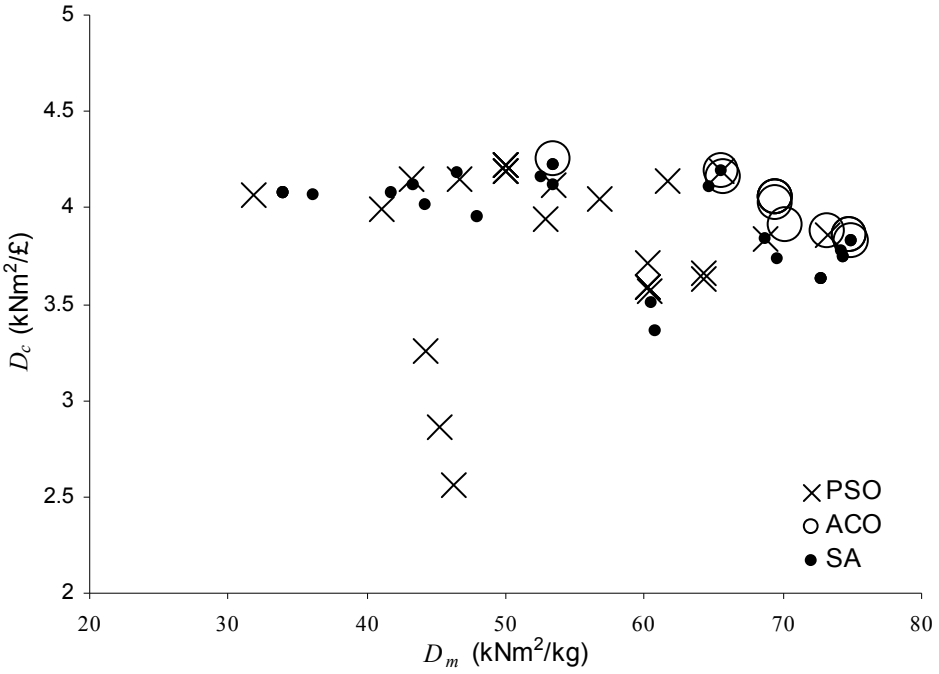

Figure 8.6. Comparative performance of the three optimisation algorithms in the fibre-reinforced polymer facing region of the accepted true Pareto-optimal front.

In order to eliminate the constrained (5 minute) solution time as a possible explanation for the relatively poorer performance of the *sandwichPSO* and *sandwichSA* algorithms, the simulations depicted in Figure 8.6 were re-run, with the same set of parameters, for as long as was needed to identify the accepted true Pareto-optimal set. Whilst the *sandwichACO* algorithm completed 10 simulation runs in an average time of 111 seconds, the SA algorithm took 11½ hours, and the *sandwichPSO* algorithm was abandoned after 24 hours having failed to find the complete Pareto-optimal set. So, on the basis of this particular study, solution time was discounted as being the limiting factor in the inferior performance of the *sandwichPSO* and *sandwichSA* algorithms. Rather, there must be some relative lack of suitability in the algorithms themselves.

One would perhaps have anticipated the poorer performance of the *sandwichSA* algorithm given the lack of information-sharing between search agents. However, the inability of the *sandwichPSO* algorithm to identify the Pareto-optimal set is more surprising and requires a more detailed comparison of the *sandwichACO* and *sandwichPSO* algorithms.

The primary controlling factor in the *sandwichACO* algorithm is the pheromone level. It is the sole factor that drives the search agents towards better solutions. The ants themselves

have no influence. They retain no memory from previous iterations. Their path decision-making is only influenced by the current levels of pheromone in the variable space. This means that they are generally good at adapting to emerging trends during the search. They have no inherent resistance or reluctance to shun such trends.

Conversely, the particles themselves in the *sandwichPSO* algorithm do have a significant influence on the progress of the algorithm. A particle's current position, its inertial term (i.e. how fast it is flying through the variable space and in which direction), and its cognitive term (i.e. its memory of its own previous best solutions) are important factors. This is in contrast to the *sandwichACO* algorithm which is entirely social. Overall, it would appear that the information that a given particle retains from previous iterations has a restricting effect on its overall searching capability, at least for the sandwich problem considered here. Figure 8.3 contains slight supporting evidence for this fact in that the *sandwichPSO* algorithm's generational distance metric showed a noticeable improvement for simulations with a large number of particles (>1,000). In these cases, given the fixed 5 minute runtime, there was a correspondingly lower number of completed iterations. This meant that a significantly greater proportion of the searching took place in the more random early iterations, before the various influence factors had a chance provide a significant effect. Figure 8.7 illustrates the overall weakness exhibited by *sandwichPSO* in comparison to *sandwichACO*. It can be seen that, over the course of a typical simulation, there is a tendency for the particles in *sandwichPSO* to be drawn towards the more easily identifiable mono-material sandwich facing solutions in the top-left of the objective space, at the expense of the more complex fibre-reinforced polymer facing region in the bottom-right. Conversely, *sandwichACO* is better able to resist this pull.

The contrasting performance of the algorithms is also illustrated by the local performance metric data summarised in Table 8.3. It can be seen that in the mono-material facing region, all the algorithms are able to identify all the Pareto-optimal solutions leading to error ratio and generational distance metrics of zero. However, in the fibre-reinforced polymer facing region, the *sandwichPSO* (and *sandwichSA*) algorithms perform poorly, with an error ratio of around 90% indicating that only around 1 in 10 of the solutions identified was Pareto-optimal.

