

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

LÂM PHÁT THUẬN

**PHÁT TRIỂN CÁC PHƯƠNG PHÁP TỐI ƯU HÓA
THÔNG MINH CHO CÁC BÀI TOÁN CƠ HỌC**

**CHUYÊN NGÀNH: CƠ KỸ THUẬT
MÃ NGÀNH: 62140101**

TÓM TẮT LUẬN ÁN

TP. HỒ CHÍ MINH, 01/2021

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI
ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

GVHD 1: PGS. TS. NGUYỄN HOÀI SƠN

GVHD 2: PGS. TS. LÊ ANH THẮNG

Luận án được bảo vệ trước

HỘI ĐỒNG ĐÁNH GIÁ LUẬN ÁN

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM,

Ngày tháng năm

NHỮNG ĐÓNG GÓP CHÍNH CỦA LUẬN ÁN

- Thuật toán DE cải tiến đã được áp dụng lần đầu tiên để giải bài toán thiết kế tối ưu của cấu trúc tấm composite gia cường và kết quả cho thấy hiệu quả và độ chính xác tốt.
- Một sự hiệu chỉnh trong bước lựa chọn của thuật toán Jaya ban đầu sử dụng kỹ thuật lựa chọn tinh hoa (Elitist Selection Technique) được đề xuất để tạo thành một phiên bản cải tiến của thuật toán. Thuật toán Jaya cải tiến sau đó được áp dụng để giải quyết bài toán tối ưu hóa cấu trúc dầm composite Timoshenko và thu được kết quả rất tốt.
- Thuật toán Jaya cải tiến lần đầu tiên được kết hợp với Phương pháp xác định vòng lặp đơn nghiệm toàn cục (SLDM) để tạo ra một bộ công cụ mới có tên (SLMD-iJaya) để giải quyết bài toán Tối ưu hóa thiết kế dựa trên độ tin cậy của các mô hình dầm composite liên tục. Các thiết kế tối ưu thu được tốt hơn và an toàn hơn nhiều so với các thiết kế không có xét đến yếu tố độ tin cậy.
- Mạng nơ-ron nhân tạo được sử dụng để xấp xỉ đáp ứng của tấm composite gia cường và ANN được tích hợp với thuật toán Differential Evolution cải tiến để tạo thành thuật toán mới gọi là thuật toán ABDE. Thuật toán mới này sau đó được áp dụng để tìm kiếm thiết kế tối ưu của các cấu trúc tấm composite gia cường. Bài toán đầu tiên là tối ưu hóa các góc hướng sợi của tấm composite gia cường và vấn đề thứ hai là giải tìm độ dày tối ưu của tấm composite gia cường. Kết quả thu được cho thấy tính hiệu quả cao của bộ công cụ ABDE được đề xuất.

ABSTRACT

Almost all design problems in engineering can be considered as optimization problems and thus require optimization techniques to solve. During the past decades, many optimization techniques have been proposed and applied to solve a wide range of various problems. Among them, metaheuristic algorithms have gained huge

popularity in recent years in solving design optimization problems of many types of structure with different materials. These metaheuristic algorithms include genetic algorithms, particle swarm optimization, bat algorithm, cuckoo search, differential evolution, firefly algorithm, harmony search, flower pollination algorithm, ant colony optimization, bee algorithms, Jaya algorithm and many others. Among the methods mentioned above, the Differential Evolution is one of the most widely used methods. Since it was first introduced by Storn and Price [1], many studies have been carried out to improve and apply DE in solving structural optimization problems. The DE has demonstrated excellent performance in solving many different engineering problems. Besides the Differential Evolution algorithm, the Jaya algorithm recently proposed by Rao [2] is also an effective and efficient method that has been widely applied to solve many optimization problems and showed its good performance. It gains dominant results when being tested with benchmark test functions in comparison with other population-based methods. However, like many other population-based optimizations, one of the disadvantages of DE and Jaya is that the optimal computational time is much slower than the gradient-based optimization methods. This is because DE and Jaya takes a lot of time in evaluating the fitness of individuals in the population. To overcome this disadvantage, artificial neural networks (ANN) are proposed to combine with the metaheuristic algorithms, such as Differential Evolution, to form a new approach that help solve the design optimization effectively. Moreover, one of the most important issues in engineering design is that the optimal designs are often affected by uncertainties which can be occurred from various sources, such as manufacturing processes, material properties and operating environments. These uncertainties may cause structures to improper performance as in the original design, and hence may result in risks to structures [3]. Therefore, reliability-based design optimization (RBDO) can be considered as a comprehensive strategy for finding an optimal design.

In this dissertation, an improved version of Differential Evolution has been first time utilized to solve for optimal fiber angle and thickness of the stiffened composite. Secondly, the Artificial Neural Network is integrated to the optimization process of the improved Differential Evolution algorithm to form a new algorithm call ABDE (ANN-based Differential Evolution) algorithm. This new algorithm is then applied to solve optimization problems of the stiffened composite plate structures. Thirdly, an elitist selection technique is utilized to modify the selection step of the original Jaya algorithm to improve the convergence of the algorithm and formed a new version of the original Jaya called iJaya algorithm. The improved Jaya algorithm is then applied to solve for optimization problem of the Timoshenko composite beam and obtained very good results. Finally, the so-called called (SLMD-iJaya) algorithm which is the combination of the improved Jaya algorithm and the Global Single-Loop Deterministic Methods (SLDM) has been proposed as a new tool set for solving the Reliability-Based Design Optimization problems. This new method is applied to look for optimal design of Timoshenko composite beam structures.

TÓM TẮT

Hầu như các bài toán thiết kế trong kỹ thuật có thể được coi là những bài toán tối ưu và do đó đòi hỏi các kỹ thuật tối ưu hóa để giải quyết. Trong những thập kỷ qua, nhiều kỹ thuật tối ưu hóa đã được đề xuất và áp dụng để giải quyết một loạt các vấn đề khác nhau. Trong số đó, các thuật toán metaheuristic đã trở nên phổ biến trong những năm gần đây trong việc giải quyết các vấn đề tối ưu hóa thiết kế của nhiều loại cấu trúc với các vật liệu khác nhau. Các thuật toán metaheuristic này bao gồm Genetic Algorithms, Particle Swarm Optimization, Bat Algorithm, Cuckoo Search, Differential Evolution, Firefly Algorithm, Harmony Search, Flower Pollination Algorithm, Ant Colony Optimization, Bee Algorithms, Jaya Algorithm và nhiều thuật toán khác. Trong số các phương pháp được đề cập ở trên, Differential Evolution là một trong những phương pháp được sử

dụng rộng rãi nhất. Kể từ khi được Storn và Price [1] giới thiệu lần đầu tiên, nhiều nghiên cứu đã được thực hiện để cải thiện và áp dụng DE trong việc giải quyết các vấn đề tối ưu hóa cấu trúc. DE đã chứng minh hiệu suất tuyệt vời trong việc giải quyết nhiều vấn đề kỹ thuật khác nhau. Bên cạnh thuật toán Differential Evolution, thuật toán Jaya được Rao [2] đề xuất gần đây cũng là một phương pháp hiệu quả và đã được áp dụng rộng rãi để giải quyết nhiều vấn đề tối ưu hóa và cho thấy hiệu suất tốt. Nó đạt được kết quả vượt trội khi được thử nghiệm với các hàm test benchmark so với các phương pháp dựa trên dân số khác. Tuy nhiên, giống như nhiều thuật toán tối ưu hóa dựa trên dân số khác, một trong những nhược điểm của DE và Jaya là thời gian tính toán tối ưu chậm hơn nhiều so với các phương pháp tối ưu hóa dựa trên độ dốc (gradient-based algorithms). Điều này là do DE và Jaya mất rất nhiều thời gian để đánh giá hàm mục tiêu của các cá thể trong bộ dân số. Để khắc phục nhược điểm này, các mạng nơ ron nhân tạo (Artificial Neural Networks) được đề xuất kết hợp với các thuật toán metaheuristic, như Differential Evolution, để tạo thành một phương pháp cách tiếp cận mới giúp giải quyết các bài toán tối ưu hóa thiết kế một cách hiệu quả. Bên cạnh đó, một trong những vấn đề quan trọng nhất trong thiết kế kỹ thuật là các thiết kế tối ưu thường bị ảnh hưởng bởi sự không chắc chắn có thể xảy ra từ nhiều nguồn khác nhau, chẳng hạn như quy trình sản xuất, tính chất vật liệu và môi trường vận hành. Những yếu tố không chắc chắn này có thể khiến các cấu trúc hoạt động không đúng như trong thiết kế ban đầu, và do đó có thể dẫn đến rủi ro cho các cấu trúc [3]. Do đó, tối ưu hóa thiết kế dựa trên độ tin cậy (Reliability-Based Design Optimization) có thể được coi là một chiến lược toàn diện để tìm kiếm một thiết kế tối ưu.

Trong luận án này, lần đầu tiên một phiên bản cải tiến của phương pháp Differential Evolution đã được sử dụng để tìm góc hướng sợi tối ưu và độ dày của tấm gia cường vật liệu composite. Thứ hai, Mạng nơ ron nhân tạo (ANN) được tích hợp vào quy trình tối ưu hóa thuật toán Differential Evolution cải tiến để hình thành thuật toán mới gọi là thuật toán ABDE (Artificial Neural Network-Based

Differential Evolution). Thuật toán mới này sau đó được áp dụng để giải quyết các bài toán tối ưu hóa của các cấu trúc tấm composite gia cường. Thứ ba, một kỹ thuật lựa chọn tinh hoa (Elitist Selection Technique) được sử dụng để hiệu chỉnh bước lựa chọn của thuật toán Jaya ban đầu để cải thiện sự hội tụ của thuật toán và hình thành một phiên bản mới của thuật toán Jaya được gọi là thuật toán iJaya. Thuật toán Jaya cải tiến (iJaya) sau đó được áp dụng để giải quyết bài toán tối ưu hóa dầm Timoshenko vật liệu composite và thu được kết quả rất tốt. Cuối cùng, thuật toán mới SLMD-iJaya được tạo thành từ sự kết hợp giữa thuật toán Jaya cải tiến và phương pháp vòng lặp đơn xác định (Single-Loop Deterministic Method) đã được đề xuất như một công cụ mới để giải quyết các vấn đề Tối ưu hóa thiết kế dựa trên độ tin cậy. Phương pháp mới này được áp dụng để tìm kiếm thiết kế tối ưu của các cấu trúc dầm composite Timoshenk và cho kết quả vượt trội.

CHƯƠNG 1

GIỚI THIỆU TỔNG QUAN

1.1 Tổng quan về Artificial Neural Network, thuật toán Metaheuristic và tối ưu hóa cấu trúc vật liệu composite

Hầu hết tất cả các vấn đề thiết kế trong kỹ thuật có thể được coi là các vấn đề tối ưu hóa và do đó cần các kỹ thuật tối ưu hóa để giải quyết. Tuy nhiên, vì hầu hết các bài toán trong thực tế đều có tính phi tuyến tính cao, các phương pháp tối ưu hóa truyền thống thường không đạt hiệu quả tốt. Xu hướng hiện nay để giải quyết các vấn đề tối ưu hóa phi tuyến như vậy là sử dụng các thuật toán tiến hóa và các phương pháp tối ưu hóa Metaheuristic. Các thuật toán Metaheuristic đã trở nên phổ biến trong những năm gần đây. Sự phổ biến của các thuật toán metaheuristic có thể là do các ưu điểm của chúng là sự đơn giản, linh hoạt, hiệu quả, dễ thích nghi và dễ thực hiện. Những ưu điểm như vậy làm cho các thuật toán này rất linh hoạt để giải quyết một loạt các bài toán tối ưu hóa, đặc biệt là các bài toán tối ưu hóa cấu trúc [4]. Tối ưu hóa cấu trúc là một lĩnh vực tiềm

năng và thu hút được sự quan tâm của nhiều nhà nghiên cứu trên thế giới. Trong suốt nhiều thập kỷ qua, nhiều kỹ thuật tối ưu hóa đã được đề xuất và áp dụng để giải quyết một loạt các vấn đề khác nhau. Trong số đó, Phương pháp Differential Evolution là một trong những phương pháp được sử dụng rộng rãi nhất. Kể từ khi nó được giới thiệu lần đầu tiên bởi Storn and Price [1], nhiều nghiên cứu đã được thực hiện để cải thiện và ứng dụng DE trong việc giải các bài toán tối ưu hóa kết cấu. DE đã thể hiện hiệu suất rất cao trong việc giải quyết nhiều bài toán kỹ thuật khác nhau.

Bên cạnh thuật toán Differential Evolution, thuật toán Jaya do Rao [2] đề xuất gần đây cũng là một phương pháp hữu hiệu đã được áp dụng rộng rãi để giải nhiều bài toán tối ưu và cho thấy hiệu quả rất tốt. Nó thu được kết quả vượt trội khi được kiểm tra với các bài toán benchmark so với các phương pháp dựa trên dân số khác. Hơn nữa, nó cũng đã được ứng dụng thành công trong việc giải quyết nhiều bài toán thiết kế tối ưu trong kỹ thuật được trình bày trong các tài liệu sau [5] - [7].

Tuy nhiên, giống như nhiều phương pháp tối ưu hóa dựa trên dân số khác, một trong những nhược điểm của DE và Jaya là thời gian tính toán tối ưu chậm hơn nhiều so với các phương pháp tối ưu hóa dựa trên gradient. Điều này là do DE và Jaya mất rất nhiều thời gian trong việc đánh giá độ phù hợp (hàm mục tiêu) của các cá thể trong quần thể. Cụ thể, trong bài toán tối ưu hóa cấu trúc, việc tính toán các giá trị hàm mục tiêu hoặc hàm ràng buộc thường được thực hiện bằng cách sử dụng phần tử hữu hạn để phân tích ứng xử của cấu trúc. Để khắc phục nhược điểm này, mạng nơron nhân tạo (ANN) được đề xuất kết hợp với thuật toán DE. Dựa trên ý tưởng mô phỏng cấu trúc não bộ, ANN có khả năng ước lượng đầu ra tương ứng với một tập dữ liệu đầu vào một cách nhanh chóng sau khi mạng đã được huấn luyện, còn được gọi là quá trình học tập. Nhờ ưu điểm vượt trội này, việc tính toán các giá trị hàm mục tiêu hoặc hàm ràng buộc trong thuật toán DE sẽ được thực hiện nhanh chóng. Kết quả là ANN sẽ giúp cải thiện đáng kể hiệu quả tính toán DE.

Hiện nay, các kết cấu từ vật liệu composite được sử dụng rộng rãi trong hầu hết các lĩnh vực như xây dựng, cơ khí, hàng hải, hàng không,... Trong đó, tấm gia cố bằng vật liệu composite là một dạng nổi bật và được sử dụng ngày càng nhiều những ưu điểm vượt trội của nó. Với sự kết hợp ưu điểm của vật liệu composite và kết cấu dầm gia cường, tấm composite gia cường có độ bền uốn rất cao với trọng lượng rất nhẹ. Do tính ứng dụng thực tế cao nên nhu cầu tối ưu hóa thiết kế kết cấu để tiết kiệm chi phí, tăng hiệu quả sử dụng cũng cao. Tuy nhiên, do sự phức tạp của việc tính toán ứng xử của loại cấu trúc đặc biệt này, việc tìm ra một thuật toán tốt để tối ưu hóa các tham số thiết kế là điều cần thiết để đảm bảo hiệu quả tính toán.

Bên cạnh đó, một trong những vấn đề quan trọng nhất trong thiết kế kỹ thuật là các thiết kế tối ưu thường bị ảnh hưởng bởi những yếu tố không chắc chắn (ngẫu nhiên) có thể xảy ra từ nhiều nguồn khác nhau, chẳng hạn như quy trình sản xuất, đặc tính vật liệu và môi trường hoạt động. Những yếu tố không chắc chắn này có thể khiến kết cấu hoạt động không đúng như thiết kế ban đầu và có thể dẫn đến rủi ro cho kết cấu [3]. Do đó, tối ưu hóa thiết kế dựa trên độ tin cậy (RBDO) có thể được coi là một chiến lược toàn diện để tìm ra một thiết kế tối ưu. Trong luận án này, các phương pháp vòng lặp đơn (Single-Loop Methods) sẽ được kết hợp với thuật toán tối ưu hóa metaheuristic để tạo thành một bộ công cụ mới để giải các bài toán RBDO của cấu trúc vật liệu composite.

Cụ thể, luận án này sẽ nghiên cứu một số kỹ thuật hiệu chỉnh và đề xuất cải tiến thuật toán Differential Evolution và thuật toán Jaya nhằm tăng độ hội tụ của thuật toán DE và Jaya. Thuật toán sửa đổi sau đó được kết hợp với ANN để phát triển một công cụ mới để giải quyết vấn đề tối ưu hóa thiết kế và vấn đề RBDO của cấu trúc vật liệu composite, chẳng hạn như tấm composite gia cường với thuật toán DE cải tiến và dầm composite với thuật toán Jaya cải tiến.

1.2 Động lực của nghiên cứu

- Phát triển / cải tiến các thuật toán hiện có nhằm nâng cao hiệu quả giải các bài toán tối ưu hóa kết cấu với độ chính xác và độ tin cậy cao.

- Nghiên cứu các ưu điểm của Mạng nơ ron nhân tạo (ANN) để kết hợp với các thuật toán tối ưu nhằm cải thiện tốc độ và hiệu suất giải các bài toán tối ưu hóa cấu trúc.

1.3 Mục tiêu của luận án

Thứ nhất, luận án tập trung vào việc phát triển các phương pháp tối ưu hóa Metaheuristic và kết hợp với ưu điểm của Mạng nơ ron nhân tạo trong việc xấp xỉ dữ liệu để xây dựng thuật toán mới giải các bài toán tối ưu hóa kết cấu vật liệu composite. Đặc biệt, thuật toán Differential Evolution gốc sẽ được hiệu chỉnh để cải thiện sự hội tụ trong việc tìm nghiệm tối ưu toàn cục và sau đó, ANN sẽ được tích hợp với thuật toán DE cải tiến (iDE) để tạo thành một thuật toán mới, được sử dụng để tìm kiếm thiết kế tối ưu của kết cấu tấm composite gia cường.

Thứ hai, luận án cũng đề xuất một bộ công cụ mới là sự kết hợp giữa thuật toán tối ưu hóa metaheuristic và các phương pháp vòng đơn để giải quyết các bài toán Tối ưu hóa thiết kế dựa trên độ tin cậy (RBDO). Đặc biệt, thuật toán Jaya ban đầu sẽ được sửa đổi để cải thiện tính hội tụ trong việc tìm kiếm giải pháp tối ưu của bài toán tối ưu hóa. Sau đó, phiên bản cải tiến của thuật toán Jaya sẽ được kết hợp với các phương pháp vòng đơn để giải quyết vấn đề Tối ưu hóa thiết kế dựa trên độ tin cậy của kết cấu dầm composite..

1.4 Phạm vi nghiên cứu của luận án

Luận án tập trung vào các vấn đề chính sau:

- Tối ưu hóa kết cấu giàn, dầm và kết cấu tấm gia cường bằng thép và vật liệu composite.
- Nghiên cứu cải tiến các phương pháp tối ưu hóa dựa trên dân số để tăng độ chính xác và hiệu quả trong việc giải các bài toán tối ưu hóa.
- Khai thác khả năng tạo mô hình gần đúng từ các tập dữ liệu của Mạng Nơ-ron để kết hợp với các thuật toán tối ưu nhằm nâng cao hiệu suất và khả năng giải nhiều dạng bài toán khác nhau.
- Kết hợp các thuật toán tối ưu với các nhóm phương pháp đánh giá độ tin cậy để giải các bài toán RBDO.
- Các bài toán được lựa chọn để tối ưu hóa tương đối đơn giản với mục đích chính là đánh giá hiệu quả, độ chính xác và độ tin cậy của

các phương pháp tối ưu đã đề xuất. Việc áp dụng các phương pháp tối ưu được đề xuất trong luận án cho các bài toán phức tạp hơn sẽ được tiếp tục nghiên cứu trong thời gian tới.

1.5 Cấu trúc của luận án

Luận án gồm bảy chương và được kết cấu như sau:

Chương 1 trình bày tổng quan về các thuật toán metaheuristic, cấu trúc vật liệu composite và đặc biệt là mạng nơron nhân tạo, vai trò và ứng dụng của nó trong quá trình tối ưu hóa. Chương này cũng đưa ra cách tổ chức của luận án thông qua phần đại cương, tính mới và mục tiêu của luận án để có cái nhìn tổng quan về những vấn đề được nghiên cứu trong luận án này. **Chương 2** giới thiệu tổng quan về vật liệu composite và lý thuyết về dầm composite Timoshenko và tấm composite gia cường là kết cấu chính được khảo sát và nghiên cứu trong các bài toán tối ưu của luận án này. **Chương 3** trình bày về thuật toán tối ưu hóa metaheuristic liên quan đến Differential Evolution và thuật toán Jaya. Chương này cũng đưa ra giới thiệu tổng quan và công thức tính cho bài toán Tối ưu hóa thiết kế dựa trên độ tin cậy (RBDO) và các phương pháp đề xuất để giải quyết vấn đề RBDO. **Chương 4** giới thiệu về Mạng Nơ-ron Nhân tạo (ANN), một số khái niệm cơ bản liên quan đến ANN và giới thiệu về Cấu trúc Mạng Nơ-ron. Bên cạnh đó, thuật toán huấn luyện Levenberg-Marquardt và hiện tượng overfitting cũng được trình bày trong chương này. **Chương 5** minh họa hiệu quả của thuật toán DE cải tiến và Jaya cải tiến trong việc giải các bài toán tối ưu hóa. Các kết cấu được khảo sát trong phần này bao gồm kết cấu giàn phẳng, kết cấu giàn không gian, dầm composite Timoshenko và tấm composite gia cường. Chương này cũng trình bày một cách tiếp cận mới được gọi là SLDM-iJaya được hình thành bởi sự kết hợp của thuật toán Jaya cải tiến và các phương pháp vòng lặp đơn để giải quyết vấn đề RBDO của dầm composite Timoshenko và ứng dụng của Mạng nơron nhân tạo khi nó được tích hợp với một phương pháp tối ưu hóa metaheuristic, chẳng hạn như thuật toán DE. Sự tích hợp này tạo thành một bộ công cụ mới gọi là thuật toán ABDE (ANN-Based Differential Evolution) và được áp dụng để giải bài toán thiết kế tối

ưu của tấm composite gia cường. Cuối cùng, **Chương 6** khép lại với các nhận xét kết luận và đưa ra một số khuyến nghị cho sự phát triển trong tương lai.

1.6 Nhận xét

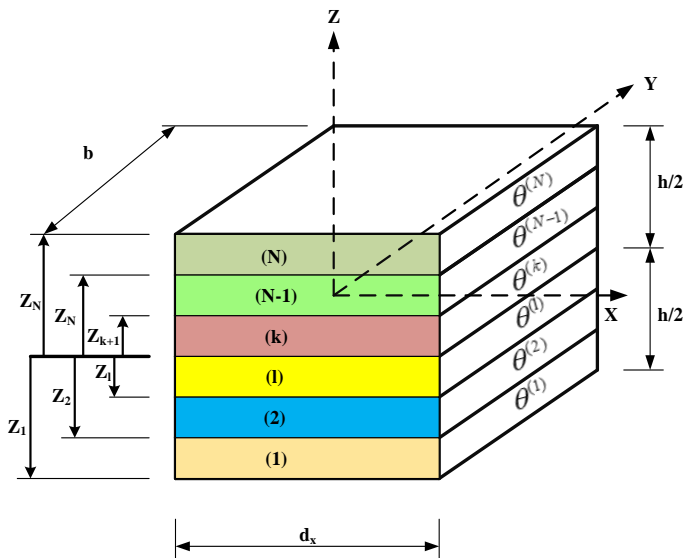
Trong chương này, các phương pháp tối ưu hóa metaheuristic, mạng nơ ron nhân tạo, cấu trúc vật liệu composite trong tối ưu hóa đã được giới thiệu tổng quan. Chương này cũng trình bày những điểm trọng tâm nghiên cứu của luận án và bố cục của luận án. Trong các chương tiếp theo, các lý thuyết cơ bản, một số phương pháp hiệu chỉnh để cải thiện hiệu quả tìm nghiệm của một số thuật toán metaheuristic và ứng dụng với kết quả số sẽ được trình bày.

CHƯƠNG 2

LÝ THUYẾT VỀ KẾT CẤU COMPOSITE TRONG THIẾT KẾ VÀ TỐI ƯU

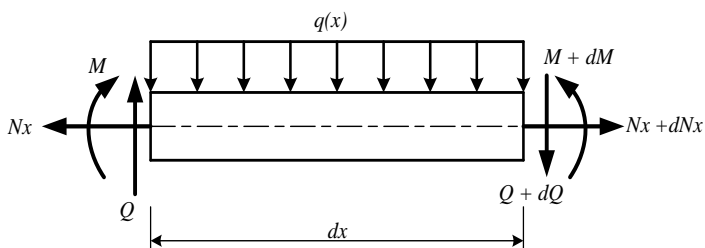
2.1 Công thức ứng suất và chuyển vị chính xác của dầm composite Timoshenko

Dầm Timoshenko composite nhiều lớp có thể được phân tích theo mô hình liên tục và mô hình rời rạc. Các mô hình rời rạc dễ thực hiện hơn nhưng khó đạt được lời giải chính xác. Ngoài ra, các mô hình rời rạc như mô hình phần tử hữu hạn không quá hiệu quả như phương pháp giải tích của mô hình liên tục. Trong phần này, việc xây dựng lời giải giải tích cho dầm composite nhiều lớp được sẽ được trình bày ngắn gọn. Để biết thêm chi tiết về phương pháp này, độc giả được có thể tham khảo công trình của Liu [8].



Hình 2. 1. Mô hình dầm composite nhiều lớp

Xét một đoạn dầm composite nhiều lớp có N lớp và hướng sợi của các lớp là $\theta_i (i = 1, \dots, N)$. Vị trí của các lớp là $z_i (i = 1, \dots, N)$. Dầm có tiết diện hình chữ nhật với chiều rộng b và chiều dài h như mô tả trong **Hình 2. 1**.



Hình 2. 2. Biểu đồ lực tác dụng lên đoạn dầm dx

Đoạn dầm dx chịu tác dụng của lực ngang như hình vẽ **Hình 2. 2**. Trường dịch vị của dầm composite nhiều lớp được tính theo công

thức giải tích dựa trên lý thuyết biến dạng cắt bậc nhất (còn gọi là lý thuyết dầm Timoshenko) như sau:

$$u_o(x) = -\bar{B} \left(\frac{q_0}{6} x^3 + \frac{1}{2} C_1 x^2 + 4C_4 x + C_5 \right) \quad (2.1)$$

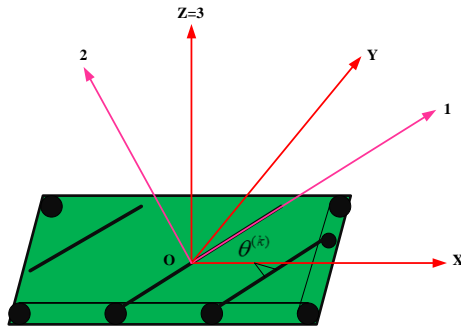
$$w_o(x) = -\bar{A} \frac{q_0}{24} x^4 - \frac{1}{6} \bar{A} C_1 x^3 - \left(\bar{C} \frac{q_0}{2} + \frac{1}{2} \bar{A} C_2 \right) x^2 + C_6 x + C_7 \quad (2.2)$$

$$\phi(x) = \bar{A} \left(\frac{q_0}{6} x^3 + \frac{1}{2} C_1 x^2 + 4C_2 x + C_3 \right) \quad (2.3)$$

Trong đó, $C_i (i=1, \dots, 7)$ là các hằng số tích phân bất định được xác định bằng cách sử dụng các điều kiện biên của các dầm composite nhiều lớp như được thể hiện trong phần sau.

$$\bar{A} = \frac{A_{11}}{b(B_{11}^2 - A_{11}D_{11})}, \bar{B} = \frac{B_{11}}{b(B_{11}^2 - A_{11}D_{11})}, \bar{C} = \frac{1}{bKA_{55}} \quad (2.4)$$

Trong đó, $A_{11}, B_{11}, D_{11}, A_{55}$ tương ứng là độ cứng chịu kéo, độ cứng kéo-uốn kết hợp, độ cứng chịu uốn và độ cứng chịu kéo của tấm composite. K là hệ số hiệu chỉnh cắt với giá trị là 5/6.



