



(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẢNG ĐỘC QUYỀN SÁNG CHẾ

(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN)
CỤC SỞ HỮU TRÍ TUỆ

(11)



1-0041847

(51)^{2022.01} G06F 30/25; G06N 3/00

(13) B

(21) 1-2023-07852

(22) 07/11/2023

(45) 25/11/2024 440

(43) 25/01/2024 430

(73) 1. Trường Đại học Bách Khoa - Đại học Quốc gia thành phố Hồ Chí Minh (VN)
268 Lý Thường Kiệt, Phường 14, Quận 10, Thành phố Hồ Chí Minh

2. Đại học Quốc gia Thành phố Hồ Chí Minh (VN)

Phường Linh Trung, thành phố Thủ Đức, Thành phố Hồ Chí Minh

(72) Hoàng Trang (VN).

(54) PHƯƠNG PHÁP, HỆ THỐNG MÁY TÍNH ĐỂ XÁC ĐỊNH THÔNG SỐ QUẢN TÍNH CỦA CÁC CÁ THỂ TRONG THUẬT TOÁN TỐI ƯU BẦY ĐÀN (PSO) ĐỂ TĂNG XÁC SUẤT TÌM ĐƯỢC GIẢI PHÁP TỐI ƯU TOÀN CỤC TRONG THIẾT KẾ VI MẠCH TƯƠNG TỰ

(57) Sáng chế đề xuất giải pháp xác định thông số quản tính của các cá thể trong thuật toán tối ưu bầy đàn, để tăng xác suất tìm được giải pháp tối ưu toàn cục trong thiết kế vi mạch tương tự. Giải pháp này tập trung vào hai nội dung chính: (1) kết hợp mô phỏng SPICE và thuật toán tối ưu bầy đàn (PSO) để lựa chọn kích thước phù hợp cho các linh kiện bên trong vi mạch tương tự, đảm bảo hiệu năng hoạt động của mạch, (2): chia quá trình tối ưu hóa thành ba giai đoạn, trong mỗi giai đoạn, thông số quản tính của thuật toán PSO được xác định bằng các biểu thức khác nhau. Quá trình thực hiện giải pháp này, về cơ bản, gồm 4 bước: (1): Vẽ sơ đồ nguyên lý của mạch cần tối ưu, xác định biến thiết kế, giới hạn trên, dưới của từng biến thiết kế, hàm mục tiêu, hàm ràng buộc, (2): Trong lõi tối ưu, khởi tạo quần thể gồm N cá thể, với tọa độ và vận tốc ngẫu nhiên, (3) Thực hiện vòng lặp để tìm nghiệm tối ưu cho đến khi thỏa mãn điều kiện dừng, (4): Trả kết quả tối ưu. Khi áp dụng vào giải quyết các vấn đề tối ưu hóa trong thiết kế vi mạch tương tự, giải pháp được đề xuất trong báo cáo này có tỉ lệ tìm thấy nghiệm tối ưu cao hơn so với các phương pháp cải thiện hiệu suất của thuật toán PSO được giới thiệu trước đó.

Lĩnh vực kỹ thuật được đề cập

Sáng chế này đề cập đến phương pháp áp dụng thuật toán tối ưu bầy đàn (thuật toán PSO) vào giải quyết bài toán tối ưu hoá trong thiết kế vi mạch tương tự, và phương pháp thiết lập giá trị quán tính của các cá thể trong thuật toán tối ưu bầy đàn tại từng vòng lặp. So với phương pháp thiết kế vi mạch dựa trên kinh nghiệm, phương pháp này cho kết quả tối ưu hơn về thời gian, chi phí và công sức. So với các phương pháp thiết lập giá trị quán tính cho thuật toán PSO trước đó, phương pháp được đề cập trong sáng chế này cho hiệu quả cao hơn về xác suất tìm thấy nghiệm tối ưu.

Tình trạng kỹ thuật của giải pháp

Cùng với sự phát triển của hệ thống mạng không dây, điện tử tiêu dùng, hệ thống tự động... là sự bùng nổ của thị trường vi mạch, trong đó vi mạch tương tự chiếm một phần quan trọng. Tuy nhiên, khác với thiết kế vi mạch số, việc thiết kế vi mạch tương tự, trong đó người kỹ sư điều chỉnh các thông số của transistor cho đến khi đạt được yêu cầu kỹ thuật, vẫn dựa phần lớn vào kiến thức và kinh nghiệm của người kỹ sư. Điều này khiến việc thiết kế vi mạch tương tự thường mất nhiều thời gian, một sơ suất nhỏ có thể ảnh hưởng đến toàn bộ quá trình thiết kế, trì hoãn thời gian đưa sản phẩm ra thị trường, dẫn đến thiệt hại lớn cho doanh nghiệp (theo một nghiên cứu, doanh nghiệp hoạt động trong lĩnh vực vi mạch có thể mất đến 14% thị phần nếu kế hoạch giới thiệu sản phẩm ra thị trường phải dời lại 4 tuần). Nhằm khắc phục các khó khăn gặp phải trong quá trình thiết kế IC tương tự, đẩy nhanh tốc độ thiết kế, nhiều tổ chức, doanh nghiệp đang đẩy mạnh nghiên cứu các phương pháp sử dụng các thuật toán tối ưu hóa để tự động hoá quá trình thiết kế vi mạch.

Bài toán tối ưu hoá trong vi mạch tương tự, nhìn chung, có thể được tổng quát hoá dưới dạng chính tắc như sau:

Tìm giá trị nhỏ nhất của hàm: $f_1(x), f_2(x), \dots, f_m(x)$

phụ thuộc vào: $c_i(x) < 0, \forall i \in \{1, 2, 3, \dots, N_C\}$,

Trong đó, $x \in R^d$ là những biến thiết kế cần được xác định ví dụ như width/length của transistor, và hàm $f_l(x) \forall l \in \{1, \dots, m\}$ là các hàm mô tả thông số ngõ ra của vi mạch.