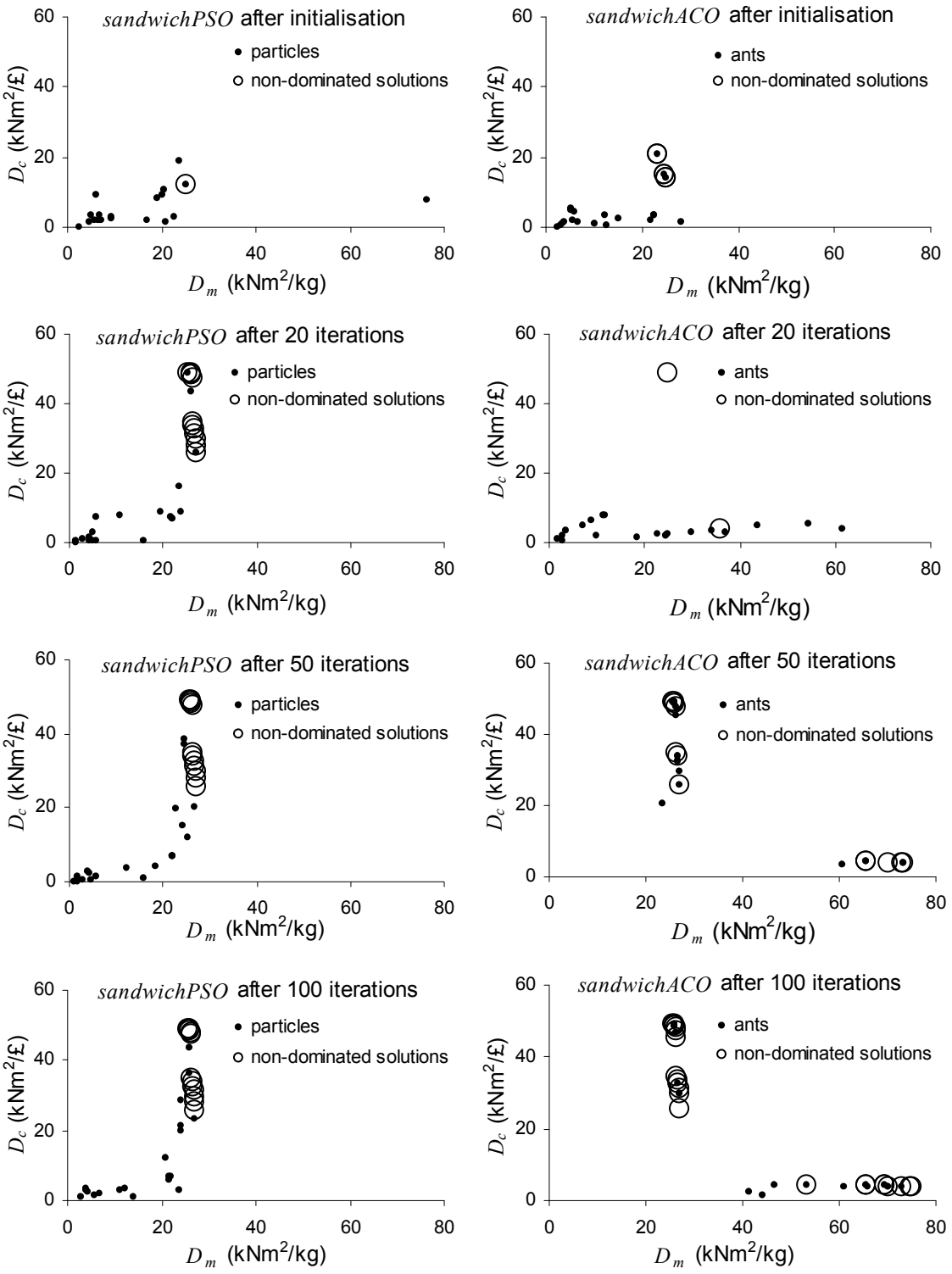

Figure 8.7. Comparative evolution of the *sandwichPSO* algorithm (left) and the *sandwichACO* algorithm (right). The graphs show the positions of the search agents at various iterations, along with current identified non-dominated solution set.

Table 8.3. Localised performance metrics for the optimisation algorithms.

		sandwichPSO	*sandwichACO*	*sandwichSA*
D_c >20 (mono-material facings)	Error ratio	0	0	0
	Generational distance	0	0	0
	Spread	157	157	157
D_c < 10 (fibre-reinforced polymer facings)	Error ratio	0.90	0	0.89
	Generational distance	2,586	0	3,694
	Spread	112	148	157

The reason why the *sandwichPSO* algorithm struggles with the fibre-reinforced polymer facings is the relatively low ratio of feasible-to-infeasible solutions in this region of the variable space. This is due primarily to the assumed constraint that only balanced, symmetric laminates should be considered. For example, for a four-ply facing laminate with available ply orientation angles of 0^o, $+45^o$, -45^o and 90^o, the ratio of feasible-to-infeasible solutions is 1.6%. For larger numbers of plies, this ratio falls still further. The overall effect is that there are far more infeasible solutions in the fibre-reinforced polymer facing region of the variable space that act to disincentivise further searching, than feasible solutions that promote it. The *sandwichACO* is less susceptible to this effect because the positions of the ants are reset (back to the nest) at the beginning of each iteration. Unlike *sandwichPSO*, their positions are not continually updated from one iteration to the next. Furthermore, lingering residual pheromone, which may still be present several iterations after which it was first deposited, also provides an incentive for *sandwichACO* ants to revisit regions with a low proportion of feasible solutions. No such incentive is provided by *sandwichPSO*.

A similar tendency was observed in a recent study by Zheng et al [17] who compared the performance of a PSO and ACO for minimising the production of nitrogen oxides from a coal-fired utility boiler. It was found that the PSO performed less well, with a marked susceptibility to becoming trapped in local minima rather than fully searching the entire variable space.

8.5 Conclusions

Of the three population-based optimisation techniques considered in this thesis, *sandwichACO* was found to be the most suitable for the optimisation of a stiff composite sandwich beam in bending with multiple objectives of low mass and low cost. Provided that the algorithm was not set-up to favour purely popular solution paths over global best solution paths (i.e. $\alpha_1 > \alpha_2$ was avoided), and that with large numbers of ants the algorithm was given sufficient time to run, *sandwichACO* proved to be a highly efficient and effective technique. Due to this, it has been decided that *sandwichACO* will be the sole technique to be carried forward and utilised for the extended case study in the next chapter.

The PSO and SA algorithms were both found to be robust tools that were largely insensitive to variations in their influencing parameters. However, both *sandwichPSO* and *sandwichSA* struggled to identify local optimum solutions in regions of the objective space in which the ratio of feasible-to-infeasible solutions was low, as characterised by multi-ply, oriented fibre-reinforced polymer sandwich facing laminates.

The extent to which sandwich design has been investigated here includes a significant proportion of the known complexities in the field of optimisation. Interestingly, the results have found the design space to contain such complexities as multimodality, deceptive optima, isolated points, discontinuities, non-uniformly distributed Pareto-optimal sets and both convex and non-convex Pareto-optimal fronts. With this in mind, considerable appreciation is given to the demands required and demonstrates the competitive ability of the *sandwichACO* algorithm.

8.6 Publications

Hudson, C.W., Carruthers, J.J., Robinson, A.M. (2010) A comparison of three population-based optimisation techniques for the design of composite sandwich materials. *Journal of Sandwich Structures and Materials*. **Accepted for publication**

8.7 References

1. (2005) CES Selector version 4.6. Granta Design Ltd.
2. Allen, H.G. (1969) *Analysis and design of structural sandwich panels*. Pergamon Press, London.
3. Gibson, R.F. (1994) *Principles of composite material mechanics*. McGraw-Hill, New York.
4. Matthews, F.L., Rawlings, R.D. (1994) *Composite materials: Engineering and science*. Chapman & Hall, London.
5. Zenkert, D. (1995) *An introduction to sandwich construction*. EMAS Publishing.
6. Ashby, M.F. (2005) *Materials selection in mechanical design*. Elsevier Butterworth-Heinemann, Italy.
7. Van Veldhuizen, D.A., Lamont, G.B. (1999) Multiobjective evolutionary algorithm test suites. *Proceedings of the ACM Symposium on Applied Computing*.
8. Ulungu, E.L., Teghem, J., Fortemps, P. H., Tuyttens, D. (1999) MOSA method: A tool for solving multiobjective combinatorial optimization problems. *Journal of Multi-Criteria Decision Analysis* **8**, 221-236.
9. Baran, B., Schaerer, M. (2003) A multiobjective ant colony system for vehicle routing problem with time windows. *IASTED International Multi-Conference on Applied Informatics*.
10. Coello Coello, C.A., Pulido, G.T., Lechuga, M.S. (2004) Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation* **8**, 256-279.
11. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**, 182-197.
12. Reddy, M.J., Kumar, D.N. (2007) An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design. *Engineering Optimization* **39**, 49-68.
13. Garcia-Martinez, C., Cordon, O., Herrera, F. (2007) A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research* **180**, 116-148.