Hình 2. 3. Hệ tọa độ tấm và hệ tọa độ vật liệu

Trường ứng suất của dầm composite nhiều lớp bao gồm các thành phần ứng suất phẳng và thành phần ứng suất cắt. Theo hệ tọa độ giữa các vật liệu (123) và dầm / lớp (xyz) như được mô tả trong **Hình 2. 3**, trong đó hướng sợi trùng với trục 1, các thành phần ứng suất phẳng được biểu thị như sau:

$$\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{pmatrix} = T^{(k)} \bar{Q}^{(k)} \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix}, z_{k+1} \leq z \leq z_k \quad (2.5)$$

Trong đó, thành phần biến dạng $\varepsilon_y = 0, \gamma_{xy} = 0$, và

$$\varepsilon_x = -\bar{B} \left(\frac{q_0}{3} x^2 + C_1 x + C_4 \right) + z \bar{A} \left(\frac{q_0}{2} x^2 + C_1 x + C_2 \right) \quad (2.6)$$

$T^{(k)}$ là ma trận chuyển đổi tọa độ và $\bar{Q}^{(k)}$ là ma trận của hệ số độ cứng của vật liệu

$$T^{(k)} = \begin{bmatrix} \cos^2 \theta^{(k)} & \sin^2 \theta^{(k)} & 2 \sin \theta^{(k)} \cos \theta^{(k)} \\ \sin^2 \theta^{(k)} & \cos^2 \theta^{(k)} & -2 \sin \theta^{(k)} \cos \theta^{(k)} \\ -\sin \theta^{(k)} \cos \theta^{(k)} & \sin \theta^{(k)} \cos \theta^{(k)} & \cos^2 \theta^{(k)} - \sin^2 \theta^{(k)} \end{bmatrix} \quad (2.7)$$

$$\bar{Q}^{(k)} = \begin{bmatrix} \bar{Q}_{11}^{(k)} & \bar{Q}_{12}^{(k)} & \bar{Q}_{61}^{(k)} \\ \bar{Q}_{21}^{(k)} & \bar{Q}_{22}^{(k)} & \bar{Q}_{26}^{(k)} \\ \bar{Q}_{16}^{(k)} & \bar{Q}_{26}^{(k)} & \bar{Q}_{66}^{(k)} \end{bmatrix} \quad (2.8)$$

Các thành phần ứng suất cắt trong hệ tọa độ vật liệu là

$$\begin{pmatrix} \tau_{23} \\ \tau_{13} \end{pmatrix} = T_s^{(k)} \bar{Q}_s^{(k)} \begin{pmatrix} \gamma_{yz} \\ \gamma_{xz} \end{pmatrix}, z_{k+1} \leq z \leq z_k \quad (2.9)$$

Trong đó, các thành phần biến dạng cắt $\gamma_{yz} = 0$ và

$$\gamma_{xz} = \bar{A} \left(\frac{1_0}{6} x^3 + \frac{1}{2} C_1 x^2 + C_2 x + C_3 \right) - \bar{A} \frac{q_0}{6} x^3 - \frac{1}{2} \bar{A} C_1 x^2 - (\bar{C} q_0 + \bar{A} C_2) x + C_6 \quad (2.10)$$

Ma trận chuyển đổi tọa độ $T_s^{(k)}$ và ma trận hệ số độ cứng $\bar{Q}_s^{(k)}$ có thể được mô tả như sau:

$$T_s^{(k)} = \begin{bmatrix} \sin \theta^{(k)} & \cos \theta^{(k)} \\ \cos \theta^{(k)} & -\sin \theta^{(k)} \end{bmatrix} \quad (2.11)$$

$$\overline{Q}_s^{(k)} = \begin{bmatrix} \overline{Q}_{44}^{(k)} & \overline{Q}_{45}^{(k)} \\ \overline{Q}_{45}^{(k)} & \overline{Q}_{55}^{(k)} \end{bmatrix} \quad (2.12)$$

Trong các phương trình trên, $\overline{Q}_{ij}^{(k)}$ là hệ số độ cứng của lớp thứ k th trong hệ tọa độ lớp và được mô tả rõ ràng trong [8].

2.2 Tóm tắt lý thuyết tấm composite gia cường

Tấm composite gia cường được tạo thành bởi tấm composite kết hợp với dầm composite Timoshenko gia cường, như minh họa ở **Figure 2.4**. Dầm được coi như một gân gia cường và được đặt song song với các trục trên bề mặt tấm. Tấm của dầm cách mặt phẳng trung tâm của tấm khoảng cách e . Hệ thống dầm-tấm được rời rạc bởi một tập hợp các nút. Bậc tự do (DOF) của mỗi nút của tấm là $\mathbf{d} = [u, v, w, \beta_x, \beta_y]^T$, trong đó u, v, w là các chuyển vị ở mặt phẳng giữa tấm và β_x, β_y là các góc quay quanh trục y và trục x . Mỗi nút của dầm có bậc tự do là $\mathbf{d}_{st} = [u_r, u_s, u_z, \beta_r, \beta_s]^T$, trong đó u_r, u_s, u_z tương ứng là chuyển vị tâm của dầm và β_r, β_s là góc quay của dầm xung quanh trục r và trục s .

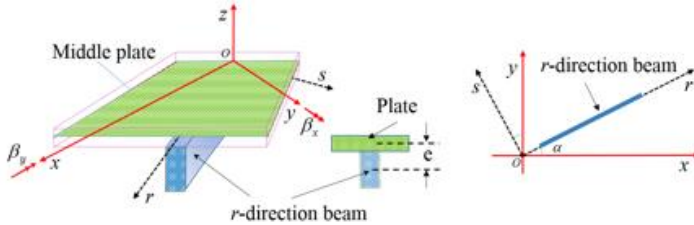


Figure 2.4. Tấm composite gia cường bởi dầm theo phương r

Sự tương thích về chuyển vị giữa tấm và dầm được đảm bảo bởi quan hệ::

$$u = u_r(r) + z\beta_r(r) \quad ; \quad v = z\beta_s(r) \quad ; \quad w = u_z(r) \quad (2.13)$$

Năng lượng biến dạng của tấm composite được cho bởi:

$$U_p = \frac{1}{2} \iint_A (\boldsymbol{\epsilon}_0^T \mathbf{D}^m \boldsymbol{\epsilon}_0 + \boldsymbol{\epsilon}_0^T \mathbf{D}^{mb} \boldsymbol{\kappa}_b + \boldsymbol{\kappa}_b^T \mathbf{D}^{mb} \boldsymbol{\epsilon}_0 + \boldsymbol{\kappa}_b^T \mathbf{D}^b \boldsymbol{\kappa}_b + \boldsymbol{\gamma}^T \mathbf{D}^s \boldsymbol{\gamma}) dA \quad (2.14)$$

Trong đó $\boldsymbol{\epsilon}_0, \boldsymbol{\kappa}_b, \boldsymbol{\gamma}$ tương ứng là các biến dạng màng, uốn và cắt của tấm composite và được biểu thị như sau

$$\boldsymbol{\epsilon}_0 = [u_{,x}, v_{,y}, u_{,y} + v_{,x}]^T; \boldsymbol{\kappa}_b = [\beta_{x,x}, \beta_{y,y}, \beta_{x,y} + \beta_{y,x}]^T; \boldsymbol{\gamma} = [w_{,x} + \beta_x, w_{,y} + \beta_y]^T \quad (2.15)$$

$\mathbf{D}^m, \mathbf{D}^{mb}, \mathbf{D}^b, \mathbf{D}^s$ là ma trận vật liệu của tấm

Năng lượng biến dạng của dầm gia cường được tính theo:

$$U_{st} = \frac{1}{2} \int_l ((\boldsymbol{\epsilon}_{st}^b)^T \mathbf{D}_{st}^b \boldsymbol{\epsilon}_{st}^b + (\boldsymbol{\epsilon}_{st}^s)^T \mathbf{D}_{st}^s \boldsymbol{\epsilon}_{st}^s) dx \quad (2.16)$$

Trong đó $\boldsymbol{\epsilon}_{st}^b, \boldsymbol{\epsilon}_{st}^s$ tương ứng là biến dạng uốn, biến dạng cắt của dầm và được biểu thị như sau

$$\boldsymbol{\epsilon}_{st}^b = [u_{r,r} + z_0 \beta_{r,r}, \beta_{r,r}, \beta_{s,r}]^T; \boldsymbol{\epsilon}_{st}^s = [u_{z,r} + \beta_r]^T \quad (2.17)$$

$\mathbf{D}_{st}^b, \mathbf{D}_{st}^s$ là ma trận vật liệu của dầm composite

Sử dụng nguyên lý chồng chất, tổng biến dạng năng lượng của tấm composite gia cường thu được:

$$U = U_p + \sum_{i=1}^{N_{st}} U_{st} \quad (2.18)$$

Trong đó N_{st} là số dầm gia cường.

Đối với phân tích tĩnh, các phương trình tổng thể cho tấm composite gia cường $[\mathbf{K}]\{\Delta\} = \{\mathbf{F}\}$ được trình bày chi tiết ở [9].

CHƯƠNG 3

PHƯƠNG PHÁP TỐI ƯU HÓA DỰA TRÊN ĐỘ TIN CẬY VỚI iJAYA VÀ iDE

3.1 Giải các bài toán Tối ưu hóa bằng cách sử dụng Differential Evolution cải tiến

Một vấn đề tối ưu hóa có thể được thể hiện như sau:

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad \begin{cases} h_i(\mathbf{x}) = 0 & i = 1, \dots, l \\ g_j(\mathbf{x}) \leq 0 & j = 1, \dots, m \end{cases} \quad (3.1)$$

Trong đó \mathbf{x} là vector của các biến thiết kế; $h_i(\mathbf{x}) = 0$ và $g_j(\mathbf{x}) \leq 0$ là những ràng buộc đẳng thức và bất đẳng thức; l, m lần lượt là số lượng các ràng buộc đẳng thức và bất đẳng thức; $f(\mathbf{x})$ là hàm mục tiêu có thể là hàm của trọng lượng, chi phí, v.v.

Tối ưu hóa thiết kế của một cấu trúc là tìm các giá trị tối ưu của các biến thiết kế trong không gian thiết kế sao cho hàm mục tiêu là nhỏ nhất. Để xử lý những vấn đề như vậy, nhiều phương pháp tối ưu hóa được sử dụng bao gồm cả phương pháp dựa trên gradient và dựa trên dân số. Trong bài báo này, Differential Evolution được sử dụng để giải bài toán tìm góc hướng sợi và độ dày tối ưu của tấm composite gia cường.

3.1.1 Tóm tắt thuật toán Differential Evolution [10], [11]

Thuật toán Differential Evolution lần đầu tiên được đề xuất bởi Storn và Price [1] đã được sử dụng rộng rãi để giải quyết nhiều loại vấn đề tối ưu hóa. Sơ đồ của thuật toán này bao gồm bốn giai đoạn như sau:

Giai đoạn 1: Khởi tạo

Tạo dân số ban đầu bằng cách lấy mẫu ngẫu nhiên từ không gian tìm kiếm

Giai đoạn 2: Đột biến

Tạo một vector đột biến mới v_i từ mỗi cá thể hiện tại x_i dựa trên các phép toán tạo đột biến.

Giai đoạn 3: Lai ghép

Tạo một vector thử nghiệm u_i bằng cách thay thế một số phần tử của vector đột biến v_i thông qua phép toán lai ghép.

Giai đoạn 4: Lựa chọn

So sánh vector thử ui với vector đích xi. Vector có giá trị hàm mục tiêu thấp hơn sẽ được lựa chọn cho thế hệ tiếp theo

Để cải thiện hiệu quả của thuật toán, *giai đoạn đột biến* và *giai đoạn lựa chọn* được hiệu chỉnh để tăng tốc độ hội tụ như sau:

Trong *giai đoạn đột biến*, các vector cha mẹ được chọn ngẫu nhiên từ bộ dân số hiện tại. Điều này có thể làm cho DE chậm khai thác nghiệm. Vì vậy, các cá thể tham gia gây đột biến nên được lựa chọn theo thứ tự ưu tiên dựa trên độ phù hợp của chúng. Bằng cách này, thông tin (đặc tính) tốt của thế hệ bố mẹ sẽ được lưu trữ ở con cái để sử dụng sau này và do đó sẽ giúp tăng tốc độ hội tụ. Để lưu trữ thông tin tốt trong quần thể con cái, các cá thể được chọn dựa trên kỹ thuật vòng quay Roulette (Roulette wheel) được đề xuất bởi Lipowski và Lipowska [12] thay vì lựa chọn ngẫu nhiên.

Trong *giai đoạn lựa chọn*, toán tử tinh hoa được giới thiệu bởi Padhye và cộng sự [13] được sử dụng cho quá trình lựa chọn thay vì lựa chọn cơ bản như trong DE thông thường. Trong quá trình lựa chọn tinh hoa, quần thể con C bao gồm các vector thử nghiệm được kết hợp với quần thể cha mẹ P của các vector mục tiêu để tạo ra một quần thể kết hợp Q. Sau đó, các cá thể tốt nhất được chọn từ quần thể kết hợp Q để tạo ra quần thể cho thế hệ tiếp theo. Làm như vậy, những cá thể tốt nhất của cả quần thể luôn được để dành cho thế hệ sau.

3.1.2 Thuật toán Roulette-wheel-Elitist Differential Evolution hiệu chỉnh

1:	Tạo bộ dân số ban đầu
2:	Đánh giá độ phù hợp của từng cá nhân trong quần thể
3:	while <điều kiện dừng chưa thỏa> do
4:	Tính xác suất lựa chọn cho mỗi cá thể
5:	for i =1 to NP do {NP: Kích thước dân số}
6:	giai đoạn đột biến dựa trên lựa chọn vòng quay
7:	Roulette
8:	$j_{rand} = randi(1,D)$ {D: số lượng biến thiết kế }
9:	for j =1 to D do

10:	if rand[0,1] < CR or j == j _{rand} then {CR: tham số
11:	điều khiển lai ghép }
12:	$u_{i,j} = x_{r1,j} + Fx(x_{r2,j} - x_{r3,j})$ {F: được chọn
13:	ngẫu nhiên trong khoảng [0,1] }
14:	else
15:	$u_{i,j} = x_{i,j}$
16:	end if
17:	end for
18:	Đánh giá véc tơ thử nghiệm u _i
	end for
	Thực hiện giai đoạn lựa chọn dựa trên toán tử lựa chọn tinh
	hoa
	end while

3.2 Improved Jaya algorithm

3.2.1 Thuật toán Jaya

Thuật toán Jaya là một kỹ thuật tối ưu hóa toàn cục dựa trên dân số, được phát triển gần đây bởi Ventaka Rao. Đây là một thuật toán đơn giản và dễ thực hiện, không yêu cầu bất kỳ tham số cụ thể của thuật toán nào [2]. Ý tưởng của thuật toán này là nó luôn cố gắng tránh thất bại (tránh xa nghiệm xấu nhất) và tiến gần hơn đến thành công (nghiệm tốt nhất) trong quá trình tìm kiếm. Phương pháp này dễ thực hiện vì nó chỉ yêu cầu các tham số kiểm soát chung (quy mô dân số và số thế hệ) để hoàn thành nhiệm vụ tối ưu hóa. Quá trình tối ưu hóa sử dụng thuật toán này có thể được tóm tắt trong bốn bước đơn giản sau đây như trong [14].

Đầu tiên, bộ dân số ban đầu bao gồm các cá thể N_p được tạo ngẫu nhiên trong không gian tìm kiếm. Mỗi ứng cử viên là một vectơ gồm n biến thiết kế $x_i = (x_1, x_2, \dots, x_n)$ và được tạo ra tuân theo các giới hạn dưới và trên như sau

$$x_{j,i} = x_{j,i}^l + rand[0,1] \times (x_{j,i}^u - x_{j,i}^l), j = (1, 2, \dots, n), i = (1, 2, \dots, N_p) \quad (3.2)$$

Trong đó x_j^u và x_j^l lần lượt là giới hạn trên và giới hạn dưới của biến thiết kế x_j ; $rand[0,1]$ tạo ra một số ngẫu nhiên trong khoảng $[0,1]$.

Gọi $f(x)$ là hàm mục tiêu của bài toán tối ưu hóa. Sự phù hợp của mỗi cá thể trong tập hợp dân số sau đó được đánh giá thông qua các giá trị của $f(x_i)$. Những cá thể nhận được giá trị phù hợp tốt nhất và kém nhất trong toàn bộ quần thể được gọi tương ứng là cá thể tốt nhất (x_{best}) và cá thể xấu nhất (x_{worst}). Nếu $x_{j,i,k}$ là giá trị của biến thứ j của ứng cử viên thứ i trong suốt vòng lặp thứ k , thì giá trị này được hiệu chỉnh ngẫu nhiên để tạo ra một vector mới $x'_{j,i,k}$

$$x'_{j,i,k} = x_{j,i,k} + r_{1,j,k} \times (x_{j,best,k} - |x_{j,i,k}|) - r_{2,j,k} \times (x_{j,worst,k} - |x_{j,i,k}|) \quad (3.3)$$

Trong đó $x_{j,worst,k}$ và $x_{j,best,k}$ là giá trị của biến thứ j tương ứng với cá thể xấu nhất (x_{worst}) và cá thể tốt nhất (x_{best}) trong bộ dân số ở vòng lặp thứ k . $r_{1,j,k}$ và $r_{2,j,k}$ là các số ngẫu nhiên trong phạm vi $[0,1]$; số hạng $r_{1,j,k} \times (x_{j,best,k} - |x_{j,i,k}|)$ cho biết xu hướng của biến thiết kế hướng tới nghiệm tốt nhất và số hạng $r_{2,j,k} \times (x_{j,worst,k} - |x_{j,i,k}|)$ cho thấy xu hướng của lời giải tránh nghiệm xấu nhất.

Tiếp theo, nếu giá trị của $x'_{j,i,k}$ nằm ngoài phạm vi giữa giới hạn dưới và giới hạn trên, một phép toán sẽ được thực hiện để điều chỉnh nó trở lại vùng cho phép

$$x'_{j,i,k} = \begin{cases} 2x_j^l - x'_{j,i,k} & \text{if } x'_{j,i,k} < x_j^l \\ 2x_j^u - x'_{j,i,k} & \text{if } x'_{j,i,k} > x_j^u \\ x'_{j,i,k} & \text{otherwise} \end{cases} \quad (3.4)$$

Cuối cùng, $x'_{j,i,k}$ được chấp nhận nếu nó mang lại giá trị hàm mục tiêu tốt hơn, nếu không, giá trị của $x_{j,i,k}$ sẽ được chọn. Tất cả các ứng cử viên được chấp nhận ở cuối vòng lặp được giữ lại và trở thành đầu vào cho vòng lặp tiếp theo.

$$x_{i,k+1} = \begin{cases} x'_{i,k} & \text{if } f(x'_{i,k}) \leq f(x_{i,k}) \\ x_{i,k} & \text{otherwise} \end{cases} \quad (3.5)$$

3.2.2 Thuật toán Jaya cải tiến

Trong thuật toán Jaya gốc, bộ dân số cho thế hệ (vòng lặp) tiếp theo được chọn dựa trên sự so sánh theo cặp của các giá trị hàm mục tiêu được tính từ x'_{ik} và x_{ik} . Điều này có thể dẫn đến việc những cá thể tốt bị loại bỏ do được so sánh với những cá thể mạnh hơn theo cặp. Mặc dù một cá thể có thể không tốt so với đối thủ của nó trong cặp, nó vẫn có thể tốt hơn những cá thể chiến thắng trong các cặp khác trong toàn bộ quần thể. Do đó, để đảm bảo những cá thể được chọn cho thế hệ tiếp theo là những cá thể tốt nhất, kỹ thuật chọn lọc tinh hoa do Padhye và cộng sự giới thiệu trong tài liệu [13] sẽ được sử dụng trong bước lựa chọn thay vì cách lựa chọn như trong thuật toán Jaya gốc. Quá trình chọn lọc được thực hiện như sau: đầu tiên, ở lần lặp thứ k , một tập hợp $2N_P$ cá thể, được gọi là tập \mathbf{A} , được tạo ra. Tập hợp \mathbf{A} này được hình thành bằng cách kết hợp tất cả các ứng cử viên x_{ik} trong tập \mathbf{X} và các ứng cử viên đã hiệu chỉnh $x'_{i,k}$ thuộc tập \mathbf{X}' . Sau đó, N_P cá thể tốt nhất trong $2N_P$ cá thể của tập \mathbf{A} được chọn để xây dựng bộ dân số cho thế hệ thứ $(k+1)$. Kỹ thuật chọn lọc tinh hoa được mô tả như sau:

1. **Đầu vào:** bộ dân số khởi tạo \mathbf{X} và bộ dân số hiệu chỉnh \mathbf{X}'
2. Tập hợp $\mathbf{A} = \mathbf{X} \cup \mathbf{X}'$ được tạo ra
3. Chọn N_{pbest} trong $2N_P$ cá thể của tập \mathbf{A} và gán cho tập \mathbf{X}_{new}
4. Đầu ra: \mathbf{X}_{new}

Bằng cách này, những cá thể tốt nhất của toàn bộ quần thể luôn được lưu trữ cho thế hệ sau. Kỹ thuật này giúp thuật toán đạt được giải pháp tối ưu với tốc độ hội tụ tốt hơn.

3.3 Tối ưu hóa dựa trên độ tin cậy sử dụng phương pháp vòng lặp đơn xác định.

3.3.1. Bài toán tối ưu hóa dựa trên độ tin cậy

Mô hình toán học của một bài toán RBDO điển hình có thể được mô tả như sau [15]:

$$\begin{aligned}
 & \text{Tìm} && \mathbf{d}, \boldsymbol{\mu}_x \\
 & \text{min} && f(\mathbf{d}, \boldsymbol{\mu}_x, \boldsymbol{\mu}_p) \\
 & \text{sao cho} && \text{Prob.}[g_i(\mathbf{d}, \mathbf{x}, \mathbf{p}) \leq 0] \geq R_i, \quad i = 1, 2, 3, \dots, m \\
 & && \mathbf{d}^{low} \leq \mathbf{d} \leq \mathbf{d}^{up}, \boldsymbol{\mu}_x^{low} \leq \boldsymbol{\mu}_x \leq \boldsymbol{\mu}_x^{up}
 \end{aligned} \tag{3.6}$$

Trong đó $f(\mathbf{d}, \boldsymbol{\mu}_x, \boldsymbol{\mu}_p)$ là hàm mục tiêu; \mathbf{d} là vector của các biến thiết kế xác định; \mathbf{d}^{low} và \mathbf{d}^{up} là các vector của giới hạn dưới và giới hạn trên của \mathbf{d} ; \mathbf{x} là vector của các biến thiết kế ngẫu nhiên; $\boldsymbol{\mu}_x$ là vector trung bình của \mathbf{x} ; $\boldsymbol{\mu}_x^{low}$ và $\boldsymbol{\mu}_x^{up}$ là các vector của giới hạn dưới và giới hạn trên của $\boldsymbol{\mu}_x$; là vector của các tham số ngẫu nhiên; $g_i(\mathbf{d}, \mathbf{x}, \mathbf{p})$ là hàm ràng buộc; R_i là độ tin cậy mong muốn thỏa mãn điều kiện ràng buộc; m số ràng buộc thỏa mãn; $\text{Prob}[\cdot]$ là toán tử xác suất, và $\text{Prob.}[g_i(\mathbf{d}, \mathbf{x}, \mathbf{p}) \leq 0] \geq R_i$ nghĩa là xác suất thỏa mãn ràng buộc $g_i(\mathbf{d}, \mathbf{x}, \mathbf{p}) \leq 0$ phải lớn hơn hoặc bằng độ tin cậy mong muốn R_i .

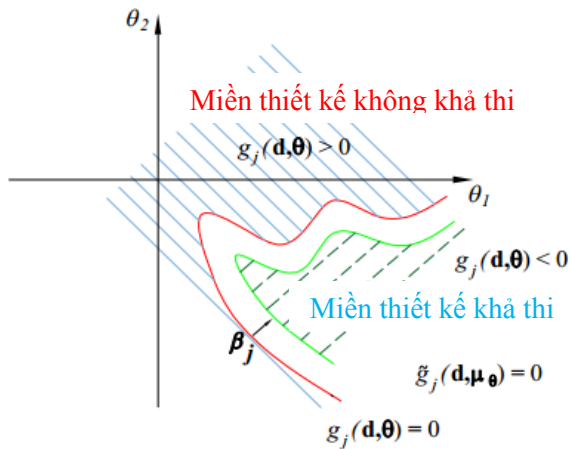
3.3.2. Thuật toán vòng lặp đơn xác định toàn cục

Phương pháp vòng lặp đơn xác định (single-loop deterministic method - SLDM) gần đây đã được giới thiệu để giải quyết các vấn đề tối ưu hóa thiết kế dựa trên độ tin cậy (RBDO) [16]. Trong SLDM, các ràng buộc xác suất được chuyển đổi thành các ràng buộc xác định gần đúng và do đó, các bài toán RBDO trở thành các bài toán tối ưu hóa xác định gần đúng, và do đó chi phí tính toán để giải các bài toán đó giảm đáng kể. Tuy nhiên, SLDM vẫn bị giới hạn ở các biến thiết kế liên tục, và các lời giải tối ưu có được từ SLDM thường

bị mắc kẹt vào cực trị địa phương. Để khắc phục hai nhược điểm này, phương pháp SLDM toàn cục được phát triển bởi Vinh và cộng sự. [3], và sau đó nó được áp dụng để giải quyết các bài toán RBDO của kết cấu giàn với cả biến thiết kế liên tục và rời rạc. Thuật toán tối ưu hóa được sử dụng trong bài báo của Vinh là thuật toán iDE, một phiên bản cải tiến của thuật toán Differential Evolution. Trong luận án này, một phương pháp tối ưu hóa metaheuristic khác, thuật toán Jaya với phiên bản cải tiến có tên iJaya, được sử dụng để tìm kiếm giải pháp tối ưu toàn cục. Phương pháp này được áp dụng để tối ưu hóa thiết kế kết cấu dầm composite.

*** Công thức của các ràng buộc xác định gần đúng**

Bước đầu tiên trong SLDM là hình thành một vùng khả thi xác định gần đúng bằng cách di chuyển biên của ràng buộc xác suất một khoảng β từ vị trí ban đầu của nó, như trong **Hình 3. 1**.



Hình 3. 1. Minh họa vùng thiết kế khả thi.

Trong Hình 3. 1, hàm trạng thái giới hạn được biểu diễn bằng đường cong màu đỏ, các đường cong biểu thị ranh giới của hàm ràng buộc xác định đã biến đổi và vùng nét đứt là vùng thiết kế khả thi xác định. Phép biến đổi đảm bảo rằng khoảng cách nhỏ nhất từ bất kỳ điểm

nào trên đường cong màu đỏ đến đường cong màu xanh lá cây là β_j , và các lời giải thu được đã được chứng minh là có thể thỏa mãn ràng buộc xác suất [16].

Sau khi vùng thiết kế khả thi được hình thành. Giả sử rằng μ_0 là một điểm nằm trên $g_i(\mathbf{d}, \mu_0)$, thì Điểm có thể xảy ra nhất (MPP), θ_{MPP} tương ứng với μ_0 có thể được xác định bằng cách dịch chuyển μ_0 ngược về phía $g_i(\mathbf{d}, \theta)$ một khoảng β

Như thể hiện trong tài liệu [17], θ_{MPP} trong vùng phá hủy trong không gian chuẩn tắc có thể được định nghĩa:

$$\theta_{j,MPP}^u = \beta_j \frac{(\partial g_i / \partial \theta_j^u)_*}{\sqrt{\sum_j (\partial g_i / \partial \theta_j^u)_*^2}}, j = 1, \dots, n, n+1, \dots, n+q \quad (3.7)$$

Trong đó u biểu thị không gian phân phối chuẩn tắc và các đạo hàm $(\partial g_i / \partial \theta_j^u)_*$ được tính tại $\theta_{j,MPP}^u$.

Mối quan hệ của các tham số ngẫu nhiên trong không gian thiết kế ban đầu và không gian chuẩn tắc thông thường được mô tả như sau

$$\theta_j^u = \frac{\theta_j - \mu\theta_j}{\sigma\theta_j} \quad (3.8)$$

Từ phương trình (3.8),

$$\frac{\partial g_i}{\partial \theta_j^u} = \frac{\partial g_i}{\partial \theta_j} \frac{\partial \theta_j}{\partial \theta_j^u} = \frac{\partial g_i}{\partial \theta_j} \sigma\theta_j \quad (3.9)$$

Kết hợp các phương trình (3.7),(3.8),(3.9), mối quan hệ giữa μ_0 và θ_{MPP} trong không gian thiết kế ban đầu được ký hiệu là

$$\theta_{j,MPP} = \mu\theta_j + \beta_j \sigma\theta_j \frac{\sigma\theta_j (\partial g_i / \partial \theta_j)_*}{\sqrt{\sum_j (\sigma\theta_j (\partial g_i / \partial \theta_j)_*)^2}} \quad (3.10)$$

Trong đó đạo hàm $(\partial g_i / \partial \theta_j)_*$ được tính tại $\theta_{j,MPP}$.

Theo Li và cộng sự [16], các đạo hàm $(\partial g_i / \partial \theta_j)_\#$ có thể được tính một cách xấp xỉ tại $\mu\theta_j$, và khi đó phương trình (3.10) có thể được viết lại thành

$$\Theta_{j,MPP} = \mu\theta_j + \beta_i \sigma\theta_j \frac{\sigma\theta_j (\partial g_i / \partial \theta_j)_\#}{\sqrt{\sum_j (\sigma\theta_j (\partial g_i / \partial \theta_j)_\#)^2}} \quad (3.11)$$

Trong đó các đạo hàm $(\partial g_i / \partial \theta_j)_\#$ được tính tại $\mu\theta_j$.

Khi Θ_{MPP} trong không gian thiết kế ban đầu đã được xác định, miền khả thi của bài toán RBDO có thể được biểu thị bằng các ràng buộc xác định gần đúng như sau:

$$g_i(\mathbf{d}, \boldsymbol{\mu}_\theta) = g_i(\mathbf{d}, \boldsymbol{\mu}_\theta + \beta_i \boldsymbol{\sigma}_\theta \mathbf{n}) \leq 0 \quad (3.12)$$

Trong đó $\mathbf{n} = (\boldsymbol{\sigma}_\theta \nabla_{\boldsymbol{\theta}} g_i(\boldsymbol{\mu}_\theta) / \|\boldsymbol{\sigma}_\theta \nabla_{\boldsymbol{\theta}} g_i(\boldsymbol{\mu}_\theta)\|)$ là vectơ gradient chuẩn hóa gần đúng được tính tại $\boldsymbol{\mu}_\theta$ on $g_i(\mathbf{d}, \boldsymbol{\theta})$.