Một cách tổng quát, các phương pháp đã được nghiên cứu nhằm giải quyết bài toán tối ưu hoá trong thiết kế vi mạch tương tự có thể phân loại vào ba nhóm: phương pháp dựa trên

phương trình (equation-based method), phương pháp dựa vào mô hình (surrogate model-based method) và phương pháp dựa vào mô phỏng (simulation-based method):

Phương pháp dựa trên phương trình (equation-based method): phương pháp này xây dựng các phương trình mô tả mối quan hệ giữa kích thước hình học của transistor (width/length) với các thông số xác định hiệu năng của mạch. Ưu điểm của phương pháp này là có thể cho kết quả nhanh chóng với chi phí tính toán thấp. Tuy nhiên, phương pháp này có tồn tại một số hạn chế. Thứ nhất, phương pháp này đòi hỏi phải xây dựng phương trình mô tả mối quan hệ giữa kích thước hình học của transistor (width/length) với các thông số xác định hiệu năng của mạch, một điều rất khó để làm được khi số lượng transistor lớn. Thứ hai, để tìm được nghiệm tối ưu, phương pháp dựa trên mô hình yêu cầu bài toán tối ưu phải là bài toán dạng lồi (convex optimization problem), trong đó tất cả các hàm mục tiêu và hàm ràng buộc phải là hàm lồi. Để khắc phục hai khó khăn trên, phương pháp xấp xỉ thường được áp dụng để đơn giản hoá quá trình lập phương trình và chuyển đổi các hàm không lồi về dạng hàm lồi. Tuy nhiên, việc xấp xỉ sẽ làm giảm độ chính xác của các phương trình, nhất là đối với các mạch phức tạp.

Phương pháp dựa trên mô hình (surrogate model-based method): phương pháp này sử dụng các thuật toán học máy như mạng neuron nhân tạo (artificial neural network - ANN), máy vector hỗ trợ (support vector machine – SVM), hồi quy quy trình Gauss (Gaussian process regression – GPR) ... để xây dựng mô hình dự đoán, với nguồn dữ liệu thu thập từ số lượng lớn mô phỏng SPICE. So sánh với phương pháp dựa trên phương trình, phương pháp này có một số ưu điểm. Phương pháp này cho kết quả chính xác hơn các kỹ thuật dựa trên phương trình truyền thống và có thể được tái sử dụng, giúp chúng hiệu quả hơn và tiết kiệm thời gian hơn. Tuy nhiên, cách tiếp cận này không phải là không có nhược điểm. Thứ nhất, để xây dựng một mô hình có chính xác cao, đặc biệt là trong không gian thiết kế lớn, việc tạo ra một lượng lớn dữ liệu từ mô phỏng SPICE là yêu cầu bắt buộc. Nhiệm vụ này đòi hỏi một lượng chi phí tính toán lớn. Hơn nữa, mô hình đã huấn luyện không thể áp dụng cho các thiết kế mạch sử dụng công nghệ vi mạch khác.

Phương pháp dựa trên mô phỏng (simulation-based method): phương pháp này coi vi mạch cần tối ưu như một hộp đen (không cần biết cấu trúc bên trong và phương trình mô tả quan hệ ngõ vào – ngõ ra), sử dụng trình mô phỏng SPICE và các thuật toán tối ưu hiện đại như kỹ thuật ủ mô phỏng (simulated annealing – SA), thuật toán di truyền (genetic algorithm - GA), thuật toán tiến hoá vi sai (differential evolution - DE), thuật toán tối ưu bầy đàn (particle swarm optimization - PSO)... để đưa ra phương án thiết kế tối ưu. Ưu điểm nổi trội của phương pháp này, so với phương pháp tối ưu dựa trên phương trình và dựa trên mô hình, là khả năng tìm ra thiết kế tối ưu toàn cục một cách nhanh chóng với độ chính xác cao, chi phí tính toán thấp, không yêu cầu hàm mô tả toán học hay yêu cầu bài toán tối ưu phải có dạng lồi. Một trong các

thuật toán tiêu biểu thuộc loại này là thuật toán tối ưu bầy đàn (PSO). Thuật toán này được ứng dụng rất rộng rãi trong việc giải quyết các bài toán tối ưu trong lĩnh vực kỹ thuật, nhờ vào những ưu điểm vượt trội như cách thức vận hành đơn giản, dễ lập trình, khả năng thực hiện số lượng lớn tính toán song song, tốc độ hội tụ nhanh, số lượng thông số cần thiết lập nhỏ... Tuy nhiên, thuật toán PSO không phải không có nhược điểm. Nhược điểm chính của thuật toán này là khi gặp phải những bài toán tối ưu phức tạp với nhiều cực trị, thuật toán có thể bị mắc kẹt tại vùng tối ưu địa phương, không tìm thấy nghiệm tối ưu toàn cục. Để khắc phục nhược điểm đó, nhiều phương pháp đã được giới thiệu để tăng cường hiệu năng của thuật toán PSO, như là kết hợp PSO với thuật toán di truyền hoặc thuật toán tìm kiếm hấp dẫn (gravitational search algorithm). Tuy nhiên, việc kết hợp hai thuật toán sẽ làm tăng số lượng tham số phải điều chỉnh, phức tạp hoá quá trình tính toán. Do đó, thay vì kết hợp hai thuật toán, một số nghiên cứu đề xuất các chiến lược mới điều chỉnh các thông số có sẵn bên trong thuật toán, vừa giúp tăng hiệu suất tìm kiếm của thuật toán và vừa không làm tăng độ phức tạp. Trong thuật toán PSO, thông số quán tính ω giữ vai trò quan trọng nhất. Theo một số nghiên cứu, thông số ω quy định hành vi tìm kiếm toàn cục hay cục bộ của thuật toán PSO: thuật toán có xu hướng chú trọng vào tìm kiếm toàn cục khi ω có giá trị lớn, và thuật toán có xu hướng chú trọng vào tìm kiếm cục bộ khi ω có giá trị nhỏ, và việc giảm dần thông số quán tính qua từng vòng lặp giúp tăng cường hiệu suất tìm kiếm của thuật toán. Khi áp dụng thuật toán PSO vào giải quyết các bài toán kỹ thuật, một số nghiên cứu đề xuất phương pháp giảm thông số quán tính qua từng vòng lặp một cách tuyến tính. Tuy nhiên, khi áp dụng phương pháp giảm tuyến tính này vào bài toán tối ưu vi mạch tương tự, phương pháp này cho hiệu quả không cao. Nhận thấy sự quan trọng của thông số quán tính ω và những hạn chế của các công trình trước đó, trong công trình này, bên cạnh việc giới thiệu quy trình sử dụng thuật toán PSO vào thiết kế vi mạch tương tự, sáng chế giới thiệu một phương pháp mới nhằm xác định giá trị của thông số ω qua từng vòng lặp, giúp nâng cao hiệu quả tính toán.

