



National Technical University of Athens

School of Electrical and Computer Engineering

Division of Communication, Electronic and Information Engineering

# Computational Methods for Automatic Analog Integrated Circuit Design

PhD Thesis

Konstantinos Touloupas

Supervisor

Paul Peter Sotiriadis, Professor, NTUA

Circuits & Systems Group

Athens, December 2022





Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών  
Τομέας Επικοινωνιών, Ηλεκτρονικής & Συστημάτων Πληροφορικής

Υπολογιστικοί Μέθοδοι για την  
Αυτόματη Σχεδίαση Αναλογικών  
Μικροηλεκτρονικών Κυκλωμάτων

Διδακτορική Διατριβή  
του

Κωνσταντίνου Τουλούπα

Επιβλέπων

Παύλος Πέτρος Σωτηριάδης, Καθηγητής, ΕΜΠ

Circuits & Systems Group

Αθήνα, Δεκέμβριος 2022







National Technical University of Athens  
School of Electrical and Computer Engineering  
Division of Communication, Electronic and Information Engineering

# Computational Methods for Automatic Analog Integrated Circuit Design

PhD Thesis by

**Konstantinos Touloupas**

**Advisory Committee:**

Paul Peter Sotiriadis      Nikolaos Maratos      Ioanna Roussaki  
Professor, NTUA      Professor Emeritus, NTUA      Assistant Professor, NTUA

Approved by the seven-member examination committee on 14/12/2022.

.....	.....	.....	.....
Paul Peter Sotiriadis	Ioanna Roussaki	Athanasios D. Panagopoulos	Dimitra I. Kaklamani
Professor	Assistant Professor	Professor	Professor
NTUA	NTUA	NTUA	NTUA

.....	.....	.....
Nuno Calado Correia Lourenço	Giorgos Stamou	Stefanos Kollias
Researcher	Professor	Professor
Instituto de Telecomunicações	NTUA	NTUA

.....

**Κωνσταντίνος Τουλούπας**

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Κωνσταντίνος Τουλούπας, 2022

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

# Abstract

For decades, the semiconductor and electronics industry have seen great progress, fueled by the continuous scaling of transistor dimensions. Integrated circuits in the sub- $\mu\text{m}$  range have been extensively utilized in the electronics industry. Nowadays, with Moore's law coming to an end, transistors' gates have reached unprecedented lengths. The eventual power and speed gains, however, come with an increase in complexity and in design considerations; random variations in the manufacturing process induce variations in circuit device parameters and effectively lead to low-yield designs. Design verification is constantly becoming more cumbersome for circuit design, especially in the case of analog circuits.

Circuit designers have traditionally resorted to Electronic Design Automation (EDA) tools for complex circuit design. Given sets of device compact models, named Process Development Kits (PDKs), EDA tools can simulate complex circuits and can be used for verification purposes. In the case of digital circuitry, established EDA tools provide automation solutions for designers to avoid cumbersome, repetitive tasks and focus on the core design. Analog and Radio-Frequency (RF) circuit design, however, has no established means of automation.

This thesis presents methodologies for analog and RF circuit automatic sizing. From a high-level perspective, the contributions of this work lie in two factors; 1) The proposal of a family of black-box optimization algorithms, which take advantage of recent machine learning developments to accelerate and improve the exploration of the circuit's design space, and 2) the development of a framework for procedural simulation execution and optimization definition, based on commercial circuit simulators. The proposed framework exposes a user-friendly Application Process Interface (API) that can be used by designers to execute ad-hoc optimization problems and guide the sizing of their circuit.

The first thrust of this thesis is the study of automatic circuit sizing in the context of black-box simulation-driven optimization. We apply and compare black-box optimization algorithms for the nominal sizing of analog and RF circuits, compare their performance and discuss their ability to provide feasible solutions within given evaluation budgets or time-frames. Taking into account that most black-box algorithms operate on continuous spaces, we define a new mutation and crossover operation for Evolutionary Algorithms (EAs) and apply it for circuit sizing. The aforementioned principles are studied for both the case of Single-Objective (SO) optimization and Multi-Objective

(MO) one, when design-space exploration and feasible performance space needs to be found.

To reduce the cost of optimization in the sense of reducing the number of costly evaluations, we consider next the case of low budget optimization algorithms. In this setting, a new SO Bayesian Optimization (BO) algorithm is introduced. The use of Gaussian Processes (GPs) and a new, batched acquisition function relying on Thompson Sampling (TS) reduces the effective time for each optimization run. Taken into account the fact Gaussian Processes require  $\mathcal{O}(n^3)$  time for inference, kernel approximations are introduced to the GPs by using inducing points. In addition to the above, a new framework is proposed in which the GP models are restricted to model certain hypercubes of the design space. This approach, which is motivated by the concept of trust-regions in the EA literature, provides exquisite constraint-handling capabilities, is scalable in terms of input parameter space and proves favorable against other BO approaches as well as other EA algorithms.

To extend the concept of local-based BO in the case of multiple objectives, a new, batched, Local Constrained Multi-Objective Bayesian Optimization (LoCoMOBO) approach is put forward. This MO optimization algorithm not only assists designers to size circuit-blocks, but also to assess the attainable performance metrics of a given topology. LoCoMOBO utilizes trust regions, and uses a Hypervolume-based acquisition function to define future query points for evaluation. In addition, TS is replaced with Random Fourier Features so as to ensure both that the design space is properly explored, in high-dimensional spaces.

To efficiently traverse mixed-variable input-spaces, where some parameters are continuous while others are integer-valued or categorical ones, a deep learning scheme that derives continuous representations of integrated devices is put forward. The core model is a Variational Autoencoder (VAE), which uses label guidance to transform the devices' input parameters to continuous-valued latent ones. The original device parameters are substituted by the VAE's latent variables in the optimization-based automatic sizing formulation, which is solved using the proposed local-based BO with satisfactory results.

In the final chapter, the conclusions of the conducted research are drawn, guidelines for future work are provided and potential impact on industry and society is discussed.

**Index Terms** Analog Integrated Circuits, automatic circuit sizing, black-box optimization, Evolutionary Algorithms, Bayesian Optimization, Gaussian Processes, Thompson Sampling, Random Fourier Features, Deep Learning, Variational Autoencoder.

# Περίληψη

Για δεκαετίες, η βιομηχανία ημιαγωγών και ηλεκτρονικών έχει επιδείξει μεγάλη ανάπτυξη, κυρίως λόγω της αδιάκοπης συρρίκνωσης των διαστάσεων των τρανζίστορ. Ολοκληρωμένα κυκλώματα σε τεχνολογίες κάτω του ενός μικρόμετρου έχουν χρησιμοποιηθεί ευρέως σε εμπορικές ηλεκτρονικές συσκευές. Στις μέρες μας, καθώς ο νόμος του Moore πλησιάζει στο τέλος του, η διάσταση των τρανζίστορ έχει φτάσει σε πολύ μικρά μεγέθη. Τα κέρδη σε ενεργειακή απόδοση και ταχύτητα που προκύπτουν, όμως, επιφυλάσσουν δυσκολίες στην σχεδίαση: οι μη ιδανικές συνθήκες κατά την διαδικασία κατασκευής προκαλούν διακυμάνσεις στις παραμέτρους των τρανζίστορ του κυκλώματος και οδηγούν ουσιαστικά σε κυκλώματα χαμηλής απόδοσης. Η επαλήθευση της λειτουργίας ενός κυκλώματος γίνεται συνεχώς πιο επίπονη για τους σχεδιαστές, ιδιαίτερα στην περίπτωση των αναλογικών ολοκληρωμένων κυκλωμάτων.