Khi đó, Bài toán RBDO trong Phương trình (3.6) có thể được định nghĩa lại bằng bài toán ADO như sau:

$$\begin{aligned} & \text{Tìm} && \mathbf{d}, \boldsymbol{\mu}_x && (3.13) \\ & \text{min} && f(\mathbf{d}, \boldsymbol{\mu}_x) \\ & \text{Sao cho} && g_i(\mathbf{d}, \boldsymbol{\mu}_x + \beta_i \boldsymbol{\sigma}_\theta \mathbf{n}) \leq 0, \quad i = 1, 2, 3, \dots, m \\ & && \mathbf{d}^{low} \leq \mathbf{d} \leq \mathbf{d}^{up}, \boldsymbol{\mu}_x^{low} \leq \boldsymbol{\mu}_x \leq \boldsymbol{\mu}_x^{up} \end{aligned}$$

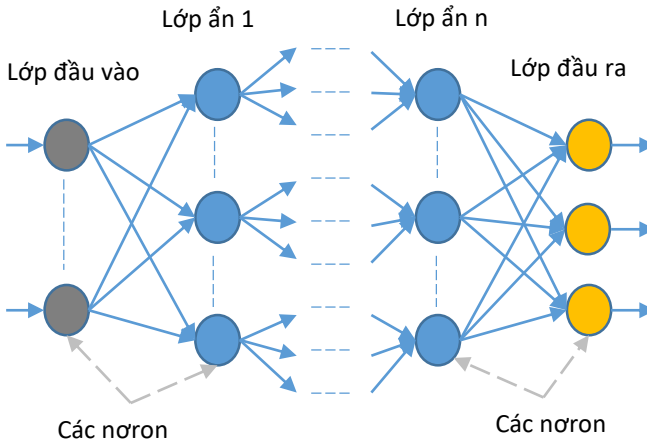
Cần lưu ý rằng các đạo hàm, $\nabla_{\boldsymbol{\theta}} g_i(\boldsymbol{\mu}_\theta)$, có thể dễ dàng đạt được thông qua đạo hàm trực tiếp của một hàm trạng thái giới hạn rõ ràng. Tuy nhiên, đối với các bài toán ứng dụng thực tế, các ứng xử cấu trúc thường được phân tích bằng các phương pháp số, và do đó hàm trạng thái giới hạn thường là ẩn. Do đó, để tính toán các đạo hàm $\nabla_{\boldsymbol{\theta}} g_i(\boldsymbol{\mu}_\theta)$, phương pháp sai phân hữu hạn, là một phương pháp đạo hàm số, được sử dụng trong nghiên cứu này.

CHƯƠNG 4 LÝ THUYẾT VỀ MẠNG NƠ-RON

4.1 Lý thuyết về mạng nơ-ron (Neural Network)

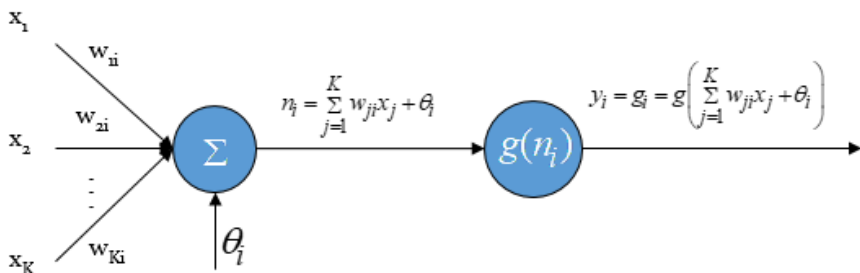
Mạng nơron bao gồm một lớp lớn các kiến trúc khác nhau và có nhiều ứng dụng khác nhau từ phân loại đối tượng, nhận dạng mẫu đến xấp xỉ. Trong nhiều trường hợp, bài toán được xem xét là xấp xỉ một hàm phi tuyến $f(x)$ bằng mạng nơron $f_{NN}(x)$. Và đó là một trong những ứng dụng hữu ích nhất của mạng nơron được sử dụng trong luận án này.

Các mạng nơron hữu ích nhất trong xấp xỉ hàm là Multi-Layer Perceptron (MLP) và Radial Basis Function (RBF). Trong luận văn này, mạng MLP được lựa chọn để nghiên cứu và ứng dụng trong việc giải các bài toán tối ưu hóa. MLP bao gồm một lớp đầu vào, một số lớp ẩn và một lớp đầu ra như trong **Hình 4. 1**.



Hình 4. 1. Mô hình mạng MLP

Trong mạng MLP, một nút đơn i , còn được gọi là nơron, bao gồm một bộ tổng và một hàm kích hoạt phi tuyến g được hiển thị trong **Hình 4. 2**.



Hình 4. 2. Nút đơn của một mạng MLP

Các đầu vào x_k , $k = 1, \dots, K$, được nhân với trọng số w_{ki} và được tổng hợp cùng với độ lệch không đổi θ_i . Tổng n_i sau đó trở thành đầu vào cho hàm kích hoạt g . Hàm kích hoạt ban đầu được chọn là một hàm *relay*, nhưng để thuận tiện trong toán học, hàm hyperbolic tangent (*tanh*) hoặc *sigmoid* thường được sử dụng phổ biến hơn.

Hàm *sigmoid* có dạng:

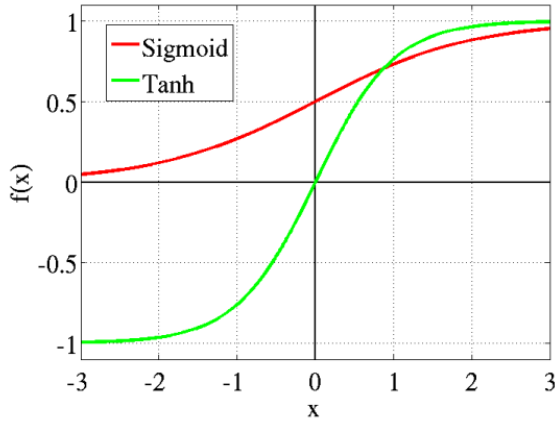
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Hàm sigmoid có giá trị giữa 0 và 1. Do đó, nó đặc biệt được sử dụng cho các mô hình cần dự đoán xác suất như một đầu ra. Vì xác suất của bất kỳ thứ gì chỉ tồn tại trong khoảng từ 0 đến 1, nên sigmoid là lựa chọn phù hợp.

Hàm *Hyperbolic tangent* được định nghĩa như sau:

$$\tanh(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

Tanh cũng giống như hàm *sigmoid* nhưng phạm vi của hàm *tanh* là từ (-1 to 1).

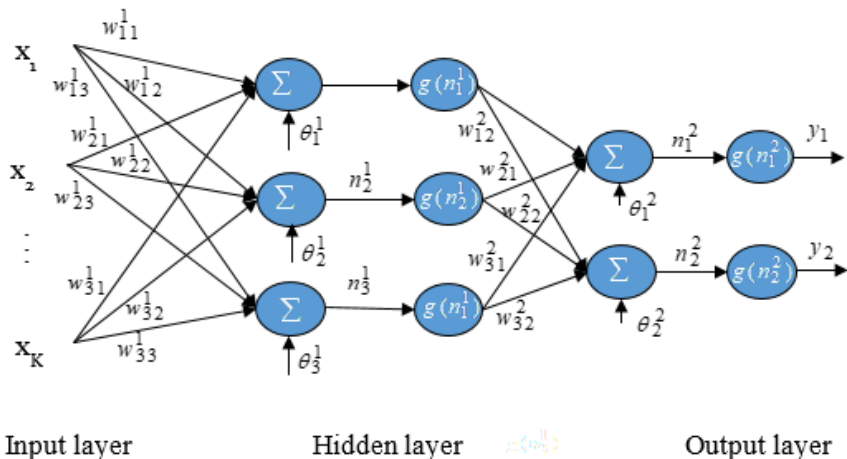


Hình 4. 3. Hàm *Tanh* và *Sigmoid*

Đầu ra của nút i :

$$y_i = g_i = g \left(\sum_{j=1}^K w_{ji} x_j + \theta_i \right) \quad (4.1)$$

Kết nối một số nút song song và nối tiếp tạo thành mạng MLP. Một mạng nơ-ron điển hình được hiển thị trong **Hình 4. 4**



Hình 4. 4. Perceptron nhiều lớp với một lớp ẩn. Cả hai lớp đều sử dụng cùng một hàm kích hoạt g

Đầu ra y_i , $i = 1, 2$ của mạng MLP

$$y_i = g\left(\sum_{j=1}^3 w_{ji}^2 g(n_j^1) + \theta_j^2\right) = g\left(\sum_{j=1}^3 w_{ji}^2 g\left(\sum_{k=1}^K w_{kj}^1 x_k + \theta_j^1\right) + \theta_j^2\right) \quad (4.2)$$

Từ (4.2) có thể kết luận rằng mạng MLP là một bản đồ tham số hóa phi tuyến từ không gian đầu vào $\mathbf{x} \in \mathbb{R}^K$ đến không gian đầu ra $\mathbf{y} \in \mathbb{R}^m$. Các tham số là trọng số w_{ji}^k và độ lệch θ_j^k . Các hàm kích hoạt g thường được giả định là giống nhau trong mỗi lớp và được biết trước.

Để tìm mạng MLP tốt nhất với tập hợp dữ liệu đầu vào - đầu ra nhất định (x_i, y_i) , $i = 1, \dots, N$, cần phải xác định các tham số (w_{ji}^k, θ_j^k) . Vấn đề trong trường hợp này được gọi là "bài toán khớp dữ liệu". Thủ tục diễn ra như sau. Đầu tiên người thiết kế phải chỉnh sửa cấu trúc của kiến trúc mạng MLP: số lượng các lớp ẩn và các nơ-ron (nút) trong mỗi lớp. Các hàm kích hoạt cho mỗi lớp cũng được chọn ở giai đoạn này, tức là chúng được giả định là đã biết. Các tham số chưa biết sẽ

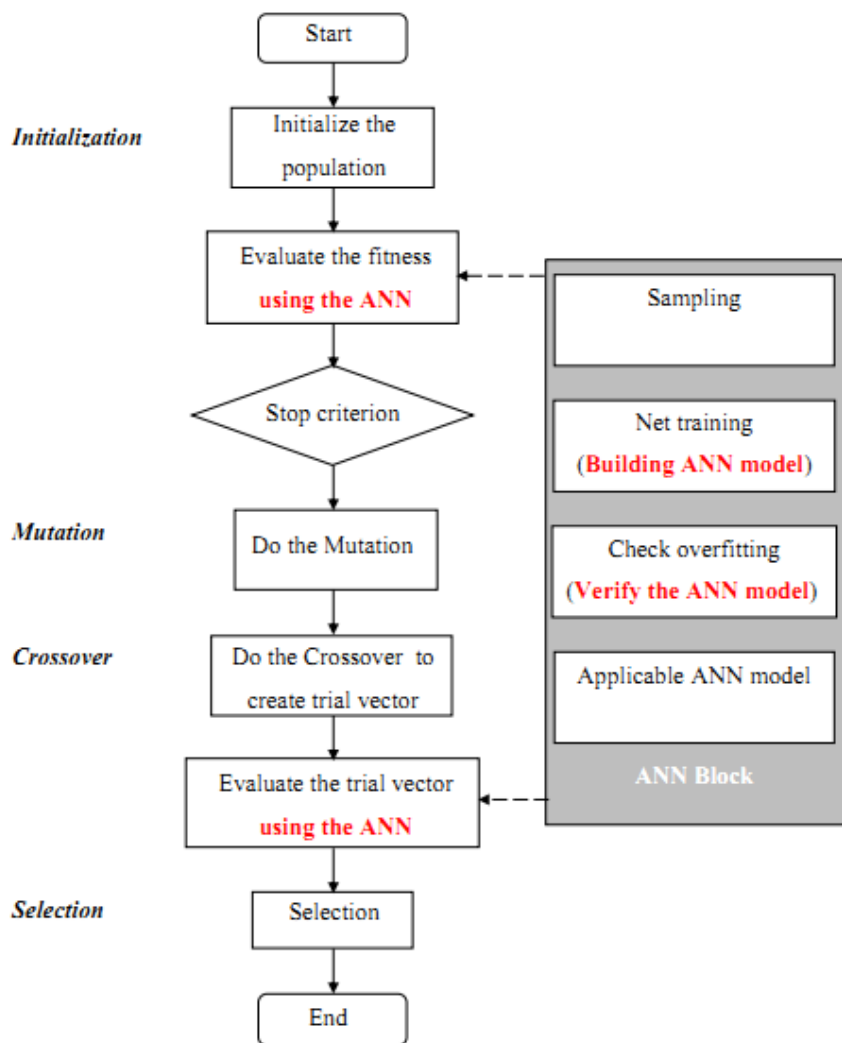
được ước tính sau đó là trọng số và độ chệch (w_{ji}^k, θ_j^k). Các thuật toán để xác định các tham số mạng được gọi là thuật toán học tập hoặc huấn luyện. Có rất nhiều thuật toán huấn luyện. Nổi tiếng nhất là lan truyền ngược và thuật toán Levenberg-Marquardt. Lan truyền ngược là một thuật toán dựa trên độ dốc, có nhiều biến thể. Levenberg-Marquardt thường hiệu quả hơn, nhưng cần nhiều bộ nhớ máy tính hơn. Trong luận án này, thuật toán Levenberg-Marquardt được sử dụng.

4.1 Các phương pháp tối ưu hóa metaheuristic dựa trên Mạng Nơron Nhân tạo

Bằng cách kết hợp các ưu điểm của Mạng nơ ron nhân tạo (ANN) và thuật toán Differential Evolution (DE), thuật toán tối ưu hóa Differential Evolution dựa trên ANN (ABDE) được đề xuất. Cụ thể, việc tính toán các giá trị hàm mục tiêu của các cá thể trong quần thể ở mỗi thế hệ và việc đánh giá các vectơ thử nghiệm trong bước lai ghép được thực hiện bằng phân tích phần tử hữu hạn như trong thuật toán DE ban đầu được thay thế bằng giá trị xấp xỉ của ANN như minh họa trong **Hình 4. 5**.

Trong **Hình 4. 5**, quá trình xây dựng một mô hình ANN thường bao gồm ba phần chính: lấy mẫu, đào tạo và xác minh mạng lưới như được minh họa trong “Khối ANN”. Trong luận án này, các mẫu được sử dụng để huấn luyện ANN được tạo bởi FEM và thuật toán được sử dụng để huấn luyện mạng là Levenberg-Marquardt.

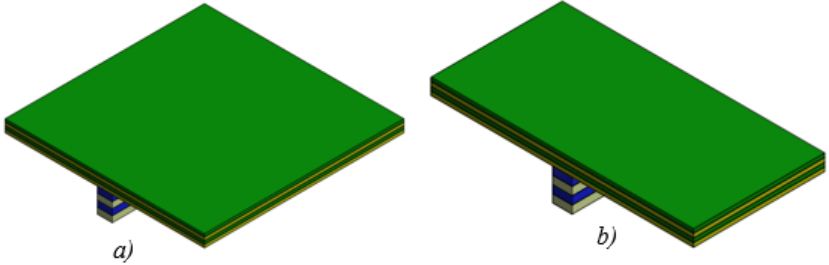
Nhờ lợi thế có thể đánh giá ngay sự phù hợp của tất cả các cá thể trong quần thể chỉ bằng một lệnh duy nhất '*net (dân số)*' thay vì sử dụng vòng lặp *for/end* và phân tích phần tử hữu hạn cho từng giá trị riêng lẻ, tốc độ tính toán của Thuật toán tối ưu hóa ABDE được tăng lên đáng kể. Mặc dù, giá trị gần đúng của ứng xử cấu trúc của tấm composite gia cường dựa trên mạng huấn luyện '*train (net, input, output)*' là có sai số, tuy nhiên sai số này là nhỏ và độ chính xác có thể chấp nhận được.



Hình 4. 5. Quá trình tối ưu hóa bằng cách sử dụng thuật toán tối ưu hóa Differential Evolution dựa trên mạng nơ ron nhân tạo (ABDE)

CHƯƠNG 5 SỰ PHÁT TRIỂN CÁC PHƯƠNG PHÁP TỐI ƯU METAHEURISTIC

5.1 Hiệu quả của thuật toán Differential Evolution cải tiến



a) Tấm vuông

b) tấm chữ nhật

Hình 5. 1. Mô hình tấm ghép cứng hình vuông và hình chữ nhật

Để chứng minh tính chính xác của thuật toán DE cải tiến (iDE), việc tối ưu hóa góc sợi của hai mô hình tấm composite gia cường hình vuông và hình chữ nhật được khảo sát trong phần này. Các dầm gia cường trong cả hai trường hợp đều theo hướng X như trong **Hình 5. 1**. Hàm mục tiêu trong bài toán này là năng lượng biến dạng của tấm.

Vấn đề tối ưu hóa được thể hiện dưới dạng:

$$\begin{cases} \min_{\theta} & U = \frac{1}{2} \mathbf{d}^T \mathbf{K} \mathbf{d} \\ \text{sao cho} & 0 \leq \theta_i \leq 180, \quad i = 1, \dots, 4 \end{cases}$$

Trong đó U là năng lượng biến dạng và θ_i là góc hướng sợi của lớp thứ i .

Để tiết kiệm thời gian tính toán nhưng vẫn đảm bảo độ chính xác, các giá trị số nguyên của các biến thiết kế đã được sử dụng trong phân tích này.

Các tham số tối ưu hóa của thuật toán Differential Evolution cải tiến được sử dụng trong phân tích này được chọn như sau:

Kích thước dân số (NP): 20 cá thể trong mỗi thế hệ

Tổng số thế hệ: 200

Tham số điều khiển lai ghép (Cr): 0.9

Yếu tố tỉ lệ (F_x): 0.8

Tolerance: 1e-6

Kết quả tối ưu của góc hướng sợi cho cả hai trường hợp được trình bày trong **Bảng 5.1**. Kết quả từ **Bảng 5.1** cho thấy các lời giải của iDE rất tương đồng với các lời giải của GA. Sai số của năng lượng biến dạng trong cả hai trường hợp là rất thấp. Mức tối đa chỉ khoảng 0,2% đối với trường hợp tấm hình chữ nhật. Tuy nhiên, thời gian tính toán của thuật toán iDE nhỏ hơn nhiều. Đặc biệt, trong trường hợp tấm hình chữ nhật, thời gian tính toán từ phương pháp GA gần gấp đôi so với phương pháp iDE, lần lượt là 4995 giây và 2851 giây. Điều này đã chứng minh tính chính xác và hiệu quả của phương pháp iDE.

Kết quả của **Bảng 5.1** cũng cho thấy các thông số hình học của kết cấu cũng có ảnh hưởng đến giá trị tối ưu của bài toán. Điều này được giải thích từ thực tế là các hướng sợi tối ưu của tấm hình vuông và hình chữ nhật là khá khác nhau trong cùng điều kiện.

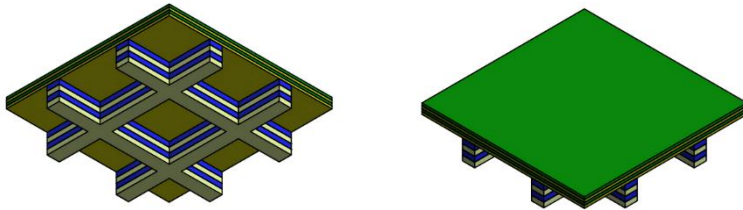
Bảng 5. 1. Kết quả tối ưu góc hướng sợi của 2 bài toán

Type of stiffened plate	Method	Optimal angle [Degree]				Strain energy (N.m)	Computational time (seconds)
		θ_1	θ_2	θ_3	θ_4		
Square	iDE	135	48	0	180	6183.2	2065
	GA	135	48	0	180	6183.1	2253
Rectangular	iDE	160	37	0	180	30366	2851
	GA	159	37	0	180	30300	4995

5.2 Tối ưu hóa tấm composite gia cường

5.2.1 Tối ưu hóa độ dày của tấm composite gia cường

Khảo sát bài toán tối ưu hóa tấm composite được gia cường bởi dầm composite như trong **Hình 5. 2** trong điều kiện tựa đơn. Các thông số bài toán đưa ra như sau: chiều dài dầm $a = 254$ mm, chiều cao dầm là h , tiết diện dầm gia cường có chiều rộng $c_x = 6,35$ mm, chiều cao là d_x . Phân tích tối ưu được thực hiện với hai trường hợp tấm hình vuông ($b = 254$ mm) và tấm hình chữ nhật ($b = 508$ mm).



Hình 5. 2. Mô hình tấm composite gia cường để tối ưu hóa

Cả tấm và dầm đều có cấu trúc bốn lớp đối xứng. Hướng sợi của mỗi lớp trên tấm tương ứng là $[90\ 45\ 45\ 90]$, và hướng sợi của lớp trên dầm là $[180\ 0\ 0\ 180]$. Dầm và tấm được làm bằng vật liệu giống nhau với các thông số: $E_1 = 144.8\text{GPa}$, $E_2 = E_3 = 9.65\text{GPa}$,

$G_{12} = G_{13} = 4.14 \text{ GPa}$, $G_{23} = 3.45 \text{ GPa}$, $\nu_{12} = \nu_{13} = \nu_{23} = 0.3$. Tấm chịu tải đều với giá trị $f = 0.6895 \text{ (N/mm}^2\text{)}$.

Bài toán có thể được biểu diễn như sau:

$$\left\{ \begin{array}{ll} \min_{t_p, t_b} & \text{Weight}(h, d_x) \\ \text{Sao cho} & \text{Disp} \leq 1 \\ & \sigma_{\text{Tsai-wu}} \leq 1 \end{array} \right.$$

Cụ thể, hàm mục tiêu là khối lượng tấm composite gia cường và chịu hai ràng buộc bao gồm chuyển vị của tấm gia cường nhỏ hơn 1 và ứng suất Tsai-Wu cũng nhỏ hơn 1.

Kết quả tối ưu hóa độ dày được tính toán trong các trường hợp dầm gia cường khác nhau (hai dầm gia cường theo hướng XY, XX, YY hoặc bốn dầm gia cường theo hướng XX-YY) và được trình bày trong **Bảng 5. 2**. Kết quả cho thấy, trong trường hợp tấm vuông, hàm mục tiêu đạt giá trị thấp nhất với trường hợp 4 dầm gia cường (XX-YY). Lý do là vì mặc dù số lượng dầm gia cường nhiều hơn trường hợp 2 dầm nhưng chiều dày tấm thu được lại nhỏ hơn nhiều so với trường hợp 2 dầm. Vì vậy trong trường hợp tấm vuông nên chọn kết cấu 4 dầm gia cường để có kết quả tốt hơn. Trong trường hợp tấm hình chữ nhật, kết quả tốt nhất thu được trong trường hợp 2 dầm gia cường được bố trí theo phương Y (Y-Y), với giá trị chiều dày tấm là nhỏ nhất. Từ các kết quả trên rút ra một nhận xét là phương án tối ưu thường đạt được với chiều dày tấm nhỏ kết hợp với số lượng dầm gia cường nhiều hơn.

Bảng 5. 2. Kết quả độ dày tối ưu cho các bài toán tấm composite gia cường

Gia cường	R/S	Theta	h	d_x	Hàm mục tiêu	Thời gian (s)
X-Y	S	90/45/180/0	13.165	58.509	1.442168	2892
X-X / Y-Y	S	90/45/180/0	10.187	74.651	1.247581	3111
XX-YY	S	90/45/180/0	8.04236	42.7559	1.104029	4989
X-Y	R	90/45/180/0	27.40554	29.32186	5.109688	12475
Y-Y	R	90/45/180/0	11.8138	106.397	3.071304	10309
X-X	R	90/45/180/0	17.53	119.837	3.68064	10706

5.2.2 Tối ưu hóa dựa trên mạng nơ-ron nhân tạo của tấm composite gia cường

Trong phần này, các thuật toán tối ưu hóa ABDE được đề xuất đã được áp dụng cho ba mô hình bài toán: tấm hình chữ nhật gia cường hướng X (RX), tấm hình chữ nhật gia cường hướng Y (RY) và tấm hình vuông được gia cường theo hai hướng X và Y (S -XY). Kết quả trong phần này thực hiện trên máy tính có cấu hình sau:

Processor: Intel® Core™ i5-2430M CPU @ 2.4GHZ

Installed Memory (RAM): 4,00 GB (3,90 GB usable)

System type: 64-bit Operating System, x64-based processor

Đầu tiên, các mẫu dùng cho huấn luyện ANN được tạo ra với 10.000 mẫu cho mỗi trường hợp. Mạng ANN sau khi huấn luyện sẽ được kiểm tra "hiện tượng overfitting" với 1.296 mẫu khác để đảm bảo độ chính xác và khả năng ứng dụng của mô hình mạng. Thời gian lấy mẫu và sai số trung bình của quá trình kiểm tra overfitting được trình bày trong **Bảng 5.3** dưới đây :

Bảng 5. 3. Tạo mẫu và sai số mô hình

Samples	Case	Total time (seconds)	Training time	Avg. Error
10000	R-X	38535	1.59	2.55 %
10000	R-Y	39188	1.30	3.55 %
10000	S-XY	19434	7.33	3.01 %

Sai số trung bình cho ba trường hợp là nhỏ, do đó, mô hình ANN có thể đảm bảo độ chính xác trong việc xấp xỉ hàm mục tiêu của tấm composite gia cường trong quá trình tối ưu hóa thuật toán DE.

Kết quả tối ưu được thể hiện trong **Bảng 5. 4**. So với kết quả tối ưu thu được từ thuật toán DE, các giá trị của biến thiết kế được tính toán từ ABDE là rất giống nhau. Sai số của các giá trị hàm mục tiêu thu được đối với ba mô hình lần lượt là 0,86%, 1,57%, 1,99%. Những sai số này bị ảnh hưởng nhiều nhất bởi mô hình huấn luyện của ANN. Tuy nhiên, với mức sai số tối đa là 1,99% thì hoàn toàn có thể chấp nhận được. Bên cạnh đó, ưu điểm vượt trội về thời gian tính toán, cũng là ưu điểm mạnh nhất của thuật toán ABDE, được thể hiện rất tốt. Chi phí cho mô hình R-X do ABDE tính toán chỉ là 8 giây so với 2851 giây của DE, nhanh hơn 356 lần. Trong 2 trường hợp còn lại của mô hình R-Y và S-XY, chi phí lần lượt là 13 giây và 5 giây so với 2903 giây và 1497 giây của thuật toán DE. Tính ưu việt này là do hàm mục tiêu có thể được xấp xỉ từ mô hình ANN bằng một lệnh duy nhất là *net (dân số)* thay vì sử dụng vòng lặp for / end như trong thuật toán DE ban đầu. Trong ba trường hợp trên, thời gian tính toán thấp nhất với sai số cao nhất là trường hợp S-XY. Điều này là do tấm hình vuông có kích thước nhỏ hơn so với tấm hình chữ nhật nên thời gian tính toán nhanh hơn. Trong khi đó, sai số có thể do nguyên nhân xấp xỉ trong tính gần đúng của mô hình ANN và độ chính xác của mô hình FEM trong việc tạo mẫu để huấn luyện mạng nơron. Do đó, để đảm bảo tính chính xác và hiệu quả của phương pháp đề xuất, bước huấn luyện và lấy mẫu ANN phải được xem xét cẩn thận.

Kết quả trong **Bảng 5. 4** có thể chứng minh độ chính xác và hiệu quả vượt trội của thuật toán ABDE so với thuật toán DE độc lập. Điều này đặc biệt hữu ích để giải quyết các vấn đề có số lượng vòng lặp lớn và các hàm mục tiêu chi phí cao.

Bảng 5. 4. So sánh độ chính xác và thời gian tính toán giữa DE và ABDE

Type of stiffened plate	Method	Optimal angle [Degree]				Strain energy (N.m)	Error (%)	Cost (seconds)
		θ_1	θ_2	θ_3	θ_4			
R-X	DE	160	37	0	180	30366	0.86	2851
	ABDE	160	38	0	180	30104		8
R-Y	DE	178	12	0	180	33039	1.57	2903
	ABDE	177	11	0	180	32520		13
S-XY	DE	135	45	0	0	4223	1.99	1497
	ABDE	134	42	0	0	4307		5

Ngoài ra, thuật toán này cũng có thể áp dụng cho nhiều dạng bài toán bao gồm cả bài toán tuyến tính và phi tuyến tính vì ý tưởng ở đây chỉ là thay thế phương pháp đánh giá hàm mục tiêu bằng cách sử dụng ANN thay vì sử dụng FEM. Do đó, thời gian tính toán được tiết kiệm hơn. Hơn nữa, mô hình ANN có thể xấp xỉ cả đáp ứng tuyến tính và phi tuyến. Vì vậy, kỹ thuật này sẽ được áp dụng cho cả bài toán tối ưu hóa tuyến tính và phi tuyến tính.

5.3 Tối ưu hóa xác định của dầm composite

Để tối ưu hóa các dầm composite nhiều lớp, các hướng sợi, chiều rộng và chiều dày (độ dày của lớp) có thể được chọn làm các biến thiết kế. Trong phần này, tối ưu hóa thiết kế được thực hiện cho dầm nhiều lớp composite Timoshenko một chiều với điều kiện ràng buộc độ cứng, độ bền và tách lớp được đưa vào công thức toán cho mô hình tối ưu hóa. Do đó, như đã giải thích rõ ràng trong công trình của

Liu [8], chỉ có chiều rộng và chiều dày của dầm được chọn làm biến số thiết kế để có được thiết kế nhẹ của dầm.

Xét mô hình tối ưu hóa thiết kế của dầm composite có tính đến ràng buộc liên quan tiêu chí hư hỏng về độ cứng, độ bền và tiêu chí về hư hỏng do phân tách. Bài toán có thể được diễn đạt như sau [8]:

$$\text{Tìm } \mathbf{d} = [b, h]^T$$

$$\text{Minimize } W(\mathbf{d})$$

$$\text{Sao cho } g_j = \left(\frac{\sigma_1^2}{X_t X_c} - \frac{\sigma_1 \sigma_2}{\sqrt{X_t X_c Y_t Y_c}} + \frac{\sigma_2^2}{Y_t Y_c} + \frac{\tau_{12}^2}{S^2} + \frac{X_c - X_t}{X_t X_c} \sigma_1 + \frac{Y_c - Y_t}{Y_t Y_c} \sigma_2 - 1 \right) < 0$$

$$f_j = \left(\frac{\tau_{13}^2}{S_{13}^2} + \frac{\tau_{23}^2}{S_{23}^2} - 1 \right) < 0$$

$$r = w_0(\alpha L) - \bar{w}_0 \leq 0$$

$$\underline{b} \leq b \leq \bar{b}$$

$$\underline{h} \leq h \leq \bar{h}$$

Trong đó $W(\mathbf{d})$ là khối lượng của dầm nhiều lớp composite. g, f và r lần lượt là hàm phá hủy độ bền, hàm phá hủy phân tách và hàm phá hủy độ cứng. \underline{b} và \bar{b} là giới hạn dưới và giới hạn trên của chiều rộng của dầm. \underline{h} và \bar{h} lần lượt là giới hạn dưới và giới hạn trên của độ dày dầm. αL xác định vị trí theo phương x nơi theo dõi độ lệch của dầm. α khác nhau đối với các loại điều kiện biên (PP: $\alpha=1/2$, FF: $\alpha=1/2$, FP: $\alpha=505/873$ and CL: $\alpha=1$). $w_0(\alpha L)$ là độ lệch của dầm tại vị trí αL . \bar{w}_0 là giới hạn về độ lệch của dầm. Chỉ số dưới ($j = 1, 2, \dots, N_m$) cho biết điểm được giám sát thứ j trong tập N_m điểm được giám sát của độ bền và độ tách lớp. X_t và X_c lần lượt là cường độ kéo và cường độ nén dọc theo trục 1 của hệ tọa độ vật liệu. Y_t và Y_c lần lượt là cường độ kéo và cường độ nén dọc theo trục 2 của hệ tọa độ vật liệu. S là độ bền cắt trên mặt phẳng 102 của hệ tọa độ

vật liệu. S_{12} và S_{23} là độ bền cắt trên mặt phẳng 103 và 203 của hệ tọa độ vật liệu. Trong luận án này, cho $S_{12} = S_{23}$.