Quy trình thiết kế vi mạch tương tự có thể được tổng quát hoá thành ba bước như sau:

- (1) Chọn sơ đồ cấu trúc (topology) cho mạch.
- (2) Tối ưu các thông số thiết kế như: thông số kích thước cho những linh kiện có trong mạch bao gồm: width/length của transistor, giá trị điện dung của tụ điện, giá trị điện trở có trong mạch.
- (3) Thiết kế Layout cho mạch.

Rất nhiều đề tài nghiên cứu đã được thực hiện nhằm vào quy trình tối ưu hóa thông số của những thành phần trong mạch bởi vì những thông số này ảnh hưởng lớn đến hoạt động và công suất tiêu thụ của mạch. Bài toán tối ưu hoá trong vi mạch tương tự, nhìn chung, có thể được tổng quát hoá như sau:

Tìm giá trị nhỏ nhất của hàm: $f_1(x), f_2(x), \dots, f_m(x)$

phụ thuộc vào: $c_i(x) < 0, \forall i \in \{1, 2, 3, \dots, N_C\}$,

Trong đó, $x \in R^d$ là những biến thiết kế cần được xác định ví dụ như width/length của transistor, và hàm $f_l(x) \forall l \in \{1, \dots, m\}$ là các hàm mô tả thông số ngõ ra của vi mạch.

Nhìn chung, các phương pháp tối ưu hoá đã được nghiên cứu có thể phân loại vào ba nhóm: phương pháp dựa trên phương trình, phương pháp dựa vào mô hình, và phương pháp dựa vào mô phỏng. Phương pháp dựa vào mô phỏng được ưa chuộng hơn cả do những ưu điểm vượt trội như chi phí tính toán thấp và khả năng giải quyết những bài toán phức tạp.

Phương pháp dựa vào mô phỏng để giải quyết bài toán tối ưu hoá trong thiết kế vi mạch tương tự có thể được tổng quát hoá như Hình 1. Hình 1 gồm 2 khối chính: khối “sơ đồ nguyên lý” và khối “lỗi tối ưu”. Khối “sơ đồ nguyên lý” bao gồm bản vẽ nguyên lý của mạch với các thông số hình học W/L của các linh kiện được đặt dưới dạng biến và trình mô phỏng SPICE (được sử dụng để mô phỏng hiệu năng của mạch). Khối “lỗi tối ưu” là loạt tác vụ thực hiện các thuật toán tối ưu (ví dụ: thuật toán di truyền, thuật toán tối ưu bầy đàn, thuật toán tiến hoá vi phân...). Đóng vai trò cầu nối giữa hai khối này là một phần mềm script viết bằng ngôn ngữ lập trình Ocean (phần mềm Ocean). Phần mềm này nhận giá trị của các biến tối ưu từ lỗi tối ưu, gán các giá trị này vào các biến thiết kế trong trình mô phỏng, kích hoạt trình mô phỏng thực hiện mô phỏng. Sau khi trình mô phỏng chạy xong, phần mềm Ocean thu thập kết quả mô phỏng từ trình mô phỏng và trả kết quả này về khối lỗi tối ưu. Quá trình này lặp đi lặp lại cho đến khi điều kiện dừng thoả mãn.

Trong nhóm các thuật toán tối ưu, thuật toán tối ưu bầy đàn (PSO) được đánh giá cao hơn cả, nhờ vào những ưu điểm vượt trội như cách thức vận hành đơn giản, dễ lập trình, khả năng thực hiện số lượng lớn tính toán song song, tốc độ hội tụ nhanh, số lượng thông số cần thiết lập nhỏ. Trong thuật toán PSO, mỗi giá trị của biến thiết kế được mô hình hoá thành toạ độ của cá thể, và cách thức thay đổi giá trị của biến thiết kế sau mỗi vòng lặp được mô hình hoá bằng khái niệm “vận tốc của cá thể”.

Vận tốc và toạ độ của cá thể thứ i , tại vòng lặp thứ $t + 1$ được xác định bằng công thức:

$$V_i^{t+1} = \underbrace{\omega V_i^t}_{\text{quán tính}} + \underbrace{c_1 r_1 (P_{\text{best}_i^t} - P_i^t)}_{\text{kinh nghiệm}} + \underbrace{c_2 r_2 (G_{\text{best}}^t - P_i^t)}_{\text{tập thể}} \quad (1)$$

$$P_i^{t+1} = P_i^t + V_i^{t+1} \quad (2)$$

Ta thấy, trong thuật toán PSO, vận tốc của một hạt phụ thuộc vào ba yếu tố: quán tính (hạt có xu hướng giữ nguyên vận tốc), kinh nghiệm (hạt có xu hướng di chuyển tới vị trí tối ưu mà nó đã tìm thấy trước đó), và tập thể (hạt có xu hướng di chuyển tới vị trí tối ưu nhất cả đàn đã tìm thấy trước đó). Ảnh hưởng của ba yếu tố này lần lượt được quy định bởi ba thông số ω , c_1 và c_2 . Trong ba thông số này, thông số quán tính ω giữ vai trò quan trọng nhất. Theo một

số nghiên cứu, thông số ω quy định hành vi tìm kiếm toàn cục hay cục bộ của thuật toán PSO: thuật toán có xu hướng chú trọng vào tìm kiếm toàn cục khi ω có giá trị lớn, và thuật toán có xu hướng chú trọng vào tìm kiếm cục bộ khi ω có giá trị nhỏ, và việc giảm dần thông số quán tính qua từng vòng lặp giúp tăng cường hiệu suất tìm kiếm của thuật toán. Khi áp dụng thuật toán PSO vào giải quyết các bài toán tối ưu hoá trong thiết kế vi mạch tương tự, nghiên cứu “Area Optimisation of Two Stage Miller Compensated Op-Amp in 65 nm Using Hybrid PSO” đề xuất phương pháp giảm thông số quán tính qua từng vòng lặp một cách tuyến tính. Tuy nhiên, phương pháp này cho hiệu quả không cao.

