



**UNIVERSITATEA NAȚIONALĂ
DE ȘTIINȚĂ ȘI TEHNOLOGIE
POLITEHNICA BUCUREȘTI**



**Școala Doctorală de Electronică, Telecomunicații și
Tehnologia Informației**

Decizie nr. 182 din 13-09-2024

**REZUMAT TEZĂ
DE DOCTORAT**

Cătălin-Laurențiu VIȘAN

**TEHNICI DE INTELIGENȚĂ ARTIFICIALĂ PENTRU
AUTOMATIZAREA PROCESULUI DE DIMENSIONARE A
CIRCUITELOR INTEGRATE**

**ARTIFICIAL INTELLIGENCE TECHNIQUES FOR INTEGRATED
CIRCUIT DESIGN AUTOMATION**

COMISIA DE DOCTORAT

Prof. Dr. Ing. Mihai CIUC Universitatea Națională de Știință și Tehnologie POLITEHNICA București	Președinte
Prof. Dr. Ing. Corneliu BURILEANU Universitatea Națională de Știință și Tehnologie POLITEHNICA București	Coordonator
Acad. Prof. Dr. Ing. Horia-Nicolai TEODORESCU Universitatea Tehnică „Gheorghe Asachi” din Iași	Referent
Prof. Dr. rer. nat. Georg PELZ Infineon Technologies AG & Universitatea Duisburg-Essen	Referent
Conf. Dr. Ing. Horia CUCU Universitatea Națională de Știință și Tehnologie POLITEHNICA București	Referent

BUCUREȘTI 2024

Cuprins

1	Introducere	1
1.1	Prezentarea domeniului tezei de doctorat	1
1.2	Subiectul tezei de doctorat	2
1.3	Conținutul tezei de doctorat	4
2	Fundamente Tehnice	7
2.1	Breviar Teoretic	7
2.1.1	Formularea problemei	7
2.2	Sinteză a Literaturii	9
3	Algoritmi Evolutivi pentru Dimensionarea Circuitelor	10
3.1	Calibrare pe Benchmark-uri Sintetice	10
3.2	Evaluation on Real Circuits	11
3.3	Cel Mai Promițător Algoritm pentru Dimensionarea Circuitelor	11
4	Un Algoritm Evolutiv Îmbunătățit prin Învățare Automată	12
4.1	Modele Surogat	12
4.1.1	Utilizarea Proceselor Gaussiene	13
4.2	Optimizare Multi-Obiectiv bazată pe Evoluție Diferențială și Inferență Bayesiană (MODEBI)	13
4.2.1	Evoluție Diferențială Generalizată Îmbunătățită prin Învățare Automată 3	13
4.3	Evaluarea Algoritmului MODEBI	15
4.3.1	Rezultatele Algoritmului MODEBI	15
4.4	Puncte Tari și Limitări ale Algoritmului MODEBI	15
4.4.1	Concluzii	15
5	Optimizare Bayesiană Evolutivă (EBO)	16
5.1	Metodologie	17
5.2	Algoritmi EBO	17
5.3	Evaluarea Algoritmilor EBO	18
5.4	Puncte Tari și Limitări ale Metodologiei EBO	18

6	Gestionarea Colțurilor de Operare	19
6.1	Dimensionarea Circuitelor în Două Etape	19
6.1.1	Abordarea „designer-like”	20
6.2	Mecanismul de Selecție Periodică a Colțului Nefavorabil (PWCS) . . .	20
6.2.1	Metodă	21
6.3	Dimensionare Centrată pe Eficiență	21
6.3.1	Metodă	21
6.4	Programatorul de Simulări	22
6.4.1	Metodă	23
7	Concluzii	24
7.1	Contribuții originale	26
7.2	Lista lucrărilor publicate	28
7.3	Perspectivă de dezvoltare ulterioară	29
	Bibliografie	31

Capitolul 1

Introducere

1.1 Prezentarea domeniului tezei de doctorat

Figura 1.1 prezintă etapele inițiale ale procesului de dezvoltare a unui circuit analogic. Procesul începe cu o listă de specificații pentru circuit, definite de un arhitect sau de un inginer de aplicații, pe baza funcționalităților dorite. Ulterior, proiectantul selectează o topologie adecvată pentru circuit. Pasul următor, dimensionarea circuitului, constă în determinarea valorilor optime ale componentelor circuitului pentru a satisface constrângerile funcționale specificate și pentru a îmbunătăți performanța generală. În această etapă, se efectuează de obicei o verificare sumară pentru a asigura robustețea de bază a designului. Etapa de verificare începe cu o evaluare detaliată a performanței, luând în considerare diferiți factori de mediu. Dacă circuitul eșuează în anumite condiții, procesul de dimensionare este reluat pentru a corecta aceste deficiențe.

După ce circuitul trece de verificare, se generează manual schema de layout. În acest punct, performanța circuitului este reevaluată, luând în calcul efectele parazite generate de layout. Dacă aceste efecte parazite degradează semnificativ performanța, este necesară ajustarea layout-ului. Când circuitul îndeplinește criteriile de performanță și robustețe, acesta intră în producție. După fabricație, se efectuează verificarea post-siliciu pentru a valida comportamentul circuitului în condiții reale, asigurându-se că acesta corespunde specificațiilor originale de proiectare.

Odată cu apariția inteligenței artificiale, a crescut interesul pentru automatizarea proceselor de proiectare a circuitelor analogice. Astfel, cercetările recente au explorat pe larg diverse sarcini de proiectare, precum verificarea pre-siliciu [1, 2], generarea layout-ului [3] și verificarea post-siliciu [4]. Mai mult, modelarea circuitelor [5, 6] a fost utilizată pentru a crește gradul de reutilizare. Această teză abordează sarcina de dimensionare a circuitelor și include procesul de verificare sumară specific acestei etape.

În mod obișnuit, în dimensionarea circuitelor, inginerii cu înaltă calificare aleg valorile adecvate pentru componentele circuitului. Această activitate necesită efort manual și îi ocupă proiectantului de circuite o perioadă semnificativă de timp. Deși

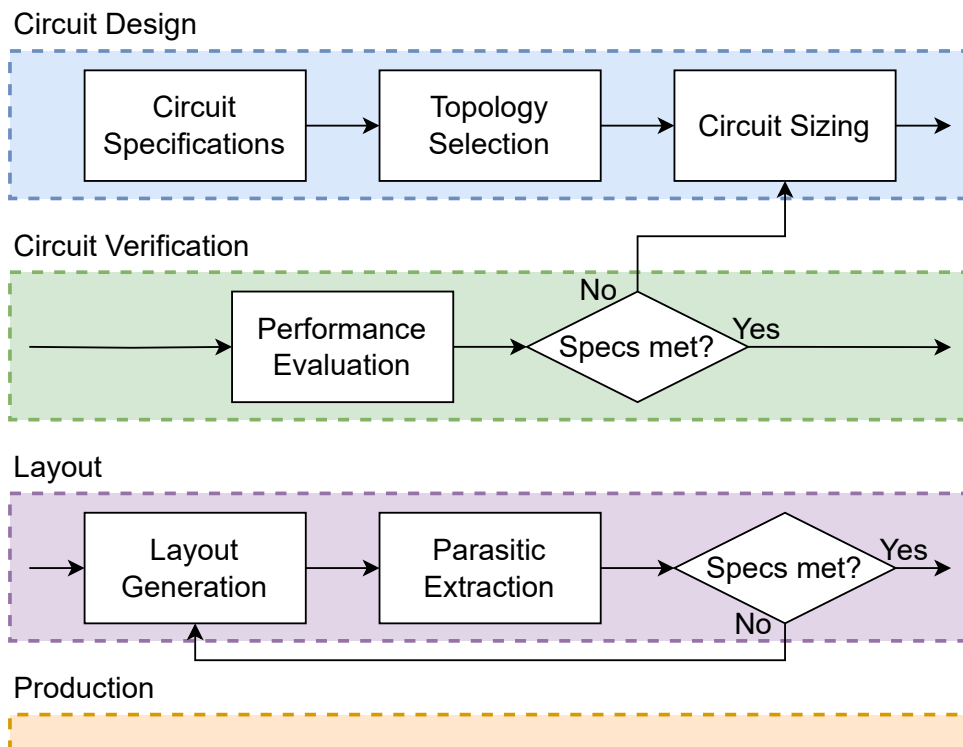


Fig. 1.1 Etapele procesului de proiectare a unui circuit analogic

dimensionarea manuală a circuitului este o sarcină iterativă, nu poate fi automatizată cu ușurință, deoarece depinde în mare măsură de experiența proiectantului. Această provocare a devenit un subiect popular în domeniul Electronic Design Automation, iar de-a lungul anilor au fost propuse numeroase tehnici. Scopul acestor metode este de a reduce efortul depus de proiectanți, permițându-le să se concentreze pe sarcini mai creative. Astfel, productivitatea și satisfacția angajaților din companiile de semiconductori pot fi influențate pozitiv.

1.2 Subiectul tezei de doctorat

Tehnicile automate de dimensionare a circuitelor pot fi împărțite în metode calitative și cantitative. O metodă calitativă este proiectarea bazată pe reguli [7], care presupune crearea unor algoritmi și ecuații pe baza cunoștințelor prealabile ale proiectantului. Această metodă oferă un anumit grad de automatizare, însă necesită în continuare un efort semnificativ din partea proiectantului. O altă clasă de metode calitative este dimensionarea bazată pe modele. Acestea utilizează ecuații de model simplificate, fiind rapide și reutilizabile. Un exemplu este Programarea Geometrică [8–10], unde răspunsurile circuitului sunt exprimate ca ecuații posinomiale ale parametrilor de proiectare. Un alt exemplu este utilizarea Programării Semidefinite (SDP) [11], care folosește modele de

regresie bazate pe date de simulare pentru a exprima răspunsurile circuitului. Deși aceste metode oferă avantaje importante, metodele de dimensionare bazate pe modele sunt nesigure din cauza abaterii intrinseci a modelului de la circuitul real. [12]

Metodele cantitative se bazează pe simulări ale circuitului. Drept urmare, acestea mențin fidelitatea față de circuitul real fără a introduce deviații. Circuitul este modelat ca o problemă de optimizare multi-obiectivă cu constrângeri, în care parametrii de proiectare sunt considerați drept intrări, iar răspunsurile circuitului sunt ieșiri. Cele mai utilizate metode pentru a aborda această problemă sunt algoritmi evolutivi (EAs) [13]. Aceștia sunt motoare de căutare stocastică globală care imită procesele naturale de supraviețuire. Alte metode cantitative utilizate în dimensionarea automată a circuitelor sunt Simulated Annealing (SA) [14, 15] și Particle Swarm Optimization (PSO) [16, 17]. Dezavantajul acestor metode constă în numărul mare de simulări necesare pentru circuit.

Pentru a depăși limitările metodelor pur cantitative, au fost dezvoltate metode hibride. Acestea combină puterea de căutare a algoritmilor evolutivi cu modele de circuit pentru a reduce numărul de simulări. Creșterea disponibilității puterii de calcul din ultimele două decenii a făcut ca modelele de învățare automată (ML) să devină o opțiune populară. Inițial, metodele au vizat proiectarea circuitelor de radiofrecvență [18, 19], iar dezvoltările recente s-au extins către dimensionarea circuitelor analogice de uz general [20, 21].

Discuția despre algoritmi evolutivi îmbunătățiți prin ML se axează pe modelul ML și pe algoritmul utilizat. O parte semnificativă a soluțiilor propuse [19, 21, 22] utilizează Procese Gaussiene [23, 24]. Mai recent, se încearcă integrarea rețelelor neuronale artificiale [20] pentru a reduce costurile computaționale. Din păcate, acestea necesită mai multe date de antrenament. Pe partea de algoritmi, multe lucrări [19, 20, 22] se bazează pe Evoluția Diferențială [25]. Altele folosesc algoritmi genetici [26–28], precum algoritmul genetic cu sortare nedominată II (NSGA-II) [29, 30].

Deși algoritmi evolutivi îmbunătățiți prin ML au fost intens dezbătuți în literatură, există unele limitări care trebuie abordate. Majoritatea aplicațiilor de dimensionare a circuitelor necesită optimizarea unor obiective conflictuale. Cu toate acestea, multe metode fie se concentrează pe rezolvarea problemelor cu un singur obiectiv [19, 31], fie folosesc scalarizarea pentru a trata obiective multiple [32, 12].