14. Zitzler, E., Deb, K., Thiele, L. (2000) Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation* **8**, 173-195.
15. Hudson, C.W., Carruthers, J.J., Robinson, A.M. (2009) Application of particle swarm optimisation to sandwich material design. *Plastics, Rubber and Composites* **38**, 106-110
16. Kennedy, J., Eberhart, R. (1995) Particle swarm optimization. *IEEE International Conference on Neural Networks - Conference Proceedings*.
17. Zheng, L.-G., Zhou, H., Cen, K.-F., Wang, C.-L. (2009) A comparative study of optimization algorithms for low NOx combustion modification at a coal-fired utility boiler. *Expert Systems with Applications* **36**, 2780-2793.

9 Optimisation of a rail vehicle floor panel using ant colony optimisation (ACO)

The comparison study in the previous chapter showed that the ant colony optimisation technique (ACO) was the most competitive out of the three algorithms investigated on the sandwich beam problem. Due to this, it will be utilised further and implemented on a more demanding problem. The case study in this chapter will investigate the application of *sandwichACO* on a sandwich plate for use as a rail vehicle floor panel.

9.1 Introduction

Within the rail industry, lightweighting is becoming an increasingly important topic. Recent studies (e.g. [1]) have indicated that rail vehicles have generally become heavier over the last thirty years. Whilst these increases in vehicle mass can often be attributed to enhanced passenger environments (e.g. the provision of air-conditioning, improved accessibility, crashworthiness, etc.), there are clearly undesirable side-effects of heavier trains. Everything else being equal, a heavier vehicle will consume more energy in operation than a lighter one, thereby making it more costly to run. Increased energy

consumption also implies a likelihood of higher CO_2 emissions at some point in the energy supply chain. Furthermore, heavier vehicles are likely to cause more damage to the track, thereby resulting in higher costs for infrastructure maintenance and renewal. In some countries, heavier vehicles also attract higher track access charges for operators.

A recent investigation [2] by a cross-industry consortium of rail vehicle manufacturers examined some of the issues surrounding the increased use of lightweight materials in metro vehicles. As part of this work, a number of applications were identified that were considered to have a high potential for lightweighting through material substitution. One such application was interior floor panels.

A typical six-car metro vehicle will have around 250 m^2 of flooring material as part of its interior (Figure 9.1). This is likely to weigh a total of around 4 tonnes, thereby representing a significant lightweighting opportunity. In terms of functionality, the most fundamental requirement of a floor construction is that it is capable of supporting the loads induced by passengers without excessive deflection or failure. Additionally, floor constructions must also provide the required level of insulation. It can be seen from Figure 9.2 that current interior floor constructions are often quite complex multi-material assemblies employing woods, inserts, elastomers and insulative materials. Is there a material configuration that would provide a lighter solution at a competitive cost?

Figure 9.1. Typical floor panels in a metro vehicle interior.

Given the combined requirements of high stiffness, low weight and good insulation, it seemed interesting to investigate the concept of a sandwich design.

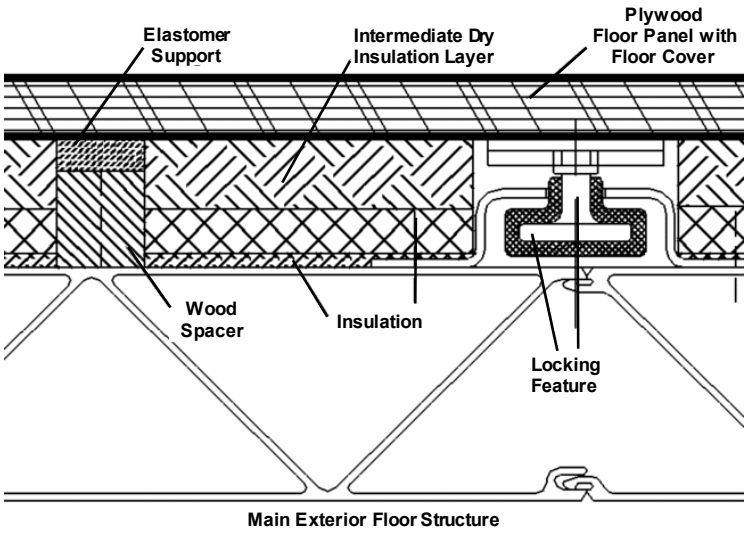

Figure 9.2. A cross-section of a typical current interior floor construction employing an assembly of different materials.

Using a metro vehicle floor panel as a case study, the ACO algorithm (*sandwichACO*) will be used to optimise a multiple objective sandwich material design.

9.2 Problem definition

An interior flooring arrangement similar to that depicted in Figure 9.2 will be considered. It consists of a series of sandwich floor panels supported by an underlying timber framework (Figure 9.3). The optimisation will consider both the construction of the sandwich floor panels and the spacing of the supporting timber joists. So there will be a trade-off between having a more substantial supporting framework and less structural panels, or having larger supporting spans and stiffer panels. The main (exterior) structural floor, which is part of the vehicle bodyshell structure, is not considered in the analysis. In sections 9.2.1 to 9.2.4 that follow, the optimisation problem is defined in terms of the objectives, the variables, the constraints and the governing physical equations.

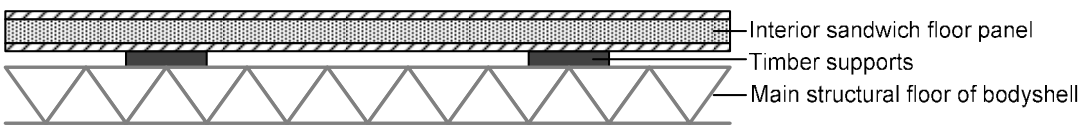

Figure 9.3. A cross-section of the assumed configuration of the sandwich flooring.

9.2.1 Objectives of the optimisation

For the floor system considered here, the objective was to find the Pareto-optimal set of sandwich constructions that are optimal for both low mass and low cost. The two objective functions to be minimised therefore are given in Equation (9.1) and (9.2).

Mass objective function:

$$M_{area} = \left\{\rho_{f1}t_{f1} + \rho_{f2}t_{f2} + \rho_c t_c\right\} + \left\{\frac{m_s}{Lb}\left[L + (b - b_s)\right]\right\} \tag{9.1}$$

where M_{area} is the total mass per unit area of the sandwich panel and its supports, ρ is density, t is thickness, L is the length of the sandwich panel, b is the width of the sandwich panel, m_s is the mass per unit length of the supporting timbers, b_s is the width of the supporting timbers, and subscripts *f1*, *f2*, *c* and *s* pertain to the upper sandwich facing, lower sandwich facing, sandwich core and timber supports respectively. So the first curly bracketed term in Equation (9.1) represents the mass contribution of the sandwich panel, and the second curly bracketed term represents the mass contribution of the supporting timber framework.

Cost objective function:

$$C_{area} = \left\{\rho_{f1}t_{f1}c_{f1} + \rho_{f2}t_{f2}c_{f2} + \rho_c t_c c_c\right\} + \left\{\frac{m_s c_s}{Lb}\left[L + (b - b_s)\right]\right\} \tag{9.2}$$

Where C_{area} is the total cost per unit area of the sandwich panel and its supports, and c is the cost per unit mass of an individual component in the system.