Οι σχεδιαστές κυκλωμάτων παραδοσιακά χρησιμοποιούν εργαλεία αυτοματοποίησης ηλεκτρονικού σχεδιασμού (Electronic Design Automation - EDA) για την σχεδίαση περίπλοκων κυκλωμάτων. Με δεδομένο ένα σύνολο από μοντέλα των διατάξεων ημιαγωγών, τα οποία αναφέρονται ως Process Development Kits (PDKs), τα εργαλεία αυτά μπορούν να προσομοιώνουν περίπλοκα κυκλώματα και να χρησιμοποιηθούν για επαλήθευση. Στην περίπτωση των ψηφιακών ολοκληρωμένων κυκλωμάτων, καθιερωμένα εργαλεία σχεδίασης EDA χρησιμοποιούνται από τους σχεδιαστές για να την εκπόνηση επαναλαμβανόμενων διεργασιών, γεγονός που διασφαλίζει την αποδοτική την εργασία τους και την ποιότητα του τελικού προϊόντος. Στην περίπτωση των αναλογικών και τηλεπικοινωνιακών κυκλωμάτων, όμως, παρόμοια εργαλεία δεν είναι διαθέσιμα.

Η διατριβή αυτή παρουσιάζει μεθοδολογίες για την αυτόματη διαστασιοποίηση αναλογικών και τηλεπικοινωνιακών κυκλωμάτων. Από γενικής άποψης, η συνεισφορά της δουλειάς που παρουσιάζεται εντοπίζεται σε δύο σημεία: 1) Την πρόταση μιας οικογένειας αλγορίθμων βελτιστοποίησης μαύρου κουτιού (black box), οι οποίοι χρησιμοποιούν πρόσφατες τεχνικές μηχανικής μάθησης για να επιταχύνουν και να βελτιώσουν την εξερεύνηση του χώρου λύσεων των προβλημάτων βελτιστοποίησης κυκλωμάτων και 2) την ανάπτυξη ενός εργαλείου για την διαδικαστική εκτέλεση προσομοιώσεων και τον ορισμό προβλημάτων βελτιστοποίησης, με χρήση εμπορικών προσομοιωτών. Το εργαλείο προσφέρει μία φιλική προς τον χρήστη διεπαφή διαδικασίας εφαρμογής (API) που μπορεί να χρησιμοποιηθεί από τους σχεδιαστές για την εκτέλεση προβλημάτων βελτιστοποίησης και να καθοδηγήσει την διαστασιοποίηση των στοιχείων του κυκλώματος.

Το πρώτο μέρος της διατριβής περιλαμβάνει την μελέτη της αυτόματης διαστασιοποι-

ησης κυκλωμάτων με χρήση βελτιστοποίηση μαύρου κουτιού και εμπορικών προσομοιωτών. Εφαρμόζουμε και συγκρίνουμε αλγόριθμους μαύρου κουτιού για την τυπική, χωρίς διακυμάνσεις, διαστασιοποίηση αναλογικών και τηλεπικοινωνιακών κυκλωμάτων και σχολιάζουμε τις αποδόσεις τους ως προς την εύρεση εφικτών λύσεων μέσα σε χρονικά ή υπολογιστικά όρια. Δεδομένου ότι η πλειοψηφία των αλγορίθμων μαύρου κουτιού εφαρμόζεται σε συνεχείς χώρους αναζήτησης, ορίζουμε καινούριους τελεστές διασταύρωσης και μετάλλαξης για Εξελικτικούς Αλγόριθμους και τους χρησιμοποιούμε στη διαστασιοποίηση κυκλωμάτων. Οι τελεστές αυτοί εξετάζονται τόσο στα πλαίσια βελτιστοποίησης ενός όσο και πολλαπλών στόχων, όπου ισχύουν περιορισμοί για την απόδοση των κυκλωμάτων.

Για να μειωθεί το κόστος της βελτιστοποίησης όσον αφορά τον αριθμό των κοστοβόρων προσομοιώσεων, εξετάζουμε επίσης την κλάση των αλγορίθμων βελτιστοποίησης χαμηλής δειγματοληψίας του χώρου αναζήτησης. Σε αυτή την περίπτωση, ένας νέος αλγόριθμος Μπεύζιανής βελτιστοποίησης ενός στόχου προτείνεται. Η χρήση Γκαουσιανών διαδικασιών και μιας νέας συνάρτησης απόκτησης που βασίζεται στην τεχνική Thompson Sampling μειώνει αποτελεσματικά τον χρόνο εκτέλεσης της βελτιστοποίησης. Δεδομένου ότι οι Γκαουσιανές διαδικασίες έχουν  $O(n^3)$  πολυπλοκότητα για πρόβλεψη, περιλαμβάνουμε ένα προσεγγιστικό πυρήνα με χρήση σημείων επιρροής. Επιπρόσθετα, προτείνουμε μία καινούρια μεθοδολογία στην οποία οι Γκαουσιανές διαδικασίες περιορίζονται εντός υπερκύβων στον χώρο αναζήτησης. Αυτή η προσέγγιση, που αντλεί έμπνευση από την ιδέα της περιοχής εμπιστοσύνης στην βιβλιογραφία των Εξελικτικών Αλγορίθμων, παρουσιάζει εξαιρετική δυνατότητα στην εύρεση εφικτών λύσεων, κλιμακώνεται σε σχέση με την διάσταση του χώρου αναζήτησης και αποδεικνύεται προτιμιτέα σε σχέση με άλλους αλγόριθμους Μπεύζιανής βελτιστοποίησης καθώς και άλλους Εξελικτικούς αλγόριθμους.

Για την επέκταση της ιδέας της τοπικής Μπεύζιανής βελτιστοποίησης σε προβλήματα πολλαπλών στόχων, προτείνεται μία νέα, παράλληλη, τοπική Μπεύζιανή Βελτιστοποίηση Πολλαπλών στόχων με περιορισμούς (Local Constrained Multi Objective Bayesian Optimization - LoCoMOBO). Ο αλγόριθμος αυτός πολλαπλών στόχων βοηθά τους σχεδιαστές όχι μόνο να διαστασιοποιήσουν τα κυκλώματά τους αυτόματα, αλλά και να έχουν εκτιμήσεις για τις εφικτές αποδόσεις διαφόρων κυκλωματικών τοπολογιών. Ο LoCoMOBO χρησιμοποιεί περιοχές εμπιστοσύνης και μια συνάρτηση απόκτησης που χρησιμοποιεί τιμές υπερόγκου για να ορίσει τα επόμενα σημεία που θα γίνουν προσομοίωση. Επιπλέον, η διαδικασία Thompson Sampling χρησιμοποιεί Τυχαία Χαρακτηριστικά Φουριέ έτσι ώστε να γίνεται καλύτερη εξερεύνηση του χώρου εισόδου, ακόμη και σε πολλές διαστάσεις.

Για την αποτελεσματική εξερεύνηση χώρων αναζήτησης με μικτού τύπου μεταβλητές, όπου ορισμένες παράμετροι είναι συνεχείς ενώ άλλες είναι ακέραιοι αριθμοί ή κατηγορικές μεταβλητές, προτείνεται μία μέθοδος βαθιάς μηχανικής μάθησης που παρέχει συνεχείς αναπαραστάσεις για τα κυκλωματικά στοιχεία. Το κυρίως μοντέλο είναι ένας εναλλασσόμενος

---

κωδικοποιητής που χρησιμοποιεί καθοδήγηση επιγραφής για να μετατρέψει τις παραμέτρους των στοιχείων σε συνεχείς μεταβλητές. Οι λανθάνουσες μεταβλητές του μοντέλου, που είναι συνεχείς, αντικαθιστούν τις μεταβλητές των στοιχείων στον ορισμό του προβλήματος βελτιστοποίησης. Η επίλυση του προβλήματος με τον νέο ορισμό γίνεται με την προτεινόμενη Μπεϋζιανή Βελτιστοποίηση και επιδεικνύει ικανοποιητικά αποτελέσματα.

Στο τελικό κεφάλαιο, τα συμπεράσματα της έρευνας που διεξήχθη διατυπώνονται, κατευθύνσεις για μελλοντικές εξελίξεις της δουλειάς αναφέρονται και συζητείται η πιθανή επιρροή της δουλειάς αυτής στην βιομηχανία και την κοινωνία.