Deși algoritmi evolutivi îmbunătățiți prin ML sunt cea mai comună abordare pentru dimensionarea circuitelor, au fost dezvoltate și metode alternative. Un exemplu care a câștigat multă tracțiune în ultimii ani este Optimizarea Bayesiană [33]. În loc să utilizeze modelul ML pentru a minimiza numărul de simulări necesare de un algoritm evolutiv, Optimizarea Bayesiană folosește modelul ca suport pentru un optimizator. Optimizatorul caută optimul modelului, așa cum este descris de o funcție de achiziție. Astfel, următoarele puncte de evaluare sunt optimele modelului, în loc de evoluții ale soluțiilor anterioare.

O metodă populară de Optimizare Bayesiană pentru dimensionarea automată a circuitelor este WEIBO [12], proiectată special pentru a aborda problemele ce au constrângeri. Cu toate acestea, nu este o abordare într-adevăr multi-obiectiv, deoarece folosește scalarizarea Tchebysheff pentru a construi o funcție de merit. O metodă de ultimă generație de Optimizare Bayesiană este MACE [34]. Aceasta construiește de asemenea un model bazat pe o funcție de merit, dar în loc de o singură funcție de achiziție, folosește trei funcții diferite. Optimizarea acestora este cu adevărat multi-obiectiv. Avantajul său constă în găsirea compromisurilor între funcțiile de achiziție, deoarece niciuna nu le poate depăși pe celelalte în toate problemele [35].

Un aspect adesea neglijat în literatura de dimensionare automată a circuitelor este gestionarea colțurilor de operare (PVT). De obicei, proiectanții selectează cel mai nefavorabil colț PVT în avans, bazându-se pe experiență, și dimensionează circuitul pentru această situație. La final, designul este verificat în toate colțurile PVT pentru a-i demonstra robustețea, însă acest proces poate duce la cicluri multiple de proiectare-verificare. Alternativ, dimensionarea poate include verificarea simultană a tuturor colțurilor PVT pentru fiecare configurație, dar acest lucru este costisitor computațional, putând epuiza bugetul de simulare fără a găsi soluții fezabile.

1.3 Conținutul tezei de doctorat

Având în vedere stadiul actual al cercetării în domeniul dimensionării automate a circuitelor, există încă spațiu semnificativ pentru îmbunătățiri în mai multe direcții. Obiectivul principal al acestei teze este de a reduce timpul necesar pentru dimensionarea circuitelor în comparație cu metodele tradiționale. Pentru a atinge acest obiectiv, ne concentrăm pe două aspecte cheie ale algoritmului: eficiența și robustețea. În acest context, stabilim următoarele obiective specifice:

- (a) Identificarea celui mai promițător algoritm evolutiv pentru sarcinile de dimensionare a circuitelor. Deși calculul evolutiv stă la baza multor metode automate de dimensionare a circuitelor, selecția algoritmului pare a fi arbitrară în majoritatea studiilor. Prin urmare, este necesară o analiză detaliată pentru a identifica cel mai potrivit algoritm pentru această aplicație.
- (b) Dezvoltarea unei metode de dimensionare a circuitului care să reducă bugetul de simulare necesar pentru a găsi configurații de circuit care să îndeplinească specificațiile. Ne propunem o reducere a numărului de simulări cu cel puțin 50% față de algoritmii evolutivi clasici.
- (c) Explorarea metodelor de Optimizare Bayesiană și a algoritmilor evolutivi îmbunătățiți prin Învățare Automată, pentru a dezvolta un algoritm robust, cu performanță consistentă pe diverse circuite.

- (d) Identificarea unor metode alternative de reducere a bugetului de simulare. Dezvoltarea continuă a unor algoritmi de optimizare din ce în ce mai avansați poate să nu fie cea mai eficientă abordare. În schimb, redefinirea problemei poate accelera procesul de dimensionare a circuitelor.
- (e) Integrarea unui grad avansat de verificare în procesul de dimensionare automată a circuitelor, pentru a minimiza numărul de cicluri de proiectare-verificare.

Pentru a răspunde acestor obiective, teza este structurată după cum urmează:

Începem prin a prezenta baza teoretică, literatura de specialitate și descrierea circuitelor în Capitolul 2. Baza teoretică (Secțiunea 2.1) include formularea problemei, descrierea metricilor utilizate și prezentări detaliate ale conceptelor cheie precum algoritmi evolutivi, Procesele Gaussiene și Optimizarea Bayesiană. Secțiunea 2.2 prezintă stadiul actual al cercetării în dimensionarea automată a circuitelor.

Comparăm algoritmi evolutivi (EAs) în Capitolul 3 pentru a identifica cel mai promițător dintre ei pentru sarcinile de dimensionare a circuitelor, conform obiectivului (a). Pornim de la cinci algoritmi evolutivi de ultimă generație și folosim benchmark-uri sintetice pentru a găsi cele mai bune configurații ale hiperparametrilor. Apoi aplicăm algoritmi pe două probleme de optimizare care vizează circuite reale, mai exact regulatoare de tensiune. În timp ce primul circuit are mai mulți parametri de proiectare (27, comparativ cu 8), al doilea are mai multe obiective de optimizare (6, în loc de 3). Pe lângă evaluarea performanței, analizăm și diversitatea soluțiilor oferite de diferiții algoritmi și tragem concluzii privind care dintre ei aduce mai multe beneficii proiectantului de circuite.

În Capitolul 4 începem să abordăm limitările abordărilor pur cantitative folosind metode hibride. Mai exact, introducem un nou algoritm evolutiv îmbunătățit prin învățare automată (ML) pentru a accelera procesul de optimizare, în conformitate cu obiectivul (b). Este vorba de un algoritm de optimizare multi-obiectiv care combină cel mai promițător algoritm evolutiv descoperit în Capitolul 3 cu Procesele Gaussiene. Acesta este conceput să funcționeze bine pe probleme cu mulți parametri, obiective și specificații de constrângeri. Ca parte a dezvoltării sale, propunem politici inovatoare de supraviețuire a populației și algoritmi de selecție a soluțiilor care reduc numărul de simulări reale necesare procesului de optimizare și păstrează diversitatea soluțiilor. Din câte știm, metoda propusă este prima care abordează în mod specific diversitatea soluțiilor prin căutarea directă în spațiul multi-obiectiv folosind dominanța Pareto.

Abordarea problemei dimensionării circuitelor într-o manieră multi-obiectiv este unul dintre pilonii principali ai metodelor propuse în această teză. De asemenea, spre deosebire de lucrările anterioare, circuitele pe care le evaluăm au numeroși parametri de proiectare și răspunsuri care trebuie să îndeplinească constrângeri. Majoritatea cercetărilor din acest domeniu au efectuat teste pe circuite cu mai puțin de 10 parametri de proiectare. Chiar dacă metode foarte recente [36] au considerat probleme cu 24 și 20

de parametri de proiectare, acestea au avut doar un obiectiv de optimizare și șase sau două constrângeri.

În Capitolul 5 aprofundăm metodele hibride concentrându-ne pe optimizarea Bayesiană. Mai precis, prezentăm o paradigmă nouă de dimensionarea automată a circuitelor care combină algoritmi evolutivi îmbunătățiți prin învățare automată cu optimizarea Bayesiană, în conformitate cu obiectivul (c). Această strategie nouă se numește Optimizare Bayesiană Evolutivă (EBO). Pe de o parte, urmează strategia algoritmilor evolutivi îmbunătățiți prin ML de căutare globală folosind calcul evolutiv, care s-a dovedit a fi eficientă. Pe de altă parte, ca în Optimizarea Bayesiană, tehnica noastră profită pe deplin de puternicul model ML, căutând optimele sale în loc să îl folosească doar ca preselector. Mai mult, metoda noastră valorifică infrastructura de simulare paralelă.

Conceptul de gestionare a colțurilor PVT este prezentat în Capitolul 6. Abordăm verificarea sumară a PVT — specifică etapei de dimensionare a circuitului — într-un mod inteligent, prin utilizarea unor strategii de gestionare a colțurilor de operare în timpul optimizării (obiectiv (d)). Propunem trei strategii diferite și, de asemenea, combinăm două dintre ele pentru a obține eficiența maximă. Prima strategie este o abordare în două etape (Secțiunea 6.1), în care algoritmul identifică candidați promițători pentru dimensionare considerând inițial doar colțul nominal. Apoi, în a doua etapă, începe să ia în considerare condițiile extreme de operare. Această tehnică simplă conduce la mai puține simulări ale circuitului cheltuite pe candidați nepromițători în pașii inițiali de optimizare. A doua strategie este un mecanism periodic de selecție a celor mai defavorabile colțuri (Secțiunea 6.2). Acesta permite obținerea de rezultate similare optimizării în toate colțurile, reducând drastic bugetul de simulare. Cele două strategii pot fi combinate pentru a îmbunătăți și mai mult eficiența (Secțiunea 6.3).

A treia strategie de gestionare a colțurilor utilizează o listă prioritară de colțuri, ordonate în funcție de nivelul lor de dificultate (Secțiunea 6.4). În funcție de caz, lista poate include sau nu colțul nominal. Soluțiile sunt evaluate secvențial în aceste colțuri, iar evaluarea fiecărei soluții se oprește imediat ce aceasta nu îndeplinește specificațiile pentru un anumit colț. Prin urmare, această abordare minimizează simulările cheltuite pe soluții de calitate inferioară, concentrând eforturile de optimizare pe colțurile mai dificile. Mai mult, această abordare permite algoritmului de dimensionare a circuitului să ia în considerare multe colțuri PVT, ceea ce poate duce la o reducere a numărului de bucle de proiectare-verificare necesare pentru a obține un circuit robust, așa cum este subliniat în obiectivul (e). Acest aspect este esențial în îmbunătățirea ratei de adoptare a metodelor automate de dimensionare a circuitelor, deoarece reduc timpul de lansare a noilor produse.

În final, în Capitolul 7 tragem concluziile și discutăm limitările cercetării noastre. În plus, parcurgem câteva dintre direcțiile care pot fi explorate în viitor.

Capitolul 2

Fundamente Tehnice

2.1 Breviar Teoretic

2.1.1 Formularea problemei

În proiectarea de circuite, inginerii aleg o anumită topologie și ajustează manual parametrii de proiectare pentru a îndeplini anumite specificații și pentru a îmbunătăți performanța generală. De obicei, parametrii de proiectare includ valori de rezistențe și condensatoare, împreună cu parametri ai tranzistorului precum multiplicitatea, lungimea și lățimea canalului. Specificațiile se referă la măsuri de performanță, cunoscute sub denumirea de răspunsuri ale circuitului, care trebuie să atingă anumite praguri. De exemplu, specificațiile pentru regulatoarele de tensiune de tip low-dropout pot include marginea de fază, raportul de respingere al sursei de alimentare (PSRR) și timpul de stabilizare. Mai mult, unele răspunsuri ale circuitului necesită optimizare suplimentară peste îndeplinirea specificațiilor, cum ar fi minimizarea suprafeței pe cip și reducerea consumului de curent.

Astfel, dimensionarea circuitului poate fi formulată ca o problemă de optimizare multi-obiectivă cu constrângeri (Ecuția 2.1).

$$\begin{aligned} & \text{minimizeaza } f_i(X), i \in N_{\text{obiective}} \\ & \text{respectand condițiile} \\ & g_j(X) < \text{prag}_j, j \in N_{\text{specificatii}} \end{aligned} \tag{2.1}$$

unde X este vectorul de parametri de proiectare, $g_j(X)$ sunt constrângerile care reprezintă specificațiile circuitului, iar $f_i(X)$ sunt obiectivele care necesită optimizare suplimentară. Trebuie consemnat că Ecuția 2.1 prezintă cazul unei probleme de minimizare. În practică, obiectivele și constrângerile care necesită maximizare pot fi transformate în

probleme de minimizare prin negarea valorilor (Ecuația 2.2).

$$\begin{aligned} \text{maximizeaza } f_i(x) &\equiv \text{minimizeaza } -f_i(x) \\ g_j(X) > \text{prag}_j &\equiv -g_j(X) < -\text{prag}_j \end{aligned} \quad (2.2)$$

De obicei, parametrii de proiectare sunt numere întregi sau reale. Cu toate acestea, uneori putem avea de a face și cu enumerări, intervale discrete sau chiar variabile categorice. De asemenea, trebuie remarcat că un obiectiv $f_i(X)$ poate avea și o constrângere asociată $g_j(X)$. De exemplu, consumul de curent trebuie să fie sub un anumit prag, dar este dorit să fie cât mai mic posibil.