Bản chất kỹ thuật của sáng chế

Mục đích của sáng chế là đề xuất giải pháp xác định thông số quán tính của các cá thể trong thuật toán tối ưu bầy đàn để tăng xác suất tìm được giải pháp tối ưu toàn cục trong thiết kế vi mạch tương tự, và khắc phục được nhược điểm của các giải pháp đã biết.

Để khắc phục nhược điểm đó, sáng chế đề xuất phương pháp xác định giá trị thông số quán tính của các cá thể trong thuật toán tối ưu bầy đàn (PSO) theo giai đoạn như sau:

$$\omega(t) = \begin{cases} \omega_{\max}, & 0 \leq t \leq \frac{3}{5}T \\ \omega_{\max} - \frac{1}{2} \frac{\omega_{\max}}{T} \left(t - \frac{3}{5}T \right), & \frac{3}{5}T \leq t \leq \frac{4}{5}T \\ 0.9\omega_{\max} - \frac{0.9 \times \omega_{\max} - \omega_{\min}}{T} (5t - 4T), & \frac{4}{5}T \leq t \leq T \end{cases} \quad (3)$$

Trong đó: $\omega_{\max}$ và $\omega_{\min}$ lần lượt là giá trị cực đại và cực tiểu của thông số quán tính, t là vòng lặp đang xét, T là tổng số vòng lặp ($0 \leq t \leq T$).

Theo một phương án ưu tiên, sáng chế đề xuất phương pháp xác định thông số quán tính của các cá thể trong thuật toán tối ưu bầy đàn, để tăng xác suất tìm được giải pháp tối ưu toàn cục trong thiết kế vi mạch tương tự, phương pháp này được thực hiện bằng phần mềm chạy trên máy tính, phần mềm này bao gồm các khối cơ bản sau:

- khối sơ đồ nguyên lý: gồm sơ đồ nguyên lý, netlist của mạch cần tối ưu với các thông số hình học W/L của các linh kiện được đặt dưới dạng biến và chương trình mô phỏng hiệu năng của mạch SPICE (trình mô phỏng),
- khối lỗi tối ưu: được xây dựng dựa trên thuật toán tối ưu bầy đàn, với phương pháp xác định thông số quán tính theo từng giai đoạn được mô tả trong bước 3,
- phần mềm script viết bằng ngôn ngữ lập trình Ocean (phần mềm Ocean) đóng vai trò cầu nối giữa hai khối trên, phần mềm này nhận giá trị của các biến tối ưu từ lỗi tối ưu, gán các giá trị này vào các biến thiết kế trong trình mô phỏng, kích hoạt trình mô phỏng thực hiện mô phỏng, sau khi trình mô phỏng chạy xong, phần mềm Ocean thu thập kết quả mô phỏng từ trình

mô phỏng và trả kết quả này về khối lõi tối ưu, quá trình này lặp đi lặp lại cho đến khi điều kiện dừng thoả mãn;

quá trình thực hiện phương pháp xác định thông số bao gồm các bước cơ bản sau:

bước 1: vẽ sơ đồ nguyên lý của mạch cần tối ưu, xác định biến thiết kế, giới hạn trên, dưới của từng biến thiết kế, hàm mục tiêu, hàm ràng buộc;

bước 2: trong lõi tối ưu, khởi tạo quần thể gồm n cá thể, với toạ độ và vận tốc ngẫu nhiên;

bước 3: thực hiện vòng lặp gồm các bước sau, cho đến khi thoả mãn điều kiện dừng:

bước 3.1: gán giá trị toạ độ của từng hạt với các biến thiết kế tương ứng trong trình mô phỏng qua một phần mềm viết bằng ngôn ngữ Ocean script (phần mềm Ocean),

bước 3.2: thực hiện, bởi trình mô phỏng chạy mô phỏng hiệu năng của mạch và trả kết quả mô phỏng về phần mềm Ocean,

bước 3.3: trả kết quả, bởi phần mềm Ocean, về lõi tối ưu, lõi tối ưu thực hiện tính toán giá trị fitness tương ứng,

bước 3.4: cập nhật, bởi lõi tối ưu, các giá trị Pbest (là toạ độ mà tại đó cá thể ghi nhận giá trị fitness tốt nhất) và Gbest (là toạ độ mà tại đó quần thể ghi nhận giá trị fitness tốt nhất), giá trị Pbest của hạt thứ i và giá trị Gbest tại vòng lặp t được xác định như sau:

$$Pbest_i^t = (fitness_i^t < Pbest_i) * fitness_i^t + (fitness_i^t > Pbest_i) * Pbest_i$$

$$G_{best}^t = \min_{1 \leq i \leq N} (Pbest_i^t)$$

với $fitness_i^t$ là giá trị fitness của hạt tại vòng lặp hiện tại, $Pbest_i$ là giá trị Pbest đã được ghi nhận trong các vòng lặp trước đó,

bước 3.5: đánh giá điều kiện dừng, bởi lõi tối ưu, nếu điều kiện dừng được thoả mãn, chuyển đến bước 4,

bước 3.6: tính toán giá trị của thông số quán tính theo công thức:

$$\omega(t) = \begin{cases} \omega_{max}, & 0 \leq t \leq \frac{3}{5}T \\ \omega_{max} - \frac{1}{2} \frac{\omega_{max}}{T} \left(t - \frac{3}{5}T \right), & \frac{3}{5}T \leq t \leq \frac{4}{5}T \\ 0.9\omega_{max} - \frac{0.9 \times \omega_{max} - \omega_{min}}{T} (5t - 4T), & \frac{4}{5}T \leq t \leq T \end{cases}$$

bước 3.7 cập nhật, bởi lõi tối ưu, vận tốc và vị trí của từng hạt theo các công thức:

$$V_i^{t+1} = \underbrace{\omega V_i^t}_{\text{quán tính}} + \underbrace{c_1 r_1 (Pbest_i^t - P_i^t)}_{\text{kinh nghiệm}} + \underbrace{c_2 r_2 (G_{best}^t - P_i^t)}_{\text{tập thể}}$$

$$P_i^{t+1} = P_i^t + V_i^{t+1};$$

bước 4: trả kết quả, bởi lõi tối ưu, giá trị tối ưu của biến thiết kế tương ứng với toạ độ Pbest, giá trị hiệu năng tối ưu tương ứng với giá trị fitness gắn với toạ độ Pbest.