**Λέξεις Κλειδιά** Αναλογικά ολοκληρωμένα κυκλώματα, αυτόματη διαστασιοποίηση κυκλωμάτων, βελτιστοποίηση μαύρου κουτιού, Εξελικτικοί Αλγόριθμοι, Μπεϋζιανή Βελτιστοποίηση, Γκαουσιανές Διαδικασίες, Thompson Sampling, Βαθιά Μάθηση, Εναλλασόμενος Κωδικοποιητής





# Εκτεταμένη Περίληψη

Η παρούσα διατριβή πραγματεύεται την διατύπωση της διαδικασίας διαστασιοποίησης ολοκληρωμένων αναλογικών και υψίσυχνων κυκλωμάτων ως πρόβλημα βελτιστοποίησης καθώς και την ανάπτυξη αλγορίθμων για την αυτόματη επίλυσή του.

Ολοκληρωμένα κυκλώματα, η αλλιώς τσιπ, είναι σύνολα από ηλεκτρονικά στοιχεία τα οποία κατασκευάζονται σε ένα μικρό, επίπεδο κομμάτι ημιαγωγού, συνήθως πυριτίου. Στις μέρες μας, μεγάλοι αριθμοί μικροσκοπικών MOSFET (τρανζίστορ πεδίου δράσης οξειδίου μετάλλου-ημιαγωγών) ενσωματώνονται σε μικρά τσιπ, οδηγώντας σε κυκλώματα που είναι τάξεις μεγέθους μικρότερα, ταχύτερα και λιγότερο ακριβά από αυτά που κατασκευάζονται από παραδοσιακά διακριτά ηλεκτρονικά εξαρτήματα. Στα πλεονεκτήματά τους συγκαταλέγεται η δυνατότητα μαζικής και αξιόπιστης παραγωγής, γεγονός που έχει διασφαλίσει την ταχεία υιοθέτηση τυποποιημένων τσιπ στη θέση των σχεδίων που χρησιμοποιούν διακριτά τρανζίστορ. Τα τσιπ χρησιμοποιούνται πλέον σχεδόν σε όλο τον ηλεκτρονικό εξοπλισμό και έχουν φέρει επανάσταση στον κόσμο των ηλεκτρονικών. Οι υπολογιστές, τα κινητά τηλέφωνα και άλλες οικιακές συσκευές αποτελούν πλέον αναπόσπαστα μέρη των σύγχρονων κοινωνιών, και γίνονται εφικτά από το μικρό μέγεθος και το χαμηλό κόστος ηλεκτρονικών συστημάτων, όπως οι σύγχρονοι επεξεργαστές υπολογιστών και οι μικροελεγκτές.

Η ενσωμάτωση πολύ μεγάλης κλίμακας (Very Large Scale Integration - VLSI), δηλαδή η κατασκευή ολοκληρωμένων κυκλωμάτων με εκατοντάδες χιλιάδες έως δισεκατομμύρια τρανζίστορ, είναι δυνατή χάρη στις τεχνολογικές εξελίξεις στην κατασκευή στοιχείων ημιαγωγών μετάλλου-οξειδίου-πυριτίου (MOS). Από την κατασκευή του πρώτου τσιπ την δεκαετία του 1960, η ταχύτητα και το μέγεθός τους βελτιώνονται σταθερά. Ο κύριος λόγος για αυτό το φαινόμενο είναι η συρρίκνωση της ελάχιστης διάστασης των τρανζίστορ που μπορούν να κατασκευαστούν. Εμπειρικά, τα στοιχεία που περιέχονται στα τσιπ συνεχώς γίνονται πιο μικρά, γεγονός που οδηγεί στον διπλασιασμό του αριθμού των τρανζίστορ που περιλαμβάνει ένα τσιπ κάθε διετία. Η πιο πυκνή ολοκλήρωση τρανζίστορ στα τσιπ οδηγεί σε πλεονεκτήματα όπως χαμηλή κατανάλωση και υψηλότερη ταχύτητα. Για την αγορά ημιαγωγών, όμως, η συρρίκνωση των διαστάσεων επέφερε σημαντική οικονομική ανάπτυξη, λόγω της δυνατότητας για ενσωμάτωση πολλών, διαφορετικών και πολύπλοκων λειτουργιών στα συγχρόνα τσιπ και της υιοθέτησής τους από πολλές διαφορετικές βιομηχανίες.

Υπάρχουν δύο κύριοι τύποι ηλεκτρονικών κυκλωμάτων που μπορούν να ολοκληρωθούν σε τσιπ, τα αναλογικά και τα ψηφιακά κυκλώματα. Τα ψηφιακά κυκλώματα λειτουργούν σε σήματα που έχουν διακριτές τιμές, κυρίως δυαδικές. Αντίθετα, τα αναλογικά κυκλώματα λειτουργούν σε ηλεκτρικά σήματα με συνεχείς τιμές. Ο ρόλος των αναλογικών κυκλωμάτων

στα τσιπ είναι η υλοποίηση διεπαφών με το ψηφιακό κομμάτι και το εξωτερικό περιβάλλον. Αντιθέτως, τα ψηφιακά κυκλώματα υλοποιούν τις περισσότερες λειτουργίες και καταλαμβάνουν μεγαλύτερη επιφάνεια σε σχέση με τα αναλογικά. Όμως, τα αναλογικά κυκλώματα είναι ο δυσκολότερος και πιο απαιτητικός παράγοντας κατά το σχεδιασμό ολοκληρωμένων συστημάτων, γεγονός που οφείλεται στην φύση τους. Ενώ τα ψηφιακά κυκλώματα χρησιμοποιούν σήματα με δύο διακριτές τιμές και είναι πιο ανθεκτικά σε μη-ιδανικά φαινόμενα, η σχεδίαση αναλογικών κυκλωμάτων πρέπει να γίνεται με βάση τη φυσική που διέπει την κατασκευή των ολοκληρωμένων κυκλωμάτων και απαιτεί ειδικές δεξιότητες.

Η σχεδίαση των αναλογικών ολοκληρωμένων κυκλωμάτων είναι απαιτητική και για έναν επιπλέον λόγο. Ο χώρος σχεδίασης, δηλαδή το σύνολο των τιμών των διαστάσεων των ηλεκτρονικών στοιχείων που περιλαμβάνουν τα αναλογικά κυκλώματα καθώς και ο τρόπος με τον οποίο αυτά διασυνδέονται, είναι πρακτικά απεριόριστος. Στην πράξη, υπάρχουν πολλές τοπολογίες αναλογικών κυκλωμάτων που προσφέρουν παρόμοιες λειτουργίες, και η απόφαση για την επιλογή μιας έναντι των υπολοίπων δεν είναι προφανής, καθώς εξαρτάται από την φύση της εκάστοτε εφαρμογής. Επιπλέον, η διαστασιοποίηση των κυκλωματικών στοιχείων απαιτεί την επιλογή τιμών από ένα συνεχές εύρος, ενώ ο χώρος επίδοσης των κυκλωμάτων, δηλαδή το σύνολο των τιμών των μετρικών που περιγράφουν την λειτουργία των κυκλωμάτων, σχετίζεται άμεσα από την επιλογή των διαστάσεων των στοιχείων. Ηλεκτρικές ιδιότητες, όπως για παράδειγμα τιμές τάσεων και ρευμάτων, παίζουν σημαντικό ρόλο στην επίδοση του εκάστοτε κυκλώματος, γεγονός που δεν αφήνει περιθώρια για την υιοθέτηση ιερικών μεθόδων σχεδίασης. Αντίθετα, συμπεριφορικά μοντέλα μπορούν να χρησιμοποιηθούν στην περίπτωση των ψηφιακών κυκλωμάτων, αποχρύπτοντας από τον σχεδιαστή φαινόμενα ή λογικές χαμηλού επιπέδου. Τα παραπάνω συνηγορούν στο γεγονός ότι η σχεδίαση αναλογικών ολοκληρωμένων κυκλωμάτων απαιτεί σοβαρό αριθμό εργατο-ωρών, παρόλο που η λειτουργία τους στα ολοκληρωμένα κυκλώματα έχει περιοριστεί σημαντικά.