Gestionarea constrângerilor este un aspect crucial în dimensionarea circuitelor. Aceasta implică îndeplinirea tuturor specificațiilor pentru ca o soluție să fie fezabilă. Tehnicile de gestionare a constrângerilor se împart în trei categorii principale [37]: cele care prioritizează soluțiile fezabile pentru a supraviețui până în generația următoare, cele care urmăresc să echilibreze compromisul între fezabilitate și convergență, și cele care sunt folosite pentru a repara soluțiile nefezabile. Această teză se concentrează pe tehnicile din prima categorie, deoarece acestea sunt robuste, bine documentate și testate.

Prin urmare, rezolvarea unei probleme de optimizare multi-obiectiv cu constrângeri necesită o abordare în două etape. Prima dată, algoritmul trebuie să găsească soluții care îndeplinesc specificațiile, numite soluții fezabile. Apoi, algoritmul trebuie să optimizeze obiectivele. Astfel, considerăm că o optimizare are două faze: căutarea soluțiilor fezabile și optimizarea obiectivelor. Această strategie în două etape pentru optimizarea cu constrângeri este bine documentată în literatură [38].

Ca în orice problemă de optimizare multi-obiectiv, optimul nu este reprezentat de un singur punct în spațiul multidimensional al obiectivelor (numit adesea hiperspațiu). În schimb, este reprezentat de un front Pareto, ceea ce face procesul de optimizare mai dificil. Calitatea unui algoritm este măsurată în funcție de cât de bine reprezintă frontul Pareto al problemei la finalul optimizării. Pe de o parte, este important să genereze soluții cât mai apropiate de frontul Pareto. Pe de altă parte, soluțiile trebuie să fie distribuite de-a lungul întregului front, nu doar într-o anumită regiune. În dimensionarea circuitelor este esențial ca soluțiile să fie distribuite pe tot frontul Pareto. Astfel, proiectantul are acces la mai multe variante de compromis.

În practică, sarcinile de dimensionare a circuitelor diferă de problemele clasice de optimizare multi-obiectiv, deoarece circuitele sunt supuse condițiilor de operare. Condițiile de operare sunt intrări ale circuitului care nu pot fi controlate de algoritm. Acestea reprezintă variabile precum colțul de proces, temperatura, tensiunea de referință etc. Similar cu parametrii de proiectare, condițiile de operare pot avea diferite tipuri de date. O anumită combinație de valori ale condițiilor de operare reprezintă un colț de operare, adesea denumit colț proces-tensiune-temperatură (PVT). De obicei, proiectanții stabilesc colțurile de operare relevante după efectuarea simulărilor Monte Carlo. Un

design este considerat robust dacă îndeplinește specificațiile în toate colțurile de operare sau se încadrează într-o anumită toleranță [39].

2.2 Sinteză a Literaturii

Algoritmi Evolutivi Algoritmii evolutivi (EAs) sunt metode de căutare stocastică prin care se imită procesul natural de supraviețuire. Diagrama de principiu a algoritmilor evolutivi pentru dimensionarea circuitelor este prezentată în Figura 2.1. EAs se diferențiază prin politicile de supraviețuire și strategiile utilizate pentru generarea soluțiilor. Cele mai relevante metode din domeniul dimensionării automate a circuitelor folosesc algoritmi genetici sau Evoluția Diferențială pentru generarea soluțiilor. Politicile de supraviețuire ale diferiților EAs variază considerabil, însă pot fi împărțite în două categorii principale: cele care urmează o abordare elitistă și cele care promovează diversitatea.

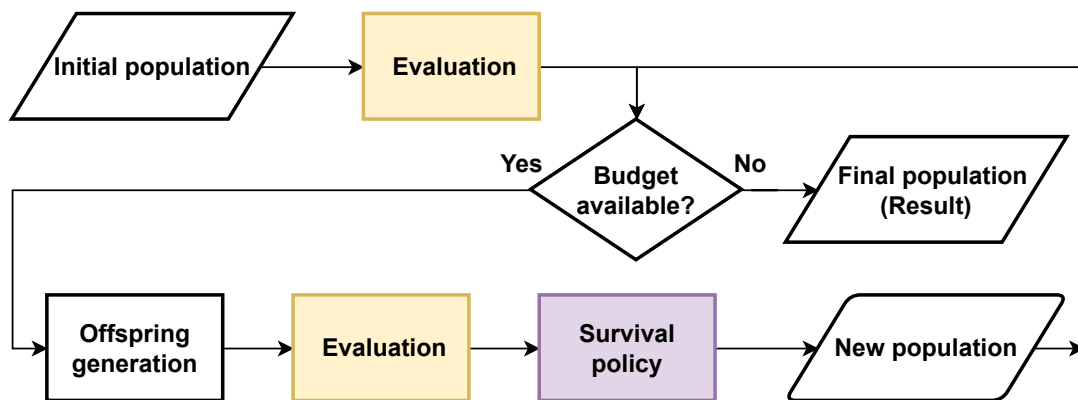


Fig. 2.1 Diagrama de principiu a algoritmilor evolutivi pentru dimensionarea circuitelor

Există numeroase exemple de succes în utilizarea EAs pentru sarcinile de dimensionare a circuitelor. Unul dintre cei mai utilizați algoritmi este NSGA-II [30], folosit ca atare pentru dimensionarea automată a circuitelor integrate analogice în [26, 27] sau prin adăugarea unei metode de clustering pentru a reduce numărul de simulări necesare [28]. Alți algoritmi populari utilizați pentru dimensionarea circuitelor includ "Particle Swarm Optimization" și diverse variante ale evoluției diferențiale [40].

Modele Surogat Deși EAs pot genera soluții de calitate, eficiența în raport cu timpul de rulare este crucială, deoarece simulările sunt costisitoare computațional. Astfel, modelele surogat sunt folosite pentru a înlocui o parte din aceste simulări, necesitând mult mai puțină putere de calcul decât un simulator integrat în algoritmul de optimizare. Cele mai populare modele surogat sunt Procesele Gaussiene (GPs) și Rețelele Neuronale Artificiale (NNs).

Capitolul 3

Algoritmi Evolutivi pentru Dimensionarea Circuitelor

Acest capitol este un studiu asupra algoritmilor evolutivi clasici utilizați în contextul dimensionării automate a circuitelor. Se bazează în principal pe [41] și [42].

În metodele bazate pe simulare, circuitul este tratat ca o cutie neagră, iar optimizarea sa se bazează exclusiv pe simulări. Există mulți algoritmi metaeuristici potriviți pentru sarcinile de dimensionare a circuitelor, însă cei mai populari sunt, de departe, algoritmi evolutivi. În acest capitol discutăm despre cei mai promițători algoritmi evolutivi în contextul dimensionării circuitelor. Mai întâi, folosim benchmark-uri sintetice pentru a găsi cele mai bune configurații de hiperparametri și aplicăm algoritmi pe un regulator de tensiune de tip low-dropout (LDO) real. Scopul nostru este să identificăm cei mai promițători algoritmi evolutivi și să evaluăm impactul proceselor aleatoare asupra sarcinii de dimensionare a circuitului. Ulterior extindem analiza folosind un alt LDO. În acest caz punem mai mult accent pe diversitatea populației și versatilitatea algoritmilor evolutivi.

3.1 Calibrare pe Benchmark-uri Sintetice

Într-un scenariu ideal, cei cinci algoritmi evolutivi ar fi calibrați pe o problemă de dimensionare a circuitului reală. Din păcate, evaluarea unei singure configurații de hiperparametri a unui singur algoritm poate dura mai mult de o zi. Astfel, configurațiile de hiperparametri utilizate sunt rezultatul calibrării algoritmilor pe benchmark-uri sintetice.

Rezultatele experimentale din această subsecțiune au fost obținute pe cele 9 benchmark-uri sintetice din toolkit-ul Walking Fish Group (WFG) [43]. Acest toolkit este scalabil și acceptă orice număr de obiective. Mai mult, acesta încorporează o varietate de caracteristici importante care se regăsesc în probleme reale.

Toate optimizările pe funcții sintetice (fiecare algoritm pe fiecare problemă) au fost efectuate de 10 ori, folosind populații inițiale diferite, iar rezultatele au fost mediate pe aceste 10 rulări. Am ales probleme cu 5 și 10 obiective pentru a putea compara modul în

care algoritmi de optimizare se comportă pe probleme cu mai puține obiective (5) și cu mai multe obiective (10).

3.2 Evaluation on Real Circuits

In this subchapter, we evaluate the target algorithms on two voltage regulators. The focus of the comparison is not only performance, but also robustness and population diversity.

3.3 Cel Mai Promițător Algoritm pentru Dimensionarea Circuitelor

Scopul acestui capitol este de a evalua algoritmi evolutivi din punct de vedere al performanței, versatilității și diversității populației. În acest context, prin versatilitate înțelegem capacitatea unui algoritm de a obține performanțe similare pe diferite circuite. Performanța algoritmului GDE3 [44] este remarcabilă și constantă în toate situațiile analizate. Astfel, putem argumenta că acesta este cel mai versatil algoritm dintre cei cinci studiați. Este adevărat că și performanța algoritmului IBEA [45] este relativ bună pentru problemele considerate, însă imposibilitatea de a-l utiliza pentru probleme cu un număr mare de obiective îi reduce versatilitatea.

În ceea ce privește diversitatea soluțiilor, GDE3 obține cele mai bune rezultate, datorită capacităților sale intrinseci și potențialului de a performa bine cu o populație mare. În concluzie, considerăm că GDE3 este cel mai promițător algoritm evolutiv pentru automatizarea dimensionării circuitelor. Performanța sa este foarte bună, s-a dovedit a fi versatil, și menține cea mai diversă populație dintre toți algoritmi considerați. Deși natura sa duce uneori la o convergență mai lentă, în aplicațiile de dimensionare a circuitelor, unde nivelul de complexitate al problemei poate fluctua considerabil de la un circuit la altul, considerăm că o convergență mai lentă, dar mai robustă, este de dorit.

Algoritmi evolutivi sunt eficienți pentru eșantionarea hiperspațiului parametrilor de proiectare, însă utilizarea lor de sine stătătoare duce la aplicații care necesită un număr mare de simulări. Soluțiile de ultimă generație pentru dimensionarea circuitelor folosesc modele surogat bazate pe învățare automată pentru a reduce numărul total de simulări. Prin urmare, ne propunem să dezvoltăm un algoritm eficient de dimensionare a circuitelor prin combinarea GDE3 cu un astfel de model surogat.

Capitolul 4

Un Algoritm Evolutiv Îmbunătățit prin Învățare Automată

Acest capitol se bazează pe următoarele articole: [46] și [47]

Algoritmii evolutivi (EAs) sunt metode puternice de căutare stocastică. Totuși, în forma lor originală, aceștia necesită un număr foarte mare de simulări de circuit pentru a găsi soluții optime. În acest capitol, combinăm GDE3 [44], considerat cel mai promițător algoritm evolutiv pentru dimensionarea circuitelor, cu Procesele Gaussiene (GPs) pentru a reduce numărul de simulări necesare. Algoritmul de dimensionare rezultat este denumit "Optimizare Multi-obiectiv bazată pe Evoluție Diferențială și Inferență Bayesiană" (MODEBI).

Principalele contribuții ale acestui capitol sunt următoarele. În primul rând, propunem o metodă de optimizare multi-obiectiv bazată pe combinarea unui algoritm inspirat de GDE3 cu GPs, concepută pentru a funcționa eficient pe probleme cu multe variabile, obiective și specificații de constrângeri. În al doilea rând, propunem politici inovatoare de supraviețuire a populației și algoritmi de selecție a soluțiilor care reduc numărul de simulări necesare pentru a finaliza procesul de optimizare. În final, performanța metodei noastre este comparată cu algoritmi evolutivi și algoritmi de optimizare bayesiană pe două regulatoare de tensiune reale.

4.1 Modele Surogat

Optimizarea eficienței timpului de rulare este esențială. Pentru a realiza acest lucru, modelele surogat sunt utilizate pentru a înlocui simulatorul real al circuitului pentru o parte din evaluări. Construirea și utilizarea unui model surogat pentru a prezice valorile funcției necesită mult mai puțin efort computațional decât integrarea directă a simulatorului în algoritmul de optimizare. În literatură, cele mai folosite modele surogat sunt Procesele Gaussiene și rețelele neuronale artificiale. Abordările recente, inclusiv cea propusă de noi, utilizează antrenarea online, în care modelul este actualizat

incremental în timpul optimizării cu fiecare nouă simulare. Deși utilizarea modelelor surogat introduce un anumit cost computațional suplimentar, acestea ghidează mai eficient selecția soluțiilor promițătoare, conducând la o convergență mai rapidă.