9.2.2 Design variables

Table 9.1 summarises the main design variables. These are the parameters that the ant colony optimisation algorithm sought to obtain optimal values for.

Table 9.1. Sandwich flooring design variables.

Variable	Range	Notes
Facing thickness	0.5 - 5 mm.	Upper and lower facings can have different thicknesses. Different discrete thicknesses within the stated range were permitted for different materials to reflect availability.
Facing material	Selected from a material database.	Upper and lower facings can be of different materials (Table 9.2). For fibre-reinforced polymer facings, there are further variables relating to the laminate construction (fibre material, matrix material, fibre volume fraction, and the orientation angle of each ply).
Core material	Selected from a material database.	A range of different densities were available for each core material option (Table 9.3)
Span (spacing between timber supports)	0.1 – 2.4 m (longitudinal) 0.1 – 1.4 m (transverse).	The support span can be different in the longitudinal and transverse directions.

For the facing materials, the optimisation algorithm was provided with a range of options to choose from including various aluminiums, steels, fibre-reinforced polymers and wood products. Furthermore, for the fibre-reinforced polymer facings, the algorithm could select between a range of fibre and matrix materials, as well as specifying the fibre volume fraction, the number of plies in the laminate, and the orientation angle of each ply (0^o, $+45^o$, -45^o or 90^o). Similarly, a number of core material options were available, including a variety of polymer foams, honeycombs and balsa woods of different densities. In total, there were 40 different facing and core materials for the algorithm to choose from (Table 9.2 and 9.3). This material database, when coupled with the fibre-reinforced polymer laminate design options, provided a very large number of potential sandwich material combinations. The upper limits on the sandwich floor span (i.e. the spacing between the underlying timber supports) were defined by a typical maximum panel size that can be manufactured in an industrial press.

Table 9.2. A list of all facing materials used in the rail vehicle floor panel optimisation

Facing materials	Cost (€/kg)	Density (kg/m^3)	Young's Modulus (GPa)		Shear Modulus (GPa)	Tensile Strength (MPa)		Compressive Strength (MPa)		Shear Strength (MPa)	Poisson's Ratio	Thermal Conductivity (W/mK)	Ref
Metal													
Aluminium													
5000 Series	1.4	2700	70		26	210		190		105	0.33	150	[3, 4]
6000 Series (High strength)	1.4	2700	70		27	280		250		140	0.34	170	[3]
7000 Series (Very high strength)	1.4	2800	73		28	460		400		230	0.34	170	[3]
Steel													
Medium strength	0.43	7900	210		82	420		310		210	0.29	52	[3]
High strength	2.3	7800	200		78	520		310		260	0.28	27	[3]
Very high strength	0.48	7900	210		82	680		360		340	0.29	46	[3]
Ultra high strength	0.63	7900	210		82	1000		690		500	0.29	40	[3, 5]
Fibres													
High Strength Carbon Fibre	74	1800	240		110	4700		5000		2350	0.11	140	[3]
High Modulus Carbon Fibre	24	1800	380		170	2400		3700		1200	0.11	140	[3]
E Glass	1.6	2600	79		33	2000		4500		1000	0.22	1.30	[3]
Matrix													
Epoxy	1.9	1300	4.1		1.4	63		150		31	0.40	0.10	[6, 7, 8]
Phenolic	1.1	1300	4.4		1.6	55		93		28	0.39	0.10	[6, 7, 8]
Polyester	2.4	1300	2.9		1.0	65		170		33	0.39	0.20	[6, 7, 8]
Wood			(long)	(trans)		(long)	(trans)	(long)	(trans)				
Plywood	1.2	500	11.0	4.8	0.25	53	45	28	26	27	0.25	0.14	[3, 9]
Hardboard	0.34	930	5.7	3.7	0.25	6.3	30	45	30	3.1	0.25	0.13	[3]

Table 9.3. A list of all core materials used in the rail vehicle floor panel optimisation.

Core materials	Cost (€/kg)	Density (kg/m^3)	Young's Modulus (GPa)		Shear Modulus (GPa)		Compressive Strength (MPa)	Shear Strength (MPa)		Thermal Conductivity (W/mK)	Ref
Polymer foams											
Polyetherimide 60	25	60	0.045		0.018		0.70	0.80		0.036	[10]
Polyetherimide 80	25	80	0.054		0.023		1.1	1.1		0.037	[10]
Polyetherimide 110	25	110	0.064		0.030		1.4	1.4		0.040	[10]
Polymethacrylimide 32	50	32	0.036		0.013		0.40	0.40		0.030	[11]
Polymethacrylimide 52	46	52	0.070		0.019		0.90	0.80		0.030	[11]
Polymethacrylimide 75	41	75	0.092		0.029		1.5	1.3		0.032	[11]
Polymethacrylimide 110	34	110	0.16		0.050		3.0	2.4		0.032	[11]
Extruded Polystyrene 40	4.8	40	0.021		0.010		0.40	0.40		0.025	[12]
Extruded Polystyrene 45	8.0	45	0.029		0.014		0.70	0.50		0.025	[12]
Polyurethane 48	4.8	48	0.01		0.0030		0.31	0.27		0.022	[13]
Polyurethane 80	4.8	80	0.026		0.0070		0.73	0.58		0.027	[13]
Polyurethane 160	4.8	160	0.088		0.019		2.3	1.7		0.037	[13]
Polyvinylchloride 48	10	48	0.060		0.015		0.60	0.56		0.026	[14]
Polyvinylchloride 60	10	60	0.075		0.020		0.90	0.76		0.027	[14]
Polyvinylchloride 80	10	80	0.10		0.027		1.4	1.2		0.029	[14]
Polyvinylchloride 100	10	100	0.13		0.035		2.0	1.6		0.031	[14]
Polyvinylchloride 130	10	130	0.18		0.050		3.0	2.2		0.032	[14]
Aluminium (3000 Series)			(long)	(trans)	(long)	(trans)		(long)	(trans)		
Honeycomb 21	27	21	1.1E-04	1.1E-04	0.10	0.050	0.59	0.45	0.31	3.4	[15]
Honeycomb 29	27	29	1.7E-04	1.7E-04	0.14	0.080	0.90	0.69	0.45	3.8	[15]
Honeycomb 37	27	37	2.8E-04	2.8E-04	0.19	0.10	1.4	0.97	0.55	4.0	[15]
Honeycomb 53	27	53	6.3E-04	6.3E-04	0.31	0.15	2.6	1.6	0.90	4.8	[15]
Honeycomb 77	27	77	1.0E-03	1.0E-03	0.48	0.26	4.6	2.5	1.5	6.6	[15]
Balsa wood											
Balsa 90	6.9	90	2.1	0.060	0.096	0.0080	5.4	1.6	0.23	0.052	[16]
Balsa 155	6.9	160	3.4	0.11	0.17	0.01	13	3.0	0.525	0.064	[16]
Balsa 220	6.9	220	5.2	0.19	0.24	0.02	22	4.5	0.60	0.086	[16]

9.2.3 Design constraints

Clearly, for the optimisation algorithm to be useful, it must be capable of discriminating between those sandwich constructions that are fit-for-purpose and those that are not. This fitness-for-purpose was defined by a number of design constraints or requirements that any prospective sandwich must satisfy. The constraints employed for the sandwich floor application were as follows:

- The sandwich must be sufficiently stiff, i.e. it must not deflect excessively under passenger loading. The limiting deflection was set at a maximum of 1 mm under a distributed load, q, of 6000 N/m^2.