Οι σχεδιαστές ολοκληρωμένων κυκλωμάτων παραδοσιακά χρησιμοποιούν εξειδικευμένα εργαλεία λογισμικού, τα οποία βοηθούν στην διαχείριση σχεδίων με εξαιρετικά μεγάλο αριθμό στοιχείων. Τα εργαλεία αυτά αναφέρονται ως εργαλεία *Ηλεκτρονικής Αυτόματης Σχεδίασης* (Electronic Design Automation - EDA) και εισήχθησαν στην βιομηχανία την δεκαετία του 1980. Στις μέρες μας, τα εργαλεία EDA παρέχουν στους σχεδιαστές ψηφιακών κυκλωμάτων την δυνατότητα να ορίζουν κυκλώματα με τρόπο αφαιρετικό, που τους επιτρέπει να εστιάζουν στην λειτουργία του κυκλώματος και στον περιορισμό επαναλαμβανόμενων διαδικασιών με δομημένο τρόπο. Το αποτέλεσμα αυτής της δυνατότητας είναι η κατακόρυφη αύξηση στην αποδοτικότητα των σχεδιαστών, και η σχεδίαση ολοκληρωμένων ψηφιακών κυκλωμάτων με δισεκατομμύρια ενσωματωμένα στοιχεία.

Στην περίπτωση των αναλογικών κυκλωμάτων, όμως, τα εργαλεία EDA δεν έχουν προ-

σφέρει παρόμοιες δυνατότητες. Η χρήση αφαιρετικής λογικής για την σχεδίαση αναλογικών κυκλωμάτων δεν είναι δυνατή, καθώς παράγοντες όπως η φυσική σχεδίαση, η τοπολογία και η διαστασιοποίηση αλληλεπιδρούν και παίζουν ρόλο στην λειτουργία των κυκλωμάτων ταυτόχρονα. Η χειροκίνητη σχεδίαση, επομένως, αποτελεί τον μοναδικό τρόπο για την σχεδίαση και υλοποίηση αναλογικών λειτουργιών σε υλικό, ενώ τα EDA εργαλεία προσφέρουν μόνο δυνατότητες προσομοίωσης και επαλήθευσης των κυκλωμάτων. Αυτός ο περιορισμός έχει ως αποτέλεσμα η ροή σχεδίασης αναλογικών ολοκληρωμένων κυκλωμάτων να μην μπορεί να ανταποκριθεί σε σύγχρονα, πολύπλοκα συστήματα, τα οποία απαιτούν πολύ χρόνο και επαναλαμβανόμενες διαδικασίες για να υλοποιηθούν. Δεδομένου, λοιπόν, του περιορισμένου αυτοματισμού στη σχεδίαση αναλογικών ολοκληρωμένων κυκλωμάτων, απαιτούνται μέτρα για να διευκολυνθεί η δουλειά των αναλογικών σχεδιαστών.

Αυτή η εργασία επικεντρώνεται στην ανάπτυξη μεθόδων και αλγορίθμων για την εισαγωγή αυτοματισμών στον κλάδο της σχεδίασης αναλογικών ολοκληρωμένων κυκλωμάτων. Πιο συγκεκριμένα, μελετώνται τρόποι για την αυτόματη διαστασιοποίηση των στοιχείων που απαρτίζουν τα ολοκληρωμένα κυκλώματα, μέσω του ορισμού του προβλήματος διαστασιοποίησης ως πρόβλημα βελτιστοποίησης. Παρουσιάζονται μέθοδοι και αλγόριθμοι για την αυτοματοποίηση μιας σειράς από διαδικασίες που απασχολούν τους σχεδιαστές, όπως η βελτιστοποίηση της απόδοσης, η εξερεύνηση του σχεδιαστικού χώρου, η αυτόματη αλλαγή τεχνολογίας κατασκευής και η επαλήθευση της λειτουργίας των κυκλωμάτων.

### Ορισμός του Προβλήματος Διαστασιοποίησης

Θεωρούμε ένα αναλογικό κύκλωμα με δεδομένη τοπολογία, τεχνολογία κατασκευής καθώς και ένα σύνολο παραμέτρων  $\mathbf{x} = [x_1, x_2, \dots, x_N]$  που αντιστοιχούν στις γεωμετρικές διαστάσεις των στοιχείων του. Επιπλέον, θεωρούμε ως δεδομένα τα εύρη των τιμών που επιτρέπονται να λάβουν οι παράμετροι του  $\mathbf{x}$ , και τα συμβολίζουμε ως  $[\mathbb{S}_i]_{i=1}^N$ . Ο χώρος αναζήτησης (ή σχεδίασης)  $\mathbb{S}$  ορίζεται ως το υποσύνολο του  $\mathbb{R}^N$  από το οποίο μπορεί να παίρνει τιμές το  $\mathbf{x}$  και ισχύει  $\mathbb{S} = \mathbb{S}_1 \times \mathbb{S}_2 \times \dots \times \mathbb{S}_N$ .

Για το δεδομένο κύκλωμα, θεωρούμε επίσης ένα σύνολο από  $k$  μετρικά  $Y = \{y_i(\mathbf{x})\}_{i=1}^k$  που καθορίζουν την λειτουργία και την απόδοση του κυκλώματος. Για παράδειγμα, στην περίπτωση ενός ενισχυτή χαμηλού θορύβου, αυτά θα μπορούσε να είναι η γραμμικότητα, το κέρδος μικρού σήματος, η θορυβική απόδοση, κλπ. Η σχεδίαση ενός αναλογικού κυκλώματος απαιτεί την εύρεση ενός καλού συμβιβασμού μεταξύ μετρικών που είναι αντικρουόμενα, δηλαδή η βελτίωση του ενός επιφέρει την χειροτέρευση του άλλου. Για αυτό τον λόγο, θεωρούμε σαν δεδομένο μια λίστα από προδιαγραφές, που ορίζουν περιορισμούς στις τιμές των μετρικών  $\mathbf{y}$ , καθώς και επιθυμητούς στόχους για μεγιστοποίηση ή ελαχιστοποίηση.

Με τα παραπάνω δεδομένα, μπορούμε να ορίσουμε τη διαστασιοποίηση του κυκλώματος

ως ένα πρόβλημα βελτιστοποίησης:

$$\begin{aligned} \min \quad & F(\mathbf{x}), \quad \mathbf{x} \in \mathbb{S} \\ \text{s.t.} \quad & g_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, l. \end{aligned} \quad (1)$$

Ανάλογα με τις προδιαγραφές, η συνάρτηση  $F(\mathbf{x})$  μπορεί να είναι βαθμωτή ή διανυσματική, δηλαδή να περιλαμβάνει ένα ή και περισσότερα μετρικά από το  $Y$ . Οι  $l$  συναρτήσεις  $g_j(\mathbf{x})$  υλοποιούν τους περιορισμούς που επιβάλλουν οι προδιαγραφές στα μετρικά του  $Y$  ως ανισότητες. Η επίλυση του προβλήματος αυτού οδηγεί σε ένα βέλτιστο διάνυσμα παραμέτρων  $\mathbf{x}^*$ , που υποδηλώνει τις διαστάσεις των στοιχείων του κυκλώματος που ικανοποιούν βέλτιστα τις προδιαγραφές.

### Αυτόματη Διαστασιοποίηση με Χρήση Προσομοίωσης

Για την επίλυση του προβλήματος (1), δύο παράγοντες παίζουν καθοριστικό ρόλο. Ο πρώτος είναι ο αλγόριθμος με τον οποίο θα προσεγγιστεί το βέλτιστο, ενώ ο δεύτερος είναι ο τρόπος με τον οποίο οι συναρτήσεις  $F(\mathbf{x})$ ,  $[g_i(\mathbf{x})]_{i=1}^l$ , και συνεπώς τα μετρικά  $Y$ , υπολογίζονται.