Theo một khía cạnh của phương án ưu tiên nêu trên, sáng chế còn đề xuất một hệ thống máy tính để thiết kế vi mạch tương tự, trong đó, hệ thống máy tính này được cài đặt phần mềm chứa các câu lệnh và được tạo cấu hình để thực hiện phương pháp nêu trên.

Sáng chế đề xuất cách xác định thông số quán tính ω theo giai đoạn vì lí do sau: giá trị của ω đặc định cực kỳ quan trọng đối với khả năng của thuật toán PSO trong việc tìm kiếm cục bộ hoặc toàn cục. Theo nghiên cứu “Particle swarm optimization algorithm: an overview” của Dongshu Wang, Dapei Tan và Lei Liu, khi ω lớn, thuật toán có xu hướng tìm kiếm toàn cục, và khi ω nhỏ, thuật toán chú trọng vào tìm kiếm cục bộ. Trong những vòng lặp đầu $(0 \leq t \leq \frac{3}{5}T)$, để tăng khả năng phát hiện vùng có tiềm năng cực trị, sáng chế đề xuất giữ ω ở mức cao. Ngược lại, khi thuật toán đi sâu hơn vào quá trình tìm kiếm nghiệm tối ưu, sáng chế đề xuất giảm dần giá trị của ω . Tuy nhiên, vì chúng ta không biết rõ về độ rộng của vùng có nghiệm tối ưu so với toàn bộ không gian thiết kế, việc giảm ω đột ngột là không phù hợp, có thể làm thuật toán khó tìm được nghiệm tối ưu. Do đó, sáng chế đã thiết kế một hàm để điều chỉnh giá trị ω , được mô tả cụ thể bởi công thức (3) và minh hoạ bởi Hình 2, bắt đầu với mức cao ở các vòng lặp đầu $(0 \leq t \leq \frac{3}{5}T)$, giảm nhẹ ở những vòng lặp tiếp theo $(\frac{3}{5}T \leq t \leq \frac{4}{5}T)$, và giảm mạnh ở những vòng lặp cuối cùng $(\frac{4}{5}T \leq t \leq T)$. Hiệu quả của vượt trội phương pháp đề xuất, so với các phương pháp đã có trước đó, được thể hiện trong phần “Ví dụ về thực hiện phương pháp”.

Tóm lại, bản chất kỹ thuật của giải pháp này có thể tóm tắt ở hai điểm chính sau:

- (1): Kết hợp mô phỏng SPICE và thuật toán tối ưu bầy đàn (PSO) để chọn kích thước phù hợp cho các linh kiện bên trong vi mạch tương tự, đảm bảo hiệu năng hoạt động của mạch.
- (2): Quá trình tìm nghiệm tối ưu được chia thành 3 giai đoạn, mỗi giai đoạn có một cách xác định thông số quán tính khác nhau.

Mô tả vắn tắt các hình vẽ

Hình 1: Lưu đồ tổng quát hoá phương pháp tối ưu hoá trong thiết kế vi mạch tương tự dựa trên mô phỏng.

Hình 2: Minh hoạ giá trị của thông số quán tính, trong trường hợp $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$, $T = 100$.

Hình 3: Lưu đồ mô tả quá trình tối ưu hoá trong thiết kế vi mạch tương tự dựa trên thuật toán tối ưu bầy đàn.

Mô tả chi tiết sáng chế

Sau đây, sáng chế sẽ được mô tả chi tiết theo các phương án ưu tiên thực hiện sáng chế và dựa vào các hình vẽ minh họa kèm theo. Các hình vẽ và nội dung giải thích nhằm làm hiểu rõ hơn giải pháp theo sáng chế, chúng không nhằm giới hạn phạm vi bộc lộ của sáng chế.

Từ phương pháp tổng quát ứng dụng thuật toán tối ưu vào thiết kế vi mạch tương tự mô tả trong hình 1, kết hợp với thuật toán tối ưu bầy đàn và phương pháp xác định giá trị thông số quán tính đã được giới thiệu trong công thức (2), sáng chế đề xuất một phương pháp ứng dụng thuật toán PSO và phương pháp xác định giá trị thông số quán tính theo giai đoạn, bao gồm các bước cơ bản sau:

bước 1: vẽ sơ đồ nguyên lý của mạch cần tối ưu, xác định biến thiết kế, giới hạn trên, dưới của từng biến thiết kế, hàm mục tiêu, hàm ràng buộc;

bước 2: trong lõi tối ưu, khởi tạo quần thể gồm n cá thể, với tọa độ và vận tốc ngẫu nhiên;

bước 3: thực hiện vòng lặp gồm các bước sau, cho đến khi thỏa mãn điều kiện dừng:

bước 3.1: gán giá trị tọa độ của từng hạt với các biến thiết kế tương ứng trong trình mô phỏng qua một phần mềm viết bằng ngôn ngữ Ocean script (phần mềm Ocean),

bước 3.2: thực hiện, bởi trình mô phỏng chạy mô phỏng hiệu năng của mạch và trả kết quả mô phỏng về phần mềm Ocean,

bước 3.3: trả kết quả, bởi phần mềm Ocean, về lõi tối ưu, lõi tối ưu thực hiện tính toán giá trị fitness tương ứng,

bước 3.4: cập nhật, bởi lõi tối ưu, các giá trị Pbest (là tọa độ mà tại đó cá thể ghi nhận giá trị fitness tốt nhất) và Gbest (là tọa độ mà tại đó quần thể ghi nhận giá trị fitness tốt nhất), giá trị Pbest của hạt thứ i và giá trị Gbest tại vòng lặp t được xác định như sau:

$$Pbest_i^t = (fitness_i^t < Pbest_i) * fitness_i^t + (fitness_i^t > Pbest_i) * Pbest_i$$

$$G_{best}^t = \min_{1 \leq i \leq N} (Pbest_i^t)$$

với $fitness_i^t$ là giá trị fitness của hạt tại vòng lặp hiện tại, $Pbest_i$ là giá trị Pbest đã được ghi nhận trong các vòng lặp trước đó,