- The sandwich must provide sufficient thermal insulation. The maximum allowable thermal conductance, Λ_{total}, of the sandwich was set at 0.0025 W/K, which is equivalent to the performance that might be expected from a conventional non-sandwich floor construction consisting of a 20 mm plywood panel with 30 mm of glass wool insulation.

- The upper facing must be sufficiently resilient to high localised loadings (e.g. heeled shoes). This aspect was arbitrarily handled by stipulating that the product of the upper facing Young's modulus and the upper facing thickness should be greater than 100 MN/m.

- The maximum allowable sandwich thickness, h, was set at 20 mm. Again, for equivalence with a typical existing plywood panel.

- The maximum allowable panel dimension was set at 2.5 m x 1.5 m – the dimensions of a typical industrial panel press.

- The sandwich must not fail under passenger loading. The failure modes considered for the sandwich included tensile and compressive failure of the facings due to bending, shear and compressive failure of the core, and wrinkling of the facings.

The supporting timber joists were also assumed to be constant in terms of their material and geometry and were therefore constrained. They had a mass per unit length (m_s) of 0.9 kg/m and a panel-supporting width (b_s) of 100 mm.

9.2.4 Governing equations

As with the previous chapter, the properties and performance of the sandwich materials and their constituents were estimated using analytical "textbook" solutions. The fibre-reinforced polymer facing stiffness properties were estimated using classical laminate theory. This is well described in many standard texts (e.g. Gibson [17], Matthews & Rawlins [6]). To simplify the laminate equations, only balanced, symmetric laminates were considered, although orthotropic constructions were permitted.

The mechanics of the sandwich panels were estimated using sandwich plate theory, as described, for example, by Allen [18] and Zenkert [19]. Each facing was considered separately, so that the upper sandwich facing could be of a different material and thickness to the lower sandwich facing. The analytical expression employed for maximum panel deflection, δ, assumed that a given section of sandwich was simply-supported around its periphery (as a worst case boundary condition from a deflection perspective). The governing equation was [18]:

$$\delta = \frac{qb^4}{D_x}\beta_1 \tag{9.3}$$

where β_1 is a sandwich coefficient [18], and D_x is the sandwich flexural rigidity in the x direction (parallel to the length, L, of the panel) given by:

$$D_x = d^2\left(\frac{1}{E_{xf1}t_{f1}} + \frac{1}{E_{xf2}t_{f2}}\right)^{-1} \tag{9.4}$$

where d is the distance between centrelines of opposing facings and E_{xf} is the Young's modulus in the x direction. The subscripts 1 and 2 refer to the upper and lower facings

respectively. The stiffness expression in Equation (9.4) is applicable for sandwich panels with orthotropic faces of unequal thickness and different materials. A similar expression was also used for the *y* direction (parallel to the width, *b*, of the sandwich).

With respect to failure prediction, the compressive and tensile stresses due to bending in the faces, and the shear and compressive stresses in the core were compared against the respective material strengths.

The equation for the compressive stress generated due to bending in the *x* direction of the upper facing, σ_{x1}, was calculated as:

$$\sigma_{x1} = \frac{\left(qb^2\beta_2\right)E_{xf1}E_{xf2}t_{f2}d}{D_x\left(E_{xf1}t_{f1} + E_{xf2}t_{f2}\right)} \tag{9.5}$$

where β_2 is a sandwich coefficient [18]. Similar checks in the *y* direction were also performed, as well as on the tensile stresses due to bending in the lower facing. Similarly to the benchmark in Chapter 8, the von Mises failure criterion was used to predict failure for the non-reinforced faces. Also, for the fibre-reinforced polymer facings, first ply failure was estimated using the Tsai-Hill criterion.

The maximum resulting shear stress, Q_c, of the core was calculated as:

$$Q_c = \frac{qb\beta_5}{d} \tag{9.6}$$

where β_5 is a sandwich coefficient [18]. A similar check in the *yz* direction of the core was also performed.

Local facing wrinkling was also considered using the expression provided by Zenkert [6]:

$$\sigma_{wrinkling} = \frac{\sqrt[3]{E_{xf1}E_cG_c}}{2} \tag{9.7}$$

where G_c is the shear modulus. The critical wrinkling stress, $\sigma_{wrinkling}$, was compared against the facing compressive stress to determine the onset of this mode of failure. Equation (9.7) was specifically used to check for wrinkling of the upper facing in the x direction. Similar checks were applied for the y direction (parallel to the width, b, of the sandwich).

The overall through-thickness thermal conductance of the sandwich, Λ_{total}, was estimated using the expression provided by Ashby [20].

$$\Lambda_{total} = h\left(\frac{t_{f1}/h}{\lambda_{f1}} + \frac{t_{f2}/h}{\lambda_{f2}} + \frac{t_c/h}{\lambda_c}\right)^{-1} \tag{9.8}$$

As with the benchmarking process, whilst more complex and accurate methods of predicting the behaviour of sandwich panels are available, their use would not have fundamentally altered the manner in which the optimisation was performed. It would just have required the substitution of one sandwich design algorithm for another within the optimisation process. For the purposes of this study, the textbook analytical solutions were considered sufficient for the purposes of demonstrating the application of ant colony optimisation for sandwich design.

9.3 Results and discussion

The detailed procedure for the *sandwichACO* algorithm has been explained previously in Chapter 6. For this problem, the ACO algorithm conducted a search of all the variables stated in Table 9.1. Optimal values of these variables were sought which maximised the mass and cost of the sandwich (Equations (9.1) and (9.2)). In addition, the ACO algorithm will need negotiate the constraints (Equations (9.3) - (9.8)) of the problem to ensure that the designs being tested are feasible.

For this study, suitable algorithm parameters were selected for *sandwichACO* on an observational basis and from the results of the previous chapter. The key parameters employed were as follows:

- Number of ants = 10.
- Number of iterations = 200,000.
- Maximum size of Pareto-optimal set = 50.
- Evaporation rate = 0.1 (i.e. during each iteration, the pheromone level for each variable reduces naturally by 90%).
- $\alpha_1 = \alpha_2 = 1$ (parameters controlling the pheromone levels of currently popular paths and Pareto-optimal solutions respectively).