Η διαδικασία υπολογισμού των μετρικών του κυκλώματος μπορεί να γίνει με δύο κύριους τρόπους, με τη χρήση εξισώσεων κλειστού τύπου ή την χρήση προσομοιωτών. Οι κυκλωματικές εξισώσεις απαιτούν ανάλυση του κυκλώματος με το χέρι, συνυπολογισμό φαινομένων όπως παρασιτικά μεγαλύτερων τάξεων και σημαντική προσπάθεια από τον σχεδιαστή. Επιπρόσθετα, στις περιπτώσεις κυκλωμάτων με σεβαστό αριθμό από μετρικά, οι εξισώσεις καταλήγουν να είναι δυσεπίλυτες.

Αντιθέτως, με τη χρήση προσομοιωτών για τον υπολογισμό των κυκλωματικών μετρικών μετρικών μπορούμε να έχουμε 1) μεγάλη ακρίβεια, η οποία εξαρτάται από τα μοντέλα των ημιαγωγών που χρησιμοποιούνται κατά την προσομοίωση και όχι από τον ίδιο τον σχεδιαστή και, 2) ένα πιο εύχρηστο περιβάλλον αυτόματης διαστασιοποίησης που δεν απαιτεί σημαντικές εισόδους από τους σχεδιαστές. Στην εργασία αυτή αναπτύχθηκε λογισμικό που υλοποιεί διεπαφή με τον εμπορικό προσομοιωτή Cadence Spectre και είναι υπεύθυνο για την αυτόματη εκκίνηση προσομοιώσεων, την ανάγνωση και επεξεργασία των αποτελεσμάτων του προσομοιωτή, την ανανέωση των παραμέτρων των κυκλωμάτων πριν από κάθε προσομοίωση, τον ορισμό αναλύσεων που θα εφαρμοστούν από τον προσομοιωτή στο κύκλωμα καθώς και την παραλληλοποίησή τους. Η μοναδική είσοδος που απαιτείται από αυτή την προσέγγιση είναι παραμετροποιημένα σχηματικά του κυκλώματος (testbenches), καθώς και η λίστα με τις προδιαγραφές για το κύκλωμα. Με αυτή την προσέγγιση μαύρου κουτιού, ο σχεδιαστής μπορεί να εστιάσει στην επιθυμητή απόδοση του κυκλώματος, ανεξάρτητα από την τοπολογία ή τις εξισώσεις που το διέπουν, καθώς αυτές λαμβάνονται

εμμέσως υπόψη μέσω του σχηματικού και του προσομοιωτή.

### Διαστασιοποίηση με Χρήση Εξελικτικών Αλγόριθμων

Ο κλάδος των Εξελικτικών Αλγορίθμων μιμείται φυσικές αρχές και φαινόμενα για να πετύχει μια μορφή αυτοματισμού κατά την επίλυση προβλημάτων όπως η βελτιστοποίηση. Οι Εξελικτικοί Αλγόριθμοι αντλούν έμπνευση από την θεωρία εξέλιξης του Δαρβίνου καθώς και από την γενετική, και θεωρούν δύο βασικές υποθέσεις. Πρώτον, υπάρχει ένα περιβάλλον που μπορεί να υποστηρίξει συγκεκριμένο αριθμό από οντότητες, και δεύτερον κάθε οντότητα στοχεύει στην αναπαραγωγή της. Οι οντότητες έχουν ξεχωριστά χαρακτηριστικά, τα οποία καθορίζουν την ικανότητά τους να προσαρμόζονται στο περιβάλλον και να αναπαράγονται. Η δυνατότητα αυτή ποσοτικοποιείται από το μετρικό της καταλληλότητας. Συνέπεια των παραπάνω είναι ότι οι οντότητες ανταγωνίζονται για να αναπαραχθούν και να επιζήσουν, καθώς ο συνολικός πληθυσμός τους είναι περιορισμένος. Κατά την αναπαραγωγή, τα χαρακτηριστικά των απογόνων προκύπτουν από συνδυασμό των γονέων-οντοτήτων και ορισμένων τυχαίων παραλλαγών. Η εξελικτική διαδικασία που ακολουθούν οι αλγόριθμοι αυτοί, λοιπόν, βασίζεται τόσο στον ανταγωνισμό μεταξύ των οντοτήτων όσο και στην ποικιλία των χαρακτηριστικών που προκύπτουν κατά την αναπαραγωγή, έτσι ώστε να καταλήξουν σε οντότητες που προσαρμόζονται καλύτερα στο περιβάλλον και είναι διαφορετικές μεταξύ τους.

Υπάρχει μεγάλη ποικιλία από Εξελικτικούς αλγόριθμους που ακολουθούν την παραπάνω μεθοδολογία στην βιβλιογραφία. Ένας διαχωρισμός που μας είναι χρήσιμος στο πλαίσιο της διατριβής αυτής είναι με βάση τα προβλήματα που επιλύουν. Στην περίπτωση όπου οι προδιαγραφές για ένα κύκλωμα είναι σε μορφή επίτευξης ενός μοναδικού στόχου, για παράδειγμα μεγιστοποίηση του εύρους ζώνης ενός ενισχυτή, το πρόβλημα ονομάζεται Μοναδικού Στόχου και η συνάρτηση  $F(\mathbf{x})$  στην εξίσωση (1) είναι βαθμωτή. Αντίθετα, στην περίπτωση που οι προδιαγραφές για ένα κύκλωμα απαιτούν την εύρεση της καλύτερης συμβιβαστικής λύσης μεταξύ δύο ή περισσότερων στόχων, όπως για παράδειγμα το κέρδος χαμηλού σήματος και το εύρος ζώνης ενός ενισχυτή, η  $F(\mathbf{x})$  είναι διανυσματική συνάρτηση. Συνεπώς, το βέλτιστο ενδέχεται να μην είναι μοναδικό και η καταλληλότητα των οντοτήτων του αλγορίθμου ορίζεται με τρόπο ώστε να λαμβάνει υπόψη το πόσο καλό συμβιβασμό μεταξύ των δύο μετρικών προσφέρουν, καθώς και πόσο διαφορετικά είναι από τον υπόλοιπο πληθυσμό. Τα προβλήματα αυτά αναφέρονται ως Πολλαπλών Στόχων και χρησιμοποιούνται στα πλαίσια της διατριβής για την εξερεύνηση του σχεδιαστικού χώρου των κυκλωμάτων.

Για την επιτυχή έκβαση της διαστασιοποίησης μέσω του ορισμού (1), είναι σημαντικό ο σχεδιαστής να συμπεριλάβει μια σειρά από περιορισμούς που, όταν πληρούνται, εξασφαλί-

ζουν την ομαλή λειτουργία του κυκλώματος. Οι Εξελικτικοί Αλγόριθμοι, όμως, πρέπει να προσαρμοστούν στο νέο πρόβλημα για να διαχειρίζεται τους περιορισμούς αυτούς. Όταν υπάρχουν περιορισμοί, ο στόχος του αλγόριθμου γίνεται διπλός: να οδηγήσει την αναζήτηση, δηλαδή τις οντότητες, σε μια περιοχή του χώρου αναζήτησης όπου οι περιορισμοί ικανοποιούνται και εκεί να βρει τις βέλτιστες λύσεις. Για αυτόν τον λόγο, στην εργασία αυτή, οι Εξελικτικοί Αλγόριθμοι τροποποιούνται ώστε η επιβίωση ή μη μιας οντότητας να καθορίζεται από ένα σύστημα προτιμήσεων, το οποίο λαμβάνει υπόψη τόσο την καταλληλότητα της κάθε οντότητας όσο και τον βαθμό παραβίασης περιορισμών της.