4.1.1 Utilizarea Proceselor Gaussiene

Regresia prin Procese Gaussiene (GP) se bazează pe inferența bayesiană, prin care un model statistic poate fi combinat cu datele observate pentru a obține un model statistic mai precis. O distribuție inițială poate fi specificată printr-o medie și o funcție kernel. În absența cunoștințelor a priori, media este considerată a fi zero. Funcția kernel reprezintă măsura de similaritate între valorile funcției în două puncte diferite. Antrenarea GP-ului pe baza punctelor de date disponibile presupune codificarea informației în medie și kernel. Pentru fiecare nou punct, GP-ul va prezice media (μ), care reprezintă valoarea estimată a funcției, și varianța (σ^2), care reprezintă incertitudinea predicției. Aceste predicții sunt rafinate incremental pe măsură ce sunt observate noi date.

O funcție de achiziție este folosită în mod tipic pentru a explora hiperspațiul folosind modelul GP. Pot fi folosite diferite tipuri de funcții de achiziție pentru a echilibra explorarea și exploatarea pe parcursul optimizării. În algoritmul propus, folosim Lower Confidence Bound (LCB) astfel (Ecuația 4.1):

$$LCB(x) = \mu(x) - K\sigma(x) \quad (4.1)$$

În problemele de minimizare, un K mai mare corespunde unui comportament puternic exploratoriu.

4.2 Optimizare Multi-Obiectiv bazată pe Evoluție Diferențială și Inferență Bayesiană (MODEBI)

În acest capitol descriem algoritmul MODEBI pornind de la inspirația sa, GDE3. Capitolul include diferitele mecanisme de selecție (și supraviețuire) a soluțiilor și versiunile algoritmului MODEBI.

4.2.1 Evoluție Diferențială Generalizată Îmbunătățită prin Învățare Automată 3

Diversitatea Populației Diversitatea populației este un aspect critic al algoritmilor evolutivi multi-obiectiv, datorită riscului de a ignora anumite zone ale spațiului de intrare și a scopului de a găsi multiple puncte pe frontul Pareto. Algoritmul GDE3 original dispune de un mecanism eficient de menținere a diversității populației: fiecare descendent este comparat doar cu părintele său pentru a determina soluția potrivită care

să fie inclusă în generația următoare. Totuși, integrarea modelului surogat GP introduce câteva modificări în abordarea inițială, fiind necesară o intervenție suplimentară pentru conservarea diversității. Algoritmul GDE3 original folosește distanța de aglomerare (*Crowding Distance*) [30] în faza de reducere a populației. În schimb, noi am aplicat *Distribution Metric* [48] nu doar pentru a monitoriza diversitatea pe parcursul optimizării, ci și pentru a selecta indivizii adecvați.

Preselecție Am utilizat modelele Gaussian Process (GP) ca mecanism de preselecție. Pentru a maximiza beneficiul potențial, generatorul de soluții GDE3 a fost folosit pentru a genera de zece ori mai mulți descendenți ($10N$) decât mărimea populației (N). În același timp, am urmărit să reducem numărul de simulări efectuate la fiecare epocă la un sfert din mărimea populației ($N/4$ în loc de N). Astfel, am adoptat modelul surogat pentru a selecta cei mai promițători $N/4$ descendenți din cei $10N$ generați. Prin înlocuirea a cel mult unui sfert din populație la fiecare pas, MODEBI menține un echilibru între explorare și exploatare. Comportamentul exploratoriu al evoluției este promovat prin menținerea unui număr mare de indivizi în populație. Simultan, doar descendenții cu cele mai bune rate de îmbunătățire sunt considerați pentru a face parte din generația următoare.

Pentru fiecare răspuns al circuitului, se antrenează câte un Proces Gaussian folosind parametrii de proiectare și condițiile de operare ca intrări. Această abordare presupune un cost computațional mai redus și permite antrenarea și interogarea paralelă a Proceselor Gaussiene. Cu toate acestea, există o limitare, și anume faptul că nu se iau în considerare corelațiile dintre răspunsuri, aspect ce poate fi abordat în cercetări viitoare. Folosim un kernel de tip "radial basis function" scalat, deoarece este o alegere populară în ML, iar complexitatea sa redusă face posibilă antrenarea modelului pe toate punctele de date disponibile (până la 20,000 în cazurile noastre de test). Mai mult, o bună acuratețe poate fi atinsă cu un cost computațional rezonabil.

Cei mai promițători $N/4$ descendenți (așa cum sunt prezicți de modelele Gaussian Process și selectați prin una dintre funcțiile de selecție introduse mai jos) sunt evaluați folosind simulatorul circuitului. În final, noua populație este creată prin selectarea celor mai buni indivizi din populația anterioară plus acești descendenți. Metoda actualizată este prezentată în Algoritmul 1.

Concluzie MODEBI poate fi considerat o extensie a GDE3 care integrează modele surogat Gaussian Process pentru preselecția descendenților, accelerând convergența fără a pierde timp pe simulări reale ale circuitului. Preselecția trebuie efectuată cu atenție pentru a păstra diversitatea populației și a promova cei mai promițători descendenți.

Algorithm 1: MODEBI pentru dimensionarea circuitelor

Input: O populație aleatorie **POP** de mărime **N**

- 1 **while** *bugetul de simulare este disponibil* **do**
- 2 GP = antrenează(simulări disponibile)
- 3 descendenți = 10 × generează(POP, GDE3Operator)
- 4 evaluează(descendenți, GP)
- 5 bestOffs = selectează(descendenți, $N/4$)
- 6 simulează(bestOffs)
- 7 POP = supraviețuire(POP, bestOffs, N)
- 8 **end while**

4.3 Evaluarea Algoritmului MODEBI

În această secțiune evaluăm algoritmul MODEBI. Mai întâi, evaluăm cele trei variante ale algoritmului pentru a determina care este cea mai promițătoare. Apoi, comparăm cea mai bună variantă MODEBI cu GDE3 și cu metoda de Optimizare Bayesiană MACE [34]. În final, analizăm impactul proceselor aleatorii asupra algoritmului MODEBI.

4.3.1 Rezultatele Algoritmului MODEBI

Am realizat trei experimente independente pentru a evalua performanța algoritmului MODEBI în optimizarea a două regulatoare de tensiune de tip low-dropout. Experimentele au inclus trei variante ale algoritmului propus. Folosirea unui model bazat pe Procese Gausiene ca surogat pentru simulatorul circuitului ar trebui să permită obținerea mai rapidă a soluțiilor fezabile. Prin urmare, variantele algoritmului au fost concepute pentru a determina combinația optimă de metode de selecție a descendenților și de supraviețuire a populației, care să producă cele mai bune rezultate.

4.4 Puncte Tari și Limitări ale Algoritmului MODEBI

4.4.1 Concluzii

În acest capitol, am propus un algoritm inovator de optimizare multi-obiectiv pentru dimensionarea automată a circuitelor, numit MODEBI. Acesta este conceput pentru a aborda circuite cu un număr mare de variabile de proiectare (în ordin de zeci), mai multe colțuri PVT și numeroase (10+) răspunsuri de circuit ce trebuie să îndeplinească specificațiile asociate. În acest context, MODEBI este o metodă de optimizare a designului care utilizează un algoritm evolutiv inspirat din GDE3 pentru a explora hyperspațiul complex al variabilelor de proiectare și al colțurilor PVT. De asemenea, MODEBI folosește Procese Gausiene ca model surogat pentru simulările costisitoare ale circuitelor, accelerând convergența algoritmului evolutiv.

Capitolul 5

Optimizare Bayesiană Evolutivă (EBO)

În acest capitol, introducem o strategie nouă pentru dimensionarea automată a circuitelor, denumită Optimizare Bayesiană Evolutivă (EBO). Această metodologie integrează punctele forte ale celor mai bune două abordări din literatură. Pe de o parte, adoptă abordarea de căutare globală prin intermediul algoritmilor evolutivi îmbunătățiți cu învățare automată (ML-enhanced EAs), care s-a dovedit a fi eficientă atât în studii mai vechi [19], cât și în dezvoltări recente (Capitolul 4). Pe de altă parte, similar cu Optimizarea Bayesiană, metoda noastră valorifică pe deplin modelul ML, căutând activ optimul acestuia, în loc să îl folosească doar ca instrument de preselecție. În plus, EBO realizează optimizare multi-obiectiv și caută pe baza unor populații de soluții, ceea ce îi permite să profite pe deplin de infrastructura de simulare paralelă.

Metodologia constă în două mecanisme imbricate de căutare. Cel interior, numit evoluție virtuală, caută optimul modelului ML. Cel exterior folosește simulatorul real al circuitului pentru a confirma calitatea soluțiilor produse de căutarea internă. De asemenea, actualizează modelul ML cu noile puncte de date disponibile. Acest proces se repetă până când bugetul de simulare este epuizat.

Pentru a demonstra capabilitățile EBO, introducem doi algoritmi dezvoltati pe baza metodologiei propuse. Ambii folosesc motoare de generare de soluții bazate pe Evoluție Diferențială și Procese Gaussiene, dar diferă prin politicile lor de supraviețuire.

Contribuțiile principale ale acestui capitol sunt următoarele: (I) Propunem o metodologie inovativă pentru dimensionarea automată a circuitelor; (II) Dezvoltăm doi algoritmi care respectă liniile directoare ale acestei metodologii; (III) Analizăm performanța algoritmilor pe două circuite complexe proprietare pentru a demonstra că strategia noastră obține rezultate mai bune decât alte metode de ultimă generație; (IV) Creăm două studii de caz pentru circuite din literatură, pentru a demonstra practicitatea metodologiei propuse.

5.1 Metodologie

EBO reprezintă o combinație între un algoritm evolutiv și Optimizarea Bayesiană. Conceptul cheie al acestei metodologii este integrarea evoluției virtuale. Algoritmii EBO utilizează două tipuri de pași de optimizare: reali și virtuali. În pașii virtuali, soluțiile sunt evaluate pe modelul ML, ce este folosit ca surogat pentru simulatorul circuitului. Când se îndeplinește o anumită condiție, algoritmul efectuează un pas real de optimizare. Mai precis, evaluează ultima populație a evoluției virtuale folosind simulatorul real al circuitului. Pașii reali sunt efectuați pentru a se asigura că modelul ML rămâne precis și pentru a verifica calitatea soluțiilor. Datorită filosofiei sale intrinseci de evoluție bazată pe populație, EBO profită pe deplin de infrastructura de simulare paralelă.

Diagrama de flux a unui algoritm EBO generic este prezentată în Figura 5.1. Mai întâi, populația inițială este generată aleatoriu și evaluată. În al doilea rând, algoritmul efectuează evoluția virtuală până când se îndeplinește o anumită condiție. Apoi, algoritmul efectuează un pas de evoluție reală. De fiecare dată când simulatorul real este utilizat, modelul surogat ML este reantrenat folosind noile puncte de date disponibile.