bước 3.5: đánh giá điều kiện dừng, bởi lõi tối ưu, nếu điều kiện dừng được thỏa mãn, chuyển đến bước 4,

bước 3.6: tính toán giá trị của thông số quán tính theo công thức:

$$\omega(t) = \begin{cases} \omega_{max}, & 0 \leq t \leq \frac{3}{5}T \\ \omega_{max} - \frac{1}{2} \frac{\omega_{max}}{T} \left(t - \frac{3}{5}T \right), & \frac{3}{5}T \leq t \leq \frac{4}{5}T \\ 0.9\omega_{max} - \frac{0.9 \times \omega_{max} - \omega_{min}}{T} (5t - 4T), & \frac{4}{5}T \leq t \leq T \end{cases}$$

bước 3.7 cập nhật, bởi lõi tối ưu, vận tốc và vị trí của từng hạt theo các công thức:

$$V_i^{t+1} = \underbrace{\omega V_i^t}_{\text{quán tính}} + \underbrace{c_1 r_1 (Pbest_i^t - P_i^t)}_{\text{kinh nghiệm}} + \underbrace{c_2 r_2 (G_{best}^t - P_i^t)}_{\text{tập thể}}$$

$$P_i^{t+1} = P_i^t + V_i^{t+1};$$

bước 4: trả kết quả, bởi lỗi tối ưu, giá trị tối ưu của biến thiết kế tương ứng với tọa độ P_{best} , giá trị hiệu năng tối ưu tương ứng với giá trị fitness gắn với tọa độ P_{best} . Quy trình vừa được mô tả được tóm tắt trong Hình 3.

Ví dụ thực hiện sáng chế

Sau đây là ví dụ về việc sử dụng thuật toán PSO và phương pháp xác định giá trị thông số quán tính theo giai đoạn theo sáng chế để tối ưu hoá thông số PSRR của mạch Bandgap Reference. PSRR (power supply rejection ratio) là tỉ lệ giữa gợn sóng (ripple) của điện áp tại nguồn so với gợn sóng của điện áp tại ngõ ra của mạch. Tín hiệu gợn sóng gây nên nhiều tác hại không mong muốn trong các mạch điện tử, làm giảm hiệu quả hoạt động của mạch. Vì thế, PSRR là một thông số đặc biệt quan trọng cần phải xét đến khi thiết kế vi mạch tương tự. PSRR càng cao thì khả năng khử gợn sóng càng tốt.

Bài toán tối ưu hoá thông số PSRR của mạch Bandgap Reference có thể được mô tả dưới dạng chính tắc như Bài toán 1 sau:

Bài toán 1: Bài toán tối ưu hoá thông số PSRR của mạch Bandgap Reference

Tìm giá trị lớn nhất của: $PSRR(x)$

phụ thuộc vào: hệ số nhiệt độ $\leq 8 \text{ ppm/}^\circ\text{C}$,
 798 mV $\leq$ điện áp ngõ ra $\leq$ 802 mV,
 công suất tiêu thụ $\leq 1 \text{ mW}$,
 phase margin $\geq 60^\circ$.

Trong đó, x là biến thiết kế, tập hợp các giá trị width/length của các transistor có trong mạch.

Để thuận lợi cho quá trình áp dụng thuật toán PSO vào giải quyết bài toán tối ưu hoá, hàm mục tiêu (PSRR) và các điều kiện ràng buộc (hệ số nhiệt độ, điện áp ngõ ra, công suất tiêu thụ, phase margin) được kết hợp vào một hàm mục tiêu duy nhất: hàm fitness. Hàm fitness được xác định bởi công thức (4) như sau:

$$\text{Fitness} = -\text{PSRR} + 1000 \times \text{penalty} \quad (4)$$

với penalty sẽ tăng 1 đơn vị khi mỗi khi điều kiện ràng buộc của mạch Bandgap Reference bị vi phạm, và bằng 0 khi tất cả các ràng buộc thiết kế được thoả mãn. Thông thường, thông số PSRR có giá trị trong khoảng từ 20 đến 150 dB. Theo công thức (4), hàm fitness sẽ mang giá trị dương khi bất cứ ràng buộc thiết kế nào bị vi phạm, và có giá trị âm khi tất cả các ràng buộc thiết kế được thoả mãn.

Theo công thức (4), hàm fitness sẽ đạt giá trị nhỏ nhất khi PSRR đạt giá trị lớn nhất và tất cả các điều kiện ràng buộc được thoả mãn. Bài toán 1 được rút gọn thành bài toán tìm giá trị nhỏ nhất cho hàm fitness.

Bài toán 1, sau khi đã được biến đổi thành bài toán tìm giá trị nhỏ nhất cho hàm fitness, được sử dụng để so sánh giữa phương pháp đề xuất với các phương pháp truyền thống như phương pháp giữ nguyên giá trị ω , phương pháp giảm dần ω theo hàm tuyến tính (gọi tắt là phương pháp PSO-LD), và phương pháp kết hợp giữa thuật toán PSO với thuật toán tìm kiếm hấp dẫn (gọi tắt là phương pháp GPSOA). Kết quả cho thấy phương pháp được đề xuất trong sáng chế này có tỉ lệ thành công vượt trội (một lần thực thi được xác định là thành công khi hàm fitness có giá trị âm, tương đương với việc tất cả các ràng buộc thiết kế được thoả mãn).