The sequential graphs in Figure 9.4 illustrate the dynamic evolution of the ACO over an increasing number of iterations of the algorithm. Each graph shows both the position of the individual ants during the given iteration, and the current non-dominated Pareto-optimal solutions. In Figure 9.4a, the initial (random) distribution of calculated objective functions is shown. During the early stages of the optimisation (the first 10 iterations, Figure 9.4b), rapid progression was observed, with the Pareto-optimal solutions showing marked improvements between successive iterations as they moved towards the low mass and low cost regions of the design space. After around 500 iterations (Figure 9.4c), incremental improvements to existing best solutions had become smaller, and a larger and more diverse set of Pareto-optimal solutions had been identified. As the number of iterations continued to increase, changes to the Pareto-optimal set became less and less significant, with few improvements beyond 100,000 iterations. The final distribution, after 200,000 iterations, is shown in Figure 9.4d

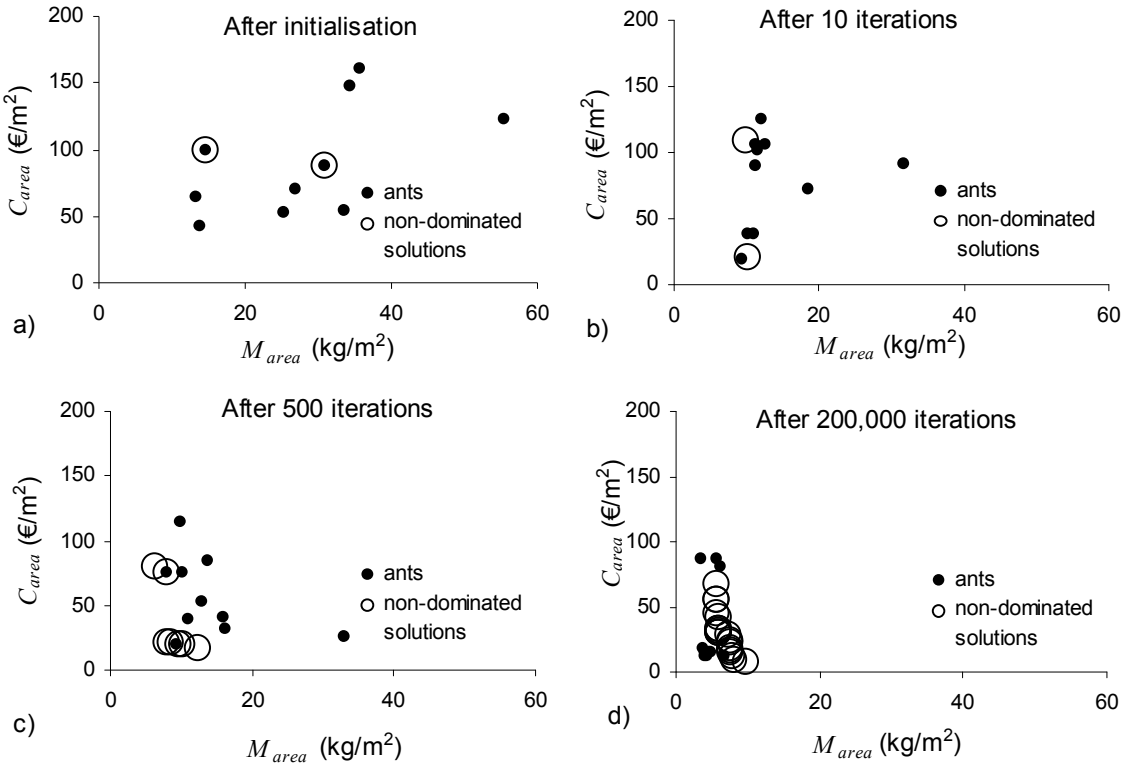

Figure 9.4. The progression of the ant colony optimisation after various iterations (a) 1 iteration, (b) 10 iterations, (c) 500 iterations and (d) 200,000 iterations.

After 200,000 iterations, the ACO had identified a total of 32 non-dominated Pareto-optimal solutions (those plotted in Figure 9.4d). For validation purposes, a random sample of these solutions were verified manually using the governing equations in order to confirm that the algorithm had performed reliably. A pleasingly broad range of optimal material solutions had been found including extruded polystyrene and polymethacrylimide cores of various densities, and a wide variety of facing materials: carbon fibre-reinforced phenolics, steel and stainless steel for the upper facing; carbon and glass fibre-reinforced phenolics, aluminium, plywood and hardboard for the lower facing. Furthermore, for the fibre-reinforced materials, a range of fibre volume fractions and lay-ups were identified. In terms of the support geometry, the maximum longitudinal span of 2.4 m was preferred in all cases, but for the transverse spans an optimal range of 0.4 – 0.52 m was suggested.

The full list of optimal design solutions is shown in Table 9.4.

Table 9.4. The full list of obtained Pareto-optimal solutions acquired for the rail vehicle floor panel optimisation. Solutions in grey represent a low mass, a low cost and an intermediate option. Superscripts relate to laminate lay-ups; (1) [0°, 90°]s, (2) [90°/ 0°]s, (3) [90°/ 90°] and (4) [90°/ 90°]s. PMI = Polymethacrylimide, XPS = Extruded Polystyrene

Obtained Pareto-optimal solutions

Core	Uper facing (*f1*)	t_{f1} (mm)	Lower facing (*f2*)	t_{f2} (mm)	l (m)	b (m)	M_{area} (kg/m^2)	C_{area} (€/m^2)
PMI 32	HM Carbon/phenolic[1, 2] υ = 0.5	1	HM Carbon/phenolic[4] υ = 0.3	1	2.4	0.50	5.6 56%	67 (344)%
PMI 32	HM Carbon/phenolic[1, 2] υ = 0.5	1	Aluminium 5000 & 6000	0.5	2.4	0.46	5.7 55%	56 (272)%
XPS 40	HM Carbon/phenolic[1, 2] υ = 0.5	1	HM Carbon/phenolic[4] υ = 0.3	1	2.4	0.46	5.9 54%	42 (178)%
XPS 40	HM Carbon/phenolic[1, 2] υ = 0.5	1	Aluminium 5000 & 6000	0.5	2.4	0.42	6.0 53%	30 (99)%
XPS 40	HM Carbon/phenolic[1, 2] υ = 0.5	1	Plywood	3	2.4	0.42	6.0 53%	29 (93)%
XPS 40	HM Carbon/phenolic[1, 2] υ = 0.5	1	Hardboard	3	2.4	0.40	7.4 42%	28 (90)%
XPS 40	Steel (medium strength)	0.5	HM Carbon/phenolic[3] υ = 0.3 & 0.4	0.5	2.4	0.4 - 0.5	7.6 - 7.7 39 - 40%	14 - 16 7 - (7)%
XPS 40	Steel (medium strength)	0.5	Glass/phenolic[3] υ = 0.35	0.5	2.4	0.40	8.0 37%	9 40%
XPS 40	Steel (medium strength)	0.5	Hardboard	3	2.4	0.40	9.8 22%	8 (47)%
XPS 45	HM Carbon/phenolic[1, 2] υ = 0.5	1	HM Carbon/phenolic[4] υ = 0.3	1	2.4	0.52	5.8 55%	44 (196)%
XPS 45	HM Carbon/phenolic[1, 2] υ = 0.5	1	Aluminium 5000 & 6000	0.5	2.4	0.46	5.9 53%	33 (119)%
XPS 45	HM Carbon/phenolic[1, 2] υ = 0.5	1	Plywood	3	2.4	0.46	5.9 53%	32 (112)%
XPS 45	Steel (medium strength)	0.5	Glass/phenolic[3] υ = 0.3	0.5	2.4	0.42	8.0 37%	12 (22)%
XPS 45	Steel (medium strength)	0.5	HM Carbon/phenolic[3] υ = 0.3 & 0.55	0.5	2.4	0.50	7.6 40%	17 - 22 (13) - (47)%
XPS 45	Steel (high strength stainless)	0.5	HM Carbon/phenolic[3] υ = 0.3	0.5	2.4	0.50	7.5 41%	25 (64)%