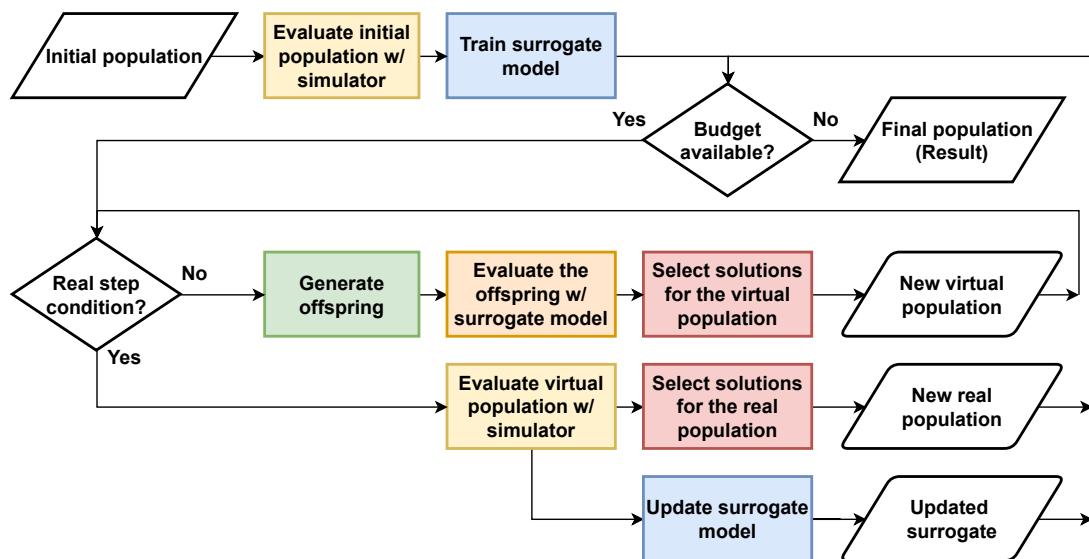


Fig. 5.1 Diagrama de principiu a unui algoritm EBO

5.2 Algoritmi EBO

Propunem doi algoritmi EBO particulari, care se diferențiază prin politicile de supraviețuire folosite. Algoritmii utilizează cel mai comun operator de Evoluție Diferențială, DE/rand/1/bin [44], ca motor de recombinare. În ceea ce privește modelele ML, algoritmi folosesc Procese Gaussiene (GPs) separate pentru fiecare răspuns al circuitului.

5.3 Evaluarea Algoritmilor EBO

Această secțiune este structurată după cum urmează: Începem prin a prezenta premisele pentru evaluare. Apoi oferim o comparație generală între cei doi algoritmi de Optimizare Bayesiană Evolutivă (EBO) propuși. În continuare, parcurgem procesul de ajustare a numărului de pași ai evoluției virtuale. Ulterior, realizăm o comparație între metoda noastră și algoritmi de ultimă generație, precum și o evaluare temporală. În final, analizăm robustețea algoritmilor EBO propuși.

5.4 Puncte Tari și Limitări ale Metodologiei EBO

În cadrul acestui capitol, propunem Optimizarea Bayesiană Evolutivă (EBO), o nouă strategie pentru dimensionarea automată a circuitelor. Această metodologie reprezintă o sinergie între cele mai promițătoare două abordări existente în literatură, valorificând în mod eficient cele mai bune aspecte ale fiecăreia. Dintr-o perspectivă, EBO urmează strategia de căutare globală a algoritmilor evolutivi îmbunătățiți cu învățare automată. Din altă perspectivă, metoda noastră valorifică pe deplin modelul ML prin căutarea optimelor sale, exact ca în Optimizarea Bayesiană. EBO utilizează o secvență alternativă de pași reali și virtuali de evoluție pentru a identifica optimul problemei.

Metodologia EBO are câteva beneficii intrinseci. În primul rând, este cu adevărat multi-obiectiv, eliminând dezavantajele asociate cu scalarea răspunsurilor și integrarea lor într-o funcție de cost. În al doilea rând, folosește în mod inerent loturi mari de simulări, valorificând pe deplin infrastructura de simulare paralelă. În al treilea rând, fiind o metodă orientată către diversitate, se concentrează nu doar pe găsirea unei soluții bune, ci pe descrierea întregului front Pareto. Astfel, oferă proiectanților acces la diferite compromisuri. În al patrulea rând, algoritmi dezvoltati în cadrul EBO nu folosesc măsurile de incertitudine ale modelelor ML. Aceasta nu doar elimină riscul asociat cu o varianță mare, ci face și metodologia compatibilă cu modele ML care nu furnizează astfel de metrice.

Capitolul 6

Gestionarea Colțurilor de Operare

Acest capitol se bazează pe următoarele articole: [49], [50], [51], [52]

Colțurile de operare sunt obținute prin combinarea valorilor relevante ale condițiilor de operare. Gestionarea eficientă a simulărilor pentru colțuri poate duce la o reducere semnificativă a bugetului de simulare. Mai mult, aceasta poate face diferența între respectarea sau nerespectarea constrângerilor.

O abordare naivă, dar comună în literatură, este optimizarea pentru colțul nominal și verificarea celorlalte colțuri la finalul procesului de optimizare. Această metodă minimizează numărul de simulări, însă algoritmul rezultat este unul lipsit de robustețe. O altă abordare constă în identificarea celui mai nefavorabil colț și optimizarea acestuia. Totuși, din cauza interacțiunilor între elementele circuitului în diverse condiții de operare, colțul cel mai nefavorabil se poate schimba pe parcursul optimizării. Cea mai sigură abordare, așa cum s-a demonstrat în capitolele anterioare, este evaluarea continuă a tuturor colțurilor. Cu toate acestea, aceasta consumă cea mai mare parte a bugetului de simulare pe colțuri care, de cele mai multe ori, sunt irelevante.

În acest capitol prezentăm diverse tehnici de gestionare a colțurilor: Mai întâi, discutăm abordarea de dimensionare în două etape în Secțiunea 6.1. Apoi, în Secțiunea 6.2, introducem mecanismul de selecție periodică a celor mai nefavorabile colțuri (*Periodic Worst-Corners Selection*). În Secțiunea 6.3, combinăm cele două metode pentru a construi un algoritm de dimensionare a circuitului orientat pe eficiență. În final, discutăm o altă paradigmă pentru gestionarea colțurilor în Secțiunea 6.4.

6.1 Dimensionarea Circuitelor în Două Etape

Un optimizator eficient, bazat pe algoritmi evolutivi îmbunătățiți prin modele de învățare automată, este MODEBI, prezentat în Capitolul 4. Acesta utilizează metrice diferite pentru soluțiile care îndeplinesc specificațiile (fezabile) și cele care nu le îndeplinesc, ceea ce duce la o abordare autentică de optimizare multi-obiectiv după identificarea

soluțiilor fezabile. Mai mult, algoritmul MODEBI folosește modele separate pentru fiecare răspuns al circuitului, rezultând în predicții mai precise.

Optimizarea pentru toate colțurile, așa cum se face în algoritmul MODEBI, este o abordare sigură. Aceasta garantează că optimizarea ia în considerare cel mai nefavorabil scenariu. Totuși, optimizarea pentru cel mai nefavorabil colț nu este cea mai eficientă strategie. Mai important, aceasta necesită un număr mare de simulări.

Proiectanții de circuite adoptă o abordare diferită atunci când dimensionează manual. Ei încep prin optimizarea pentru colțul nominal pentru a identifica candidați promițători pentru dimensionare. Apoi, iau în considerare condițiile de operare extreme. În această secțiune, propunem o metodă automată de dimensionare a circuitelor care imită această abordare, specifică proiectanților. Ca rezultat, în etapele inițiale ale optimizării, sunt utilizate mai puține simulări pentru candidații nepromițători, ceea ce este deosebit de important pentru circuite foarte instabile.

6.1.1 Abordarea „designer-like”

Optimizarea „designer-like” este o tehnică simplă inspirată de modul în care proiectanții de circuite analogice abordează dimensionarea circuitelor. Simplitatea acestei metode constă în procesul său în două etape. În prima etapă, circuitul este optimizat în colțul nominal de operare, permițând algoritmului să convergă către o zonă stabilă cu un număr redus de simulări. După ce o anumită condiție este îndeplinită, algoritmul intră în a doua fază de optimizare și începe să ia în considerare colțurile de operare extreme.

6.2 Mecanismul de Selecție Periodică a Colțului Nefavorabil (PWCS)

Reducerea numărului de cicluri proiectare-verificare este esențială pentru a asigura un timp de lansare pe piață cât mai scurt pentru noile circuite. De obicei, proiectanții selectează în avans cel mai nefavorabil colț PVT, bazându-se pe expertiza lor. Ulterior, se realizează dimensionarea circuitului pentru această condiție, iar la final designul este verificat în toate colțurile PVT pentru a-i demonstra robustețea. Această abordare minimizează numărul de simulări necesare pentru dimensionarea circuitului, dar adesea conduce la multiple cicluri de proiectare-verificare, deoarece soluția care era fezabilă pentru colțul PVT ales inițial se dovedește a nu fi fezabilă în toate celelalte colțuri.

O abordare „designer-like” adaptată pentru dimensionarea automată a circuitelor necesită o a doua etapă de optimizare conștientă de PVT, așa cum este prezentată în Secțiunea 6.1. Cu toate acestea, această abordare este costisitoare din punct de vedere computațional. În acest context, propunem integrarea verificării PVT în etapa de dimensionare a circuitelor într-un mod mai inteligent, selectând periodic colțurile PVT

cele mai nefavorabile și efectuând dimensionarea doar în acestea. Astfel, se pot obține rezultate similare cu optimizarea în toate colțurile, respectând bugetul de simulare.

6.2.1 Metodă

În MODEBI, am propus evaluarea tuturor colțurilor pentru toate soluțiile la fiecare iterație, în efortul de a minimiza numărul de cicluri proiectare-verificare care au loc înainte de a găsi o soluție adecvată. În mod obișnuit, în sarcinile de dimensionare a circuitelor, găsirea de soluții fezabile în unele colțuri poate necesita semnificativ mai multe simulări decât în altele. În esență, procesul de optimizare ajunge într-un punct în care majoritatea soluțiilor din populație sunt deja fezabile în unul sau mai multe colțuri PVT, dar întâmpină dificultăți în a găsi soluții fezabile în alte colțuri, mai dificile.

Astfel, propunem algoritmul MODEBI-CM, o îmbunătățire a algoritmului MODEBI care folosește un mecanism de *selecție a colțului nefavorabil*, astfel încât simulările să nu fie irosite pe colțuri ușoare care probabil respectă deja specificațiile, dacă acele *colțuri nefavorabile* le respectă.

6.3 Dimensionare Centrată pe Eficiență

Deși tehnicile menționate anterior (Secțiunile 6.1 și 6.2) au demonstrat eficiență, nu au existat încercări de a combina diferite strategii de accelerare a optimizării într-un singur algoritm de dimensionare. În această secțiune, introducem un algoritm de dimensionare automată centrat pe eficiență, care combină multiple abordări într-o singură metodă.

6.3.1 Metodă

Algoritmul de dimensionare automată a circuitelor centrat pe eficiență este versatil, aplicabil oricăror metode evolutive. Acesta se extinde dincolo de algoritmi evolutivi (EAs) tradiționali pentru a include și EAs îmbunătățiți prin învățare automată. Metoda eficientizează procesul de dimensionare pe trei niveluri: inițial, folosind eșantionarea Latin Hypercube (LHS) [53] pentru selecția populației inițiale, pentru a diminua impactul proceselor aleatorii și a reduce bugetul mediu necesar pentru optimizare [54]; începând optimizarea cu colțul nominal de operare pentru a obține un "start rapid" (*warm start*) (Secțiunea 6.1) la un cost minim de simulare; și apoi optimizând circuitul pe un subset actualizat periodic al celor mai dificile colțuri de operare (Secțiunea 6.2), validând cele mai promițătoare soluții în raport cu întregul set de colțuri.

Algoritmul 2 prezintă modul de lucru al algoritmului de dimensionare a circuitelor centrat pe eficiență. Deși mulți pași sunt simpli, unii necesită explicații suplimentare. „Arhiva” funcționează ca un depozit care filtrează automat soluțiile dominate, unde o soluție este considerată dominată dacă există alta cu performanțe superioare pe toate

obiectivele. Procesul de reducere (pruning) se realizează conform aceluiași criteriu de dominație. Pentru a diferenția între mai multe soluții nedominate, *Crowding Distance* [30] este folosită pentru a identifica cel mai divers set de soluții. Pasul de optimizare poate continua pentru un număr predeterminat de epoci sau până la atingerea unui număr specificat de soluții fezabile. Pentru pasul de selecție a celor mai nefavorabile colțuri, utilizatorii au flexibilitatea de a specifica numărul de colțuri dificile asupra cărora doresc să se concentreze.

Algorithm 2: Algoritm de dimensionare a circuitelor centrat pe eficiență

```

1 soluții = LHS.generează(dim_pop);
2 soluții = EA.optimizează(soluții, colț_nominal);
3 arhivă.initializează();
4 while True do
5     elite_pop = filtrează(soluții, dim_elită);
6     evaluează(elite_pop, toate_colțurile);
7     arhivă.adaugă(elite_pop);
8     if bugetul_maxim_atins then
9         | break;
10    end if
11    colțuri_nefavorabile = selectează(toate_colțurile, elite_pop);
12    soluții = filtrează(soluții+arhivă, dim_pop);
13    soluții = EA.optimizează(soluții, colțuri_nefavorabile);
14 end while
15 return arhivă

```

6.4 Programatorul de Simulări

Deși tehnicile prezentate anterior în acest capitol au demonstrat eficiență, ele împărtășesc un dezavantaj semnificativ: tratează toate soluțiile dintr-o populație în mod egal, indiferent de calitatea lor. Ideal ar fi să folosim simulări pentru soluții promițătoare, pentru a le valida calitatea, în timp ce eliminăm soluțiile de calitate scăzută pentru a economisi simulări.