Bảng 1. So sánh các giá trị fitness đạt được sau 10 lần thực thi giữa phương pháp đề xuất với các phương pháp truyền thống, trong bài toán tối ưu mạch Bandgap Reference

Lần thực thi	Phương pháp truyền thống	Phương pháp PSO-LD	Phương pháp GPSOA	Phương pháp đề xuất
1	-99,8006	-98,3181	901,7207	-99,009
2	903,6636	905,8816	-79,4015	-99,9896
3	904,2904	-98,8251	901,0472	899,6047
4	905,5045	906,551	901,1983	904,9483
5	902,9503	-98,4502	-99,0421	-99,9868
6	900,6525	-99,7719	900,0975	-99,9098
7	904,2006	901,0167	906,2871	900,7729
8	-98,7331	900,0905	-98,7723	-97,5106
9	906,2794	904,5398	900,5086	-99,8476
10	-99,677	906,3641	910,0853	-99,1143
Tỉ lệ thành công	3/10	4/10	3/10	7/10
Giá trị fitness tốt nhất đạt được	-99,8006	-99,7719	-99,0421	-99,9896
Giá trị fitness trung bình (cho các lần thực thi thành công)	-99,4036	-98,8413	-92,4053	-99,3382
Độ lệch chuẩn (cho các lần thực thi thành công)	0,5839	0,6565	9,1957	0,9064

Ngoài ra, phương pháp được đề xuất có tiềm năng rất lớn để giải quyết các bài toán tối ưu khác gặp phải trong thiết kế vi mạch tương tự, chẳng hạn như bài toán tối ưu hoá diện tích và độ lợi của vi mạch khuếch đại hai tầng, hay bài toán giảm thiểu dòng điện tĩnh của mạch tích hợp nguồn tuyến tính (low dropout regulator).

Hiệu quả đạt được của sáng chế

Phương pháp được đề cập đến trong sáng chế này có những lợi ích như:

- Tiết kiệm hơn về thời gian, chi phí và công sức so với việc thiết kế vi mạch tương tự dựa trên kinh nghiệm.
- Xét về mặt hiệu năng tính toán, phương pháp này hứa hẹn khả năng tìm thấy nghiệm tối ưu toàn cục cho các bài toán phức tạp cao hơn so với các phương pháp xác định giá trị thông số quán tính truyền thống (như giữ nguyên giá trị thông số quán tính, hoặc giảm tuyến tính giá trị thông số này qua suốt quá trình thực thi thuật toán).
- Có khả năng áp dụng rộng rãi trong thiết kế vi mạch tương tự, tự động hoá quá trình thiết kế các mạch phức tạp, có số lượng biến thiết kế lớn.

YÊU CẦU BẢO HỘ

1. Phương pháp xác định thông số quán tính của các cá thể trong thuật toán tối ưu bầy đàn, để tăng xác suất tìm được giải pháp tối ưu toàn cục trong thiết kế vi mạch tương tự, phương pháp này được thực hiện bằng phần mềm chạy trên máy tính, phần mềm này bao gồm các khối cơ bản sau:

- (a) khối sơ đồ nguyên lý: gồm sơ đồ nguyên lý, netlist của mạch cần tối ưu với các thông số hình học W/L của các linh kiện được đặt dưới dạng biến và chương trình mô phỏng hiệu năng của mạch SPICE (trình mô phỏng),
- (b) khối lõi tối ưu: được xây dựng dựa trên thuật toán tối ưu bầy đàn, với phương pháp xác định thông số quán tính theo từng giai đoạn được mô tả trong bước 3,
- (c) phần mềm script viết bằng ngôn ngữ lập trình Ocean (phần mềm Ocean) đóng vai trò cầu nối giữa hai khối trên, phần mềm này nhận giá trị của các biến tối ưu từ lõi tối ưu, gán các giá trị này vào các biến thiết kế trong trình mô phỏng, kích hoạt trình mô phỏng thực hiện mô phỏng, sau khi trình mô phỏng chạy xong, phần mềm Ocean thu thập kết quả mô phỏng từ trình mô phỏng và trả kết quả này về khối lõi tối ưu, quá trình này lặp đi lặp lại cho đến khi điều kiện dừng thỏa mãn;

quá trình thực hiện phương pháp bao gồm các bước cơ bản sau:

bước 1: vẽ sơ đồ nguyên lý của mạch cần tối ưu, xác định biến thiết kế, giới hạn trên, dưới của từng biến thiết kế, hàm mục tiêu, hàm ràng buộc;

bước 2: trong lõi tối ưu, khởi tạo quần thể gồm n cá thể, với tọa độ và vận tốc ngẫu nhiên;

bước 3: thực hiện vòng lặp gồm các bước sau, cho đến khi thỏa mãn điều kiện dừng:

bước 3.1: gán giá trị tọa độ của từng hạt với các biến thiết kế tương ứng trong trình mô phỏng qua một phần mềm viết bằng ngôn ngữ Ocean script (phần mềm Ocean),

bước 3.2: thực hiện, bởi trình mô phỏng chạy mô phỏng hiệu năng của mạch và trả kết quả mô phỏng về phần mềm Ocean,

bước 3.3: trả kết quả, bởi phần mềm Ocean, về lõi tối ưu, lõi tối ưu thực hiện tính toán giá trị fitness tương ứng,

bước 3.4: cập nhật, bởi lõi tối ưu, các giá trị Pbest (là tọa độ mà tại đó cá thể ghi nhận giá trị fitness tốt nhất) và Gbest (là tọa độ mà tại đó quần thể ghi nhận giá trị fitness tốt nhất), giá trị Pbest của hạt thứ i và giá trị Gbest tại vòng lặp t được xác định như sau:

$$Pbest_i^t = (fitness_i^t < Pbest_i) * fitness_i^t + (fitness_i^t > Pbest_i) * Pbest_i$$

$$G_{best}^t = \min_{1 \leq i \leq N} (Pbest_i^t)$$

với $fitness_i^t$ là giá trị fitness của hạt tại vòng lặp hiện tại, $Pbest_i$ là giá trị Pbest đã được ghi nhận trong các vòng lặp trước đó,

bước 3.5: đánh giá điều kiện dừng, bởi lỗi tối ưu, nếu điều kiện dừng được thoả mãn, chuyển đến bước 4,

bước 3.6: tính toán giá trị của thông số quán tính theo công thức:

$$\omega(t) = \begin{cases} \omega_{\max}, & 0 \leq t \leq \frac{3}{5}T \\ \omega_{\max} - \frac{1}{2} \frac{\omega_{\max}}{T} \left(t - \frac{3}{5}T \right), & \frac{3}{5}T \leq t \leq \frac{4}{5}T \\ 0.9\omega_{\max} - \frac{0.9 \times \omega_{\max} - \omega_{\min}}{T} (5t - 4T), & \frac{4}{5}T \leq t \leq T \end{cases}$$