Στα πλαίσια της διατριβής, σχεδιάζονται δύο ταλαντωτές LC σε τεχνολογία κατασκευής TSMC 90nm και οι επιθυμητές προδιαγραφές τους ορίζονται με τρόπο που εξασφαλίζεται η επιθυμητή συχνότητα ταλάντωσης. Για την διερεύνηση των διαφορών μεταξύ των δύο τοπολογιών, εξετάζονται οι χώροι απόδοσης τους, όσον αφορά το συμβιβασμό μεταξύ θορύβου φάσης και κατανάλωσης ισχύος. Χρησιμοποιώντας Εξελικτικούς Αλγόριθμους, φτάνουμε στο συμπέρασμα ότι η μία τοπολογία υπερέχει της άλλης, όσον αφορά τον ελάχιστο εφικτό θόρυβο φάσης για παρόμοιες καταναλώσεις ισχύος.

Για την περαιτέρω βελτίωση της απόδοσης των Εξελικτικών Αλγορίθμων, όσον αφορά τις λύσεις που αποδίδουν σε προβλήματα Πολλαπλών Στόχων με περιορισμούς, προτείνουμε μια παραλλαγή του Εξελικτικού Αλγορίθμου NSGA-II, ώστε να διαχειρίζεται χώρους αναζήτησης με μικτές συνεχείς-διακριτές μεταβλητές όταν υπάρχουν περιορισμοί. Ο μηχανισμός επιλογής οντοτήτων που επιβιώνουν τροποποιείται, ώστε να λαμβάνει υπόψη την καταλληλότητα λύσεων που δεν ικανοποιούν τους περιορισμούς. Η διαχείριση του χώρου αναζήτησης με μικτές μεταβλητές γίνεται με την υιοθέτηση ενός συστήματος παραλλαγής, όπου οι διακριτές μεταβλητές αλλάζουν τιμές ακολουθώντας μια κατανομή Poisson. Ο προτεινόμενος αλγόριθμος εφαρμόζεται σε έναν ενισχυτή χαμηλού θορύβου και έναν τελεστικό ενισχυτή, για την εξερεύνηση του χώρου απόδοσης τους. Και στις δύο περιπτώσεις καταφέρνουμε να πάρουμε καλύτερους συμβιβασμούς μεταξύ των αντικρουόμενων μετρικών των κυκλωμάτων.

### **Διαστασιοποίηση με Χρήση Αλγόριθμων Χαμηλού Κόστους**

Η αυτόματη διαστασιοποίηση πολύπλοκων κυκλωμάτων, με χώρους αναζήτησης πολλών διαστάσεων και αυστηρούς περιορισμούς ως προδιαγραφές, είναι ένα δύσκολο πρόβλημα που απαιτεί, στην περίπτωση των Εξελικτικών Αλγορίθμων, πολλές προσομοιώσεις για την αξιολόγηση των πληθυσμών των οντοτήτων. Στην πράξη, η προσομοίωση κυκλωμάτων ενδέχεται να είναι μια κοστοβόρα χρονικά διαδικασία, καθώς μια προσομοίωση μπορεί να χρειάζεται ώρες για να ολοκληρωθεί, γεγονός που καθιστά τις πληθυσμιακές μεθόδους, όπως τους Εξελικτικούς Αλγόριθμους, μη πρακτικούς. Για τους λόγους αυτούς, μια νέα

προσέγγιση στην αυτόματη διαστασιοποίηση αναλογικών προβλημάτων με χρήση αλγορίθμων χαμηλής δειγματοληψίας του χώρου αναζήτησης προτείνεται. Η προσέγγιση βασίζεται στην Μπεϋζιανή εκτίμηση και αναφέρεται ως Μπεϋζιανή Βελτιστοποίηση.

Η Μπεϋζιανή Βελτιστοποίηση είναι μια μέθοδος βελτιστοποίησης αποδοτικής δειγματοληψίας του χώρου αναζήτησης, που στοχεύει κυρίως σε υπολογιστικά ακριβά προβλήματα. Στην τυπική μορφή της, όπου δεν λαμβάνονται υπόψη περιορισμοί, της παρέχονται παρατηρήσεις από μια συνάρτηση  $f$  και αυτή δημιουργεί ένα γρήγορο στην αξιολόγηση μοντέλο. Τα σημεία του χώρου αναζήτησης για προσομοίωση, ή αλλιώς σημεία ερωτήματος, επιλέγονται σειριακά σε κάθε επανάληψη. Η Μπεϋζιανή Βελτιστοποίηση αποτελείται από δύο βασικά συστατικά: 1) ένα πιθανοτικό μοντέλο που προσεγγίζει την άγνωστη συνάρτηση  $f$  και 2) μια συνάρτηση απόκρισης  $a : \mathbb{S} \rightarrow \mathbb{R}$ , που ορίζει ένα μέτρο για την καταλληλότητα κάθε σημείου στο χώρο αναζήτησης, με βάση το πιθανοτικό μοντέλο. Συνήθως, το μοντέλο αυτό είναι μια Γκαουσιανή Διαδικασία, που εκπαιδεύεται και χρησιμοποιείται σε κάθε επανάληψη.

Για να προσαρμόσουμε την Μπεϋζιανή Βελτιστοποίηση στα προβλήματα βελτιστοποίησης που προκύπτουν κατά την αυτόματη διαστασιοποίηση αναλογικών κυκλωμάτων, προτείνουμε μια σειρά από τροποποιήσεις στη βασική της μορφή:

1. Οι Γκαουσιανές Διαδικασίες χρησιμοποιούν συναρτήσεις πυρήνα για να ορίσουν μια συνάρτηση πυκνότητας πιθανότητας στο χώρο των συναρτήσεων. Οι συναρτήσεις πυρήνα, όμως, χρησιμοποιούν την Ευκλείδεια απόσταση μεταξύ των σημείων παρατηρήσεων για να ορίσουν με την σειρά τους την συνδιακύμανση μεταξύ τιμών της άγνωστης συνάρτησης που μοντελοποιούν. Ο σημαντικός περιορισμός που προκύπτει είναι η απώλεια πληροφορίας σε χώρους με πολλές διαστάσεις, όπου η Ευκλείδεια απόσταση δεν παρέχει ιδιαίτερα χρήσιμες πληροφορίες. Για την εξομάλυνση του φαινομένου αυτού, προτείνεται μια τοπική προσέγγιση στο πρόβλημα της βελτιστοποίησης, όπου η Μπεϋζιανή Βελτιστοποίηση λειτουργεί και μοντελοποιεί την άγνωστη συνάρτηση σε περιφέρειες εμπιστοσύνης, αντί στο σύνολο του χώρου αναζήτησης.
2. Η σειριακή φύση του τυπικού αλγορίθμου δεν είναι αποδοτική από την ακόλουθη σκοπιά: Σε κάθε επανάληψη, ένα μοναδικό σημείο ερωτήματος επιλέγεται, προσομοιώνεται και χρησιμοποιείται για την ανανέωση των Γκαουσιανών μοντέλων. Επομένως, το υπολογιστικό κόστος της εκπαίδευσης των μοντέλων μας επιβαρύνει για μία μοναδική παρατήρηση, ενώ επίσης δεν χρησιμοποιούνται στο έπακρο οι διαθέσιμοι πόροι που επιτρέπουν την παραλληλοποίηση πολλών προσομοιώσεων. Για αυτό τον λόγο, ορίζουμε μια νέα συνάρτηση απόκρισης, η οποία επιτρέπει την επιλογή πολλών σημείων ερωτήματος σε κάθε επανάληψη, με αποτέλεσμα να έχουμε στη διάθεσή μας περισσότερη πληροφορία στο ίδιο χρονικό παράθυρο.