În acest context, introducem un mecanism alternativ pentru gestionarea colțurilor, capabil să reducă numărul total de simulări necesare algoritmilor de dimensionare cu peste 70%. Această metodă folosește o listă de colțuri, ordonată în funcție de nivelul lor de dificultate. Soluțiile sunt evaluate secvențial pe aceste colțuri, iar evaluarea fiecărei soluții se oprește imediat ce aceasta nu îndeplinește specificațiile pentru un anumit colț. Prin urmare, această abordare minimizează numărul de simulări efectuate pentru soluțiile de calitate inferioară, concentrând eforturile de optimizare pe colțurile cele mai dificile.

6.4.1 Metodă

La baza tehnicii noastre de gestionare a colțurilor stă un programator de simulări. Pentru a funcționa optim, este esențială aranjarea listei de colțuri în ordine descrescătoare a nivelului lor de dificultate. Metoda propusă folosește Violarea Constrângerilor (CV) [55] pentru a clasifica colțurile de operare în funcție de dificultate. CV reprezintă media normalizată a abaterilor de la specificații a unei anumite soluții într-un colț de operare (Ecuția 6.1).

$$CV(x) = \frac{1}{N_{\text{constr}}} \sum_{i=1}^{N_{\text{constr}}} \frac{\text{deviation}_i(x)}{\text{reference}_i} \quad (6.1)$$

Pentru un set dat de soluții, se calculează media CV pentru fiecare colț. Colțurile cu cele mai mari valori ale CV sunt considerate colțuri dificile și sunt plasate la începutul clasamentului. Spre deosebire de mecanismul de Selecție Periodică a Colțului Nefavorabil (Secțiunea 6.2), metoda propusă realizează această clasificare o singură dată, la începutul procesului de optimizare, folosind setul inițial de soluții generate aleatoriu.

Funcționarea de bază a metodei noastre de programare este structurată după cum urmează: După ce algoritmul evolutiv produce un nou lot (populație) de soluții, acestea sunt evaluate mai întâi în colțul cel mai dificil, conform clasificării din lista de dificultate. Oricărei soluții care nu îndeplinește specificațiile pentru acest colț i se alocă o anumită valoare. Doar soluțiile care respectă cerințele în cel mai dificil colț continuă să fie evaluate în următorul colț din lista de dificultate. Procesul continuă până când fiecare soluție primește o valoare sau a fost evaluată în toate colțurile de pe listă. Dacă o soluție îndeplinește specificațiile pentru fiecare colț, aceasta primește o valoare egală cu zero.

Metoda propusă calculează această valoare (metrică) pe baza numărului de colțuri evaluate pentru o anumită soluție (Ecuția 6.2).

$$\text{Metric}(x) = N_{\text{corners}} - N_{\text{passed}}(x) + CV(x, N_{\text{passed}} + 1) \quad (6.2)$$

Mai exact, metrica se obține prin scăderea numărului de colțuri parcurse cu succes de soluție din numărul total de colțuri. Ulterior, CV (Ecuția 6.1) este calculat pentru colțul care a fost evaluat, dar nu a fost parcurs cu succes de soluție. Acest CV este apoi inclus în metrică pentru a face distincția între soluțiile care parcurg cu succes un număr identic de colțuri.

Capitolul 7

Concluzii

Dimensionarea automată a circuitelor este un subiect de actualitate datorită potențialului său de a reduce timpul de lansare pe piață a circuitelor electronice și de a îmbunătăți satisfacția angajaților. Această abordare automată își propune să înlocuiască procesul iterativ bazat pe experiența proiectanților cu algoritmi de optimizare multi-obiectiv. Cele mai populare tehnici pentru dimensionarea automată a circuitelor sunt metode hibride care combină modele de învățare automată cu motoare de căutare metaeuristice, precum algoritmi evolutivi. În mod particular, cele mai avansate metode se bazează pe algoritmi evolutivi îmbunătățiți prin învățare automată și pe optimizare bayesiană. Eforturile recente s-au concentrat pe îmbunătățirea acestor metode prin introducerea de tehnici noi care reduc bugetul de simulare, cum ar fi strategiile de gestionare a colțurilor.

În această teză, ne propunem să avansăm stadiul actual al cercetării prin prezentarea a patru studii-cheie. În primul rând, în Capitolul 3, analizăm cinci algoritmi evolutivi de ultimă generație pentru a identifica cel mai eficient pentru sarcinile de dimensionare a circuitelor. În al doilea rând, în Capitolul 4, propunem un nou algoritm evolutiv îmbunătățit prin învățare automată, care folosește Procese Gaussiene pentru a selecta cele mai promițătoare soluții generate pe baza Evoluției Diferențiale. În al treilea rând, în Capitolul 5, introducem o metodologie nouă pentru dimensionarea automată a circuitelor, care combină punctele forte ale algoritmilor evolutivi îmbunătățiți prin învățare automată cu cele ale optimizării bayesiene. În cele din urmă, în Capitolul 6, abordăm provocarea gestionării colțurilor de operare pentru a îmbunătăți eficiența metodelor de dimensionare a circuitelor. Pentru a demonstra performanța soluțiilor propuse, le comparăm cu algoritmi de ultimă generație folosind cinci circuite, dintre care trei sunt proiecte proprietare și două sunt circuite din literatură. Pe parcursul cercetării, ne-am concentrat nu doar pe performanță, ci și pe practicabilitatea și robustețea metodelor propuse.

Rezultatele prezentate în Capitolul 3 demonstrează că algoritmi evolutivi (EAs) reprezintă o bază eficientă și de încredere pentru sarcinile de dimensionare a circuitelor. Am evaluat cinci EAs, inspirați din biologie [56]: Algoritm Genetic cu Sortare Non-dominantă II (NSGAI) [30], Algoritm Genetic cu Sortare Non-dominantă III (NS-

GAlIII) [57], Evoluția Diferențială Generalizată 3 (GDE3) [44], Algoritm Evolutiv Bazat pe Indicatori (IBEA) [45], și Algoritm Evolutiv bazat pe Forță Pareto 2 (SPEA2) [58]. Evaluarea noastră a luat în considerare performanța, nivelul de conservare a diversității și versatilitatea acestora. Rezultatele evidențiază faptul că GDE3 prezintă performanțe consecvente în toate scenariile și excelează în conservarea diversității soluțiilor, în special datorită potențialului său de a performa bine folosind o populație numeroasă. Astfel, concluzionăm că GDE3 este cel mai promițător EA pentru dimensionarea automată a circuitelor. Deși EAs sunt foarte eficienți pentru explorarea spațiului parametrilor de design, folosirea lor exclusivă duce la un număr ridicat de simulări. Prin urmare, obiectivul nostru este de a dezvolta un algoritm de dimensionare a circuitelor mai eficient prin integrarea GDE3 cu un model surogat.

În Capitolul 4, am introdus un nou algoritm de optimizare multi-obiectiv pentru dimensionarea automată a circuitelor, numit Optimizare Multi-obiectiv bazată pe Evoluția Diferențială și Inferență Bayesiană (MODEBI). Această metodă utilizează un EA inspirat din GDE3 pentru a explora spațiul complex al variabilelor de design și al colțurilor de operare. În plus, folosește Procese Gaussiene ca model surogat pentru a reduce necesitatea simulărilor costisitoare ale circuitelor, ceea ce îmbunătățește semnificativ convergența. MODEBI este conceput în mod special pentru a aborda circuite cu multe variabile de design, multiple colțuri de operare și numeroase răspunsuri care trebuie să îndeplinească anumite specificații. Spre deosebire de majoritatea algoritmilor de ultimă generație, care convertesc problemele multi-obiectiv în unele mono-obiectiv, MODEBI utilizează dominația Pareto pentru a explora direct spațiul multi-obiectiv. Rezultatele demonstrează performanța sa superioară comparativ cu GDE3 și metoda de optimizare bayesiană, MACE [34]. Deși MODEBI este un algoritm puternic, acesta prezintă anumite limitări. În primul rând, are tendințe elitiste, ceea ce îl face mai puțin versatil și robust. În al doilea rând, performanța sa este destul de sensibilă la ajustarea hiperparametrilor, fapt care îi afectează practicalitatea.

Pentru a depăși limitările MODEBI și pentru a crea un algoritm care să fie atât eficient, cât și extrem de robust, am dezvoltat metodologia Evolutionary Bayesian Optimization (EBO) în Capitolul 5. Acest cadru integrează cele două abordări cele mai promițătoare din literatura de specialitate, combinându-le eficient avantajele. Dintr-o perspectivă, adoptă strategia de căutare globală a algoritmilor evolutivi îmbunătățiți prin învățare automată. Pe de altă parte, profită de puterea modelelor de învățare automată prin căutarea optimelor acestora, similar cu abordarea din Optimizarea Bayesiană. Metodologia EBO se bazează pe conceptul-cheie de „evoluție virtuală”. Acesta alternează între pași de evoluție reală și virtuală pentru a identifica soluții optime. La fel ca MODEBI, EBO folosește dominația Pareto pentru optimizarea multi-obiectiv și utilizează loturi mari de simulări. Pe de altă parte, spre deosebire de MODEBI, EBO rămâne fidel filozofiei intrinseci a EA, fără a modifica fluxul său de lucru. Această abordare minimizează numărul de hiperparametri și asigură conservarea diversității și robustețea atunci când

este asociată cu un EA potrivit. Rezultatele obținute pe patru circuite demonstrează superioritatea EBO față de MODEBI, GDE3 și MACE. Deși dezvoltarea de noi algoritmi de optimizare pentru dimensionarea circuitelor este interesantă, ne concentrăm atenția și spre îmbunătățirea performanței prin strategii precum gestionarea colțurilor de operare.

În Capitolul 6, propunem mai multe metode de îmbunătățire a performanței care intră sub umbrela „gestionării colțurilor de operare.” Colțurile de operare sunt obținute prin combinarea valorilor relevante ale condițiilor de operare. Pentru o optimizare robustă, fiecare configurație a circuitului trebuie validată în mai multe colțuri de operare. Gestionarea eficientă a acestor simulări poate reduce semnificativ bugetul de simulare. Abordarea sigură folosită în MODEBI și EBO validează toate configurațiile circuitului în fiecare colț de operare. Totuși, această abordare consumă rapid bugetul de simulare.

O tehnică de bază (Secțiunea 6.1) folosește optimizarea în colțul nominal până când este găsită regiunea de fezabilitate, după care se verifică toate colțurile pentru restul procesului de optimizare. Această abordare oferă un „start rapid” pentru optimizarea în colțuri, cu un cost minim de simulare. O metodă alternativă este mecanismul de selecție periodică a colțului nefavorabil (Secțiunea 6.2), care asigură robustetea algoritmului reducând semnificativ bugetul de simulare. În plus, combinarea acestor două metode, așa cum este prezentat în Secțiunea 6.3, maximizează eficiența.

Deși aceste metode sunt eficiente, ele au un dezavantaj-cheie: tratează toate configurațiile circuitului în mod egal, indiferent de calitate. Ideal, simulările ar trebui folosite pentru validarea soluțiilor promițătoare, eliminând rapid soluțiile de calitate inferioară. Programatorul de simulări descris în Secțiunea 6.4 evaluează secvențial configurațiile circuitului în funcție de colțuri, oprind evaluarea unei anumite configurații imediat ce aceasta nu îndeplinește specificațiile într-unul dintre colțurile de operare.

Principalul avantaj al tehnicilor de gestionare a colțurilor propuse este flexibilitatea lor — ele nu sunt legate de un algoritm specific de dimensionare. Acestea funcționează ca metode generale de îmbunătățire a performanței pentru orice metodă de dimensionare a circuitului bazată pe calcul evolutiv.

În concluzie, această teză avansează stadiul de dezvoltare a domeniului pe mai multe niveluri. Evaluăm performanța algoritmilor evolutivi în sarcini de dimensionare a circuitelor pentru a identifica abordarea cea mai promițătoare. Pe baza acestei evaluări, propunem două metode inovatoare de dimensionare a circuitelor care abordează limitările tehnicilor existente. În cele din urmă, dezvoltăm mai multe strategii de îmbunătățire a performanței pentru a crește și mai mult eficiența metodelor propuse.