5.5.1 Tối ưu hóa các biến thiết kế: b và h

Xét các dầm composite có các đặc tính vật liệu cho trong **Bảng 5. 5**. Các dầm có $N = 8$ lớp với góc hướng sợi đối xứng là $[0/90/45 / -45]$ s. Nhịp của dầm composite là $L = 7.2\text{m}$. Các dầm chịu tải trọng phân bố đều $q_0 = 10^5 \text{ N/m}$ và được khảo sát dưới nhiều loại ràng buộc khác nhau bao gồm PP, FF, FP và CL. Thiết kế ban đầu của dầm composite nhiều lớp là $b = 0.3 \text{ m}$ và $h = 0.48 \text{ m}$ (độ dày của mỗi lớp là 0.06), khối lượng $W = 1597 \text{ kg}$. Giới hạn dưới và giới hạn trên của các biến thiết kế là $0.1\text{m} \leq b \leq 2\text{m}$, $0.2\text{m} \leq h \leq 2\text{m}$.

Bảng 5. 5. Đặc tính vật liệu các lớp

Thông số	T300/5208
E_1 (GPa)	136.00
$E_2 = E_3$ (GPa)	9.80
$G_{12} = G_{13}$ (GPa)	4.70
G_{23} (GPa)	5.20
$\nu_{12} = \nu_{13}$	0.28
ν_{23}	0.15
ρ (kg/m ³)	1540
X_t (Mpa)	1550
X_c (Mpa)	1090
Y_t (Mpa)	59
Y_c (Mpa)	59
S (Mpa)	75
$S_{13} = S_{13}$ (Mpa)	75

Các bài toán thiết kế tối ưu hóa được giải quyết với các loại điều kiện biên khác nhau (P-P, F-F, F-P và C-L) bằng cách sử dụng bốn thuật toán dựa trên dân số khác nhau bao gồm DE, IDE, Jaya và iJaya và một thuật toán dựa trên gradient từ công trình của Liu. Kết quả số được trình bày trong **Bảng 5. 6**. Như trong bảng, khối lượng tối ưu

thu được từ iJaya rất phù hợp với các giải pháp khác. Tuy nhiên, thuật toán iJaya tiêu tốn ít thời gian nhất để đạt được nghiệm tối ưu so với các cách tiếp cận khác. Trong số năm phương pháp, thuật toán SQP (được triển khai bằng lệnh *fmincon* trong Matlab) được sử dụng trong công trình của Liu đã đạt được giải pháp tối ưu rất nhanh nhưng nó có thể bị mắc kẹt ở nghiệm tối ưu cục bộ và điều đó ảnh hưởng đến giá trị của lời giải tối ưu. So với các phương pháp tối ưu hóa toàn cục khác (DE, IDE, Jaya), phương pháp iJaya hoàn toàn vượt trội hơn. Đặc biệt, iJaya đã giảm thời gian tính toán ít nhất 13% đối với trường hợp điều kiện C-L đến tối đa 27% đối với trường hợp điều kiện P-P so với thuật toán Jaya ban đầu. Số lần phân tích hàm trung bình giảm đến 21% khi sử dụng phương pháp iJaya thay vì Jaya. Phương pháp iJaya cũng nhanh hơn cách tiếp cận iDE trong việc đạt được các nghiệm tối ưu và tốc độ hội tụ của iJaya xấp xỉ gấp đôi tốc độ của DE trong việc đạt được các giá trị tối ưu của các biến thiết kế. Từ những phân tích trên, iJaya có thể được coi là thuật toán hiệu quả nhất vì nó tìm được nghiệm toàn cục ở tốc độ hội tụ cao nhất.

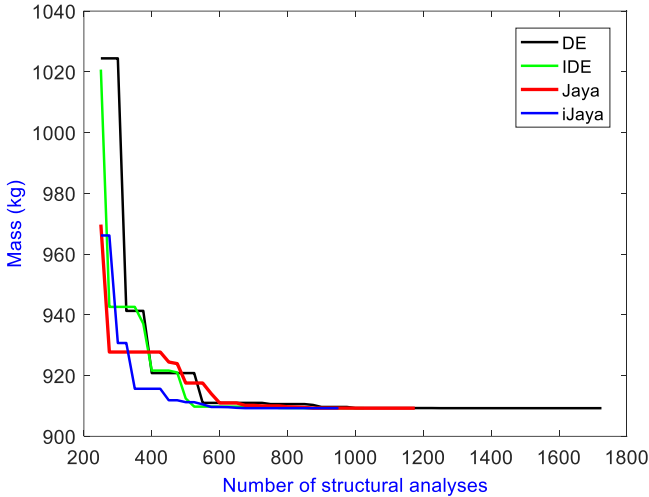
Bảng 5. 6. So sánh thiết kế tối ưu với các biến thiết kế liên tục

BC	Optimal results	Liu [39]	DE	IDE	Jaya	iJaya
P-P	Mass	909.2634	909.2634	909.2634	909.2634	909.2634
	$[h, b]$	[0.1000, 0.8200]	[0.1000, 0.8200]	[0.1000, 0.8200]	[0.1000, 0.8200]	[0.1000, 0.8200]
	Worst mass	-	909.2640	909.2642	909.2638	909.2639
	Mean mass	-	909.2636	909.2637	909.2636	909.2636
	Std.	-	0.0001	0.0002	0.0001	0.0001
	Average f-count	7	1660	1075	1181	933
	CPU time (s)	45 [0.54]*	1.06	0.69	0.82	0.60
F-F	Mass	560.7427	560.7428	560.7428	560.7428	560.7427
	$[h, b]$	[0.1000, 0.5057]	[0.1000, 0.5057]	[0.1000, 0.5057]	[0.1000, 0.5057]	[0.1000, 0.5057]
	Worst mass	-	560.7430	560.7433	560.7430	560.7430
	Mean mass	-	560.7429	560.7429	560.7428	560.7428
	Std.	-	0.0001	0.0001	0.0001	0.0001
	Average f-count	9	1654	1084	1144	921

CPU time (s)	6 [0.17]	1.15	0.73	0.84	0.70
Mass	706.5145	706.5145	706.5145	706.5145	706.5145
$[a, b]$	[0.1000, 0.6372]	[0.1000, 0.6372]	[0.1000, 0.6372]	[0.1000, 0.6372]	[0.1000, 0.6372]
Worst mass	-	706.5148	706.5150	706.5151	706.5149
F-P Mean mass	-	706.5146	706.5147	706.5146	706.5147
Std.	-	0.0001	0.0001	0.0001	0.0001
Average f-count	32	1641	1061	1153	924
CPU time (s)	11 [0.11]	1.07	0.67	0.77	0.61
Mass	2065	2064.9645	2064.9645	2064.9646	2064.9645
$[a, b]$	[0.1000, 1.8623]	[0.1000, 1.8623]	[0.1000, 1.8623]	[0.1000, 1.8623]	[0.1000, 1.8623]
Worst mass	-	2064.9658	2064.9658	2064.9654	2064.9655
C-L Mean mass	-	2064.9649	2064.9650	2064.9649	2064.9648
Std.	-	0.0003	0.0003	0.0002	0.0003
Average f-count	16	1591	954	1039	863
CPU time (s)	38 [0.13]	0.51	0.29	0.31	0.27

[]*: Thời gian chạy CPU trong nghiên cứu này bằng cách sử dụng thuật toán SQP tính bằng *fmincon* Matlab.

Tốc độ hội tụ của bốn thuật toán tối ưu hóa toàn cục dựa trên dân số cho trường hợp điều kiện PP có thể được thể hiện trong **Hình 5. 3**. Như chúng ta có thể dễ dàng thấy, số lượng phân tích cấu trúc để đạt được thiết kế nhẹ của dầm P-P của thuật toán iJaya ít hơn nhiều so với DE, IDE và Jaya gốc. Chỉ mất khoảng 450 phân tích cấu trúc để hội tụ đến khối lượng tối ưu trong khi DE và Jaya dành khoảng 600 phân tích và hơn 500 phân tích cho trường hợp iDE. Những con số này đã chứng minh rằng thuật toán iJaya vượt trội hơn những thuật toán khác trong việc đạt được lời giải tối ưu.



Hình 5. 3. Các đường cong hội tụ của DE, IDE, Jaya và iJaya đối với dầm có điều kiện P-P

5.5.2 Thiết kế tối ưu với biến: b và t_i

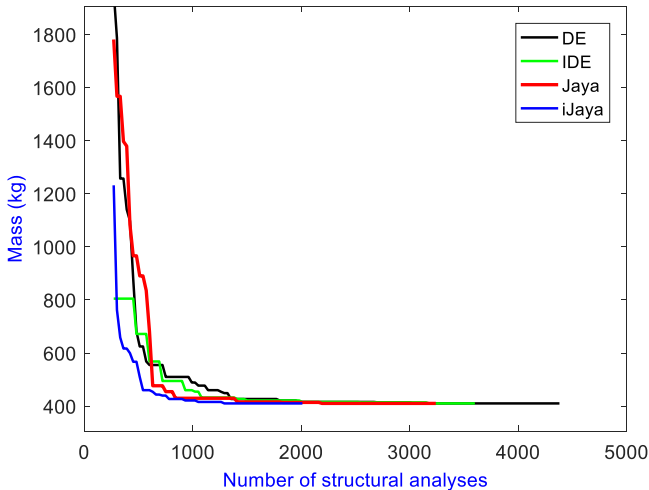
Trong phần này, chiều dày của dầm composite nhiều lớp (h) được chia thành độ dày của các lớp của dầm để tối ưu hóa. Điều này được thực hiện với mục đích cải thiện thiết kế tối ưu của dầm nhiều lớp composite và đạt được trọng lượng nhẹ hơn cho dầm. Độ dày của mỗi lớp, được biểu thị bằng $[t_1, t_2, t_3, t_4]$ s, được chọn làm biến thiết kế trong trường hợp này có thể được coi là biến thiết kế rời rạc.

Bảng 5.7. So sánh thiết kế tối ưu với các biến thiết kế rời rạc

BC	Optimal results	Liu [39]	DE	IDE	Jaya	iJaya
P-P	Mass	909.2634	410.256	410.256	410.256	410.256
	b	0.1000	0.100	0.100	0.100	0.100
	$[t_1, t_2, t_3, t_4]_s$	[0.1025, 0.1025, 0.1025, 0.1025] _s	[0.190, 0.060, 0.060, 0.060] _s	[0.190, 0.060, 0.060, 0.060] _s	[0.190, 0.060, 0.060, 0.060] _s	[0.190, 0.060, 0.060, 0.060] _s
	Worst mass	-	415.800	415.800	415.800	415.800
	Mean mass	-	410.533	414.341	411.088	410.872
	Std.	-	1.240	2.508	2.031	1.793
	Average f-count	7	3375	2225	3132	2190
	CPU time (s)	45	2.75	1.81	2.24	1.51
	Mass	560.7427	260.568	260.568	260.568	260.568
	b	0.1000	0.100	0.100	0.100	0.100
F-F	$[t_1, t_2, t_3, t_4]_s$	[0.0632, 0.0632, 0.0632, 0.0632] _s	[0.105, 0.040, 0.040, 0.050] _s	[0.085, 0.050, 0.050, 0.055] _s	[0.100, 0.045, 0.040, 0.050] _s	[0.100, 0.045, 0.040, 0.050] _s
	Worst mass	-	260.568	266.112	260.568	266.112
	Mean mass	-	260.568	260.860	260.568	260.876
	Std.	-	0.000	1.272	0.000	1.307
	Average f-count	9	2675	2325	2534	1663
	CPU time (s)	6	2.18	1.90	1.81	1.15
	Mass	706.5145	327.096	327.096	327.096	327.096
	b	0.1000	0.100	0.100	0.100	0.100
	$[t_1, t_2, t_3, t_4]_s$	[0.0796, 0.0796, 0.0796, 0.0796] _s	[0.130, 0.060, 0.055, 0.050] _s	[0.130, 0.055, 0.060, 0.050] _s	[0.130, 0.050, 0.050, 0.065] _s	[0.125, 0.055, 0.055, 0.060] _s
	Worst mass	-	327.096	327.096	327.096	327.096
Mean mass	-	327.096	327.096	327.096	327.096	
Std.	-	0.000	0.000	0.000	0.000	
Average f-count	32	2650	2225	2499	1657	
CPU time (s)	11	2.17	1.83	1.83	1.17	
C-L	Mass	2065	942.480	942.480	942.480	942.480
	b	0.1000	0.100	0.100	0.100	0.100
	$[t_1, t_2, t_3, t_4]_s$	[0.2328, 0.2328, 0.2328, 0.2328] _s	[0.430, 0.140, 0.140, 0.140] _s	[0.400, 0.145, 0.150, 0.160] _s	[0.425, 0.140, 0.145, 0.140] _s	[0.425, 0.140, 0.140, 0.145] _s
	Worst mass	-	942.480	948.024	942.480	942.480
	Mean mass	-	942.480	945.415	942.480	942.480
	Std.	-	0.000	2.852	0.000	0.000
	Average f-count	16	3500	2350	3129	2190

Kết quả thu được được trình bày trong **Bảng 5.7**. Khối lượng tối ưu thu được từ tất cả các phương pháp tối ưu hóa dựa dân số với các biến thiết kế rời rạc chỉ bằng một nửa so với kết quả thu được từ công trình của Liu sử dụng thuật toán SQP với các biến thiết kế liên tục. Và một lần nữa, phương pháp iJaya lại chiếm ưu thế so với các phương pháp khác về cả số lần phân tích hàm và thời gian CPU. Kết quả từ **Bảng 5.7** cũng cho thấy rằng việc tối ưu hóa với các biến thiết kế rời rạc hiệu quả hơn nhiều so với việc giải quyết vấn đề với các biến thiết kế liên tục.

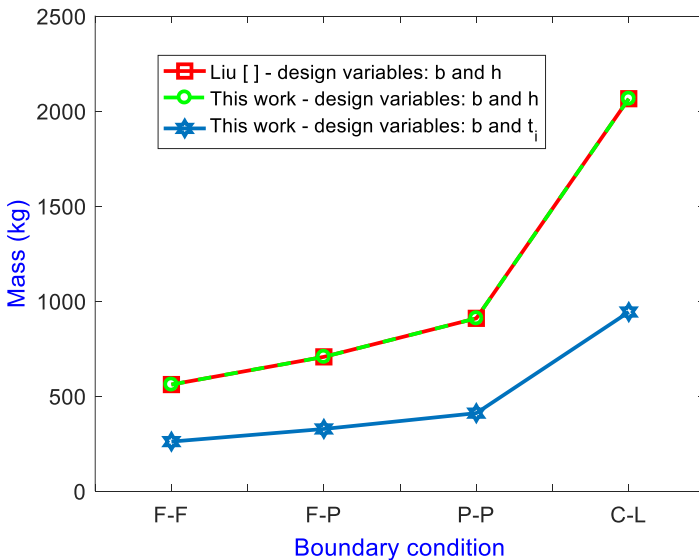
Sự hội tụ của bốn thuật toán tối ưu hóa toàn cục dựa trên dân số cho trường hợp điều kiện P-P được vẽ trong **Hình 5.4**. Biểu đồ cho thấy rằng phương pháp iJaya đạt được thiết kế nhẹ tối ưu với số lượng phân tích cấu trúc ít nhất so với nghiệm được giải bằng DE, IDE và Jaya gốc.



Hình 5.4. Các đường cong hội tụ của DE, IDE, Jaya và iJaya đối với dầm có điều kiện P-P.

Phương pháp iJaya chỉ thực hiện khoảng 900 phân tích cấu trúc để tiến gần tới khối lượng tối ưu trong khi những phương pháp khác mất khoảng 1400 phân tích cấu trúc để đạt được giải pháp tối ưu. Điều này cho thấy iJaya là phương pháp hội tụ nhanh nhất với độ chính xác cao trong số các phương pháp so sánh.

Trong **Hình 5. 5** thể hiện sự so sánh giữa các phương pháp tối ưu hóa sử dụng các biến thiết kế chiều dày của lớp và các phương pháp sử dụng các biến thiết kế chiều dày của dầm với các loại điều kiện biên khác nhau. So sánh đầu tiên giữa phương pháp tối ưu hóa toàn cục dựa trên dân số (đường màu xanh lá cây) và phương pháp tối ưu hóa dựa trên độ dốc trong công trình của Liu (đường màu đỏ) đã cho thấy sự đồng thuận tốt về kết quả tối ưu. So sánh thứ hai là giữa các phương pháp tối ưu hóa toàn cục với các biến thiết kế rìa rạc (đường màu xanh lam) và thuật toán SQP với các biến thiết kế liên tục trong công trình của Liu (đường màu đỏ).



Hình 5. 5. Comparison of different design approaches with different boundary conditions.

Kết quả cho thấy rằng phương pháp đầu tiên đã làm giảm đáng kể giá trị của khối lượng tối ưu của dầm, đặc biệt, tới một nửa giá trị thu so với giá trị thu được từ phương pháp thứ hai. Điều này chứng tỏ mạnh mẽ hiệu quả và độ chính xác của phương pháp đề xuất sử dụng thuật toán iJaya với các biến thiết kế rời rạc trong việc tối ưu hóa các dầm nhiều lớp tổng hợp.

5.4 Tối ưu hóa dựa trên độ tin cậy dầm composite Timoshenko

5.4.1 Kiểm nghiệm phương pháp SLDM-iJaya

Một mô hình toán học đơn giản, đã được nghiên cứu rộng rãi trong tài liệu, được sử dụng để kiểm tra phương pháp đề xuất một cách chi tiết. Bài toán chỉ có hai biến ngẫu nhiên phân phối chuẩn x_1 , x_2 , ràng buộc xác suất và không có biến thiết kế xác định và tham số ngẫu nhiên. Bài toán RBDO được mô tả như sau [18], [19]:

$$\text{Tìm } \mathbf{x} = \{x_1, x_2\}$$

$$\text{min } f(\mathbf{x}) = \mu_{x_1} + \mu_{x_2}$$

$$\text{sao cho } \text{Prob}[g_i(\mathbf{x}) \leq 0] \geq \Phi(\beta_i), \quad i = 1, \dots, 3$$

$$g_1(\mathbf{x}) = 1 - \frac{x_1^2 x_2}{20},$$

$$g_2(\mathbf{x}) = 1 - \frac{(x_1 + x_2 - 5)^2}{30} - \frac{(x_1 - x_2 - 12)^2}{120}$$

$$g_3(\mathbf{x}) = 1 - \frac{80}{(x_1^2 + 8x_2 + 5)},$$

$$0 \leq \mu_{x_j} \leq 10, \quad \sigma_j = 0.3, \quad j = 1, 2,$$

$$\beta_i = 3, \quad i = 1, \dots, 3.$$

Các nghiệm cho bài toán RBDO ở trên có thể thu được bằng cách sử dụng đánh giá độ tin cậy và tối ưu hóa tuần tự (SORA) trong công trình của Du và Chen [20], cách tiếp cận không gian thiết kế đáng tin cậy do Shan và Wang [19] đề xuất và vòng lặp đơn và phương pháp lặp kép kết hợp với thuật toán IDE được trình bày trong bài báo của Ho-Huu và cộng sự [3]. Kết quả thu được từ các phương pháp trên được so sánh với kết quả của phương pháp đề xuất trong nghiên

cứ này và được trình bày trong **Bảng 5. 8**. Phương pháp chính được đề xuất trong bài báo này là sự kết hợp giữa SLDM và thuật toán iJaya. Tuy nhiên, các kết quả thu được từ thuật toán DLM-iJaya, được hình thành bằng cách kết hợp DLM với thuật toán iJaya, cũng được trình bày để so sánh rộng hơn. Như được trình bày trong **Bảng 5. 8**, các lời giải đạt được của SLDM-iJaya và DLM-iJaya rất đồng thuận với các giải nghiệm được bằng các phương pháp khác. Tuy nhiên, chi phí tính toán đã giảm đáng kể khi sử dụng phương pháp DLM-iJaya và phương pháp SLDM-iJaya thay vì sử dụng DLM-IDE và SLDM-IDE như trong nghiên cứu của Ho-Huu et al. [3]. Đặc biệt, số lượng đánh giá hàm ràng buộc giảm gần 16% đối với trường hợp sử dụng DLM-iJaya và xuống còn 20% đối với trường hợp sử dụng SLDM-iJaya. Mức giảm chi phí tính toán lần lượt là 52% và 36%. Những con số này đã xác minh mạnh mẽ hiệu quả và độ chính xác của phương pháp SLDM-iJaya được đề xuất.

Bảng 5. 8. So sánh kết quả tối ưu hóa giữa các phương pháp

Design variables (mm)	Du and Chen [146]		Shan and Wang [145]		Ho-Huu et al. [3]		This study	
	SORA	RDS	DLM-IDE	SLDM-IDE	DLM-iJaya	SLDM-iJaya		
x_1	3.4409	3.4406	3.4391	3.4406	3.4391	3.4406		
x_2	3.2909	3.2800	3.2866	3.2800	3.2866	3.2800		
Objective	6.7318	6.7205	6.7256	6.7205	6.7256	6.7205		
β	β_1^{FORM}	3.011	2.995	3.000	2.995	3.000	2.995	
	β_2^{FORM}	3.011	2.978	3.000	2.978	3.000	2.978	
	β_3^{FORM}	10.026	10.047	10.038	10.047	10.038	10.047	
	β_1^{MCS}	2.998	2.969	2.980	2.969	2.980	2.969	
	β_2^{MCS}	3.059	3.022	3.061	3.022	3.061	3.022	
	β_3^{MCS}	Infinite	Infinite	Infinite	Infinite	Infinite	Infinite	
Computational cost	No. of constraint function evaluations	-	-	9917	4600	8388	3700	
	Time (s)	-	-	0.912	0.390	0.4365	0.251	

* Lưu ý rằng xác suất thất bại bằng 0% với $\beta > 5$.

5.4.2 Thiết kế trọng lượng nhẹ dựa trên độ tin cậy

Trong phần này, phương pháp đề xuất được áp dụng để giải bài toán thiết kế tối ưu hóa dựa trên độ tin cậy trọng lượng nhẹ của các dầm composite nhiều lớp với các loại điều kiện biên khác nhau. Kết quả tối ưu được tính toán với ba chỉ số độ tin cậy yêu cầu khác nhau, $\beta_{r,i}$. Kết quả được liệt kê trong **Bảng 5.9**. Khối lượng tối ưu thu được

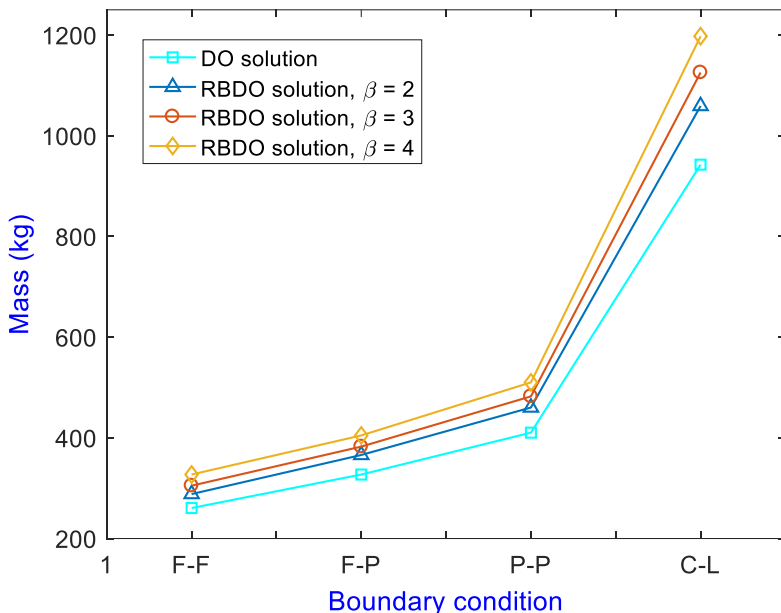
được so sánh giữa phương pháp DLM-iJaya và SLDM-iJaya. Như chúng ta có thể dễ dàng thấy, các nghiệm tối ưu về khối lượng thu được giữa 2 phương pháp là rất tương đồng, nhưng số phân tích cấu trúc và thời gian CPU của SLDM-iJaya vượt trội hơn so với DLM-iJaya trong mọi trường hợp. Đặc biệt, thời gian CPU tiêu thụ bởi SLDM-iJaya ít hơn 5 lần so với cái sau trong trường hợp điều kiện P-P. Con số này về mặt số lần phân tích cấu trúc thậm chí còn nhỏ hơn 10 lần đối với trường hợp của $\beta_{r,i} = 4$. Các hiệu suất tương tự cũng xảy ra đối với các điều kiện biên còn lại của F-F, F-P và C-L. Hơn nữa, phương pháp SLDM-iJaya cũng có thể đạt được khối lượng tối ưu tương tự mà DLM-iJaya thu được với giá trị chỉ số độ tin cậy nhỏ hơn, $\beta_{o,1}$. Ví dụ, trong trường hợp điều kiện F-P với chỉ số độ tin cậy yêu cầu $\beta_{r,i} = 2$, giá trị của $\beta_{o,1}$ được tính toán bởi DLM-iJaya là 2.009 trong khi con số này đạt được bởi SLDM-iJaya là 1.997. Những kết quả này chứng minh rằng phương pháp SLDM-iJaya có thể thu được nghiệm tối ưu với độ tin cậy cao hơn và ít chi phí hơn.

Hình 5. 6 mô tả sự so sánh các kết quả tối ưu của bài toán RBDO với các mức độ tin cậy khác nhau. Đường màu lục lam biểu thị kết quả thu được từ bài toán DO mà không cần phân tích độ tin cậy, và trọng lượng tối ưu của nó chắc chắn là các giá trị nhỏ nhất. Ba dòng trên cùng đại diện cho các lời giải tối ưu đạt được của các bài toán RBDO với các giá trị khác nhau của chỉ số độ tin cậy $\beta = 2, 3$ và 4 tương ứng.

Như chúng ta thấy, các giá trị của β càng lớn thì các giá trị của khối lượng càng lớn, có nghĩa là nếu chúng ta muốn nâng cao độ tin cậy của một kết cấu, chúng ta cần phải gia cố kết cấu bằng nhiều vật liệu hơn và điều đó làm cho trọng lượng của kết cấu tăng lên. Điều này cũng ngụ ý rằng việc tối ưu hóa mà không có phân tích độ tin cậy có thể dẫn đến cấu trúc rủi ro cao, mặc dù chi phí tiết kiệm được rất nhiều. Do đó, để cân bằng giữa chi phí và an toàn, phương pháp RBDO cần được tính đến trong việc tối ưu hóa thiết kế kết cấu. Và SLMD-iJaya là một trong những phương pháp hiệu quả nhất đã được chứng minh để giải quyết những vấn đề như vậy.

Bảng 5. 9. Kết quả tối ưu thiết kế nhẹ dựa trên độ tin cậy với các mức độ tin cậy khác nhau.

BC	Optimal results	Required reliability index, $\beta_i = 2$		Required reliability index, $\beta_i = 3$		Required reliability index, $\beta_i = 4$	
		DLM-iJaya	SLDM-iJaya	DLM-iJaya	SLDM-iJaya	DLM-iJaya	SLDM-iJaya
P-P	Mass	460.152	460.152	482.328	482.328	510.048	510.048
	b	0.100	0.100	0.100	0.100	0.100	0.100
	$[t_1, t_2, t_3, t_4]_s$	[0.200, 0.070, 0.070, 0.075] _s	[0.200, 0.070, 0.075, 0.070] _s	[0.225, 0.070, 0.070, 0.070] _s	[0.220, 0.075, 0.070, 0.070] _s	[0.230, 0.075, 0.075, 0.080] _s	[0.225, 0.075, 0.075, 0.085] _s
	f-count	319885	52416	471364	58590	449279	43776
	CPU time (s)	167.834	41.480	220.78	39.35	327.325	43.753
	$\beta_{b,1}$	2.026	2.026	3.012	2.978	4.006	3.978
	$\beta_{b,2}$	Infinite	Infinite	Infinite	Infinite	Infinite	Infinite
	$\beta_{b,3}$	Infinite	Infinite	Infinite	Infinite	Infinite	Infinite
	Mass	288.288	288.288	304.920	304.920	327.096	327.096
	b	0.100	0.100	0.100	0.100	0.100	0.100
F-F	$[t_1, t_2, t_3, t_4]_s$	[0.200, 0.070, 0.070, 0.075] _s	[0.200, 0.070, 0.075, 0.070] _s	[0.135, 0.045, 0.050, 0.045] _s	[0.130, 0.050, 0.050, 0.045] _s	[0.125, 0.055, 0.065, 0.050] _s	[0.120, 0.065, 0.055, 0.055] _s
	f-count	402721	49968	375080	46080	316688	50688
	CPU time (s)	182.614	33.308	165.33	28.26	183.42	36.66
	$\beta_{b,1}$	2.034	2.034	3.046	2.992	4.076	3.994
	$\beta_{b,2}$	Infinite	Infinite	Infinite	Infinite	Infinite	Infinite
	$\beta_{b,3}$	Infinite	Infinite	Infinite	Infinite	Infinite	Infinite
	Mass	365.904	360.360	382.536	382.536	404.712	404.712
F-P	b	0.100	0.100	0.100	0.100	0.100	0.100
	$[t_1, t_2, t_3, t_4]_s$	[0.200, 0.070, 0.070, 0.075] _s	[0.200, 0.070, 0.075, 0.070] _s	[0.165, 0.060, 0.060, 0.060] _s	[0.165, 0.060, 0.060, 0.060] _s	[0.175, 0.070, 0.060, 0.060] _s	[0.175, 0.060, 0.070, 0.060] _s
	f-count	327635	59616	661090	47664	354492	42480
	CPU time (s)	130.49	36.69	1381.7	35.50	156.90	26.737
	$\beta_{b,1}$	2.009	1.997	3.018	3.018	4.021	4.031
	$\beta_{b,2}$	Infinite	Infinite	Infinite	Infinite	Infinite	Infinite
	$\beta_{b,3}$	Infinite	Infinite	Infinite	Infinite	Infinite	Infinite
	Mass	1058.904	1058.904	1125.432	1125.432	1197.504	1197.504
	b	0.100	0.100	0.100	0.100	0.100	0.100
	C-L	$[t_1, t_2, t_3, t_4]_s$	[0.490, 0.155, 0.155, 0.155] _s	[0.485, 0.155, 0.155, 0.160] _s	[0.500, 0.165, 0.175, 0.175] _s	[0.490, 0.165, 0.190, 0.170] _s	[0.500, 0.175, 0.205, 0.200] _s
f-count		448839	58752	403747	43920	424718	33120
CPU time (s)		100.80	17.58	135.25	16.71	104.69	10.36
$\beta_{b,1}$		2.009	1.996	3.003	2.979	4.003	3.999
$\beta_{b,2}$		Infinite	Infinite	Infinite	Infinite	Infinite	Infinite
$\beta_{b,3}$		Infinite	Infinite	Infinite	Infinite	Infinite	Infinite



Hình 5. 6. So sánh các kết quả tối ưu RBDO với các mức độ tin cậy khác nhau

CHƯƠNG 6 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Nhận xét kết luận.