Table 9.5 summarises these optimised design variables for three representative Pareto-optimal solutions – a low mass option, a low cost option and an intermediate option. The savings in mass and cost are in comparison to a typical existing 2.5 m x 0.5 m x 20 mm plywood / 30 mm glass wool construction with a mass of 12.7 kg/m^2 and a cost of 15 €/m^2 (including timber supports). It can be seen that, from a cost perspective, only the "low cost" option is cheaper than an equivalent plywood panel. Furthermore, this design also provides a 37% mass saving. The lighter "intermediate" and "low mass" optimal solutions were both more expensive than plywood, although their weight savings were also higher at 40% and 53% respectively. However, it should be noted that lightweight designs are likely to provide additional cost savings over and above those associated with materials. For example, an integrated, self-insulating sandwich might have lower installation costs than a separate plywood / glass wool insulation system. There will also be through-life operational cost savings associated with the use of lighter materials. For a single six-car metro vehicle, the estimated annual operational cost saving associated with a 53% reduction in flooring mass would be around 10,000 € [2]. Clearly, for a fleet of vehicles over a 40 year life, such operational cost savings would be very significant.

Table 9.5. Representative Pareto-optimal solutions for the metro vehicle floor panels.

	Low mass design	Low cost design	Intermediate design
Upper facing material	Carbon fibre-reinforced phenolic	Steel	Steel
Upper facing lay-up	[0°/90°]s	-	-
Upper facing fibre volume fraction	0.5	-	-
Upper facing thickness, t_{f1}	1 mm	0.5 mm	0.5 mm
Core material	Extruded polystyrene	Extruded polystyrene	Extruded polystyrene
Core density	40 kg/m^3	40 kg/m^3	45 kg/m^3
Lower facing material	Plywood	Glass fibre-reinforced phenolic	Carbon fibre-reinforced phenolic
Lower facing lay-up	-	[90°/90°]	[90°/90°]
Lower facing fibre volume fraction	-	0.35	0.3
Lower facing thickness, t_{f2}	3 mm	0.5 mm	0.5 mm
Longitudinal span	2.40 m	2.40 m	2.40 m
Transverse span	0.42m	0.40 m	0.50 m
M_{area} (kg/m^2)	6.0 (53 % reduction)	8.0 (37% reduction)	7.6 (40% reduction)
C_{area} (€/m^2)	29 (93% increase)	9 (40% reduction)	17 (13% increase)

Finally, an important point to note is that whilst the ACO algorithm attempts to identify the non-dominated (i.e. "best") set of optimal solutions, it cannot be absolutely known that the set it generates does indeed match the *true* Pareto-optimal set to the problem. However, by using a large number of iterations (200,000), and by running the simulation multiple times from different random starting positions, an acceptable level of confidence in the results can be obtained.

9.4 Conclusions

The *sandwichACO* algorithm has been applied to the design of a sandwich panel for a rail vehicle interior flooring application in which multiple objectives of low mass and low cost were considered. The problem definition and the associated implementation of the algorithm allowed considerable freedom in the choice of both materials and geometry subject to certain constraints associated with fitness-for-purpose.

A broad range of optimal solutions were identified by the *sandwichACO* technique. These included sandwich constructions that provided a significant (approximately 40%) saving in both mass and cost compared to the plywood panels that are currently used, as well as designs that provided more significant mass savings (of over 40%), albeit at a cost premium.

Overall, *sandwichACO* has shown to be successful at optimising a rail vehicle floor sandwich panel. Similarly to the case study in Chapter 8, the technique was able to rapidly identify a non-dominated set of solutions with good repeatability. Also, as a result of its good performance throughout, it has proven to be robust. Additionally, given the relative ease at which the *sandwichACO* handled the extra complexity of the problem regarding the additional sandwich mechanics (compared with Chapter 8), the *sandwichACO* algorithm presents itself as an extremely competitive technique and offers good scope for further utilization on many other engineering aspects.

9.5 Publications

Hudson, C.W., Carruthers, J.J., Robinson, A.M. (2009) Multiple objective optimisation of composite sandwich structures for rail vehicle floor panels. *Composite Structures*. **In press, corrected proof**

9.6 References

1. Ford, R. (2007) Transport mass. *Institution of mechanical engineers seminar: Weight saving and structural integrity of rail vehicles*. Derby, UK.
2. Carruthers, J.J., Calomfirescu, M., Ghys, P., Prockat, J. (2009) The application of a systematic approach to material selection for the lightweighting of metro vehicles. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* **223**, 427-437.
3. (2005) CES Selector version 4.6. Granta Design Limited.
4. AluSelect (2001) Physical and elastic properties of aluminium. *http://aluminium.matter.org.uk/aluselect/03_physical_browse.asp*.
5. Rukki (2009) Hot rolled steel plates, sheets and coils. *http://www.ruukki.com/www/materials.nsf/0/8A66087E679A2F06C2257689002D9CCB/$File/Optim%20QC%20HR_12%202009_EN.pdf?openElement*.
6. Matthews, F.L., Rawlings, R.D. (1994) *Composite materials: Engineering and science*. Chapman & Hall, London.
7. Hull D., Clyne T. W. (1996) *An introduction to composite materials* Cambridge University Press, Cambridge.
8. Plastics Technology Online (2009) Pricing. *http://www.ptonline.com/*.
9. Canadian Plywood Association (2010) Plywood design fundamentals. *http://www.canply.org/pdf/main/plywood_designfund.pdf*.
10. Alcan (2005) Airex 82 high performance structural foam. *http://files.alcancomposites.com/downloads/2_1_en_/r82_data_sheet.pdf*.

11. Evonic Industries (2010) Rohacell IG product information. *http://www.rohacell.com/sites/dc/Downloadcenter/Evonik/Product/ROHACELL/product-information/ROHACELL%20IG_IG-F%20Product%20Information.pdf*.
12. Dow (2006) STYROFOAM technical data. *http://www.composite-panel.co.uk/images/Styrofoam%20in%20Composite%20Panels.pdf*.
13. General Plastics Manufacturing Company (2010) LAST-A-FOAM data sheet. *http://www.generalplastics.com/products/product_detail.php?pid=15&*.
14. DIAB (2009) Divinycell H technical data. *http://www.diabgroup.com/europe/literature/e_pdf_files/ds_pdf/H_DS_EU.pdf*.
15. Hexcel (2006) HexWeb ACG honeycomb product data. *http://www.hexcel.com/NR/rdonlyres/6A65CF17-B8FA-474D-A92B-8E5F8C1BFDFC/0/HexWeb_ACG_us.pdf.*
16. DIAB (2009) ProBalsa technical data. *http://www.diabgroup.com/europe/literature/e_pdf_files/ds_pdf/PB_DS.pdf*.
17. Gibson, R.F. (1994) *Principles of composite material mechanics*. McGraw-Hill, New York.
18. Allen, H.G. (1969) *Analysis and design of structural sandwich panels*. Pergamon Press, London.
19. Zenkert, D. (1995) *An introduction to sandwich construction*. EMAS Publishing.
20. Ashby, M.F. (2005) *Materials selection in mechanical design*. Elsevier Butterworth-Heinemann, Italy.