3. Η τυπική Μπεύζιανή Βελτιστοποίηση δεν λαμβάνει υπόψη περιορισμούς κατά την επίλυση προβλημάτων. Για να προσαρμοστεί στον ορισμό της σχέσης (1), επεκτείνουμε την προτεινόμενη συνάρτηση απόκτησης ώστε να συνυπολογίζει την πιθανότητα τα σημεία αναζήτησης να ικανοποιούν τους περιορισμούς.
4. Για να περιορίσουμε το κόστος εκπαίδευσης των Γκαουσιανών Διαδικασιών, όταν τα σημεία ερωτήματος είναι πολλά σε αριθμό, χρησιμοποιούμε μια μέθοδο από χαρακτηριστικά σημεία, τα οποία αντικαθιστούν μέλη από το αρχικό σύνολο δεδομένων που θα έπρεπε διαφορετικά να χρησιμοποιηθεί.
5. Προτείνουμε την χρήση Τυχαίων Χαρακτηριστικών Φουριέ, ώστε να βελτιώσουμε την επίδοση της συνάρτησης απόκτησης σε προβλήματα μεγάλων διαστάσεων, αντικαθιστώντας την διαδικασία δειγματοληψίας από Γκαουσιανές διαδικασίες με μία νέα, όπου χτίζονται αναλυτικές συναρτήσεις ως δείγματα.
6. Προτείνουμε παραλλαγές Μπεύζιανής Βελτιστοποίησης που μπορούν να λύσουν προβλήματα Μοναδικού και Πολλαπλών Στόχων.

Οι παραπάνω μέθοδοι εφαρμόζονται για την αυτόματη διαστασιοποίηση μια σειράς αναλογικών κυκλωμάτων, που περιλαμβάνουν τελεστικούς ενισχυτές πολλών σταδίων και ενισχυτές χαμηλού θορύβου. Τα προβλήματα διαστασιοποίησης λαμβάνουν υπόψη την επίδοση του εκάστοτε κυκλώματος τόσο σε τυπικές συνθήκες, όσο και σε συνθήκες με διακυμάνσεις τάσης, θερμοκρασίας και κατασκευής. Η χρήση της Μπεύζιανής μεθόδου για την διαστασιοποίηση επιφέρει καλύτερα αποτελέσματα σε σχέση με μία σειρά άλλων μεθόδων στη βελτιστοποίηση και την εξερεύνηση του χώρου επίδοσης, για παρόμοιο αριθμό προσομοιώσεων.

### Εκμάθηση Αναπαραστάσεων Κυκλωματικών Στοιχείων

Για την εύρωστη μοντελοποίηση διακριτών χώρων αναζήτησης μέσω Γκαουσιανών διαδικασιών, προτείνεται μια μέθοδος αναπαράστασης κυκλωματικών στοιχείων με συνεχείς μεταβλητές, με χρήση Βαθιάς Μάθησης. Οι γεωμετρικές στοιχείων όπως τα ολοκληρωμένα πηνία παραμετροποιούνται τόσο με συνεχείς, όσο και με διακριτές μεταβλητές, γεγονός που καθιστά αδύνατη την μοντελοποίηση του χώρου αναζήτησης από Γκαουσιανές διαδικασίες με τυπικές συναρτήσεις πυρήνα. Για τον λόγο αυτό, προτείνουμε την εκμάθηση αναπαραστάσεων που λαμβάνουν συνεχείς τιμές, με χρήση δεδομένων από προσομοίωση. Με την αντικατάσταση των αρχικών πεδίων ορισμού, που αντιστοιχούν σε στοιχεία με μικτές συνεχείς-διακριτές μεταβλητές, από τις συνεχείς αναπαραστάσεις τους, καθιστούμε το πρόβλημα βελτιστοποίησης συνεχές.



Η μέθοδός μας περιλαμβάνει ένα γενετικό μοντέλο λανθανουσών μεταβλητών, το οποίο αντιστοιχεί σημεία από έναν χώρο  $\mathbb{Z}$  στον αρχικό χώρο αναζήτησης του προβλήματος,  $\mathcal{S}$ . Για να εξασφαλίσουμε ότι ο λανθάνων χώρος  $\mathbb{Z}$  είναι δομημένος με τρόπο που θα εξυπηρετεί την βελτιστοποίηση, περιλαμβάνεται ένα επιπλέον μοντέλο καθοδήγησης το οποίο εκπαιδεύεται παράλληλα με το γενετικό μοντέλο. Το μοντέλο καθοδήγησης εκπαιδεύεται ώστε να αντιστοιχεί σημεία από τον  $\mathbb{Z}$  σε πραγματικές διαστάσεις των στοιχείων που μοντελοποιούνται. Οι αναπαραστάσεις των κυκλωματικών στοιχείων φτιάχνονται με τρόπο τέτοιο ώστε να εξασφαλίζεται ότι λειτουργικά παρόμοιες γεωμετρίες είναι κοντά στον  $\mathbb{Z}$ . Αυτό επιτυγχάνεται με τη χρήση δεδομένων προσομοίωσης που αντικατοπτρίζουν τις επιδόσεις του εκάστοτε στοιχείου, ως εισόδους του γενετικού μοντέλου. Κατά την βελτιστοποίηση, ο αλγόριθμος αναζητά τιμές στον λανθάνων χώρο, οι οποίες περνούν από το δίκτυο καθοδήγησης για να μετατραπούν σε πραγματικές γεωμετρίες και έπειτα λαμβάνει χώρα η προσομοίωση.

Για την επαλήθευση της προτεινόμενης μεθόδου έγιναν τα εξής πειράματα: 1) Μοντελοποίηση ολοκληρωμένων σπειροειδών πηνίων και 2) μοντελοποίηση ολοκληρωμένων μεταλλίων πυκνωτών. Και τα δύο στοιχεία περιλαμβάνουν διακριτές μεταβλητές κατά τον ορισμό της γεωμετρίας τους. Μια σειρά από κυκλώματα έγιναν βελτιστοποίηση με χρήση των αναπαραστάσεων των προαναφερθέντων στοιχείων, με αποτελέσματα που υποδηλώνουν την ακριβή μοντελοποίηση και την βελτίωση της απόδοσης των Μπεϋζιανών μεθόδων σε πλήρης συνεχείς χώρους αναζήτησης.



# Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Professor Pavlos-Petros P. Sotiriadis. His assistance, insightful comments, suggestions and continuous support were crucial for the completion of my Ph.D. research. With his immense expertise, Professor Sotiriadis gracefully offered me his knowledge in Mathematics and Engineering and guided me through the paths of my research. What an amazing journey I was lucky and glad to have with him. It has been an honor to have worked under his supervision and to have received his support.

I would also like to deeply thank Professor Nikolaos Maratos and Professor Ioanna Roussaki for serving as members of my advisory committee. Also, I would like to express my gratitude to Professors Stefanos Kollias, Dimitra Kaklamani, Giorgos Stamou, Athanasios Panagopoulos and Nuno Lourenço, for participating in my examination committee.

I am also grateful to all my colleagues and students in the Circuits and Systems Group of Professor Sotiriadis, with whom I have shared many wonderful and memorable moments during the past years. Our collaboration and friendship were crucial in making the Ph.D. experience productive and enjoyable. Special thanks to my kind friends from the lab, Kostas Asimakopoulos, Dimitris Baxevanakis, Dr. Chris Dimas, Dr. Kostas Papafotis and Dr. Nikos Temenos.

Getting through my Ph.D. studies required more than academic support, and I have many, many people to thank for being next to me over the past years. For their support and for many memorable evenings and vacations, I would like to thank my friends from Highschool and the University and wish them the best in their endeavors.

My family was most important to me in my pursuit of the Ph.D. I would like to thank my parents, Thanasis and Vasiliki, whose love and guidance are always with me in whatever I pursue, and my beloved sister, Semina, who cares for me and supports me in my endeavors.