7.1 Contribuții originale

Contribuțiile referitoare la analiza algoritmilor evolutivi în contextul dimensionării circuitelor electronice prezentate în Capitolul 3 sunt următoarele:

- Ajustarea hiperparametrilor pe benchmark-uri sintetice [41, 42]. Această activitate are ca scop determinarea celor mai bune configurații pentru algoritmi evolutivi de ultimă generație folosind probleme ușor de evaluat care se aseamănă cu sarcinile de dimensionare a circuitelor. Rezultatele furnizează o resursă valoroasă pentru orice cercetare axată pe dimensionarea automată a circuitelor prin calcul evolutiv.
- Analiza aprofundată a algoritmilor evolutivi de ultimă generație pe circuite analogice proprietare de ultimă generație [41, 42]. Această activitate oferă informații despre modul în care anumite caracteristici specifice ale algoritmilor evolutivi îi fac potriviți pentru dimensionarea circuitelor. Analiza ia în considerare atât performanța, cât și versatilitatea și robustețea algoritmilor.

Principalele contribuții referitoare la algoritmi evolutivi îmbunătățiți prin învățare automată, prezentate în Capitolul 4, sunt următoarele:

- O metodă inovatoare de optimizare multi-obiectiv care combină un algoritm bazat pe Evoluția Diferențială cu Procese Gaussiene. Această metodă este concepută pentru a excela în probleme cu un număr mare de variabile, obiective și restricții [46]. Algoritmul obține o accelerare de peste trei ori comparativ cu algoritmi evolutivi clasici și performează mai bine decât metodele de dimensionare a circuitelor de ultimă generație.
- Mai multe politici inovatoare de supraviețuire a populației și algoritmi de selecție a soluțiilor care minimizează numărul de simulări reale necesare pentru a finaliza procesul de optimizare [46, 47].

Contribuțiile referitoare la aplicarea optimizării bayesiene în dimensionarea circuitelor, prezentate în Capitolul 5, sunt următoarele:

- O metodologie nouă pentru dimensionarea automată a circuitelor, care combină algoritmi evolutivi îmbunătățiți prin ML cu optimizarea bayesiană. Metoda propusă integrează cele două abordări, exploatând punctele forte ale fiecăreia.
- Doi algoritmi de dimensionare a circuitelor dezvoltați conform acestei metodologii. Rezultatele arată îmbunătățiri semnificative în ceea ce privește performanța și robustețea, comparativ cu metodele de ultimă generație.
- Două studii de caz care implică circuite din literatură pentru a demonstra practicabilitatea acestei metodologii. Deși rezultatele pe circuitele proprietare arată că algoritmi pot face față problemelor complexe de dimensionare, anumite detalii nu pot fi divulgate din cauza naturii proprietare a acestora. Pentru a oferi o perspectivă mai largă, algoritmi propuși sunt testați și pe circuite populare din literatură.

Contribuțiile referitoare la îmbunătățirea performanței prin gestionarea colțurilor de operare, prezentate în Capitolul 6, sunt următoarele:

- Un mecanism periodic de selecție a colțurilor nefavorabile, care asigură robustetea algoritmului, reducând în același timp bugetul de simulare necesar pentru a obține soluții optime [49].
- Un algoritm de dimensionarea a circuitelor centrat pe eficiență [52] care combină mecanismul periodic de selecție a colțurilor nefavorabile cu o tehnică de dimensionare în două etape [50]. Această tehnică, inspirată din dimensionarea manuală a circuitelor, începe cu dimensionarea pentru colțul nominal, urmată de optimizarea conștientă de colțurile de operare.
- Un programator de simulări inovator [51] care evaluează secvențial configurațiile circuitelor în funcție de colțurile de operare, reducând numărul de simulări alocate soluțiilor de calitate inferioară.

7.2 Lista lucrărilor publicate

1. [41] Stănescu, M., Vișan, C., Sandu, G., Cucu, H., Diaconu, C., Buzo, A., & Pelz, G. (2021, October). Multi-objective optimization algorithms for automated circuit sizing of analog/mixed-signal circuits. In 2021 International Semiconductor Conference (CAS) (pp. 117-120). IEEE, ISI WOS:000853482700022
2. [42] Vișan, C., Pascu, O., Stănescu, M., Cucu, H., Diaconu, C., Buzo, A., & Pelz, G. (2021, October). Versatility and population diversity of evolutionary algorithms in automated circuit sizing applications. In 2021 International Conference on Speech Technology and Human-Computer Dialogue (SpeD) (pp. 68-73). IEEE, ISI WOS:000786794700013
3. [59] Pascu, O., Visan, C., Stănescu, M., Cucu, H., Diaconu, C., Buzo, A., & Pelz, G. (2022, October). Efficient Modeling of PVT Variation for Mixed-Signal Circuit Sizing. In 2022 International Semiconductor Conference (CAS) (pp. 105-108). IEEE.
4. [46] Vișan, C., Pascu, O., Stănescu, M., Șandru, E. D., Diaconu, C., Buzo, A., Pelz, G. & Cucu, H. (2022). Automated circuit sizing with multi-objective optimization based on differential evolution and Bayesian inference. Knowledge-Based Systems, 258, 109987, ISI WOS:000883752300005
5. [47] Vișan, C., Pascu, O., Stănescu, M., Cucu, H., Diaconu C., Buzo, A. & Pelz, G., (2022). IMPROVING MODEBI: MULTI-OBJECTIVE OPTIMIZATION BASED ON DIFFERENTIAL EVOLUTION AND BAYESIAN INFERENCE.

UPB Scientific Bulletin, Series C: Electrical Engineering and Computer Science, Vol. 84, Issue 4, Pg. 181-196, ISSN 2286-3540, ISI WOS:000907279800011

6. [49] Pascu, O., Visan, C., Nicolae, G., Boldeanu, M., Cucu, H., Diaconu, C., ... & Pelz, G. (2023, August). Efficient Multi-Objective Optimization for PVT Variation-Aware Circuit Sizing Using Surrogate Models and Smart Corner Sampling. In 2023 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED) (pp. 1-6). IEEE, ISI WOS:001073659300049
7. [50] Vişan, C., Sieberer, M., & Cucu, H. (2023, October). Designer-like Automated Circuit Sizing for Multiloop LDO. In 2023 International Semiconductor Conference (CAS) (pp. 103-106). IEEE.
8. [54] Nicolae, G., Visan, C., Curavale, D., Boldeanu, M., Cucu, H., Buzo, A., & Pelz, G. (2023, October). A Study on Initial Population Sampling for Multi-Objective Optimization based on Differential Evolution and Bayesian Inference. In 2023 International Conference on Speech Technology and Human-Computer Dialogue (SpeD) (pp. 128-132). IEEE.
9. [51] Visan, C., Curavale, D., Nicolae, G., Boldeanu, M., Cucu, H. & Buzo, A. (2024, June). A novel simulations scheduler for automated circuit sizing algorithms. In 2024 International Conference on Synthesis, Modeling, Analysis and Simulation Methods, and Applications to Circuit Design (SMACD)
10. [52] Visan, C., Pleşa, C.S. & Cucu, H. (2024, November). Efficiency-centric Automatic Circuit Design for Feed-forward Operational Amplifier. (ICECS), *accepted for publication*

7.3 Perspective de dezvoltare ulterioară

Metodele propuse în această teză sunt bine definite. Totodată, anumite aspecte specifice necesită dezvoltări ulterioare. Cel mai important este găsirea unui mecanism adaptiv pentru selectarea numărului potrivit de pași virtuali în cadrul Evolutionary Bayesian Optimization (Capitolul 5). În prezent, stabilim empiric numărul de pași virtuali per pas real, între 30 și 80, în funcție de problemă. Pentru a evita posibile disfuncționalități, mecanismul ar putea fi conceput astfel încât să aleagă în acest interval. O idee preliminară ar fi oprirea evoluției virtuale dacă descendenții deviază prea mult de la orice soluție evaluată de simulatorul circuitului real. Astfel, evităm încrederea excesivă în modelul de învățare automată în zone unde acesta ar putea fi nesigur.

O altă metodă care necesită atenție este programatorul de simulări descris în Secțiunea 6.4. Deoarece acesta evaluează configurațiile circuitului în mod secvențial, în funcție de colțurile de operare, metoda nu poate exploata pe deplin o infrastructură paralelă de

simulare. Ne propunem să dezvoltăm un sistem de organizare a loturilor de simulări care să maximizeze paralelismul. O soluție simplă ar fi completarea loturilor incomplete cu simulări ale colțurilor viitoare, deoarece acestea vor fi necesare la un moment dat. În plus, programatorul de simulări a fost aplicat doar în contextul algoritmilor evolutivi clasici. Ar fi util să explorăm acest mecanism în combinație cu metode de dimensionare mai avansate, cum ar fi algoritmi evolutivi îmbunătățiți prin ML (Capitolul 4) și tehnicile de Optimizare Bayesiană (Capitolul 5).

Selectarea populației inițiale este un aspect important de discutat, deoarece influențează toate metodele introduse în această teză. Până acum, ne-am bazat în principal pe eșantionarea aleatorie, o abordare standard în algoritmi evolutivi. Cu toate acestea, tehnici mai sofisticate precum Latin Hypercube Sampling [53] sau eșantionarea Sobol [60] ar putea oferi date mai relevante, așa cum se sugerează în [54].

Trecând de la îmbunătățiri specifice la direcții mai ample de explorare, începem cu modelarea circuitelor în timp real. Procesele Gaussiene sunt modele puternice care obțin o precizie impresionantă cu seturi de date mici. Cu toate acestea, ele nu se scalează bine pentru seturi de date mari, având o complexitate temporală de $O(N^3)$ pentru antrenare și $O(N^2)$ pentru predicție. În dimensionarea circuitelor, bugetul de simulare poate ajunge la zeci de mii, în timp ce Procesele Gaussiene gestionează eficient în jur de cinci până la zece mii de simulări. Un domeniu de explorat ar fi dezvoltarea unor mecanisme de selecție a punctelor de antrenare pentru seturi de date mai mari de zece mii de simulări.

Alternativ, alte modele de învățare automată pot fi integrate în algoritmi de dimensionare a circuitelor. De exemplu, am experimentat cu Random Forest [61], iar rezultatele preliminare indică o reducere substanțială a timpului de antrenare al modelului. Cu toate acestea, modelele Random Forest sunt mai puțin precise decât Procesele Gaussiene, ceea ce poate afecta uneori convergența. O altă direcție interesantă este explorarea Rețelelor Neuronale Artificiale. Am constatat empiric că Rețelele Neuronale nu sunt suficient de precise pentru seturi de date de antrenament mai mici de zece mii de simulări. Totuși, având în vedere progresele în modelele bazate pe Rețele Neuronale, această direcție ar trebui investigată în continuare.

Recent, Reinforcement Learning și Transfer Learning au câștigat popularitate în comunitatea de proiectare automată a circuitelor. O metodă importantă combină Rețele Neuronale Convoluționale pentru Grafuri cu Reinforcement Learning, introdusă inițial în [62] și dezvoltată în continuare în [63]. La prima vedere, această abordare oferă un avantaj structural față de metodele de calcul evolutiv și de optimizare bayesiană. Prin modelarea circuitului ca un graf, un agent de Reinforcement Learning antrenat pe un circuit poate fi aplicat și pe alte topologii. Cu suficient antrenament, un agent de Reinforcement Learning ar putea depăși performanța proiectanților umani și a metodelor automate de dimensionare a circuitelor care nu folosesc Transfer Learning.