- Trong luận án này, phương pháp Differential Evolution cải tiến được giới thiệu và ứng dụng để giải bài toán tối ưu góc sợi và độ dày của tấm composite gia cường và kết quả cho thấy hiệu quả và độ chính xác tốt.
- Kỹ thuật lựa chọn tinh hoa được sử dụng để sửa đổi bước lựa chọn của thuật toán Jaya ban đầu nhằm cải thiện sự hội tụ của thuật toán. Thuật toán Jaya cải tiến sau đó được áp dụng để

giải bài toán tối ưu hóa dầm composite Timoshenko và thu được kết quả rất tốt.

- Ngoài ra, thuật toán (SLMD-iJaya), là sự kết hợp của thuật toán Jaya cải tiến và Phương pháp vòng lặp đơn xác định toàn cục (SLDM) đã được đề xuất như một bộ công cụ mới để giải quyết bài toán Tối ưu hóa Thiết kế dựa trên độ tin cậy các mô hình dầm composite liên tục. Kết quả thu được tốt hơn và đáng tin cậy hơn nhiều so với kết quả không có hệ số tin cậy.
- Mạng nơ ron nhân tạo được tích hợp với thuật toán Differential Evolution cải tiến để tạo thành một thuật toán mới gọi là thuật toán ABDE (ANN-based Differential Evolution). Thuật toán mới này sau đó được áp dụng để giải quyết hai bài toán tối ưu hóa của cấu trúc tấm gia cường. Bài toán thứ nhất là tối ưu hóa các góc sợi của tấm composite gia cường và bài toán thứ hai là tối ưu hóa độ dày của tấm composite gia cường. Kết quả thu được cho thấy hiệu suất cao của bộ công cụ ABDE được đề xuất.
- - Ngoài ra, thông qua một số kết quả số có thể nêu một số kết luận chính như sau:

+ Kết quả tối ưu của góc hướng sợi cho cả trường hợp tấm composite gia cường hình vuông và hình chữ nhật cho thấy rằng các lời giải của iDE rất phù hợp với các kết quả của GA. Sai số của năng lượng biến dạng trong cả hai trường hợp là rất thấp. Mức tối đa chỉ khoảng 0,2% đối với trường hợp tấm hình chữ nhật. Tuy nhiên, thời gian tính toán của thuật toán iDE nhỏ hơn nhiều. Điều này đã chứng minh tính chính xác và hiệu quả của phương pháp iDE. Kết quả cũng cho thấy các thông số hình học của kết cấu cũng có ảnh hưởng đến giá trị tối ưu của bài toán. Đặc biệt, hướng sợi tối ưu của tấm hình vuông và hình chữ nhật là khá khác nhau trong cùng điều kiện

+ Kết quả tối ưu hóa chiều dày cho thấy, trong trường hợp tấm vuông, hàm mục tiêu đạt giá trị thấp nhất với 4 dầm gia cường (XX-YY) vì chiều dày của tấm thu được nhỏ hơn nhiều so với trường hợp 2 dầm gia cường. Trong trường hợp các tấm hình chữ nhật, kết quả

tốt nhất thu được trong trường hợp hai dầm gia cường được bố trí theo phương Y (Y-Y), với giá trị của chiều dày tấm là nhỏ nhất. Nói cách khác, phương án tối ưu thường đạt được với chiều dày tấm nhỏ nhất kết hợp với nhiều dầm gia cường hơn đối với trường hợp tải trọng đồng đều.

+ Các vấn đề thiết kế tối ưu hóa của dầm composite Timoshenko được giải quyết với các loại điều kiện biên khác nhau (P-P, F-F, F-P và C-L) bằng cách sử dụng bốn thuật toán dựa trên dân số khác nhau bao gồm DE, IDE, Jaya và iJaya và một thuật toán dựa trên gradient từ công trình của Liu. Các kết quả số cho thấy rằng khối lượng tối ưu thu được từ iJaya phù hợp tốt với các giải pháp khác. Tuy nhiên, thuật toán iJaya tiêu tốn ít thời gian nhất để đạt được giải pháp tối ưu so với các cách tiếp cận khác. Trong số năm phương pháp, thuật toán SQP (được triển khai bằng lệnh `fmincon` trong Matlab) được sử dụng trong công trình của Liu đã đạt được giải pháp tối ưu rất nhanh nhưng nó có thể bị mắc kẹt trong tối ưu cục bộ và điều đó ảnh hưởng đến giá trị của giải pháp tối ưu. So với các phương pháp tối ưu hóa toàn cục khác (DE, IDE, Jaya), phương pháp iJaya chắc chắn vượt trội hơn. Thuật toán iJaya có thể được coi là thuật toán hiệu quả nhất vì nó đã giải được nghiệm toàn cục ở tốc độ hội tụ cao nhất. Trong trường hợp chiều dày của dầm nhiều lớp (h) được chia thành độ dày của các lớp của dầm để tối ưu hóa. Và một lần nữa, phương pháp iJaya lại chiếm ưu thế so với các phương pháp khác về cả số lượng phân tích hàm và thời gian CPU.

+ Các giải pháp cho vấn đề RBDO của dầm composite sử dụng SLDM-iJaya cũng đã được trình bày và thể hiện trong Bảng 5. 8. Các nghiệm đạt được của SLDM-iJaya và DLM-iJaya rất phù hợp với các nghiệm thu được bằng các phương pháp khác. Tuy nhiên, chi phí tính toán đã giảm đáng kể khi sử dụng phương pháp DLM-iJaya và phương pháp SLDM-iJaya thay vì sử dụng DLM-IDE và SLDM-IDE. Đặc biệt, số lượng đánh giá hàm ràng buộc giảm gần 16% đối với trường hợp sử dụng DLM-iJaya và giảm đến 20% đối với trường hợp sử dụng SLDM-iJaya. Mức giảm chi phí tính toán lần lượt là

52% và 36%. Những con số này đã chứng minh mạnh mẽ tính hiệu quả và chính xác của phương pháp SLDM-iJaya được đề xuất.

+ Phương pháp SLDM-iJaya đề xuất được áp dụng để giải quyết vấn đề thiết kế tối ưu hóa dựa trên độ tin cậy của các dầm composite nhiều lớp với nhiều loại điều kiện biên khác nhau. Kết quả tối ưu được tính toán với ba chỉ số độ tin cậy yêu cầu khác nhau, $\beta_{r,i}$. Kết quả so sánh giữa phương pháp DLM-iJaya và SLDM-iJaya được liệt kê trong Bảng 5. 9 cho thấy rằng các giải pháp tối ưu về khối lượng rất phù hợp, nhưng số lần phân tích hàm và thời gian CPU chạy của SLDM-iJaya vượt trội hơn so với DLM- iJaya trong mọi trường hợp.

+ Trong chương cuối cùng, các thuật toán tối ưu hóa ABDE được đề xuất đã được áp dụng cho ba mô hình bài toán: tấm hình chữ nhật gia cường theo hướng X (RX), tấm hình chữ nhật gia cường theo hướng Y (RY) và tấm hình vuông được gia cường theo hai hướng X và Y (S-XY). So với kết quả tối ưu thu được từ thuật toán DE, các giá trị của biên thiết kế được tính toán từ ABDE là rất giống nhau, với sai số tối đa chỉ là 1,99%. Chi phí cho mô hình R-X do ABDE tính toán chỉ là 8 giây so với 2851 giây của DE, nhanh hơn 356 lần. Với 2 trường hợp còn lại của mô hình R-Y và S-XY, thời gian tính lần lượt là 13 giây và 5 giây so với 2903 giây và 1497 giây của thuật toán DE. Tính ưu việt này là do hàm mục tiêu có thể được xấp xỉ từ mô hình ANN bằng một lệnh duy nhất: *net (dân số)*, không phụ thuộc vào dân số của mỗi thế hệ, thay vì sử dụng vòng lặp *for / end* như trong thuật toán DE ban đầu. Kết quả trong Bảng 5. 4 có thể chứng minh độ chính xác và hiệu quả vượt trội của thuật toán ABDE so với thuật toán DE độc lập. Thuật toán đơn giản này có thể dễ dàng mở rộng áp dụng để giải nhiều dạng bài toán tối ưu hóa cấu trúc cho cả mô hình tuyến tính và phi tuyến.

6.2 Đề xuất và hướng phát triển

- DE cải tiến có thể được mở rộng cho nhiều vấn đề tối ưu hóa khác của nhiều vật liệu và cấu trúc khác. Và cũng có thể được áp dụng để giải quyết các bài toán RBDO khác.

- Thuật toán Jaya cải tiến có thể được áp dụng cho các loại cấu trúc composite khác như tấm composite gia cường, tấm composite gấp, vỏ phẳng để kiểm tra và khai thác độ hiệu quả của phương pháp.
- Khảo sát các kỹ thuật có thể giúp cải thiện tốc độ huấn luyện và độ chính xác của Mạng nơron nhân tạo cũng là một hướng nghiên cứu tiềm năng từ luận án này.
- Thuật toán ABDE có thể được mở rộng cho nhiều bài toán tối ưu hóa khác để chứng minh tính hiệu quả và độ tin cậy của nó với nhiều loại vật liệu khác nhau..

DANH SÁCH BÀI BÁO XUẤT BẢN

Một phần của luận án này đã được đăng trên các tạp chí quốc tế, tạp chí trong nước hoặc được trình bày trong các hội nghị. Những bài báo này bao gồm:

Tạp chí Quốc tế

1. **T. Lam-Phat**, V. Ho-Huu, S. Nguyen-Ngoc, S. Nguyen-Hoai, Trung Nguyen-Thoi. Deterministic and reliability-based lightweight design of Timoshenko composite beams. **Engineering with Computers**, 2020, <https://doi.org/10.1007/s00366-020-00946-8>.
2. **T. Lam-Phat**, S. Nguyen-Hoai, V. Ho-Huu, Q. Nguyen, T. Nguyen-Thoi. An Artificial Neural Network-Based Optimization of Stiffened Composite Plate Using A New Adjusted Differential Evolution Algorithm. Proceedings of the International Conference on Advances in Computational Mechanics 2017 pp 229-242 (Part of the **Lecture Notes in Mechanical Engineering** book series (LNME))
Link: https://link.springer.com/chapter/10.1007/978-981-10-7149-2_16
3. Q. Nguyen, S. Nguyen-Hoai, T. Chuong-Thiet, **T. Lam-Phat**. Optimization of the Longitudinal Cooling Fin by Levenberg–Marquardt Method. Proceedings of the International Conference on Advances in Computational Mechanics 2017 pp 217-227 (Part of the **Lecture Notes in Mechanical Engineering** book series (LNME))

Link: https://link.springer.com/chapter/10.1007/978-981-10-7149-2_15

4. T. Nguyen-Thoi, T. Rabczuk, **T. Lam-Phat**, V. Ho-Huu, P. Phung-Van (2014). Free vibration analysis of cracked Mindlin plate using an extended cell-based smoothed discrete shear gap method (XCS-DSG3). **Theoretical and Applied Fracture Mechanics**. Vol.72, 150-163.

Link:

<https://www.sciencedirect.com/science/article/pii/S016784421400041X>

Tạp chí trong nước

5. **Lam Phat Thuan**, Nguyen Nhat Phi Long, Nguyen Hoai Son, Ho Huu Vinh, Le Anh Thang. Global Optimization of Lamination Composite Beams Using An Improved Differential Evolution Algorithm. **Journal of Science and Technology in Civil Engineering NUCE** 2020. 14 (1): 54–64
6. Nguyen-Thoi, T., Ho-Huu, V., Dang-Trung, H., Bui-Xuan, T., **Lam-Phat, T.** (2013) Optimization analysis of stiffened composite plate by sequential quadratic programming. **Journal of Science and Technology**, Vol. 51(4B), p. 156-165.
7. Nguyen Thoi Trung, Bui Xuan Thang, Ho Huu Vinh, **Lam Phat Thuan**, Ngo Thanh Phong. An Effective Algorithm For Reliability-Based Optimization Of Stiffened Mindlin Plate. **Vietnam Journal of Mechanics**, VAST, Vol. 35, No. 4 (2013), pp. 335 – 346

Hội nghị quốc tế

8. **Thuan Lam-Phat**, Son Nguyen-Hoai, Vinh Ho-Huu, Trung Nguyen-Thoi. Optimization of stiffened composite plate using adjusted different evolution algorithm. Proceeding of the international conference on computational methods (Vol.3, 2016), Berkeley, CA, USA.

Hội nghị trong nước

9. **Thuan Lam-Phat**, Son Nguyen-Hoai, Vinh Ho-Huu, Trung Nguyen-Thoi. Optimization analysis of stiffened composite plate by adjusted different evolution. Hội nghị Khoa học – Công nghệ toàn quốc về cơ khí 2015
10. **Lâm Phát Thuận**, Nguyễn Hoài Sơn, Lê Anh Thắng, Hồ Hữu Vịnh. Tối ưu hóa góc hướng sợi tấm Composite gia cường dùng thuật toán Differential Evolution kết hợp mạng thần kinh nhân tạo. Hội nghị cơ học toàn quốc lần thứ X, 8-9/12/2017)

**MINISTRY OF EDUCATION AND TRAINING
HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY AND EDUCATION**

LAM PHAT THUAN

**DEVELOPMENT OF METAHEURISTIC
OPTIMIZATION METHODS FOR MECHANICS
PROBLEMS**

**MAJOR: ENGINEERING MECHANICS
CODE: 62140101**

PHD THESIS SUMMARY

Ho Chi Minh City, 01/2021

THE WORK IS COMPLETED AT
HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY AND EDUCATION

Supervisor 1: Assoc. Prof. Dr. NGUYEN HOAI SON

Supervisor 2: Assoc. Prof. Dr. LE ANH THANG

PhD thesis is protected in front of
EXAMINATION COMMITTEE FOR PROTECTION OF
DOCTORAL THESIS
HCM CITY UNIVERSITY OF TECHNOLOGY AND
EDUCATION,

Date month year

NHỮNG ĐÓNG GÓP CHÍNH CỦA LUẬN ÁN

- Thuật toán DE cải tiến đã được áp dụng lần đầu tiên để giải bài toán thiết kế tối ưu của cấu trúc tấm composite gia cường và kết quả cho thấy hiệu quả và độ chính xác tốt.
- Một sự hiệu chỉnh trong bước lựa chọn của thuật toán Jaya ban đầu sử dụng kỹ thuật lựa chọn tinh hoa (Elitist Selection Technique) được đề xuất để tạo thành một phiên bản cải tiến của thuật toán. Thuật toán Jaya cải tiến sau đó được áp dụng để giải quyết bài toán tối ưu hóa cấu trúc dầm composite Timoshenko và thu được kết quả rất tốt.
- Thuật toán Jaya cải tiến lần đầu tiên được kết hợp với Phương pháp xác định vòng lặp đơn nghiệm toàn cục (SLDM) để tạo ra một bộ công cụ mới có tên (SLMD-iJaya) để giải quyết bài toán Tối ưu hóa thiết kế dựa trên độ tin cậy của các mô hình dầm composite liên tục. Các thiết kế tối ưu thu được tốt hơn và an toàn hơn nhiều so với các thiết kế không có xét đến yếu tố độ tin cậy.
- Mạng nơ-ron nhân tạo được sử dụng để xấp xỉ đáp ứng của tấm composite gia cường và ANN được tích hợp với thuật toán Differential Evolution cải tiến để tạo thành thuật toán mới gọi là thuật toán ABDE. Thuật toán mới này sau đó được áp dụng để tìm kiếm thiết kế tối ưu của các cấu trúc tấm composite gia cường. Bài toán đầu tiên là tối ưu hóa các góc hướng sợi của tấm composite gia cường và vấn đề thứ hai là giải tìm độ dày tối ưu của tấm composite gia cường. Kết quả thu được cho thấy tính hiệu quả cao của bộ công cụ ABDE được đề xuất.

ABSTRACT

Almost all design problems in engineering can be considered as optimization problems and thus require optimization techniques to solve. During the past decades, many optimization techniques have been proposed and applied to solve a wide range of various problems. Among them, metaheuristic algorithms have gained huge

popularity in recent years in solving design optimization problems of many types of structure with different materials. These metaheuristic algorithms include genetic algorithms, particle swarm optimization, bat algorithm, cuckoo search, differential evolution, firefly algorithm, harmony search, flower pollination algorithm, ant colony optimization, bee algorithms, Jaya algorithm and many others. Among the methods mentioned above, the Differential Evolution is one of the most widely used methods. Since it was first introduced by Storn and Price [1], many studies have been carried out to improve and apply DE in solving structural optimization problems. The DE has demonstrated excellent performance in solving many different engineering problems. Besides the Differential Evolution algorithm, the Jaya algorithm recently proposed by Rao [2] is also an effective and efficient method that has been widely applied to solve many optimization problems and showed its good performance. It gains dominant results when being tested with benchmark test functions in comparison with other population-based methods. However, like many other population-based optimizations, one of the disadvantages of DE and Jaya is that the optimal computational time is much slower than the gradient-based optimization methods. This is because DE and Jaya takes a lot of time in evaluating the fitness of individuals in the population. To overcome this disadvantage, artificial neuron networks (ANN) are proposed to combine with the metaheuristic algorithms, such as Differential Evolution, to form a new approach that help solve the design optimization effectively. Moreover, one of the most important issues in engineering design is that the optimal designs are often affected by uncertainties which can be occurred from various sources, such as manufacturing processes, material properties and operating environments. These uncertainties may cause structures to improper performance as in the original design, and hence may result in risks to structures [3]. Therefore, reliability-based design optimization (RBDO) can be considered as a comprehensive strategy for finding an optimal design.

In this dissertation, an improved version of Differential Evolution has been first time utilized to solve for optimal fiber angle and thickness of the stiffened composite. Secondly, the Artificial Neural Network is integrated to the optimization process of the improved Differential Evolution algorithm to form a new algorithm call ABDE (ANN-based Differential Evolution) algorithm. This new algorithm is then applied to solve optimization problems of the stiffened composite plate structures. Thirdly, an elitist selection technique is utilized to modify the selection step of the original Jaya algorithm to improve the convergence of the algorithm and formed a new version of the original Jaya called iJaya algorithm. The improved Jaya algorithm is then applied to solve for optimization problem of the Timoshenko composite beam and obtained very good results. Finally, the so-called called (SLMD-iJaya) algorithm which is the combination of the improved Jaya algorithm and the Global Single-Loop Deterministic Methods (SLDM) has been proposed as a new tool set for solving the Reliability-Based Design Optimization problems. This new method is applied to look for optimal design of Timoshenko composite beam structures.

TÓM TẮT

Hầu như các bài toán thiết kế trong kỹ thuật có thể được coi là những bài toán tối ưu và do đó đòi hỏi các kỹ thuật tối ưu hóa để giải quyết. Trong những thập kỷ qua, nhiều kỹ thuật tối ưu hóa đã được đề xuất và áp dụng để giải quyết một loạt các vấn đề khác nhau. Trong số đó, các thuật toán metaheuristic đã trở nên phổ biến trong những năm gần đây trong việc giải quyết các vấn đề tối ưu hóa thiết kế của nhiều loại cấu trúc với các vật liệu khác nhau. Các thuật toán metaheuristic này bao gồm Genetic Algorithms, Particle Swarm Optimization, Bat Algorithm, Cuckoo Search, Differential Evolution, Firefly Algorithm, Harmony Search, Flower Pollination Algorithm, Ant Colony Optimization, Bee Algorithms, Jaya Algorithm và nhiều thuật toán khác. Trong số các phương pháp được đề cập ở trên, Differential Evolution là một trong những phương pháp được sử

dụng rộng rãi nhất. Kể từ khi được Storn và Price [1] giới thiệu lần đầu tiên, nhiều nghiên cứu đã được thực hiện để cải thiện và áp dụng DE trong việc giải quyết các vấn đề tối ưu hóa cấu trúc. DE đã chứng minh hiệu suất tuyệt vời trong việc giải quyết nhiều vấn đề kỹ thuật khác nhau. Bên cạnh thuật toán Differential Evolution, thuật toán Jaya được Rao [2] đề xuất gần đây cũng là một phương pháp hiệu quả và đã được áp dụng rộng rãi để giải quyết nhiều vấn đề tối ưu hóa và cho thấy hiệu suất tốt. Nó đạt được kết quả vượt trội khi được thử nghiệm với các hàm test benchmark so với các phương pháp dựa trên dân số khác. Tuy nhiên, giống như nhiều thuật toán tối ưu hóa dựa trên dân số khác, một trong những nhược điểm của DE và Jaya là thời gian tính toán tối ưu chậm hơn nhiều so với các phương pháp tối ưu hóa dựa trên độ dốc (gradient-based algorithms). Điều này là do DE và Jaya mất rất nhiều thời gian để đánh giá hàm mục tiêu của các cá thể trong bộ dân số. Để khắc phục nhược điểm này, các mạng nơ ron nhân tạo (Artificial Neural Networks) được đề xuất kết hợp với các thuật toán metaheuristic, như Differential Evolution, để tạo thành một phương pháp cách tiếp cận mới giúp giải quyết các bài toán tối ưu hóa thiết kế một cách hiệu quả. Bên cạnh đó, một trong những vấn đề quan trọng nhất trong thiết kế kỹ thuật là các thiết kế tối ưu thường bị ảnh hưởng bởi sự không chắc chắn có thể xảy ra từ nhiều nguồn khác nhau, chẳng hạn như quy trình sản xuất, tính chất vật liệu và môi trường vận hành. Những yếu tố không chắc chắn này có thể khiến các cấu trúc hoạt động không đúng như trong thiết kế ban đầu, và do đó có thể dẫn đến rủi ro cho các cấu trúc [3]. Do đó, tối ưu hóa thiết kế dựa trên độ tin cậy (Reliability-Based Design Optimization) có thể được coi là một chiến lược toàn diện để tìm kiếm một thiết kế tối ưu.

Trong luận án này, lần đầu tiên một phiên bản cải tiến của phương pháp Differential Evolution đã được sử dụng để tìm góc hướng sợi tối ưu và độ dày của tấm gia cường vật liệu composite. Thứ hai, Mạng nơ ron nhân tạo (ANN) được tích hợp vào quy trình tối ưu hóa thuật toán Differential Evolution cải tiến để hình thành thuật toán mới gọi là thuật toán ABDE (Artificial Neural Network-Based

Differential Evolution). Thuật toán mới này sau đó được áp dụng để giải quyết các bài toán tối ưu hóa của các cấu trúc tấm composite gia cường. Thứ ba, một kỹ thuật lựa chọn tinh hoa (Elitist Selection Technique) được sử dụng để hiệu chỉnh bước lựa chọn của thuật toán Jaya ban đầu để cải thiện sự hội tụ của thuật toán và hình thành một phiên bản mới của thuật toán Jaya được gọi là thuật toán iJaya. Thuật toán Jaya cải tiến (iJaya) sau đó được áp dụng để giải quyết bài toán tối ưu hóa dầm Timoshenko vật liệu composite và thu được kết quả rất tốt. Cuối cùng, thuật toán mới SLMD-iJaya được tạo thành từ sự kết hợp giữa thuật toán Jaya cải tiến và phương pháp vòng lặp đơn xác định (Single-Loop Deterministic Method) đã được đề xuất như một công cụ mới để giải quyết các vấn đề Tối ưu hóa thiết kế dựa trên độ tin cậy. Phương pháp mới này được áp dụng để tìm kiếm thiết kế tối ưu của các cấu trúc dầm composite Timoshenk và cho kết quả vượt trội.

CHAPTER 1 LITERATURE REVIEW

1.1 Overview of Artificial Neural Network, Metaheuristic algorithms and composite material structural optimization

Almost all design problems in engineering can be considered as optimization problems and thus require optimization techniques to solve. However, as most real-world problems are highly non-linear, traditional optimization methods usually do not work well. The current trend is to use evolutionary algorithms and metaheuristic optimization methods to tackle such nonlinear optimization problems. Metaheuristic algorithms have gained huge popularity in recent years. The popularity of metaheuristic algorithms can be attributed to their good characteristics because these algorithms are simple, flexible, efficient, adaptable and yet easy to implement. Such advantages make them versatile to deal with a wide range of optimization problems, especially the structural optimization problems [4]. Structural optimization is a potential field and has

attracted the attention of many researchers around the world. During the past decades, many optimization techniques have been proposed and applied to solve a wide range of various problems. Among them, the Differential Evolution is one of the most widely used methods. Since it was first introduced by Storn and Price [1], many studies have been carried out to improve and apply DE in solving structural optimization problems. The DE has demonstrated excellently performance in solving many different engineering problems. Besides the Differential Evolution algorithm, the Jaya algorithm recently proposed by Rao [2] is also an effective and efficient methods that has been widely applied to solve many optimization problems and showed its good performance. It gains dominate results when being tested with benchmark test functions in comparison with other population-based methods. Moreover, it has been also successfully applied in solving many optimal design problem in engineering as presented in following literature [5]–[7]. However, like many other population-based optimizations, one of the disadvantages of DE and Jaya is that the optimal computational time is much slower than the gradient-based optimization methods. This is because DE and Jaya takes a lot of time in evaluating the fitness of individuals in the population. Specifically, in the structural optimization problem, the calculation of the objective function or constraint function values is usually done by using the finite element to analyse the structural response. To overcome this disadvantage, artificial neuron networks (ANN) are proposed to combine with the DE algorithm. Based on the idea of imitation of the brain structure, ANN is capable of approximating an output corresponding to a set of input data quickly after the network has been trained, also known as a learning process. Thanks to this remarkable advantage, the computation of objective function or constraint function values in the DE algorithm will be done quickly. As a result, ANN will help significantly improve the efficiency of DE calculations. At present, the structures made from composite material are widely used in almost all fields such as construction, mechanical

engineering, marine, aviation, etc. In particular, stiffened composite plates made of composite material are an outstanding form and are used increasingly by its superior advantages. By combining the advantages of composite materials and the reinforced beams structure, the reinforced composite plates have very high bending strength with very light weight. Due to its high practical applicability, the need to optimize the design of the structure to save costs and increase the efficiency of use is also high. However, because of the complexity of computing the behaviour of this particular type of structure, finding a good algorithm for optimizing design parameters is essential to ensure computational efficiency.

Moreover, one of the most important issues in engineering design is that the optimal designs are often effected by uncertainties which can be occurred from various sources, such as manufacturing processes, material properties and operating environments. These uncertainties may cause structures to improper performance as in the original design, and hence may result in risks to structures [3]. Therefore, reliability-based design optimization (RBDO) can be considered as a comprehensive strategy for finding an optimal design. In this thesis, the single-loop methods will be combined with a metaheuristic optimization algorithm to form a new tool set for solving a RBDO problems of composite structure.

In summary, in this thesis, some modifications will be investigated and propose to improve the original algorithm of Differential Evolution and Jaya algorithm to increase the increase the convergence of DE and Jaya algorithm. The modified algorithm is then combine with ANN to develop a new tool for solving design optimization problem and the RBDO problem of composite structure, such as stiffened composite plate with the improved DE algorithm and composite beam with the improve Jaya algorithm.

1.2 Motivation of the research

The thesis is motivated by:

- The development / improvement of existing algorithms to improve the efficiency of solving structural optimization problems with high accuracy and reliability.
- Studying the advantages of Artificial Neural Network (ANN) to combine with optimal algorithms to improve the speed and the performance of solving structural optimization problems.

1.3 Goal of the dissertation

Firstly, the thesis focuses on the development of metaheuristic optimization methods and combines with the advantage of the Artificial Neural Network in approximating data to build up a new algorithm for solving composite material structural optimization problems. Particularly, the original Differential Evolution will be modified to improve the convergence in solving for global optimal solution and then, the ANN will be integrated to the improve Differential Evolution (iDE) to form a new algorithm, which is used to look for optimal design of stiffened composite plate structures.

Secondly, the thesis also proposes a new tool set, which is the combination of metaheuristic optimization algorithm and the single-loop methods to deal with Reliability-Based Design Optimization (RBDO) problems. In particular, the original Jaya algorithm will be modified to improve the convergence in looking for optimal solution of the optimization problem. Then, this improved version of Jaya algorithm will be combined with single-loop methods to solve the Reliability-Based Design Optimization of composite beam structures.