10 Conclusions and recommendations for further work

10.1 Conclusions

Population-based optimisation techniques have been highlighted in this thesis as offering a novel solution to the challenge of multiple objective optimisation of sandwich materials and structures. A detailed assessment of the literature showed that three methods in particular showed considerable potential. These were particle swarm optimisation (PSO), ant colony optimisation (ACO) and simulated annealing (SA). Further investigation led to the development of three novel optimisation techniques. These were termed by the author as *sandwichPSO*, *sandwichACO* and *sandwich SA*.

A benchmark problem considered the application of these algorithms to a multiple objective sandwich beam optimisation. The free variables investigated included the facing thickness, and the facing and core materials. For the facings, multi-ply, oriented laminate constructions were considered. Furthermore, several geometric, thermal, deflection and strength constraints were placed upon the design space. Based on these inputs, the sandwich beam was optimised for stiffness, mass and cost. Results showed that, with little tuning, the ACO was the most competitive. It demonstrated superior ability to obtain all optimal solutions in most cases. Both PSO and SA struggled to identify local optimum solutions in regions of the objective space in which the ratio of feasible-to-infeasible solutions was low. This is characterised by multi-ply, oriented fibre-reinforced polymer sandwich facing laminates. However, encouragingly, PSO and SA were both found to be robust tools that were largely insensitive to variations in their influencing parameters.

From the results of the benchmark, the ACO technique was carried forward and applied to a further case study. This involved the optimisation of a sandwich plate for a rail vehicle floor panel. In addition to the benchmark, the problem was extended to allow the material and thickness of the top face to be different to the bottom. Orthotropic fibre-reinforced facing constructions were also included, as well as a localised load constraint. A broad range of optimal solutions were identified for the applied minimum mass and cost objectives. Sandwich constructions provided a significant (approximately 40%) saving in both mass and cost compared to the existing plywood design. More significant mass saving designs were also identified (of over 40%), but with a cost premium.

Overall, a significant amount of development work has been conducted to ensure each optimisation technique was suitably refined for the intended purpose. This is reflected by the complexity of the problems that have been investigated. To recall, the extent to which sandwich design has been investigated covers a significant proportion of the known complexities in the field of optimisation. In relation to the comparison case study, not only have multiple variables, objectives and constraints been included, but the results of which have found the design space to contain such complexities as multimodality, deceptive optima, isolated points, discontinuities, non-uniformly distributed Pareto-optimal sets and both convex and non-convex Pareto-optimal fronts. With this in mind, considerable appreciation is given to the ability of the *sandwichACO* algorithm. Not only in terms of dealing with those complexities recognised within the optimisation community, but the final case study especially proves the practical application of the technique to real engineering related problems.

10.2 Recommendations for further work

While every effort was made to ensure the best result from each algorithm was obtained, several aspects could be investigated to try and further improve performance. For instance, the application of hybrid algorithms for sandwich design could form an interesting topic. Here, a population-based optimiser could provide the global search of the entire solution space, then another, more primitive, method could be used to perform a search of local

areas. This could be achieved say by combining PSO or ACO with a single point gradient-based or direct search technique to improve convergence time. Alternatively, a Tabu search algorithm in conjunction with SA could be explored for this purpose.

One of the primary objectives of the project was to identify the most suitable optimisation technique for sandwich design. Although this has been achieved, a number of questions remain open regarding the specific reason why PSO did not perform as well as the ACO technique. Even though the observed affects have been analysed to an extent, this is still an area open for further investigation.

Another aspect that could be investigated further would be to examine what effect different constraint handling methods have on performance. A parameterless constraint handling approach has been adopted in this thesis as it requires no problem specific tuning. However, other methods such as those using penalty functions could be used as an alternative. While the disadvantages with these methods have already been addressed, they may provide improved performance in heavily constrained areas of the search space.

For this thesis, only problems with analytical equations have been solved. While analytical solutions offer the advantage of providing accurate and rapid solutions, to an extent, this limits the complexity of the shape that can be used. For instance, if the bodyshell of a rail vehicle cab was constructed using a sandwich structure, the extent to which analytical equations could be used to represent the given profile would be quite restricted. In these instances, a favourable option would be to turn to finite element analysis (FEA). FEA offers the advantage that almost any geometry can be modelled. However, if FEA were to be utilised here, one challenge in particular would need to be overcome. This relates to the many thousands of solutions that population-based techniques need to analyse in a given simulation to obtain the best combination of variables. Analytical equations lend themselves well as the solutions can be solved in real time. Hence, an optimisation process can be conducted within a suitable timeframe. On the other hand, a complex FEA can require several hours to obtain just one solution. In this case, the time to solve a model alone would make the use of FEA initially unsuitable. However, for such complex cases, the author recognises two methods to by-pass this problem. One approach would be to use a relatively simplistic FEA model in which the solve time is reasonably rapid. While

accuracy would be compensated for a gain in speed, it would offer a suitable early stage comparative means of testing a wide material and geometric range.

Alternatively, a computational intensive FEA model could be utilised if the optimisation algorithm was used in conjunction with a neural network say. The procedure for this would be to initially obtain a reasonably few number of complete FEA solutions using a wide range of given variables (core materials, facing thicknesses etc). Through monitoring the results of this known limited set of solutions, a neural network could then be trained to "predict" the response of the FEA for any given input. In this case, the optimisation algorithm would instead use the predicted data provided by the neural network. So by not using the FEA directly, the optimisation process becomes significantly faster. While this has proven to be successful in some applications, the practicalities for the type of sandwich design implemented here would need further investigation.

One of the main focuses of the work in this thesis has been to benchmark the developed algorithms against problems relating to sandwich optimisation. However, in the wider optimisation community, more mathematical, yet rigorous, benchmark procedures also exist. In Chapter 2, several features of an optimisation problem were recognised as causing difficulty when finding optimal solutions. So, it is no surprise that several test functions have been developed to test an algorithms ability to handle each of these features systematically. While it is recognised that optimisation methods need to be tailored to suit the particular application, further benchmarking procedures carried out on the optimisation techniques developed in this thesis would give a wider appreciation of their performance.

The sandwich floor panel case study represents just one instance where population-based multiple objective optimisation has proven to be successful. In this case, and other situations where sandwiches could replace existing (non-sandwich) components, the advantages are two fold. Not only could the advantages of lightweight sandwich alternatives be obtained, but the benefits of using population-based techniques to generate optimal designs would give a more rapid and complete assessment of suitability. In addition, for applications where sandwich structures are already used, there is the possibility that these techniques may optimise a design further.

On a final note, the previous successful application of population-based optimisation in a broad range of industries is accountable due to their excellent transferability and robustness. While a large part of this thesis has developed these techniques specifically for sandwich design, as far as possible, parameterless methods have been adopted. Generally, this is because the inclusion of parameters only narrows the applicability of the technique to particular instances. However, while parameterless methods can work equally well, they are inherently more transferability as less information needs to be provided. Due to this, it is likely that their success will extend far beyond the sandwich composite industry and their application in many other fields offers great potential.