Bibliografie

- [1] Cristian Manolache, Alexandru Caranica, Marius Stănescu, Horia Cucu, Andi Buzo, Cristian Diaconu, and Georg Pelz. Advanced operating conditions search applied in analog circuit verification. In *2022 18th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, pages 1–4. IEEE, 2022.
- [2] Alecsandra Rusu, Emilian David, Marina Țopa, Vasile Grosu, Andi Buzo, and Georg Pelz. Improvement and performance evaluation of an adaptive method for integrated circuits pre-silicon verification. In *2023 International Symposium on Signals, Circuits and Systems (ISSCS)*, pages 1–4. IEEE, 2023.
- [3] António Gusmão, Fábio Passos, Ricardo Póvoa, Nuno Horta, Nuno Lourenço, and Ricardo Martins. Semi-supervised artificial neural networks towards analog ic placement recommender. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2020.
- [4] B Carbunescu-Stoenescu, E David, M Topa, A Buzo, and G Pelz. Distribution modelling for yield analysis using variational autoencoders. In *2023 International Symposium on Signals, Circuits and Systems (ISSCS)*, pages 1–5. IEEE, 2023.
- [5] Vasile Grosu, Liviu Goras, Emilian David, and Georg Pelz. Using neural network based active learning for modelling integrated circuits behavior. In *2023 International Symposium on Signals, Circuits and Systems (ISSCS)*, pages 1–4. IEEE, 2023.
- [6] Vasile Grosu, Emilian David, Liviu Goras, and Georg Pelz. Modelling integrated circuits behavior using an active learning approach based on gaussian process regression. In *2023 International Semiconductor Conference (CAS)*, pages 245–248. IEEE, 2023.
- [7] Nuno Horta. Analogue and mixed-signal systems topologies exploration using symbolic methods. *Analog Integrated Circuits and Signal Processing*, 31:161–176, 2002.
- [8] Walter Daems, Georges Gielen, and Willy Sansen. Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(5):517–534, 2003.
- [9] Stephen P. Boyd and Seung Jean Kim. Geometric programming for circuit optimization. In *Proceedings of the 2005 International Symposium on Physical Design, ISPD '05*, page 44–46, New York, NY, USA, 2005. Association for Computing Machinery.

- [10] Abdelrahman Sayed, Ahmed Nader Mohieldin, and Mohsen Mahroos. A fast and accurate geometric programming technique for analog circuits sizing. In *2019 31st International Conference on Microelectronics (ICM)*, pages 316–319, 2019.
- [11] Ye Wang, Michael Orshansky, and Constantine Caramanis. Enabling efficient analog synthesis by coupling sparse regression and polynomial optimization. In *Proceedings of the 51st Annual Design Automation Conference*, pages 1–6, 2014.
- [12] W Lyu, P Xue, F Yang, C Yan, Z Hong, X Zeng, and D Zhou. An efficient bayesian optimization approach for automated optimization of analog circuits. *IEEE Trans. Circuits Syst. I Regul. Pap.*, 65(6):1954–1967, June 2018.
- [13] Bo Liu, Yan Wang, Zhiping Yu, Leibo Liu, Miao Li, Zheng Wang, Jing Lu, and Francisco V Fernández. Analog circuit optimization system based on hybrid evolutionary algorithms. *Integration*, 42(2):137–148, 2009.
- [14] R. Phelps, M. Krasnicki, R.A. Rutenbar, L.R. Carley, and J.R. Hellums. Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(6):703–717, 2000.
- [15] Jacopo Panerati, Donatella Sciuto, Giovanni Beltrame, et al. Optimization strategies in design space exploration. In *Handbook of Hardware/Software Codesign*, pages 189–216. Springer Netherlands, 2017.
- [16] R A Vural and T Yildirim. Analog circuit sizing via swarm intelligence. *AEU - International Journal of Electronics and Communications*, 66(9):732–740, September 2012.
- [17] Vipul Kumar Mishra and Anirban Sengupta. Mo-pse: Adaptive multi-objective particle swarm optimization based design space exploration in architectural synthesis for application specific processor design. *Advances in Engineering Software*, 67:111–124, 2014.
- [18] B Liu, N Deferm, D Zhao, P Reynaert, and G G E Gielen. An efficient High-Frequency linear RF amplifier synthesis method based on evolutionary computation and machine learning techniques. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 31(7):981–993, July 2012.
- [19] Bo Liu, Dixian Zhao, Patrick Reynaert, and Georges G. E. Gielen. Gaspad: A general and efficient mm-wave integrated circuit synthesis method based on surrogate model assisted evolutionary algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(2):169–182, 2014.
- [20] Ahmet Faruk Budak, Miguel Gandara, Wei Shi, David Z Pan, Nan Sun, and Bo Liu. An efficient analog circuit sizing method based on machine learning assisted global optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(5):1209–1221, 2021.
- [21] Sen Yin, Ruitao Wang, Jian Zhang, Xiaosen Liu, and Yan Wang. Fast surrogate-assisted constrained multi-objective optimization for analog circuit sizing via self-adaptive incremental learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022.

- [22] Bo Liu, Hadi Aliakbarian, Soheil Radiom, Guy A E Vandenbosch, and Georges Gielen. Efficient multi-objective synthesis for microwave components based on computational intelligence techniques. In *Proceedings of the 49th Annual Design Automation Conference, DAC '12*, pages 542–548, New York, NY, USA, 2012. ACM.
- [23] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.
- [24] Matthias Seeger. Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106, 2004.
- [25] Rainer Storn and Kenneth Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341, 1997.
- [26] Nuno Lourenço and Nuno Horta. GENOM-POF: multi-objective evolutionary synthesis of analog ICs with corners validation. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation, GECCO '12*, pages 1119–1126, July 2012.
- [27] Nuno Lourenço, Ricardo Martins, António Canelas, Ricardo Póvoa, and Nuno Horta. AIDA: Layout-aware analog circuit-level sizing with in-loop layout generation. *Integration*, 55:316–329, September 2016.
- [28] A Canelas, R Martins, R Povoá, N Lourenço, and N Horta. Efficient yield optimization method using a variable K-Means algorithm for analog IC sizing. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, pages 1201–1206, March 2017.
- [29] Kalyanmoy Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2):311–338, 2000.
- [30] K Deb, A Pratap, S Agarwal, and T Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.*, 6(2):182–197, April 2002.
- [31] Manuel Barros, Jorge Guilherme, and Nuno Horta. Ga-svm feasibility model and optimization kernel applied to analog ic design automation. In *Proceedings of the 17th ACM Great Lakes symposium on VLSI*, pages 469–472, 2007.
- [32] Bo Liu, Francisco V Fernández, Qingfu Zhang, Murat Pak, Suha Sipahi, and Georges Gielen. An enhanced moea/d-de and its application to multiobjective analog cell sizing. In *IEEE Congress on Evolutionary Computation*, pages 1–7. IEEE, 2010.
- [33] Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *ArXiv*, abs/1012.2599, 2010.
- [34] Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. Batch bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design. In *International Conference on Machine Learning*, pages 3306–3314. proceedings.mlr.press, 2018.
- [35] Matthew D. Hoffman, Eric Brochu, and Nando de Freitas. Portfolio allocation for bayesian optimization. *ArXiv*, abs/1009.5419, 2010.

- [36] Biao He, Shuhan Zhang, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. An efficient bayesian optimization approach for analog circuit synthesis via sparse gaussian process modeling. In *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 67–72, 2020.
- [37] Ke Li, Renzhi Chen, Guangtao Fu, and Xin Yao. Two-archive evolutionary algorithm for constrained multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 23(2):303–315, 2018.
- [38] S Venkatraman and G G Yen. A generic framework for constrained optimization using genetic algorithms. *Trans. Evol. Comp*, 9(4):424–435, August 2005.
- [39] Xin Li, Padmini Gopalakrishnan, Yang Xu, and Lawrence T. Pileggi. Robust analog/rf circuit design with projection-based performance modeling. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26:2–15, 2007.
- [40] S K Goudos, K Siakavara, T Samaras, E E Vafiadis, and J N Sahalos. Self-Adaptive differential evolution applied to Real-Valued antenna and microwave design problems. *IEEE Trans. Antennas Propag.*, 59(4):1286–1298, April 2011.
- [41] M Stanescu, C Visan, G Sandu, H Cucu, C Diaconu, A Buzo, and G Pelz. Multi-objective optimization algorithms for automated circuit sizing of analog/ mixed-signal circuits. In *2021 International Semiconductor Conference (CAS)*, pages 1–4. IEEE, October 2021.
- [42] C Visan, O Pascu, M Stanescu, H Cucu, C Diaconu, A Buzo, and G Pelz. Versatility and population diversity of evolutionary algorithms in automated circuit sizing applications. In *2021 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pages 1–6. IEEE, October 2021.
- [43] Simon Huband, Luigi Barone, Lyndon While, and Phil Hingston. A scalable multi-objective test problem toolkit. In *Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005. Proceedings 3*, pages 280–295. Springer, 2005.
- [44] S Kukkonen and J Lampinen. GDE3: the third evolution step of generalized differential evolution. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 443–450 Vol.1, September 2005.
- [45] Eckart Zitzler and Simon Künzli. Indicator-Based selection in multiobjective search. In *Parallel Problem Solving from Nature - PPSN VIII*, pages 832–842. Springer Berlin Heidelberg, 2004.
- [46] Catalin Visan, Octavian Pascu, Marius Stanescu, Elena-Diana Sandru, Cristian Diaconu, Andi Buzo, Georg Pelz, and Horia Cucu. Automated circuit sizing with multi-objective optimization based on differential evolution and bayesian inference. *Knowledge-Based Systems*, 258:109987, 2022.
- [47] Catalin Visan, Octavian Pascu, Marius Stanescu, Horia Cucu, Cristian Diaconu, Andi Buzo, and Georg Pelz. Improving modebi: Multi-objective optimization based on differential evolution and bayesian inference. *UPB Scientific Bulletin*, 84(4), 2022.
- [48] Kai Zheng, Ren-Jye Yang, Hongyi Xu, and Jie Hu. A new distribution metric for comparing pareto optimal solutions. *Struct. Multidiscip. Optim.*, 55(1):53–62, January 2017.

- [49] Octavian Pascu, Catalin Visan, Georgian Nicolae, Mihai Boldeanu, Horia Cucu, Cristian Diaconu, Andi Buzo, and Georg Pelz. Efficient multi-objective optimization for pvt variation-aware circuit sizing using surrogate models and smart corner sampling. In *2023 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 1–6, 2023.
- [50] Cătălin Vișan, Michael Sieberer, and Horia Cucu. Designer-like automated circuit sizing for multiloop ldo. In *2023 International Semiconductor Conference (CAS)*, pages 103–106, 2023.
- [51] Catalin Visan, Dan Curavale, Georgian Nicolae, Mihai Boldeanu, Horia Cucu, and Andi Buzo. A novel simulations scheduler for automated circuit sizing algorithms. In *2024 20th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, pages 1–4. IEEE, 2024.
- [52] Catalin Visan, Cosmin-Sorin Plesa, and Horia Cucu. Efficiency-centric automatic circuit design for feed-forward operational amplifier. In *Proceedings of the International Conference on Electronics, Circuits, and Systems (ICECS)*, November 2024. Accepted for publication.
- [53] Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.
- [54] Georgian Nicolae, Catalin Visan, Dan Curavale, Mihai Boldeanu, Horia Cucu, Andi Buzo, and Georg Pelz. A study on initial population sampling for multi-objective optimization based on differential evolution and bayesian inference. In *2023 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pages 128–132, 2023.
- [55] Sanyou Zeng, Ruwang Jiao, Changhe Li, Xi Li, and Jawdat S Alkasassbeh. A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization. *IEEE transactions on Cybernetics*, 47(9):2678–2688, 2017.
- [56] Sinem Akyol and Bilal Alatas. Plant intelligence based metaheuristic optimization algorithms. *Artif Intell*, 47:417–462, 2017.
- [57] K Deb and H Jain. An evolutionary Many-Objective optimization algorithm using Reference-Point-Based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Trans. Evol. Comput.*, 18(4):577–601, August 2014.
- [58] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103, 2001.
- [59] Octavian Pascu, Cătălin Visan, Marius Stănescu, Horia Cucu, Cristian Diaconu, Andi Buzo, and Georg Pelz. Efficient modeling of pvt variation for mixed-signal circuit sizing. In *2022 International Semiconductor Conference (CAS)*, pages 105–108, 2022.
- [60] Ilya M. Sobol. On the distribution of points in a cube and the approximate evaluation of integrals. *Ussr Computational Mathematics and Mathematical Physics*, 7:86–112, 1967.
- [61] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

- [62] Hanrui Wang, Jiacheng Yang, Hae-Seung Lee, and Song Han. Learning to design circuits. *arXiv preprint arXiv:1812.02734*, 2018.
- [63] Hanrui Wang, Kuan Wang, Jiacheng Yang, Linxiao Shen, Nan Sun, Hae-Seung Lee, and Song Han. Gcn-rl circuit designer: Transferable transistor sizing with graph neural networks and reinforcement learning. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2020.