1.4 Research scope of the dissertation

The thesis focuses on the following main issues:

- Optimize truss, beam and stiffened plate structures using steel and composite materials.
- Study and improve population-based optimization methods to increase accuracy and efficiency in solving optimization problems.
- Exploit the ability to create approximate models from data sets of Neural Network to combine with optimal algorithms to improve the

performance and the ability to solve many different types of problems.

- Combine optimal algorithms with groups of reliability assessment methods to solve RBDO problems.

- The problems selected for optimization are relatively simple with the main purpose of evaluating the effectiveness, accuracy and reliability of the proposed optimization methods. The application of optimal methods proposed in the thesis for more complex problems will be further studied in the future.

1.5 Outline

The dissertation contains seven chapters and is structured as follows: **Chapter 1** presents an overview on metaheuristic algorithms, composite material structure and especially artificial neural networks and its role and application in optimization process. This chapter also give out the organization of the thesis via the outline section and the novelty and goal of the thesis for quick review of what is studied in this thesis. **Chapter 2** provides an overview of composite material and theory of Timoshenko composite beam and stiffened composite plate which are the main structure under investigated and studied in optimization problems of this thesis. **Chapter 3** devotes the presentation of metaheuristic optimization related to Differential Evolution and Jaya algorithm. This chapter also gives out an overview and formulation for Reliability-Based Design Optimization (RBDO) and the proposed methods for solving RBDO problem. **Chapter 4** offers the introduction and the historical development of Artificial Neural Network (ANN), some basic concepts related to ANN and introduction of the Neural Network Structure. Moreover, the Levenberg-Marquardt training algorithm and the overfitting phenomenon are also presented in this chapter. **Chapter 5** illustrate the effectiveness and efficiency of the improve Differential Evolution and the improve Jaya in solving optimization problems. The structures investigated in this section includes planar truss structure, space truss structure, Timoshenko composite beam and stiffened composite plate. This chapter also presents a new

approach called SLDM-iJaya which is formed by the combination of the improve Jaya algorithm and the single-loop methods for solving the RBDO problem of the Timoshenko composite beam and the application of Artificial Neural Network when it is integrated to a metaheuristic optimization method, such as Differential Evolution algorithm. The integration form a new tool set call ABDE (ANN-Based Differential Evolution) algorithm and applied to solve for optimal design of the stiffened composite plate. Finally, **Chapter 6** closes the concluding remarks and give out some recommendations for future work.

1.6 Concluding remarks

In this chapter, an overview of metaheuristic optimization methods, artificial neural network, composite material structure in optimization is given out. This chapter also presents the novelty points of this dissertation, and the organization of the dissertation with eight chapters. In the next chapters, fundamental theories, some approaches for modification to improve the solution of some metaheuristic algorithm and application with numerical results will be presented.

CHAPTER 2

Fundamental theory of Composite Structure in Design and Optimization

2.1 Exact analytical displacement and stress of Timoshenko composite beam

Composite laminated Timoshenko beams can be treated as continuous models and discrete models. The discrete models are easier to be implemented but difficult to obtain the exact solution. In addition, the discrete models such as finite element approaches are not so effective as the analytical approaches of continuous models. The process to build up the analytical solution for the composite laminated beam is simply presented as in the following section. For

more details of the method, readers are encouraged to refer to Liu's work [8].

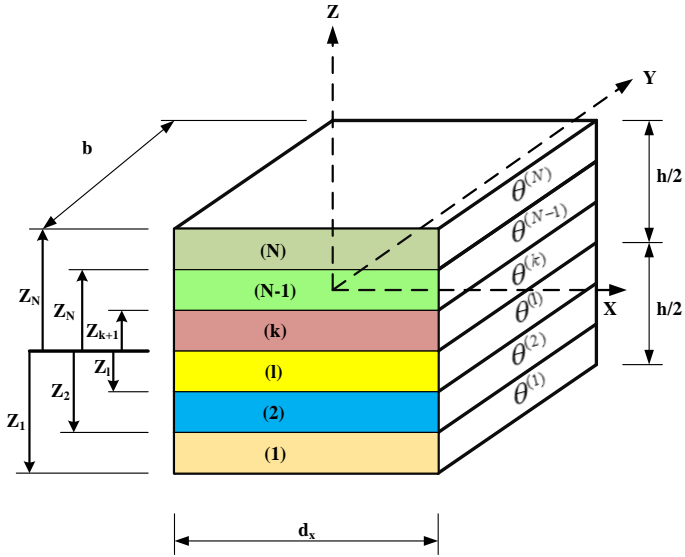


Figure 2. 1. Composite laminated beam model

Consider a segment of composite laminated beam with N layers and the fiber orientations of layers are of $\theta_i (i = 1, \dots, N)$. The positions of layers are denoted by $z_i (i = 1, \dots, N)$. The beam has rectangular cross section with the width b and the length h as depicted in **Figure 2. 1**. The beam segment dx is subjected to the transversal force as shown in **Figure 2. 2**.

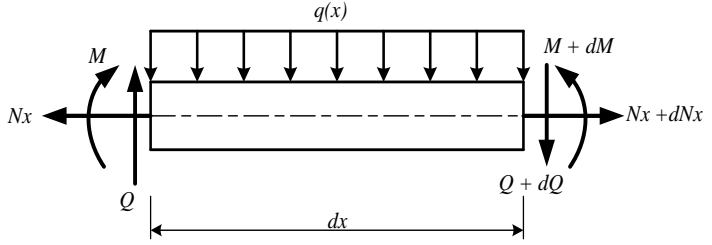


Figure 2. 2. Free-body diagram

The displacement fields of the composite laminated beam calculated analytically based on the first-order shear deformation theory (also called Timoshenko beam theory) are:

$$u_o(x) = -\bar{B} \left(\frac{q_0}{6} x^3 + \frac{1}{2} C_1 x^2 + 4C_4 x + C_5 \right) \quad (2.1)$$

$$w_o(x) = -\bar{A} \frac{q_0}{24} x^4 - \frac{1}{6} \bar{A} C_1 x^3 - \left(\bar{C} \frac{q_0}{2} + \frac{1}{2} \bar{A} C_2 \right) x^2 + C_6 x + C_7 \quad (2.2)$$

$$\phi(x) = \bar{A} \left(\frac{q_0}{6} x^3 + \frac{1}{2} C_1 x^2 + 4C_2 x + C_3 \right) \quad (2.3)$$

where $C_i (i = 1, \dots, 7)$ are indefinite integration constants determined by using the boundary conditions of the composite laminated beams as shown in the following section.

$$\bar{A} = \frac{A_{11}}{b(B_{11}^2 - A_{11}D_{11})}, \bar{B} = \frac{B_{11}}{b(B_{11}^2 - A_{11}D_{11})}, \bar{C} = \frac{1}{bKA_{55}} \quad (2.4)$$

where $A_{11}, B_{11}, D_{11}, A_{55}$ are respectively extensional stiffness, bending-extensional coupling stiffness, bending stiffness and extensional stiffness of the composite laminate. K is the shear correction factor with the value of $5/6$.

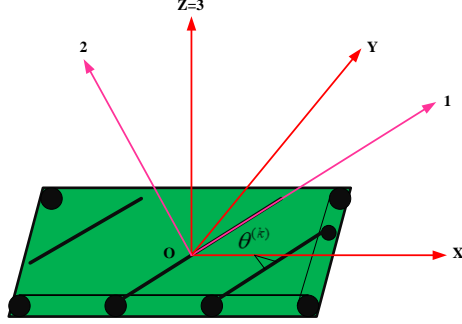


Figure 2. 3. The material and laminate coordinate system

The stress fields of the composite laminated beam include the plane stress components and the shear stress components. According to the coordinate system between the materials (123) and the beam/laminate (xyz) as depicted in **Figure 2. 3**, in which the fiber orientation coincides with the 1-axis, the plane stress components are expressed as follows

$$\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{pmatrix} = T^{(k)} \bar{Q}^{(k)} \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix}, z_{k+1} \leq z \leq z_k \quad (2.5)$$

where the strain components $\varepsilon_y = 0, \gamma_{xy} = 0$, and

$$\varepsilon_x = -\bar{B} \left(\frac{q_0}{3} x^2 + C_1 x + C_4 \right) + z \bar{A} \left(\frac{q_0}{2} x^2 + C_1 x + C_2 \right) \quad (2.6)$$

$T^{(k)}$ is the coordinate transformation matrix and $\bar{Q}^{(k)}$ is the matrix of material stiffness coefficients

$$T^{(k)} = \begin{bmatrix} \cos^2 \theta^{(k)} & \sin^2 \theta^{(k)} & 2 \sin \theta^{(k)} \cos \theta^{(k)} \\ \sin^2 \theta^{(k)} & \cos^2 \theta^{(k)} & -2 \sin \theta^{(k)} \cos \theta^{(k)} \\ -\sin \theta^{(k)} \cos \theta^{(k)} & \sin \theta^{(k)} \cos \theta^{(k)} & \cos^2 \theta^{(k)} - \sin^2 \theta^{(k)} \end{bmatrix} \quad (2.7)$$

$$\bar{Q}^{(k)} = \begin{bmatrix} \bar{Q}_{11}^{(k)} & \bar{Q}_{12}^{(k)} & \bar{Q}_{61}^{(k)} \\ \bar{Q}_{21}^{(k)} & \bar{Q}_{22}^{(k)} & \bar{Q}_{26}^{(k)} \\ \bar{Q}_{16}^{(k)} & \bar{Q}_{26}^{(k)} & \bar{Q}_{66}^{(k)} \end{bmatrix} \quad (2.8)$$

The shear stress components in the material coordinate systems are

$$\begin{pmatrix} \tau_{23} \\ \tau_{13} \end{pmatrix} = T_s^{(k)} \bar{Q}_s^{(k)} \begin{pmatrix} \gamma_{yz} \\ \gamma_{xz} \end{pmatrix}, Z_{k+1} \leq Z \leq Z_k \quad (2.9)$$

where the shear strain components $\gamma_{yz} = 0$ and

$$\gamma_{xz} = \bar{A} \left(\frac{1}{6} x^3 + \frac{1}{2} C_1 x^2 + C_2 x + C_3 \right) - \bar{A} \frac{q_0}{6} x^3 - \frac{1}{2} \bar{A} C_1 x^2 - (\bar{C} q_0 + \bar{A} C_2) x + C_6 \quad (2.10)$$

The coordinate transformation matrix $T_s^{(k)}$ and the matrix of stiffness coefficients $\bar{Q}_s^{(k)}$ can be described as

$$T_s^{(k)} = \begin{bmatrix} \sin \theta^{(k)} & \cos \theta^{(k)} \\ \cos \theta^{(k)} & -\sin \theta^{(k)} \end{bmatrix} \quad (2.11)$$

$$\bar{Q}_s^{(k)} = \begin{bmatrix} \bar{Q}_{44}^{(k)} & \bar{Q}_{45}^{(k)} \\ \bar{Q}_{45}^{(k)} & \bar{Q}_{55}^{(k)} \end{bmatrix} \quad (2.12)$$

In the above equations, $\bar{Q}_{ij}^{(k)}$ is the stiffness coefficients of the k th lamina in the laminate coordinate system and are described clearly in [8].

2.2 Analysis of stiffened composite plate

Stiffened composite plate is formed by a composite plate combining with a stiffening Timoshenko composite beam, as illustrated in **Figure 2.4**. The beam is considered as a stiffener and is set parallel with the axes in the surface of the plate. The centroid of the beam has a distance e from the middle plane of the plate. The plate-beam system is discretized by a set of node. The degree of freedom (DOF)

of each node of the plate is $\mathbf{d} = [u, v, w, \beta_x, \beta_y]^T$, in which u, v, w are the displacements at the middle of the plate and β_x, β_y are the rotations around the y -axis and x -axis. Each node of the beam has the DOF of $\mathbf{d}_{st} = [u_r, u_s, u_z, \beta_r, \beta_s]^T$, where u_r, u_s, u_z are respectively centroid displacements of beam and β_r, β_s are the rotations of beam around r -axis and s -axis.

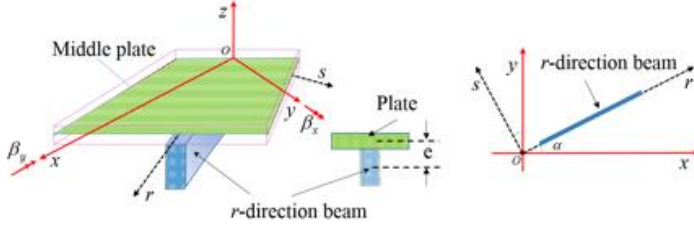


Figure 2. 4. A composite plate stiffened by an r-direction beam

The displacement compatibility between plate & beam is ensured by:

$$u = u_r(r) + z\beta_r(r) \quad ; \quad v = z\beta_s(r) \quad ; \quad w = u_z(r) \quad (2.13)$$

The strain energy of composite plate is given by:

$$U_P = \frac{1}{2} \iint_A (\boldsymbol{\epsilon}_0^T \mathbf{D}^m \boldsymbol{\epsilon}_0 + \boldsymbol{\epsilon}_0^T \mathbf{D}^{mb} \boldsymbol{\kappa}_b + \boldsymbol{\kappa}_b^T \mathbf{D}^{mb} \boldsymbol{\epsilon}_0 + \boldsymbol{\kappa}_b^T \mathbf{D}^b \boldsymbol{\kappa}_b + \boldsymbol{\gamma}^T \mathbf{D}^s \boldsymbol{\gamma}) dA \quad (2.14)$$

where $\boldsymbol{\epsilon}_0, \boldsymbol{\kappa}_b, \boldsymbol{\gamma}$ are respectively membrane, bending and shear strains of composite plate and are expressed as follows

$$\boldsymbol{\epsilon}_0 = [u_{,x}, v_{,y}, u_{,y} + v_{,x}]^T; \boldsymbol{\kappa}_b = [\beta_{x,x}, \beta_{y,y}, \beta_{x,y} + \beta_{y,x}]^T; \boldsymbol{\gamma} = [w_{,x} + \beta_x, w_{,y} + \beta_y]^T \quad (2.15)$$

$\mathbf{D}^m, \mathbf{D}^{mb}, \mathbf{D}^b, \mathbf{D}^s$ are material matrices of plate

The strain energy of composite stiffener is given by

$$U_{st} = \frac{1}{2} \int_l \left((\mathbf{e}_{st}^b)^T \mathbf{D}_{st}^b \mathbf{e}_{st}^b + (\mathbf{e}_{st}^s)^T \mathbf{D}_{st}^s \mathbf{e}_{st}^s \right) dx \quad (2.16)$$

where $\mathbf{e}_{st}^b, \mathbf{e}_{st}^s$ are respectively bending, shear strain of beam and are expressed as follows

$$\mathbf{e}_{st}^b = [u_{r,r} + z_0 \beta_{r,r}, \beta_{r,r}, \beta_{s,r}]^T; \mathbf{e}_{st}^s = [u_{z,r} + \beta_r]^T \quad (2.17)$$

$\mathbf{D}_{st}^b, \mathbf{D}_{st}^s$ are material matrices of composite beam

Using the superposition principle, total energy strain of stiffened composite plate is obtained:

$$U = U_p + \sum_{i=1}^{N_{st}} U_{st} \quad (2.18)$$

where N_{st} is the number of stiffeners.

For static analysis, the global equations for the stiffened composite plate $[\mathbf{K}]\{\Delta\} = \{\mathbf{F}\}$ can found in [9] for detail.

CHAPTER 3

Reliability-based optimization Methods with iJaya and improved Differential Evolution

3.1 Solving Optimization problems using improved Differential Evolution

An optimization problem can be expressed as follows:

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad \begin{cases} h_i(\mathbf{x}) = 0 & i = 1, \dots, l \\ g_j(\mathbf{x}) \leq 0 & j = 1, \dots, m \end{cases} \quad (3.1)$$

where \mathbf{x} is the vector of design variables; $h_i(\mathbf{x}) = 0$ and $g_j(\mathbf{x}) \leq 0$ are equality and inequality constraints; l, m are the number of

inequality and equality constraints, respectively; $f(\mathbf{x})$ is the objective function which can be the function of weight, cost, etc. Design optimization of a structure is to find optimal values of design variables in design space such that the objective function is minimum. Dealing with such problems, many optimization methods are used including gradient-based and population-based approach to find the solution. In this paper, the Differential Evolution is utilized to solve the problem of finding optimal fiber orientations and thickness of the stiffened composite plate.

3.1.1 Brief on the Differential Evolution algorithm [10], [11]

The original differential evolution algorithm firstly proposed by Storn and Price [1] has been widely used to solve many kinds of optimization problems. The scheme of this algorithm consists of four main phases as follows:

Phase 1: Initialization

Create an initial population by randomly sampling from the search space

Phase 2: Mutation

Generate a new mutant vector v_i from each current individual x_i based on mutation operations.

Phase 3: Crossover

Create a trial vector u_i by replacing some elements of the mutant vector v_i via crossover operation.

Phase 4: Selection

Compare the trial vector u_i with the target vector x_i . One with lower objective function value will survive in the next generation

To improve the effectiveness of the algorithm, the *Mutation phase* and the *Selection phase* are modified to increase the convergence rate as follow:

In the *mutation phase*, parent vectors are chosen randomly from the current population. This may make the DE be slow at exploitation of the solution. Therefore, the individuals participating in mutation should be chosen following a priority based on their fitness. By doing this, good information of parents in offspring will be stored

for later use, and hence will help to increase the convergence speed. To store good information in offspring populations, the individuals is chosen based on Roulette wheel selection via acceptant stochastic proposed by Lipowski and Lipowska [12] instead of the random selection.

In the *selection phase*, the elitist operator introduced by Padhye et al. [13] is used for the selection progress instead of basic selection as in the conventional DE. In the elitist process, the children population C consisting of trial vectors is combined with parent population P of target vectors to create a combined population Q. Then, best individuals are chosen from the combined population Q to construct the population for the next generation. By doing so, the best individuals of the whole population are always saved for the next generation.

3.1.2 The modified algorithm Roulette-wheel-Elitist Differential Evolution

1:	Generate the initial population
2:	Evaluate the fitness for each individual in the population
3:	while <the stop criterion is not met> do
4:	Calculate the selection probability for each individual
5:	for i =1 to NP do {NP: Size of population}
6:	Do mutation phase based on Roulette wheel
7:	selection
8:	$j_{rand} = \text{randi}(1,D)$ {D: number of design variables}
9:	for j =1 to D do
10:	if $\text{rand}[0,1] < CR$ or $j == j_{rand}$ then {CR: crossover
11:	control parameter}
12:	$u_{i,j} = x_{r1,j} + Fx(x_{r2,j} - x_{r3,j})$ {F:randomly
13:	chosen within [0,1] interval}
14:	else
15:	$u_{i,j} = x_{i,j}$
16:	end if
17:	end for
18:	Evaluate the trial vector u_i

	end for Doing selection phase based on Elitist selection operator end while
--	---

3.2 Improved Jaya algorithm

3.2.1 Jaya Algorithm

Jaya algorithm is a population-based global optimization technique, developed by Ventaka Rao recently. This is a simple and easy-to-implement algorithm which does not require any algorithm-specific parameters [2]. The idea of this algorithm is that it always tries to avoid the failure (moving away from the worst solution) and get closer to the success (best solution) during the searching process. The method is easy to implement because it just requires common controlling parameters (population size and number of generations) to accomplish to optimization task. The optimization process using this algorithm can be summarized in four following simple steps as in [14].

Firstly, an initial population set comprising N_p individuals is generated randomly in the search space. Each candidate is a vector of n design variables $x_i = (x_1, x_2, \dots, x_n)$ and is created obeying the lower and upper bounds as follows

$$x_{j,i} = x_{j,i}^l + rand[0,1] \times (x_{j,i}^u - x_{j,i}^l), j = (1, 2, \dots, n), i = (1, 2, \dots, N_p) \quad (3.2)$$

where x_j^u and x_j^l are respectively the upper and lower bounds of the design variable x_j ; $rand[0,1]$ generates a random number within the interval $[0,1]$.

Let $f(x)$ be the objective function of the optimization problem. The fitness of each individual in the population set is then evaluated through the values of $f(x_i)$. The individuals receiving the best and worst fitness value in the entire population are called the best (x_{best}) and the worst (x_{worst}) candidate, respectively. If $x_{j,i,k}$ is the value

of the j^{th} variable for i^{th} candidate during the k^{th} iteration, then this value is modified stochastically to generate a new vector $x'_{j,i,k}$

$$x'_{j,i,k} = x_{j,i,k} + r_{1,j,k} \times \left(x_{j,best,k} - |x_{j,i,k}| \right) - r_{2,j,k} \times \left(x_{j,worst,k} - |x_{j,i,k}| \right) \quad (3.3)$$

where $x_{j,worst,k}$ and $x_{j,best,k}$ are the values of j^{th} variable corresponding to the worst (x_{worst}) and the best (x_{best}) candidate in the whole population at k^{th} iteration. $r_{1,j,k}$ and $r_{2,j,k}$ are the random numbers in the range of $[0,1]$; the term $r_{1,j,k} \times \left(x_{j,best,k} - |x_{j,i,k}| \right)$ indicates the tendency of the design variable moving toward the best solution and the term $r_{2,j,k} \times \left(x_{j,worst,k} - |x_{j,i,k}| \right)$ shows the tendency of the solution avoiding the worst one.

Next, if the value of $x'_{j,i,k}$ is out of the range between lower bound and upper bound, an operation is carried to reflect it back to the allowable region

$$x'_{j,i,k} = \begin{cases} 2x_j^l - x'_{j,i,k} & \text{if } x'_{j,i,k} < x_j^l \\ 2x_j^u - x'_{j,i,k} & \text{if } x'_{j,i,k} > x_j^u \\ x'_{j,i,k} & \text{otherwise} \end{cases} \quad (3.4)$$

Finally, $x'_{j,i,k}$ is accepted if it gives better objective function value, otherwise, the value of $x_{j,i,k}$ is chosen. All accepted candidates at the end of iteration are maintained and become the input to the next iteration.

$$x_{i,k+1} = \begin{cases} x'_{i,k} & \text{if } f(x'_{i,k}) \leq f(x_{i,k}) \\ x_{i,k} & \text{otherwise} \end{cases} \quad (3.5)$$

3.2.2 Improvement version of Jaya algorithm

In the original Jaya algorithm, the population set for the next generation (iteration) is selected based on the paired comparison of fitness function values computed from $x'_{i,k}$ and $x_{i,k}$. This may result in good individuals being eliminated as a result of being compared to stronger individuals in pair. Although an individual may be not good compared to its competitor in the pair, it can be still better than the winners in other pairs in the entire population. Therefore, to ensure the selected individuals for the next generation are the best one, we use the elitist selection technique introduced by Padhye et al. [13] or the selection step instead of the original one in the Jaya algorithm. The process of selection is carried out as follows: firstly, at k th iteration, a set of $2N_P$ individuals, called set \mathbf{A} , is created. This \mathbf{A} set is formed by combining all the candidates $x_{i,k}$ in set \mathbf{X} and the modified candidates $x'_{i,k}$ belonging to set \mathbf{X}' . Then, N_P best individuals out of $2N_P$ candidates from set \mathbf{A} are selected to construct the population for the $(k+1)$ th generation. The elitist selection technique is depicted as follows

1. **Input:** initial population set \mathbf{X} and the modified set \mathbf{X}'
2. $\mathbf{A} = \mathbf{X} \cup \mathbf{X}'$ is generated
3. Select $N_{P_{best}}$ out of $2N_P$ individuals of \mathbf{A} and assign to \mathbf{X}_{new}
4. **Output:** \mathbf{X}_{new}

By this way, the best individuals of the entire population are always stored for the next generation. This technique helps the algorithm reach the optimal solution at better convergence rate.

3.3 Reliability-based design optimization using a global single loop deterministic method.

3.3.1. Reliability-based optimization problem formulation

The mathematical model of a typical RBDO problem can be described as follows [15]:

$$\begin{aligned}
& \text{find} && \mathbf{d}, \boldsymbol{\mu}_x \\
& \text{min} && f(\mathbf{d}, \boldsymbol{\mu}_x, \boldsymbol{\mu}_p) \\
& \text{subject to} && \text{Prob.}[g_i(\mathbf{d}, \mathbf{x}, \mathbf{p}) \leq 0] \geq R_i, \quad i = 1, 2, 3, \dots, m \\
& && \mathbf{d}^{low} \leq \mathbf{d} \leq \mathbf{d}^{up}, \boldsymbol{\mu}_x^{low} \leq \boldsymbol{\mu}_x \leq \boldsymbol{\mu}_x^{up}
\end{aligned} \tag{3.6}$$

where $f(\mathbf{d}, \boldsymbol{\mu}_x, \boldsymbol{\mu}_p)$ is the objective function; \mathbf{d} is the vector of deterministic design variables; \mathbf{d}^{low} and \mathbf{d}^{up} are the vectors of the lower and upper bounds of \mathbf{d} ; \mathbf{x} is the vector of random design variables; $\boldsymbol{\mu}_x$ is the mean vector of \mathbf{x} ; $\boldsymbol{\mu}_x^{low}$ and $\boldsymbol{\mu}_x^{up}$ are the vectors of the lower and upper bounds of $\boldsymbol{\mu}_x$; \mathbf{p} is the vector of random parameters; $g_i(\mathbf{d}, \mathbf{x}, \mathbf{p})$ are the constraint function; R_i are the desired reliabilities of constraint satisfaction; m is the number of constraint satisfactions; $\text{Prob}[\cdot]$ is a probability operator, and $\text{Prob.}[g_i(\mathbf{d}, \mathbf{x}, \mathbf{p}) \leq 0] \geq R_i$ means that the probability of constraint satisfaction $g_i(\mathbf{d}, \mathbf{x}, \mathbf{p}) \leq 0$ should be greater than or equal to the desired reliability R_i .

3.3.2. A global single-loop deterministic approach

A single-loop deterministic method (SLDM) has recently been introduced for solving reliability-based design optimization (RBDO) problems [16]. In SLDM, probabilistic constraints are converted to approximate deterministic constraints and so, the RBDO problems become the approximate deterministic optimization problems, and hence the computational cost of solving such problems is reduced significantly. However, the SLDM is still limited to continuous design variables, and the optimal solutions derived from the SLDM are often trapped into local extrema. To overcome these two disadvantages, a global single-loop deterministic approach is developed by Vinh et al.[3], and then it is applied to solve the RBDO problems of truss structures with both continuous and discrete design variables. The optimization algorithm used in Vinh's work is the IDE algorithm, an improved version of the differential evolution algorithm. In this thesis, another metaheuristic optimization method,

Jaya algorithm with an *improved* version named iJaya, is used to search for the global optimal solution. The approach is applied for design optimization of composite beam structure.

*** Formulation of approximate deterministic constraints**

The first step in SLDM is to form an approximate deterministic feasible region by moving the boundary of the probabilistic constraint a distance β from its original place, as shown in **Figure 3. 1.**

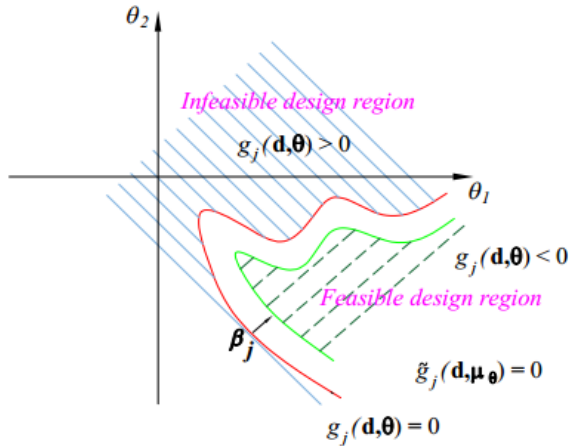


Figure 3. 1. Illustration of the feasible design region.

In **Figure 3. 1.** Illustration of the feasible design region., the limit-state function is represented by the red curve, the solid curves denotes the boundary of the transformed deterministic constraint function and the dashed area is the deterministic feasible design region. The transformation ensures that the smallest distance from any point on the red curve to the green curve is β_j , and it has been proven that the obtained solutions can satisfy the probabilistic constraint [16].

After, the feasible design region is formed. Suppose that $\boldsymbol{\mu}_0$ is a point lying on $g_i(\mathbf{d}, \boldsymbol{\mu}_0)$, then the Most Probable Point (MPP), $\boldsymbol{\Theta}_{MPP}$ corresponding to $\boldsymbol{\mu}_0$ can be determined by moving $\boldsymbol{\mu}_0$ backward to $g_i(\mathbf{d}, \boldsymbol{\Theta})$ a distance β

As shown in [17], $\boldsymbol{\Theta}_{MPP}$ on the failure in a standard normal space can be defined by

$$\boldsymbol{\Theta}_{j,MPP}^u = \beta_i \frac{(\partial g_i / \partial \theta_j^u)_*}{\sqrt{\sum_j (\partial g_i / \partial \theta_j^u)_*^2}}, j=1, \dots, n, n+1, \dots, n+q \quad (3.7)$$

where the subscript u denotes the standard normal distribution space and the derivatives $(\partial g_i / \partial \theta_j^u)_*$ are evaluated at $\boldsymbol{\Theta}_{j,MPP}^u$.

The relationship of random parameters in the original design space and the normal standard space is depicted as follows

$$\theta_j^u = \frac{\theta_j - \mu\theta_j}{\sigma\theta_j} \quad (3.8)$$

From equation (3.8),

$$\frac{\partial g_i}{\partial \theta_j^u} = \frac{\partial g_i}{\partial \theta_j} \frac{\partial \theta_j}{\partial \theta_j^u} = \frac{\partial g_i}{\partial \theta_j} \sigma\theta_j \quad (3.9)$$

By combining equation (3.7),(3.8),(3.9), the relationship between $\boldsymbol{\mu}_0$ and $\boldsymbol{\Theta}_{MPP}$ in the original design space is denoted by

$$\boldsymbol{\Theta}_{j,MPP} = \mu\theta_j + \beta_i \sigma\theta_j \frac{\sigma\theta_j (\partial g_i / \partial \theta_j)_*}{\sqrt{\sum_j (\sigma\theta_j (\partial g_i / \partial \theta_j)_*)^2}} \quad (3.10)$$

where the derivatives $(\partial g_i / \partial \theta_j)_*$ are evaluated at $\boldsymbol{\Theta}_{j,MPP}$.