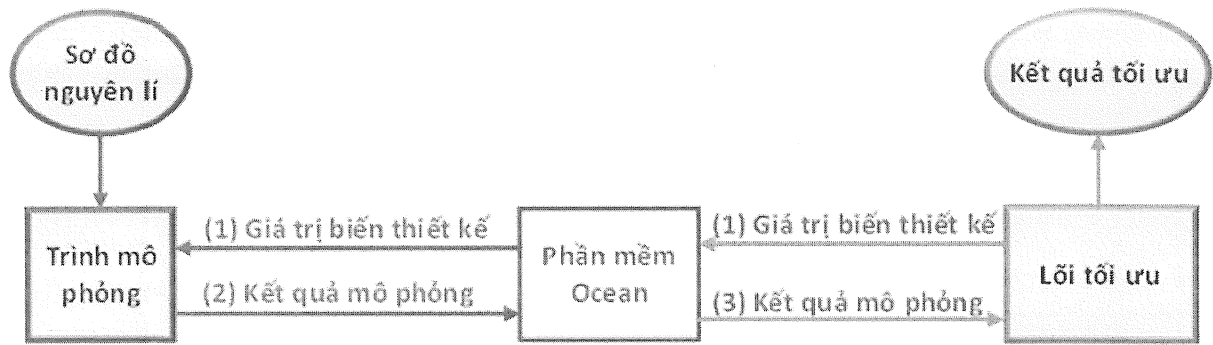
bước 3.7 cập nhật, bởi lỗi tối ưu, vận tốc và vị trí của từng hạt theo các công thức:

$$V_i^{t+1} = \underbrace{\omega V_i^t}_{\text{quán tính}} + \underbrace{c_1 r_1 (Pbest_i^t - P_i^t)}_{\text{kinh nghiệm}} + \underbrace{c_2 r_2 (G_{\text{best}}^t - P_i^t)}_{\text{tập thể}}$$

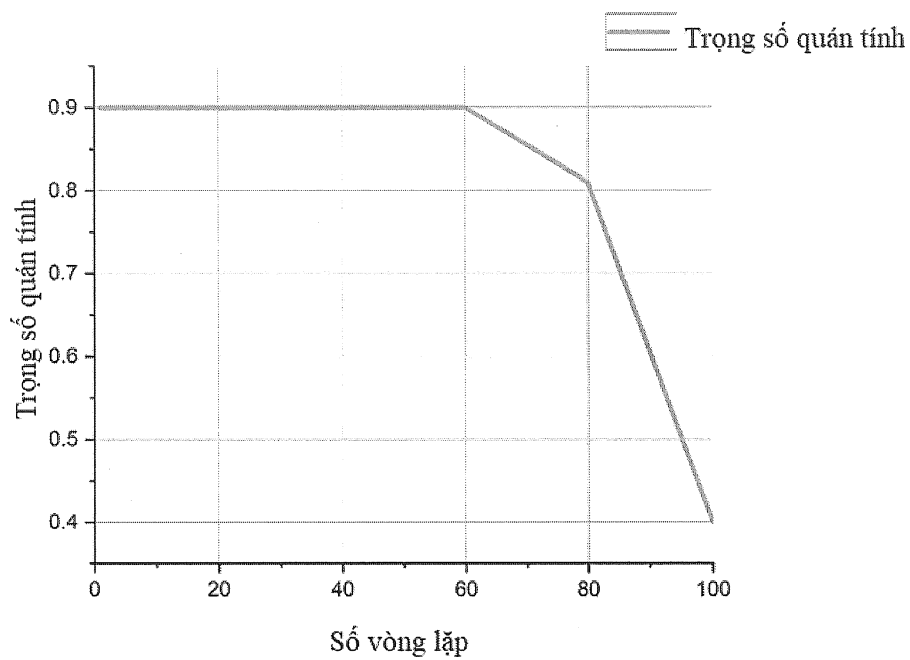
$$P_i^{t+1} = P_i^t + V_i^{t+1};$$

bước 4: trả kết quả, bởi lỗi tối ưu, giá trị tối ưu của biến thiết kế tương ứng với tọa độ Pbest, giá trị hiệu năng tối ưu tương ứng với giá trị fitness gắn với tọa độ Pbest.

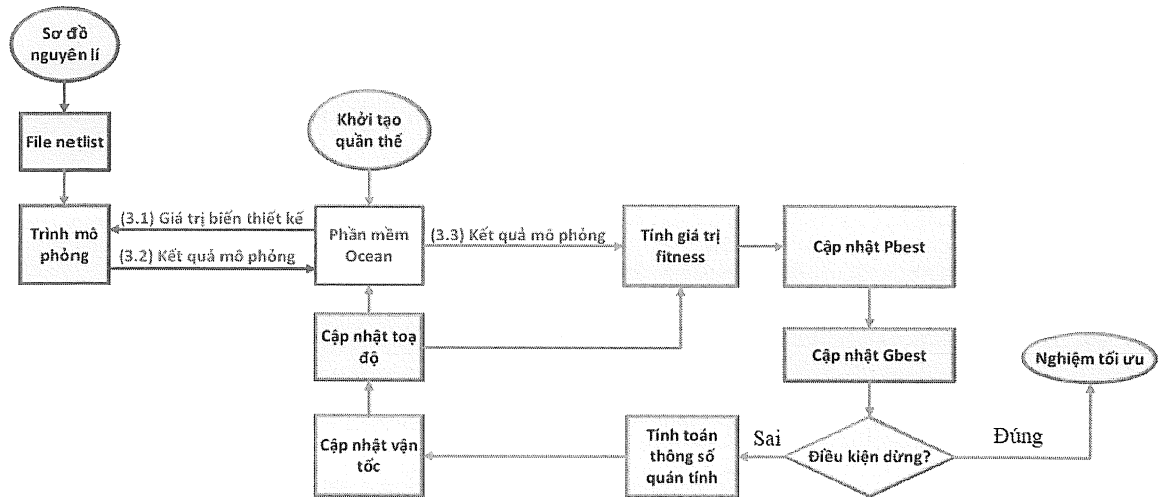
2. Hệ thống máy tính để thiết kế vi mạch tương tự, trong đó, hệ thống máy tính này được cài đặt phần mềm chứa các câu lệnh và được tạo cấu hình để thực hiện phương pháp theo điểm 1.



Hình 1



Hình 2



Hình 3