According to Li *et. Al* [16], the derivatives $(\partial g_i / \partial \theta_j)_*$ can be approximately assessed at $\mu\theta_j$, and then equation (3.10) can be rewritten as

$$\boldsymbol{\theta}_{j,MPP} = \mu\theta_j + \beta_i\sigma\theta_j \frac{\sigma\theta_j(\partial g_i / \partial \theta_j)_\#}{\sqrt{\sum_j (\sigma\theta_j(\partial g_i / \partial \theta_j)_\#)^2}} \quad (3.11)$$

where the derivatives $(\partial g_i / \partial \theta_j)_\#$ are evaluated at $\mu\theta_j$.

Once $\boldsymbol{\theta}_{MPP}$ in the original design space has been determined, the feasible domain of the RBDO problem can be denoted by approximate deterministic constraints as

$$g_i(\mathbf{d}, \boldsymbol{\mu}_\theta) = g_i(\mathbf{d}, \boldsymbol{\mu}_\theta + \beta_i \boldsymbol{\sigma}_\theta \mathbf{n}) \leq 0 \quad (3.12)$$

where $\mathbf{n} = (\sigma_\theta \nabla_{\boldsymbol{\theta}} g_i(\boldsymbol{\mu}_\theta) / \|\sigma_\theta \nabla_{\boldsymbol{\theta}} g_i(\boldsymbol{\mu}_\theta)\|)$ is the approximately normalized gradient vector evaluated at $\boldsymbol{\mu}_\theta$ on $g_i(\mathbf{d}, \boldsymbol{\theta})$.

Then, the RBDO problem in Equation (3.6) can be reformulated by an ADO problem as follows:

$$\begin{aligned} & \text{find} && \mathbf{d}, \boldsymbol{\mu}_x && (3.13) \\ & \min && f(\mathbf{d}, \boldsymbol{\mu}_x) \\ & \text{subject to} && g_i(\mathbf{d}, \boldsymbol{\mu}_x + \beta_i \boldsymbol{\sigma}_\theta \mathbf{n}) \leq 0, \quad i = 1, 2, 3, \dots, m \\ & && \mathbf{d}^{low} \leq \mathbf{d} \leq \mathbf{d}^{up}, \boldsymbol{\mu}_x^{low} \leq \boldsymbol{\mu}_x \leq \boldsymbol{\mu}_x^{up} \end{aligned}$$

It should be noted that the derivatives, $\nabla_{\boldsymbol{\theta}} g_i(\boldsymbol{\mu}_\theta)$, can be easily acquired through the direct derivative of an explicit limit-state function. Nonetheless, for the practical application problems, the structural behaviors are often analyzed by numerical methods, and hence the limit-state function are often implicit. To calculate the derivatives $\nabla_{\boldsymbol{\theta}} g_i(\boldsymbol{\mu}_\theta)$, the finite difference method, which is a numerical derivative method, is therefore used in this study.

CHAPTER 4

FUNDAMENTAL THEORY OF NEURAL NETWORK

4.1 Fundamental theory of Neural Network

Neural networks consist of a large class of different architectures and has variety of applications from object classification, pattern recognition to approximation. In many cases, the problem under consideration is to approximate a static nonlinear function $f(x)$ by a neural network $f_{NN}(x)$. And that is one of the most useful applications of neural network which is utilized in this thesis.

The most useful neural networks in function approximation are Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF). In this thesis, MLP networks are chosen to study and apply in solving optimization problems. A MLP consists of an input layer, several hidden layers, and an output layer as in Figure 4. 1.

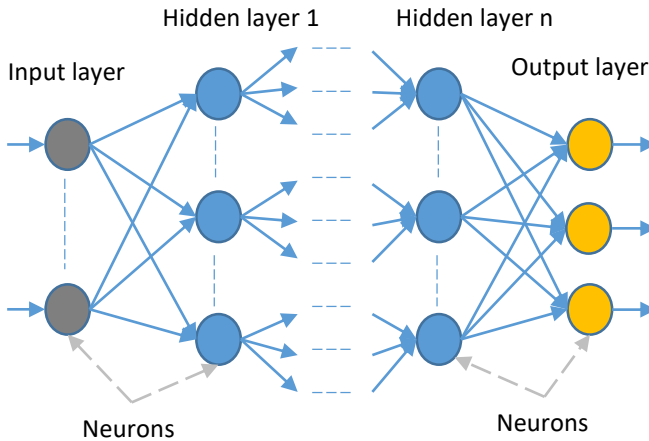


Figure 4. 1. A Multi-layer perceptron network model

In a MLP network, a single node i , also called a neuron, includes a summer and a nonlinear activation function g is shown in **Figure 4. 2.**

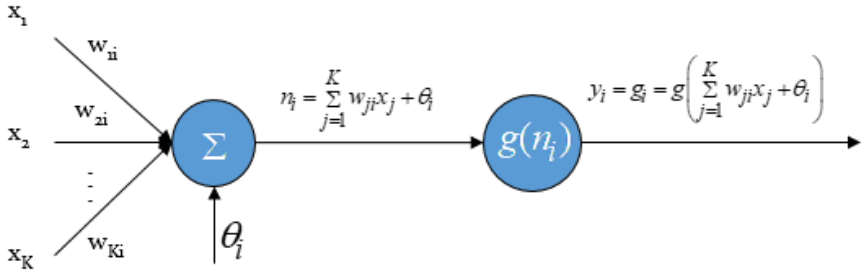


Figure 4. 2. Single node in an MLP network

The inputs x_k , $k = 1, \dots, K$ to the neuron are multiplied by weights w_{ki} and summed up together with the constant bias term θ_i . The resulting n_i is the input to the activation function g . The activation function was originally chosen to be a relay function, but for mathematical convenience a hyperbolic tangent (\tanh) or a sigmoid function are most commonly used.

And the sigmoid function is:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

The sigmoid function exists between 0 and 1. Therefore, it is especially used for models that need to predict the probability as an output. Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice.

Hyperbolic tangent is defined as

$$\tanh(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

Tanh is also like sigmoid function but the range of the *tanh* function is from (-1 to 1).

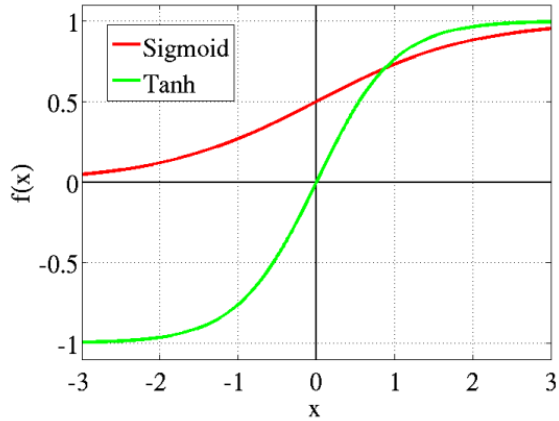


Figure 4. 3. Tanh and Sigmoid function

The output of node i becomes:

$$y_i = g_i = g \left(\sum_{j=1}^K w_{ji} x_j + \theta_i \right) \quad (4.1)$$

Connecting several nodes in parallel and series form a MLP network. A typical network is shown in **Figure 4. 4**

The unknown parameters to be estimated are then the weights and biases (w_{ji}^k, θ_j^k) . The algorithms for determining the network parameters are called learning or training algorithms. There are many training algorithms. The most well-known are back-propagation and Levenberg-Marquardt algorithm. Back-propagation is a gradient based algorithm, which has many variants. Levenberg-Marquardt is usually more efficient, but needs more computer memory. In this thesis, the Levenberg-Marquardt algorithm is used.

4.1 Artificial Neural Network based metaheuristic optimization methods

By combining the advantages of Artificial Neural Network (ANN) and the Differential Evolution (DE) algorithm, the ANN-based Differential Evolution (ABDE) optimization algorithm is proposed. Specifically, the calculation of the objective function values of individuals in the population at each generation and the evaluation of the trial vectors in the crossover step carried out by finite element analysis as in the original DE algorithm are replaced by the approximation of the ANN as illustrated in **Figure 4. 5**.

In **Figure 4. 5**, the process of building an ANN model usually consists of three main parts: sampling, training and verifying the net as illustrated in “*ANN Block*”. In this thesis, the samples used for ANN training are created by the FEM and the algorithm used to train the net is Levenberg-Marquardt.

Thanks to the advantage of being able to immediately evaluate the fitness of all individuals in the population by just a single command ‘*net (population)*’ rather than using the *for/end* loop and finite analysis for each individual value, the computational speed of the ABDE optimization algorithm is increased significantly. Although, the approximation of the structural respond of stiffened composite plate based on the training network ‘*train(net,input,output)*’ is erroneous, however this error is small and the accuracy is acceptable.

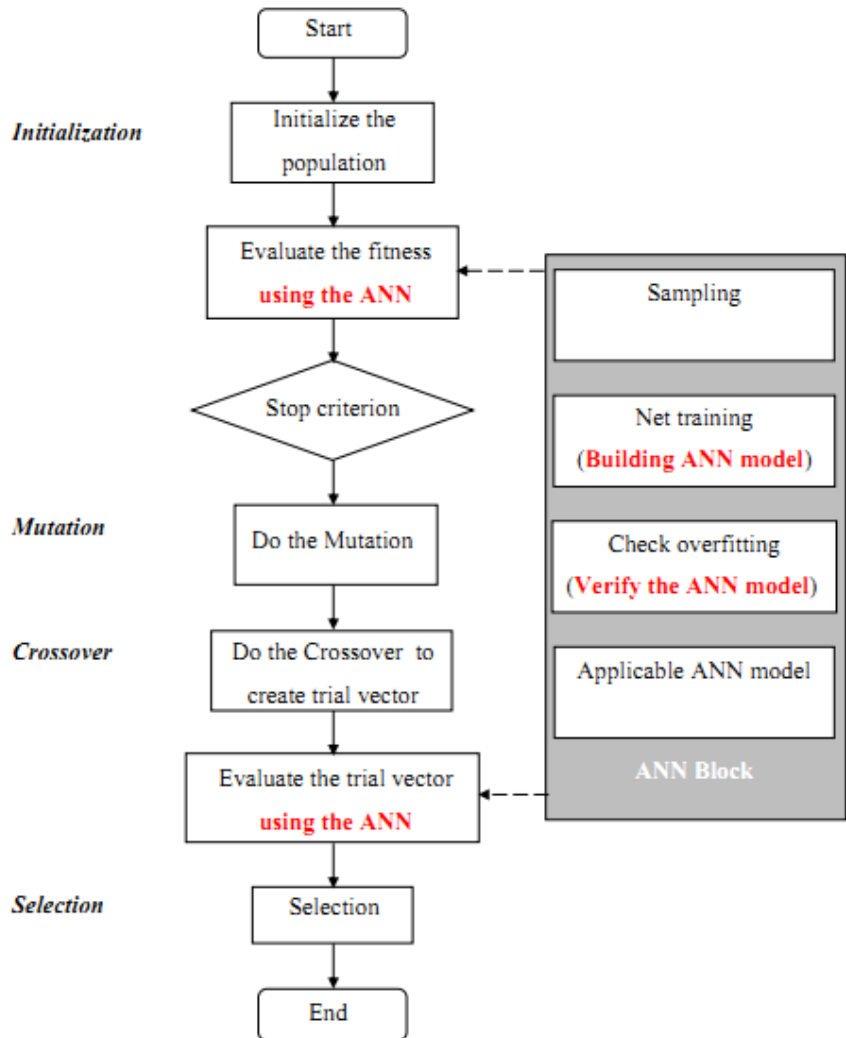
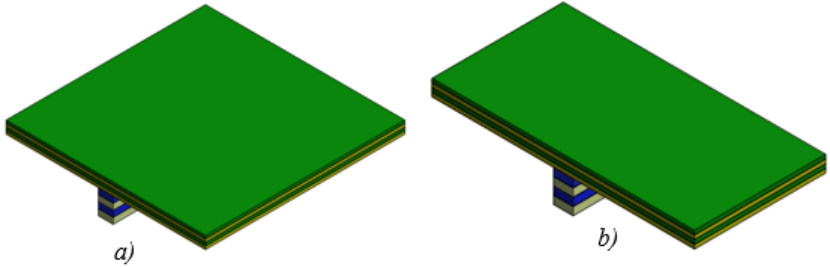


Figure 4. 5. Optimization process using Artificial Neural Network based Differential Evolution (ABDE) optimization algorithm

CHAPTER 5

DEVELOPMENTS OF METAHEURISTIC OPTIMIZATION METHODS

5.1 The effective of the improved Differential Evolution algorithm



a) square plate – b) rectangular plate

Figure 5. 1. Models of square and rectangular stiffened composite plates

To verify the accuracy of the improved DE algorithm (iDE), the fiber-angle optimization of two models of square and rectangular stiffened composite plate are considered in this section. The stiffeners in both cases are in X-direction as shown in **Figure 5. 1**. The objective function in this problem is the strain energy of the plate.

The optimization problem is expressed as:

$$\begin{cases} \min_{\theta} & \mathbf{U} = \frac{1}{2} \mathbf{d}^T \mathbf{K} \mathbf{d} \\ \text{subject to} & 0 \leq \theta_i \leq 180, \quad i = 1, \dots, 4 \end{cases}$$

where U is strain energy and θ_i is fiber orientation of i th layer.

In order to save the time of computation but still ensure accuracy, the integer values of design variables were used in this analysis.

The optimization parameters of improved Differential Evolution algorithm used in this stage are given here:

Initial Population size (NP): 20 individuals each generation

Total generation: 200

The crossover control parameter (Cr): 0.9

The scaling factor (Fx): 0.8

Tolerance: 1e-6

The optimal results of fiber orientations for both cases are presented in **Table 5. 1**. The results from the **Table 5. 1** show that the solutions by the iDE agree very well with those by the GA. The errors of strain energy in both case are very low. The maximum one is just about 0.2% for the case of rectangular plate. However, the computational time of the iDE algorithm are much smaller. Especially, in the case of rectangular plate, the time of computation from the GA method is nearly double in comparison with the one from iDE, 4995 seconds and 2851 seconds, respectively. This proved the accuracy and the effectiveness of the iDE method.

The results of **Table 5. 1** also shows that the geometric parameters of the structures also have influence to the optimal values of the problems. This is explained by the fact that the optimal fiber orientations of the square and rectangular plate are quite different under the same conditions.

Table 5. 1. The optimal results of two problems

Type of stiffened plate	Method	Optimal angle [Degree]				Strain energy (N.m)	Computational time (seconds)
		θ_1	θ_2	θ_3	θ_4		
Square	iDE	135	48	0	180	6183.2	2065
	GA	135	48	0	180	6183.1	2253
Rectangular	iDE	160	37	0	180	30366	2851
	GA	159	37	0	180	30300	4995

5.2 Optimization of reinforced composite plate

5.2.1 Optimization of thickness of stiffened composite plate

Consider the problem of optimizing composite plate reinforced by composite beams as shown in **Figure 5. 2** under a simply-supported condition. The problem parameters are given as follows: length of beam $a = 254$ mm, height of beam is h , cross section of reinforced beam has the width of $c_x = 6.35$ mm, and the height is d_x . The optimal analysis is carried out with two cases of square plate ($b = 254$ mm) and rectangular plate ($b = 508$ mm).

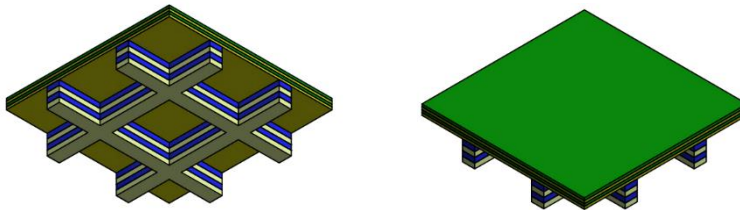


Figure 5. 2. Model of stiffened composite plate for optimization

Both plate and beam have a symmetrical four-layer structure. The fiber direction of each layer on the plate is respectively $[90 \ 45 \ 45 \ 90]$, and the layer fiber of the beam is $[180 \ 0 \ 0 \ 180]$. Beams and plate are made with the same materials with parameters $E_1 = 144.8\text{GPa}$,

$E_2 = E_3 = 9.65 \text{ GPa}$, $G_{12} = G_{13} = 4.14 \text{ GPa}$, $G_{23} = 3.45 \text{ GPa}$,
 $\nu_{12} = \nu_{13} = \nu_{23} = 0.3$. The plate is loaded evenly with the value of $f = 0.6895 \text{ (N/mm}^2\text{)}$.

The problem can be represented as follows:

$$\left\{ \begin{array}{ll} \min_{t_p, t_b} & \text{Weight}(h, d_x) \\ \text{subject to} & \text{Disp} \leq 1 \\ & \sigma_{T\text{sai-wu}} \leq 1 \end{array} \right.$$

In particular, the objective function is the stiffened composite plate mass and subjected to two constraints including the displacement of the stiffened plate less than 1 and the Tsai-Wu stress is also less than 1.

The thickness optimization results are calculated in different cases of stiffened beams (two stiffened beams with direction of XY, XX, YY or four stiffened beams with direction of XX-YY) and are presented in **Table 5. 2**. It is seen from the results that, in the case of square plate, the objective function reaches the lowest value with 4 stiffened beams (XX-YY). The reason is because although the number of stiffened beams is more than the case of 2 stiffened beams, the thickness of the plate obtained is much smaller than the case of 2 stiffened beams. So in the case of square plates, the structure of 4 stiffened beams should be chosen for better results. In the case of rectangular plates, the best results are obtained in the case of 2 stiffened beams arranged in Y-direction (Y-Y), with the value of the plate thickness being the smallest. From the above results, one observation is that the optimal option is usually achieved with the smallest plate thickness combined with the more number of stiffened beams.

Table 5. 2. The optimal thickness results for stiffened composite plate problems

Gia cường	R/S	Theta	h	d_x	Hàm mục tiêu	Thời gian (s)
X-Y	S	90/45/180/0	13.165	58.509	1.442168	2892
X-X / Y-Y	S	90/45/180/0	10.187	74.651	1.247581	3111
XX-YY	S	90/45/180/0	8.04236	42.7559	1.104029	4989
X-Y	R	90/45/180/0	27.40554	29.32186	5.109688	12475
Y-Y	R	90/45/180/0	11.8138	106.397	3.071304	10309
X-X	R	90/45/180/0	17.53	119.837	3.68064	10706

5.2.2 Artificial neural network-based optimization of reinforced composite plate

In this section, the proposed ABDE optimization algorithms were applied to three problem models: rectangular plate with X-directional stiffener (R-X), rectangular plate with Y-directional stiffener (R-Y), and square plate reinforced in two directions X and Y (S-XY). The results in this section obtained by the computer with the following configuration:

Processor: Intel® Core™ i5-2430M CPU @ 2.4GHZ

Installed Memory (RAM): 4,00 GB (3,90 GB usable)

System type: 64-bit Operating System, x64-based processor

Firstly, the samples for net training were created with 10,000 samples for each case. The net after training will be checked for “overfitting phenomenon” with other 1,296 samples to ensure the accuracy and the applicability of the net. The time for sampling and

the average errors of overfitting checking process are presented in **Table 5. 3** the below :

Table 5. 3. Sampling and overfitting checking error

Samples	Case	Total time (seconds)	Training time	Avg. Error
10000	R-X	38535	1.59	2.55 %
10000	R-Y	39188	1.30	3.55 %
10000	S-XY	19434	7.33	3.01 %

The average errors for three cases are small, so the ANN model can ensure the accuracy in approximating the objective function of the stiffener composite plate in the optimization process of the DE algorithm.

The optimal results are shown in **Table 5. 4**. Compared to the optimal results obtained from the DE algorithm, the values of the design variable computed from ABDE is very similar. The errors of objective function values obtained for the three models were 0.86%, 1.57%, 1.99%, respectively. These errors are most influenced by the training model of ANN. However, with a maximum error of 1.99%, it is perfectly acceptable. Besides, superior advantage in terms of computational time, which is also the strongest advantage of the ABDE algorithm, were very well expressed. The cost for the R-X model computed by ABDE is only 8 seconds compared to 2851 seconds of DE, which is 356 times faster. The remaining 2 cases of R-Y and S-XY models, the cost are respectively 13 seconds and 5 seconds compared to 2903 seconds and 1497 seconds of the DE algorithm. This superiority is due to the fact that the objective function can be approximated from the ANN model by a single command: *net(population)*, which is independent of the population of each generation, instead of using *for/end* loop as in the original DE algorithm. In three cases above, the lowest computational time with highest error is of S-XY case. This is because the square plate has smaller dimension in comparison with the rectangular one, so the

computational time is faster. Meanwhile, the error may be caused by the error in the approximation of ANN model and the accuracy of the FEM procedure in creating the sample for neural network training. Therefore, to ensure the accuracy and the effectiveness of the proposed method, the ANN training and sampling step must be considered carefully.

The results in **Table 5. 4** can prove the ABDE algorithm's superior accuracy and efficiency in comparison to the independent DE algorithm. This is especially useful for solving problems with large quantity of loops and high-cost objective functions.

Table 5. 4. Comparison of accuracy and computational time between DE and ABDE

Type of stiffened plate	Method	Optimal angle [Degree]				Strain energy (N.m)	Error (%)	Cost (seconds)
		θ_1	θ_2	θ_3	θ_4			
R-X	DE	160	37	0	180	30366	0.86	2851
	ABDE	160	38	0	180	30104		8
R-Y	DE	178	12	0	180	33039	1.57	2903
	ABDE	177	11	0	180	32520		13
S-XY	DE	135	45	0	0	4223	1.99	1497
	ABDE	134	42	0	0	4307		5

Additionally, this algorithm can also be applicable for many type of problems including linear and nonlinear ones because the idea here is just alternating the approach of evaluating the objective function by using ANN instead of using FEM. Therefore, the computational time is saved. Moreover, the ANN model can well approximate both linear and nonlinear response. So, this technique will be applied for both linear and nonlinear optimization problems.

5.3 Deterministic optimization of composite beam

In optimization of composite laminated beams, the fiber orientations, width and depth (thickness of layer) can be chosen as design variables. In this section, the design optimization is implemented for the one-dimensional Timoshenko composite laminated beam with the stiffness failure, strength failure and delamination failure are included in the optimization mathematical model. Therefore, as explained clearly in the work of Liu [8], only the width and the depth of the beam are chosen as the design variables to obtain the lightweight designs of the beams.

Consider the design optimization model of the composite beam taking into account the constraint of the stiffness failure criterion, strength failure criterion and delamination failure criterion. The problem can be mathematically expressed as follows [8]:

$$\text{Find } \mathbf{d} = [b, h]^T$$

$$\text{Minimize } W(\mathbf{d})$$

$$\text{Subject to } g_j = \left(\frac{\sigma_1^2}{X_t X_c} - \frac{\sigma_1 \sigma_2}{\sqrt{X_t X_c Y_t Y_c}} + \frac{\sigma_2^2}{Y_t Y_c} + \frac{\tau_{12}^2}{S^2} + \frac{X_c - X_t}{X_t X_c} \sigma_1 + \frac{Y_c - Y_t}{Y_t Y_c} \sigma_2 - 1 \right)_j < 0$$

$$f_j = \left(\frac{\tau_{13}^2}{S_{13}^2} + \frac{\tau_{23}^2}{S_{23}^2} - 1 \right)_j < 0$$

$$r = w_0(\alpha L) - \bar{w}_0 \leq 0$$

$$\underline{b} \leq b \leq \bar{b}$$

$$\underline{h} \leq h \leq \bar{h}$$

where $W(\mathbf{d})$ is the mass of the composite laminated beam. g , f and r are respectively strength failure function, delamination failure function and stiffness failure function. \underline{b} and \bar{b} are the lower and upper bound of the width of the beam. \underline{h} and \bar{h} are the lower and upper bound of the depth of the beam, respectively. αL determines the location in x -direction where the deflection of the beam is monitored. α is different for various types of boundary conditions

(PP: $\alpha=1/2$, FF: $\alpha=1/2$, FP: $\alpha=505/873$ and CL: $\alpha=1$). $w_0(\alpha L)$ is the deflection of the beam at the position αL . \bar{w}_0 is the limit on the deflection of the beam. The subscript ($j = 1, 2, \dots, N_m$) indicates the j th monitored point in the set N_m monitored points of the strength and delamination. X_t and X_c are the tensile strength and compressive strength along the 1-axis of the material coordinate system, respectively. Y_t and Y_c are the tensile strength and compressive strength along the 2-axis of the material coordinate system, respectively. S is the shear strength on the plane 102 of the material coordinate system. S_{12} and S_{23} are the shear strength on the plane 103 and 203 of the material coordinate system. In this thesis, let $S_{12} = S_{23}$.

5.5.1 Optimal design with variables: b and h

Consider the composite beams with the material properties given in **Table 5. 5**. The beams have $N=8$ layers with symmetric fiber orientations of $[0/90/45/-45]_s$. The span of the composite laminated beam is $L = 7.2\text{m}$. The beams are subjected to the uniform distributed loading $q_0 = 10^5 \text{N/m}$ and are considered under various types of constraint including PP, FF, FP and CL. The initial design of the composite laminated beams are $b=0.3 \text{ m}$ and $h = 0.48 \text{ m}$ (the thickness of each layer is 0.06), mass $W = 1597 \text{ kg}$. The lower and upper boundary of the design variables are $0.1\text{m} \leq b \leq 2\text{m}$, $0.2\text{m} \leq h \leq 2\text{m}$.

Table 5. 5. Material properties of lamina

Property	T300/5208
E_1 (GPa)	136.00
$E_2 = E_3$ (GPa)	9.80
$G_{12} = G_{13}$ (GPa)	4.70
G_{23} (GPa)	5.20
$\nu_{12} = \nu_{13}$	0.28
ν_{23}	0.15

ρ (kg/m ³)	1540
X_t (Mpa)	1550
X_c (Mpa)	1090
Y_t (Mpa)	59
Y_c (Mpa)	59
S (Mpa)	75
$S_{I3} = S_{I3}$ (Mpa)	75

The optimization design problems are solved with different types of boundary conditions (P-P, F-F, F-P and C-L) using four different population-based algorithm including DE, IDE, Jaya and iJaya and one gradient-based algorithm from Liu’s work. The numerical results are presented in **Table 5. 6**. As shown in the table, the optimal mass obtained from iJaya are agreed well with other solutions. However, the iJaya algorithm consumed least time to achieve the optimal solution in compared with other approaches. Among the five methods, the SQP (implemented by *fmincon* prompt in Matlab) algorithm used in Liu’s work reached the optimal solution very fast but it could be stuck in the local optimum and that affects the value the optimal solution. In comparison with other global optimization method (DE, IDE, Jaya), the iJaya method definitely outperforms them. In particularly, the iJaya decreased the computational time at least 13% for the case of C-L condition to the maximum of 27% for the case of P-P condition in comparison with the original Jaya algorithm. The number of average function count is reduced up to 21% when using the iJaya method instead of the Jaya. The iJaya is also faster than IDE approach in reaching the optimal solutions and the convergence speed of iJaya is approximately double the speed of DE in achieving the optimal values of design variables. From the above analyses, iJaya can be considered as the most effective and the efficient algorithm because it solved for the global solution at the highest speed of convergence.

Table 5. 6. Comparison of optimal design with continuous design variables

BC	Optimal results	Liu [39]	DE	IDE	Jaya	iJaya
P-P	Mass	909.2634	909.2634	909.2634	909.2634	909.2634
	$[h, b]$	[0.1000, 0.8200]	[0.1000, 0.8200]	[0.1000, 0.8200]	[0.1000, 0.8200]	[0.1000, 0.8200]
	Worst mass	-	909.2640	909.2642	909.2638	909.2639
	Mean mass	-	909.2636	909.2637	909.2636	909.2636
	Std.	-	0.0001	0.0002	0.0001	0.0001
	Average f-count	7	1660	1075	1181	933
	CPU time (s)	45 [0.54]*	1.06	0.69	0.82	0.60
F-F	Mass	560.7427	560.7428	560.7428	560.7428	560.7427
	$[h, b]$	[0.1000, 0.5057]	[0.1000, 0.5057]	[0.1000, 0.5057]	[0.1000, 0.5057]	[0.1000, 0.5057]
	Worst mass	-	560.7430	560.7433	560.7430	560.7430
	Mean mass	-	560.7429	560.7429	560.7428	560.7428
	Std.	-	0.0001	0.0001	0.0001	0.0001
	Average f-count	9	1654	1084	1144	921
	CPU time (s)	6 [0.17]	1.15	0.73	0.84	0.70
F-P	Mass	706.5145	706.5145	706.5145	706.5145	706.5145
	$[h, b]$	[0.1000, 0.6372]	[0.1000, 0.6372]	[0.1000, 0.6372]	[0.1000, 0.6372]	[0.1000, 0.6372]
	Worst mass	-	706.5148	706.5150	706.5151	706.5149
	Mean mass	-	706.5146	706.5147	706.5146	706.5147
	Std.	-	0.0001	0.0001	0.0001	0.0001
	Average f-count	32	1641	1061	1153	924
	CPU time (s)	11 [0.11]	1.07	0.67	0.77	0.61
C-L	Mass	2065	2064.9645	2064.9645	2064.9646	2064.9645
	$[h, b]$	[0.1000, 1.8623]	[0.1000, 1.8623]	[0.1000, 1.8623]	[0.1000, 1.8623]	[0.1000, 1.8623]
	Worst mass	-	2064.9658	2064.9658	2064.9654	2064.9655
	Mean mass	-	2064.9649	2064.9650	2064.9649	2064.9648
	Std.	-	0.0003	0.0003	0.0002	0.0003
	Average f-count	16	1591	954	1039	863
	CPU time (s)	38 [0.13]	0.51	0.29	0.31	0.27

[*]: CPU time in this study by using SQP algorithm in *fmincon* Matlab.

The speed of convergence of the four population-based global optimization algorithms for the case of P-P condition can be shown

in **Figure 5. 3**. As we can easily see, the number of structural analyses to achieve the lightweight design of P-P beam of iJaya algorithm is far less than those of DE, IDE and the original Jaya. It just took around 450 structural analyses to converge to the optimal mass while the DE and Jaya spent about 600 analyses and more than 500 analyses for the case of IDE. These numbers proved that the iJaya algorithm outperforms the others in attaining the optimal solution.

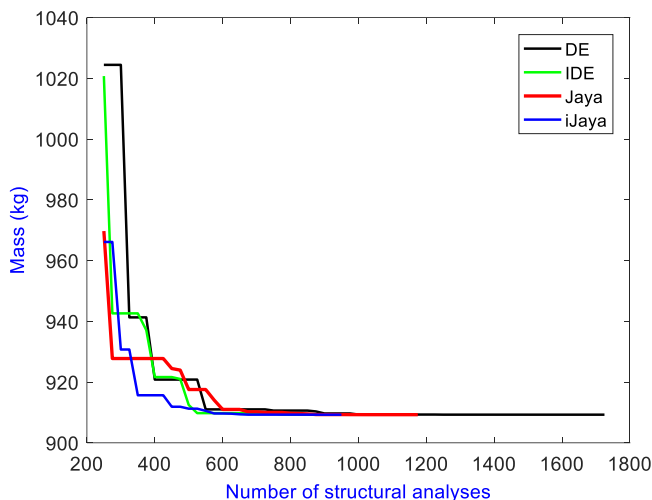


Figure 5. 3. Convergence curves of DE, IDE, Jaya and iJaya for the beam with P-P condition

5.5.2 Optimal design with variables: b and t_i

In this section, the depth of the composite laminated beam (h) are divided into thicknesses of the layers of the beam to optimize. This is implemented with the intention of improving the optimal design of the composite laminated beam and achieving lighter weight for the beam. The thicknesses of each layer, denoted by $[t_1, t_2, t_3, t_4]_s$, are chosen to be design variables in this case can be considered as the discrete design variables.

Table 5. 7. Optimal design with discrete design variables

BC	Optimal results	Liu [39]	DE	IDE	Jaya	iJaya
P-P	Mass	909.2634	410.256	410.256	410.256	410.256
	b	0.1000	0.100	0.100	0.100	0.100
	$[t_1, t_2, t_3, t_4]_s$	[0.1025, 0.1025, 0.1025, 0.1025] _s	[0.190, 0.060, 0.060, 0.060] _s	[0.190, 0.060, 0.060, 0.060] _s	[0.190, 0.060, 0.060, 0.060] _s	[0.190, 0.060, 0.060, 0.060] _s
	Worst mass	-	415.800	415.800	415.800	415.800
	Mean mass	-	410.533	414.341	411.088	410.872
	Std.	-	1.240	2.508	2.031	1.793
	Average f-count	7	3375	2225	3132	2190
	CPU time (s)	45	2.75	1.81	2.24	1.51
	Mass	560.7427	260.568	260.568	260.568	260.568
	b	0.1000	0.100	0.100	0.100	0.100
$[t_1, t_2, t_3, t_4]_s$	[0.0632, 0.0632, 0.0632, 0.0632] _s	[0.105, 0.040, 0.040, 0.050] _s	[0.085, 0.050, 0.050, 0.055] _s	[0.100, 0.045, 0.040, 0.050] _s	[0.100, 0.045, 0.040, 0.050] _s	
F-F	Worst mass	-	260.568	266.112	260.568	266.112
	Mean mass	-	260.568	260.860	260.568	260.876
	Std.	-	0.000	1.272	0.000	1.307
	Average f-count	9	2675	2325	2534	1663
	CPU time (s)	6	2.18	1.90	1.81	1.15
	Mass	706.5145	327.096	327.096	327.096	327.096
	b	0.1000	0.100	0.100	0.100	0.100
	$[t_1, t_2, t_3, t_4]_s$	[0.0796, 0.0796, 0.0796, 0.0796] _s	[0.130, 0.060, 0.055, 0.050] _s	[0.130, 0.055, 0.060, 0.050] _s	[0.130, 0.050, 0.050, 0.065] _s	[0.125, 0.055, 0.055, 0.060] _s
	Worst mass	-	327.096	327.096	327.096	327.096
	Mean mass	-	327.096	327.096	327.096	327.096
Std.	-	0.000	0.000	0.000	0.000	
Average f-count	32	2650	2225	2499	1657	
CPU time (s)	11	2.17	1.83	1.83	1.17	
C-L	Mass	2065	942.480	942.480	942.480	942.480
	b	0.1000	0.100	0.100	0.100	0.100
	$[t_1, t_2, t_3, t_4]_s$	[0.2328, 0.2328, 0.2328, 0.2328] _s	[0.430, 0.140, 0.140, 0.140] _s	[0.400, 0.145, 0.150, 0.160] _s	[0.425, 0.140, 0.145, 0.140] _s	[0.425, 0.140, 0.140, 0.145] _s
	Worst mass	-	942.480	948.024	942.480	942.480
	Mean mass	-	942.480	945.415	942.480	942.480
	Std.	-	0.000	2.852	0.000	0.000
	Average f-count	16	3500	2350	3129	2190

The results obtained are presented in **Table 5. 7**. The optimal masses, as shown in **Table 5. 7**, obtained from all the population-based optimization methods with discrete design variables are just equal to half of that derived from Liu’s work using the SQP algorithm with continuous design variables. And once again, the iJaya method dominates the other methods in both the number of function count and the CPU time. The results from **Table 5. 7** also show that the optimization with discrete design variables are much more effective than solving the problem with continuous design variables.

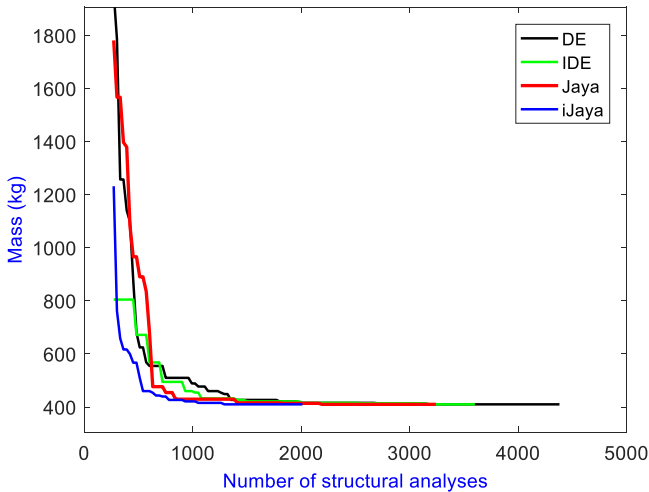


Figure 5. 4. Convergence curves of DE, IDE, Jaya and iJaya for the beam with P-P condition.

The convergence of the four population-based global optimization algorithms for the case of P-P condition are plotted in **Figure 5. 4**. The diagram shows that the iJaya method achieved the optimal lightweight design at the least number of structural analyses compared with those solved by DE, IDE and the original Jaya. The iJaya just took around 900 structural analyses to get close to the optimal mass while the others spent about 1400 structural analyses

to get close to the optimal solution. This shows that the iJaya is the fastest convergence method with high accuracy among them.

In **Figure 5. 5**, the comparison between optimization methods using thickness-of-layer design variables and the ones using depth-of-beam design variables with different types of boundary conditions is depicted. The first comparison between the population-based global optimization (the green line) and the gradient-based optimization method in Liu’s work (the red line) showed a good agreement in the optimal results.

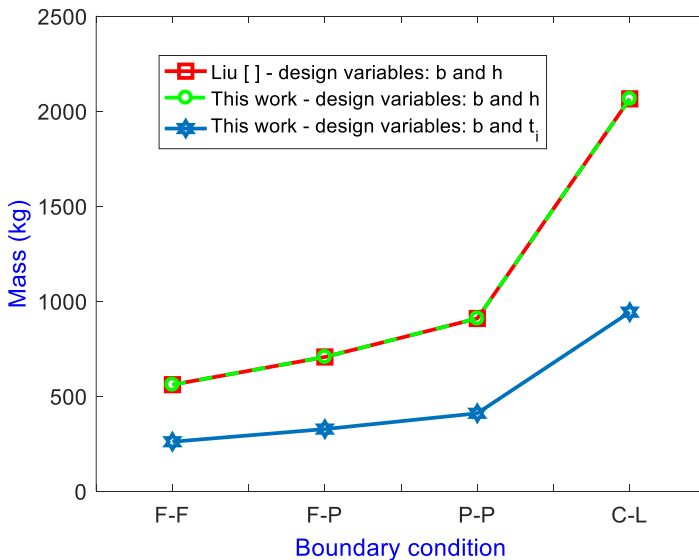


Figure 5. 5. Comparison of different design approaches with different boundary conditions.

The second comparison is between the global optimization approaches with discrete design variables (the blue line) and the SQP algorithm with continuous design variables in Liu’s work (the red line). The results showed that the former significantly decreased the values of the optimal masses of the beams, in particularly, up to half

of that obtained from the latter. This strongly proved the efficiency and the accuracy of the proposed method using iJaya algorithm with discrete design variables in optimizing the composite laminated beams.

5.4 Reliability-based optimization design of Timoshenko composite beam

5.4.1 Verification of SLDM-iJaya

A simple mathematical problem, which has been widely studied in the literature, is used to verify the proposed method in detail. The problem has only two normally distributed random variables x_1, x_2 , three probabilistic constraints, and there are no deterministic design variables and random parameters. The RBDO problem is mathematically described as follows [18], [19]:

$$\begin{aligned}
 &\text{find} && \mathbf{x} = \{x_1, x_2\} \\
 &\text{min} && f(\mathbf{x}) = \mu_{x_1} + \mu_{x_2} \\
 &\text{subject to} && \text{Prob}[g_i(\mathbf{x}) \leq 0] \geq \Phi(\beta_i), \quad i = 1, \dots, 3 \\
 &&& g_1(\mathbf{x}) = 1 - \frac{x_1^2 x_2}{20}, \\
 &&& g_2(\mathbf{x}) = 1 - \frac{(x_1 + x_2 - 5)^2}{30} - \frac{(x_1 - x_2 - 12)^2}{120} \\
 &&& g_3(\mathbf{x}) = 1 - \frac{80}{(x_1^2 + 8x_2 + 5)}, \\
 &&& 0 \leq \mu_{x_j} \leq 10, \quad \sigma_j = 0.3, \quad j = 1, 2, \\
 &&& \beta_i = 3, \quad i = 1, \dots, 3.
 \end{aligned}$$

The solutions for the above RBDO problem can be obtained by using the sequential optimization and reliability assessment (SORA) in the work of Du and Chen [20], the reliable design space approach proposed by Shan and Wang [19] and the single-loop and double-loop methods combing with the IDE algorithm presented in the paper of Ho-Huu et al [3]. The results obtained from the above methods

are compared with that of the proposed method in this study and are shown in **Table 5. 8**. The main method proposed in this paper is the combination of the SLDM and the iJaya algorithm. However, we also presented the results derived by the DLM-iJaya algorithm, which is formed by combining the DLM with the iJaya algorithm, for a broader comparison. As shown in the **Table 5. 8**, the solutions achieved by the SLDM-iJaya and DLM-iJaya are agreed well with those obtained by the other methods. However, the computational costs are significantly decreased when using the DLM-iJaya method and the SLDM-iJaya method instead of utilizing the DLM-IDE and SLDM-IDE as in the work of Ho-Huu et al. [3]. In particularly, the number of constraint function evaluations is reduced nearly 16% for the case of using the DLM-iJaya to 20% for the case of using the SLDM-iJaya. These reductions in term of computational costs are 52% and 36%, respectively. These numbers strongly verified the efficiency and effectiveness of the proposed SLDM-iJaya method.

Table 5. 8. Comparison of optimization results of the mathematical problem

Design variables (mm)	Du and Chen [146]		Shan and Wang [145]		Ho-Huu et al. [3]		This study	
	SORA	RDS	DLM-IDE	SLDM-IDE	DLM-iJaya	SLDM-iJaya		
x_1	3.4409	3.4406	3.4391	3.4406	3.4391	3.4406		
x_2	3.2909	3.2800	3.2866	3.2800	3.2866	3.2800		
Objective	6.7318	6.7205	6.7256	6.7205	6.7256	6.7205		
β	β_1^{FORM}	3.011	2.995	3.000	2.995	3.000	2.995	
	β_2^{FORM}	3.011	2.978	3.000	2.978	3.000	2.978	
	β_3^{FORM}	10.026	10.047	10.038	10.047	10.038	10.047	
	β_1^{MCS}	2.998	2.969	2.980	2.969	2.980	2.969	
	β_2^{MCS}	3.059	3.022	3.061	3.022	3.061	3.022	
	β_3^{MCS}	Infinite	Infinite	Infinite	Infinite	Infinite	Infinite	
Computational cost	No. of constraint function evaluations	-	-	9917	4600	8388	3700	
	Time (s)	-	-	0.912	0.390	0.4365	0.251	

*Note that the probability of failure is equal to 0% with $\beta > 5$.

5.4.2 Reliability-based lightweight design

In this section, the proposed method is applied to solve the lightweight reliability-based optimization design problem of the

composite laminated beams with various types of boundary conditions. The optimal results are computed with three different required reliability indexes, $\beta_{r,i}$. The results are listed in **Table 5. 9**. The optimal masses obtained, as shown in **Table 5. 9**, are compared between the DLM-iJaya and the SLDM-iJaya method. As we can see easily, the optimal solutions of mass are agreed well, but the function count and the CPU time by the SLDM-iJaya outperformed those of the DLM-iJaya in all cases.

In particularly, the CPU time consumed by the former is less than 5 times compared with the latter for the case of the P-P condition. This number in term of function count is even less than 10 times for the case of $\beta_{r,i} = 4$.

Table 5. 9. Optimal results of reliability based lightweight design with different level of reliability.

BC	Optimal results	Required reliability index, $\beta_{r,i} = 2$		Required reliability index, $\beta_{r,i} = 3$		Required reliability index, $\beta_{r,i} = 4$	
		DLM-iJaya	SLDM-iJaya	DLM-iJaya	SLDM-iJaya	DLM-iJaya	SLDM-iJaya
P-P	Mass	460.152	460.152	482.328	482.328	510.048	510.048
	b	0.100	0.100	0.100	0.100	0.100	0.100
	$[t_1, t_2, t_3, t_4]_s$	[0.200, 0.070, 0.070, 0.075] _s	[0.200, 0.070, 0.075, 0.070] _s	[0.225, 0.070, 0.070, 0.070] _s	[0.220, 0.075, 0.070, 0.070] _s	[0.230, 0.075, 0.075, 0.080] _s	[0.225, 0.075, 0.075, 0.085] _s
	f-count	319885	52416	471364	58590	449279	43776
	CPU time (s)	167.834	41.480	220.78	39.35	327.325	43.753
	$\beta_{0,1}$	2.026	2.026	3.012	2.978	4.006	3.978
	$\beta_{0,2}$	Infinite	Infinite	Infinite	Infinite	Infinite	Infinite
	$\beta_{0,3}$	Infinite	Infinite	Infinite	Infinite	Infinite	Infinite
	Mass	288.288	288.288	304.920	304.920	327.096	327.096
	b	0.100	0.100	0.100	0.100	0.100	0.100
F-F	$[t_1, t_2, t_3, t_4]_s$	[0.200, 0.070, 0.070, 0.075] _s	[0.200, 0.070, 0.075, 0.070] _s	[0.135, 0.045, 0.050, 0.045] _s	[0.130, 0.050, 0.050, 0.045] _s	[0.125, 0.055, 0.065, 0.050] _s	[0.120, 0.065, 0.055, 0.055] _s
	f-count	402721	49968	375080	46080	316688	50688
	CPU time (s)	182.614	33.308	165.33	28.26	183.42	36.66
	$\beta_{0,1}$	2.034	2.034	3.046	2.992	4.076	3.994
	$\beta_{0,2}$	Infinite	Infinite	Infinite	Infinite	Infinite	Infinite
	$\beta_{0,3}$	Infinite	Infinite	Infinite	Infinite	Infinite	Infinite

	Mass	365.904	360.360	382.536	382.536	404.712	404.712
	b	0.100	0.100	0.100	0.100	0.100	0.100
	$[t_1, t_2, t_3, t_4]_s$	[0.200, 0.070, 0.070, 0.075] _s	[0.200, 0.070, 0.075, 0.070] _s	[0.165, 0.060, 0.060, 0.060] _s	[0.165, 0.060, 0.060, 0.060] _s	[0.175, 0.070, 0.060, 0.060] _s	[0.175, 0.060, 0.070, 0.060] _s
F-P	f-count	327635	59616	661090	47664	354492	42480
	CPU time (s)	130.49	36.69	1381.7	35.50	156.90	26.737
	$\beta_{0,1}$	2.009	1.997	3.018	3.018	4.021	4.031
	$\beta_{0,2}$	Infinite	Infinite	Infinite	Infinite	Infinite	Infinite
	$\beta_{0,3}$	Infinite	Infinite	Infinite	Infinite	Infinite	Infinite
	Mass	1058.904	1058.904	1125.432	1125.432	1197.504	1197.504
	b	0.100	0.100	0.100	0.100	0.100	0.100
	$[t_1, t_2, t_3, t_4]_s$	[0.490, 0.155, 0.155, 0.155] _s	[0.485, 0.155, 0.155, 0.160] _s	[0.500, 0.165, 0.175, 0.175] _s	[0.490, 0.165, 0.190, 0.170] _s	[0.500, 0.175, 0.205, 0.200] _s	[0.500, 0.190, 0.200, 0.190] _s
C-L	f-count	448839	58752	403747	43920	424718	33120
	CPU time (s)	100.80	17.58	135.25	16.71	104.69	10.36
	$\beta_{0,1}$	2.009	1.996	3.003	2.979	4.003	3.999
	$\beta_{0,2}$	Infinite	Infinite	Infinite	Infinite	Infinite	Infinite
	$\beta_{0,3}$	Infinite	Infinite	Infinite	Infinite	Infinite	Infinite

The same performances occurred for the rest boundary conditions of F-F, F-P and C-L. Moreover, the SLDM-iJaya method can also achieve the same optimal masses obtained by the DLM-iJaya with smaller value of reliability indexes, $\beta_{0,i}$'s. For example, in the case of F-P condition with the required reliability index $\beta_{r,i} = 2$, the value of $\beta_{0,1}$ computed by the DLM-iJaya is 2.009 while this number attained by the SLDM-iJaya is 1.997. These results prove that the SLDM-iJaya method can acquire the optimal solutions with higher reliability and less effort.

Figure 5. 6 depicted the comparison of optimal results of the RBDO problem with different levels of reliability. The cyan line represents the result obtained from the DO problem without the reliability analysis, and its optimal weight is undoubtedly the smallest values. The top three line represents the optimal solutions achieved by the RBDO problems with different values of reliability indexes of $\beta=2,3$ and 4, respectively. As we can see, the larger the values of β the larger the values of masses, which means if we want to improve the

reliability of a structure, we need to reinforce the structure with more materials and that make the weight of the structure increase. This also implies that the optimization without reliability analysis can result in high-risk structure, although the cost is a lot saved. Therefore, in order to balance the costs and safety, the RBDO method should be taken into account in the optimization of structural design. And the SLMD-iJaya is one of the proven most effective method to deal with such problems.

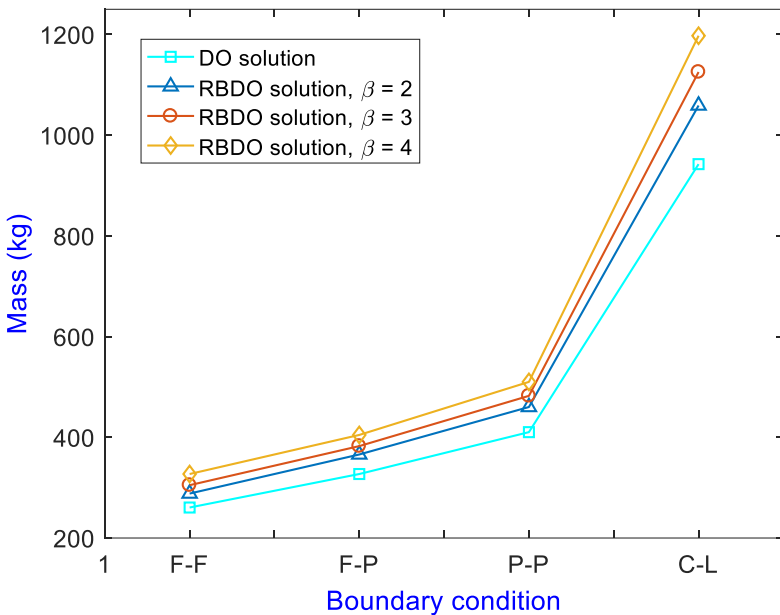


Figure 5. 6. Comparison of RBDO optimal results with different levels of reliability

CHAPTER 6 CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions and Remarks

- In this dissertation, the improved Differential Evolution is introduced and apply to solve for optimal fiber angle and thickness of the stiffened composite plate and the results showed its good effectiveness and accuracy.
- The elitist selection technique is utilized to modify the selection step of the original Jaya algorithm to improve the convergence of the algorithm. The improved Jaya algorithm is then applied to solve for optimization problem of the Timoshenko composite beam and obtained very good results.
- In addition, the so-called called (SLMD-iJaya) algorithm which is the combination of the improved Jaya algorithm and the Global Single-Loop Deterministic Methods (SLDM) has been proposed as a new tool set for solving the Reliability-Based Design Optimization of the continuous composite beam models. The results obtained is much better and more reliable than those without the reliability factor.
- The Artificial Neural Network is integrated to optimization process of the improved Differential Evolution algorithm to form a new algorithm call ABDE (ANN-based Differential Evolution) algorithm. This new algorithm is then applied to solve two optimization problems of the stiffened composite plate structures. The first one is optimizing the fiber angles of the stiffened composite plate and the second one is optimizing the thickness of the stiffened composite plate. The results obtained show a highly effective performance of the proposed ABDE tool set.
- Besides, through several numerical results, some main conclusions can be stated as follows:

+ The optimal results of fiber orientations for both cases of square and rectangular stiffened composite plate show that the solutions by the iDE agree very well with those by the GA. The errors of strain energy in both case are very low. The maximum one is just about 0.2% for the case of rectangular plate. However, the computational time of the iDE algorithm are much smaller. This proved the

accuracy and the effectiveness of the iDE method. The results also show that the geometric parameters of the structures also have influence to the optimal values of the problems. Particularly, the optimal fiber orientations of the square and rectangular plate are quite different under the same conditions

+ The thickness optimization results show that, in the case of square plate, the objective function reaches the lowest value with 4 stiffened beams (XX-YY) because the thickness of the plate obtained is much smaller than the case of 2 stiffened beams. In the case of rectangular plates, the best results are obtained in the case of two stiffened beams arranged in Y-direction (Y-Y), with the value of the plate thickness being the smallest. In other words, the optimal option is usually achieved with the smallest plate thickness combined with the more number of stiffened beams for the case of uniform loads.

+ The optimization design problems of Timoshenko composite beam are solved with different types of boundary conditions (P-P, F-F, F-P and C-L) using four different population-based algorithm including DE, IDE, Jaya and iJaya and one gradient-based algorithm from Liu's work. The numerical results show that the optimal mass obtained from iJaya are agreed well with other solutions. However, the iJaya algorithm consumed least time to achieve the optimal solution in compared with other approaches. Among the five methods, the SQP (implemented by *fmincon* prompt in Matlab) algorithm used in Liu's work reached the optimal solution very fast but it could be stuck in the local optimum and that affects the value the optimal solution. In comparison with other global optimization method (DE, IDE, Jaya), the iJaya method definitely outperforms them. The iJaya algorithm can be considered as the most effective and the efficient algorithm because it solved for the global solution at the highest speed of convergence. In the case where the depth of the composite laminated beam (h) are divided into thicknesses of the layers of the beam to optimize. And once again, the iJaya method

dominates the other methods in both the number of function count and the CPU time.

+ The solutions for the RBDO problem of the composite beam using the SLDM-iJaya has also presented and shown in **Table 5. 8**. The solutions achieved by the SLDM-iJaya and DLM-iJaya are agreed well with those obtained by the other methods. However, the computational costs are significantly decreased when using the DLM-iJaya method and the SLDM-iJaya method instead of utilizing the DLM-IDE and SLDM-IDE. In particularly, the number of constraint function evaluations is reduced nearly 16% for the case of using the DLM-iJaya to 20% for the case of using the SLDM-iJaya. These reductions in term of computational costs are 52% and 36%, respectively. These numbers strongly verified the efficiency and the effectiveness of the proposed SLDM-iJaya method.

+ The proposed SLDM-iJaya method is applied to solve the lightweight reliability-based optimization design problem of the composite laminated beams with various types of boundary conditions. The optimal results are computed with three different required reliability indexes, $\beta_{r,i}$. The results compared between the DLM-iJaya and the SLDM-iJaya method listed in **Table 5. 9** show that the optimal solutions of mass are agreed well, but the function count and the CPU time by the SLDM-iJaya outperformed those of the DLM-iJaya in all cases.

+ In the final chapter, the proposed ABDE optimization algorithms were applied to three problem models: rectangular plate with X-directional stiffener (R-X), rectangular plate with Y-directional stiffener (R-Y), and square plate reinforced in two directions X and Y (S-XY). Compared to the optimal results obtained from the DE algorithm, the values of the design variable computed from ABDE is very similar, with maximum error is just 1.99%. The cost for the R-X model computed by ABDE is only 8 seconds compared to 2851 seconds of DE, which is 356 times faster. The remaining 2 cases of R-Y and S-XY models, the cost are respectively 13 seconds and 5

seconds compared to 2903 seconds and 1497 seconds of the DE algorithm. This superiority is due to the fact that the objective function can be approximated from the ANN model by a single command: *net(population)*, which is independent of the population of each generation, instead of using for/end loop as in the original DE algorithm. The results in **Table 5. 4** can prove the ABDE algorithm's superior accuracy and efficiency in comparison to the independent DE algorithm. This simple algorithm can be easily extended to apply to solve many types of structural optimization problems for both linear and nonlinear models.

6.2 Recommendations and future works

- The improved DE can be extended to many other optimization problems of many other materials and structure. And can be applied to solve the RBDO problems, too.
- The improved Jaya algorithm can be applied to other type of composite structure such as stiffened composite plate, folded composite plate, flat shell for more verifications and test.
- Investigate in techniques that can help to improve the training speed and accuracy of the Artificial Neural Networks is also one potential direction research from this thesis
- The ABDE algorithm can be extended to many other optimization problems for proving its effectiveness and efficiency with many types of different materials as well.

LIST OF PUBLICATIONS

Parts of this dissertation have been published in international journals, national journals or presented in conferences. These papers are:

International Journal

1. **T. Lam-Phat**, V. Ho-Huu, S. Nguyen-Ngoc, S. Nguyen-Hoai, Trung Nguyen-Thoi. Deterministic and reliability-based lightweight design of Timoshenko composite beams. **Engineering with Computers**, 2020, <https://doi.org/10.1007/s00366-020-00946-8>.
2. **T. Lam-Phat**, S. Nguyen-Hoai, V. Ho-Huu, Q. Nguyen, T. Nguyen-Thoi. An Artificial Neural Network-Based Optimization of Stiffened Composite Plate Using A New Adjusted Differential Evolution Algorithm. Proceedings of the International Conference on Advances in Computational Mechanics 2017 pp 229-242 (Part of the **Lecture Notes in Mechanical Engineering** book series (LNME))
Link: https://link.springer.com/chapter/10.1007/978-981-10-7149-2_16
3. Q. Nguyen, S. Nguyen-Hoai, T. Chuong-Thiet, **T. Lam-Phat**. Optimization of the Longitudinal Cooling Fin by Levenberg–Marquardt Method. Proceedings of the International Conference on Advances in Computational Mechanics 2017 pp 217-227 (Part of the **Lecture Notes in Mechanical Engineering** book series (LNME))

Link: https://link.springer.com/chapter/10.1007/978-981-10-7149-2_15

4. T. Nguyen-Thoi, T. Rabczuk, **T. Lam-Phat**, V. Ho-Huu, P. Phung-Van (2014). Free vibration analysis of cracked Mindlin plate using an extended cell-based smoothed discrete shear gap method (XCS-DSG3). **Theoretical and Applied Fracture Mechanics**. Vol.72, 150-163.

Link:

<https://www.sciencedirect.com/science/article/pii/S016784421400041X>

National Journal

5. **Lam Phat Thuan**, Nguyen Nhat Phi Long, Nguyen Hoai Son, Ho Huu Vinh, Le Anh Thang. Global Optimization of Lamination Composite Beams Using An Improved Differential Evolution Algorithm. **Journal of Science and Technology in Civil Engineering NUCE** 2020. 14 (1): 54–64
6. Nguyen-Thoi, T., Ho-Huu, V., Dang-Trung, H., Bui-Xuan, T., **Lam-Phat, T.** (2013) Optimization analysis of stiffened composite plate by sequential quadratic programming. **Journal of Science and Technology**, Vol. 51(4B), p. 156-165.
7. Nguyen Thoi Trung, Bui Xuan Thang, Ho Huu Vinh, **Lam Phat Thuan**, Ngo Thanh Phong. An Effective Algorithm For Reliability-Based Optimization Of Stiffened Mindlin Plate. **Vietnam Journal of Mechanics**, VAST, Vol. 35, No. 4 (2013), pp. 335 – 346

International Conference

8. **Thuan Lam-Phat**, Son Nguyen-Hoai, Vinh Ho-Huu, Trung Nguyen-Thoi. Optimization of stiffened composite plate using adjusted different evolution algorithm. Proceeding of the international conference on computational methods (Vol.3, 2016), Berkeley, CA, USA.

National Conference

9. **Thuan Lam-Phat**, Son Nguyen-Hoai, Vinh Ho-Huu, Trung Nguyen-Thoi. Optimization analysis of stiffened composite plate by adjusted different evolution. Hội nghị Khoa học – Công nghệ toàn quốc về cơ khí 2015
10. **Lâm Phát Thuận**, Nguyễn Hoài Sơn, Lê Anh Thắng, Hồ Hữu Vịnh. Tối ưu hóa góc hướng sợi tấm Composite gia cường dùng thuật toán Differential Evolution kết hợp mạng thần kinh nhân tạo. Hội nghị cơ học toàn quốc lần thứ X, 8-9/12/2017)