This research is co-financed by Greece and the European Union (European Social Fund-ESF) through the Operational Programme «Human Resources Development, Education and Lifelong Learning» in the context of the project “Strengthening Human Resources Research Potential via Doctorate Research” (MIS-5000432), implemented by the State Scholarships Foundation (IKY)



# Contents

<b>1</b>	<b>Introduction</b>	<b>37</b>
1.1	Motivation . . . . .	38
1.2	Thesis Contributions . . . . .	42
1.3	Thesis Outline . . . . .	44
<b>2</b>	<b>Related Work</b>	<b>47</b>
2.1	Knowledge-Based Approaches . . . . .	47
2.2	Optimization-Based Approaches . . . . .	49
2.2.1	Model-Based Sizing . . . . .	49
2.2.2	Simulation-Based . . . . .	51
2.3	Other Approaches . . . . .	53
<b>3</b>	<b>Evolutionary Algorithm-based Sizing</b>	<b>55</b>
3.1	Evolutionary Computation Preliminaries . . . . .	55
3.1.1	Genetic Algorithms . . . . .	58
3.1.2	Differential Evolution . . . . .	59
3.1.3	Covariance Matrix Adaptation . . . . .	60
3.2	Single-Objective Sizing . . . . .	61
3.3	Multi-Objective Sizing . . . . .	61
3.4	Constraint Handling . . . . .	62
3.5	Application in Design Space Exploration . . . . .	63
3.5.1	LC-VCO topologies . . . . .	64
3.5.2	LC-VCO Optimization . . . . .	65
3.6	Proposed Multi-Objective Variant . . . . .	69
3.6.1	Handling Integer Parameters . . . . .	69
3.6.2	Constraint Handling . . . . .	70
3.6.3	Example Applications . . . . .	71
3.6.4	Inductorless Wideband LNA . . . . .	74
3.7	Summary & Concluding Remarks . . . . .	77

<b>4</b>	<b>Sample-Efficient Single-Objective Sizing</b>	<b>79</b>
4.1	Background	79
4.1.1	Gaussian Processes	79
4.1.2	Bayesian Optimization	85
4.1.3	BO & Constraint Handling	88
4.2	Proposed Approach	90
4.2.1	Local Bayesian Optimization	90
4.2.2	Scaling Up Gaussian Processes	92
4.3	Circuit Design Applications	94
4.3.1	Three-Stage Amplifier	95
4.3.2	High Linearity LNA	96
4.4	Summary & Concluding Remarks	99
<b>5</b>	<b>Sample Efficient Design Space Exploration</b>	<b>101</b>
5.1	BO Local Based Approach	102
5.1.1	Sampling Functions from GP posteriors	105
5.1.2	Acquisition Function	110
5.1.3	Implementation	114
5.1.4	Summary	115
5.1.5	MOP Example	118
5.2	Circuit Design Applications	124
5.2.1	Three Stage Amplifier	124
5.2.2	Four Stage Amplifier	130
5.2.3	Low Noise Amplifier	135
5.3	Summary & Concluding Remarks	139
<b>6</b>	<b>Device Representation Learning</b>	<b>141</b>
6.1	Motivation	141
6.2	Latent Space Optimization	142
6.2.1	Autoencoders	142
6.2.2	Variational Autoencoders	143
6.2.3	Proposed Approach	147
6.2.4	Applications on Integrated Devices	151
6.3	Circuit Design Applications	156
6.3.1	Inductively Degenrated LNA	157
6.3.2	Wideband LNA	160

---

6.4	Summary & Concluding Remarks . . . . .	164
<b>7</b>	<b>Conclusions &amp; Future Directions</b>	<b>169</b>
7.1	Thesis Contributions . . . . .	169
7.2	Future Work . . . . .	170
	<b>References</b>	<b>175</b>
	<b>A' Appendix A: Publications</b>	<b>193</b>





# List of Figures

1.1	Historic evolution of the smallest manufacturing node [2]. . . . .	38
1.2	Graphical illustration of the analog design gap [2]. . . . .	41
1.3	A sample analog IC design flow. This work focuses on device sizing selection.	42
2.1	An illustration of the optimization-based approach to analog circuit sizing.	49
3.1	A flowchart depicting the general operation of EC-based optimization algorithms. Figure adapted from [76] . . . . .	57
3.2	A depiction of the single-point crossover operation in GAs. . . . .	59
3.3	nMOS cross-coupled LC-VCO . . . . .	66
3.4	Complementary nMOS-pMOS cross-coupled LC-VCO . . . . .	66
3.5	Acquired Pareto Fronts for both LC-VCO topologies, with $f_{osc} = 5\text{GHz}$	68
3.6	Acquired Pareto Fronts for both LC-VCO topologies, with $f_{osc} = 2.4\text{GHz}$	69
3.7	The probability mass function of two sample Poisson distributions with $\lambda = 2$ and $\lambda = 4$ . . . . .	70
3.8	Top row: Optimization run for a multimodal constrained function, using the feasibility rule. Bottom row: Optimization run using the constrained handling method proposed. Initial candidate vectors are identical. . . . .	72
3.9	Nested Current Mirror amplifier proposed in [90]. Half-circuit instance names are shown, since the circuit is symmetric. . . . .	73
3.10	Pareto fronts for the NSGA-II algorithm with the proposed constraint handling method, and the typical NSGA-II with feasibility rule. . . . .	74
3.11	Inductorless, wideband low noise amplifier proposed in [92]. . . . .	75
3.12	Pareto fronts for the wideband LNA experiment. . . . .	76
4.1	Sampled functions from different GP priors. . . . .	82
4.2	A depiction of GP regression to a set of sinewave measurements. The measurements are shown as black stars, the actual sinewave is shown in red and the GP mean and confidence bounds are shown in blue. . . . .	84
4.3	Example BO execution using the LCB acquisition function on an 1D Ackley function. . . . .	89

4.4	A GP's pointwise predictions on a test function (mean and 95% confidence bounds shown), along with 3 samples from a joint predictive distribution. Past queries are shown as black dots, while next queries are marked in red circles. . . . .	92
4.5	Three stage amplifier proposed in [106]. . . . .	95
4.6	High Linearity LNA proposed in [107]. . . . .	97
5.1	A demonstration of the local based approach for minimizing $F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})]$ in a $2D$ space. . . . .	104
5.2	Matèrn kernel approximation using RFF samples. The $x$ axes depict the quantity $\tau = x - x'$ . . . . .	109
5.3	Illustration of the Hypervolume and Hypervolume Improvement concepts. Minimization is assumed here. The light-orange area is the dominated HV. Out of three candidates, $\mathbf{x}_3$ provides the largest HVI. . . . .	112
5.4	Illustration of the mean and standard deviations of the resulting Hypervolume metrics on the zdt1 experiment. . . . .	114
5.5	The PFs from the zdt1 experiment. The RFF results in better trade-offs. . . . .	115
5.6	The evolution of HV attained by the proposed approach using different batch sizes. The values in the $y$ axis are normalized to the largest HV value attained. . . . .	118
5.7	Illustration of the operation of LoCoMOBO for optimizing the MOP of Eq. 5.28. . . . .	120
5.8	Illustration of the operation of LoCoMOBO for optimizing the MOP of Eq. 5.28. . . . .	121
5.9	Illustration of the operation of LoCoMOBO for optimizing the MOP of Eq. 5.28. . . . .	122
5.10	Illustration of the operation of LoCoMOBO for optimizing the MOP of Eq. 5.28. . . . .	123
5.11	Three stage amplifier proposed in [106]. . . . .	124
5.12	Pareto Fronts for the Three Stage Amplifier. Bottom-right values result in better DC Gain vs Pdc trade-off. . . . .	127
5.13	A pareto front for the Three Stage Amplifier. . . . .	129
5.14	Four stage amplifier proposed in [116]. . . . .	130
5.15	Biasing sub-circuit for the considered four stage amplifier. . . . .	130
5.16	Pareto Fronts for the Four Stage Amplifier. Bottom-right values result in better DC Gain vs Pdc trade-off. . . . .	132
5.17	A pareto front for the Four Stage Amplifier. . . . .	135
5.18	Low Noise Amplifier examined in this subsection. . . . .	135

5.19	Pareto Fronts for the Low Noise Amplifier. Bottom-right values result in better $S_{21}$ vs Pdc trade-off. . . . .	138
5.20	A comparison between the Pareto Fronts derived accounting for PVT variations and only for nominal conditions. Nominal sizing results are depicted using light dotted lines and PVT-aware results using solid lines. . . . .	138
6.1	Directed graphical models for (a) the approximate posterior $q_{\theta}(\mathbf{z} \mathbf{x})$ and (b) the probabilistic decoder $p_{\phi}(\mathbf{x} \mathbf{z})$ . The coloured node indicates the observable random variable and the nodes outside the box are model parameters. . . . .	145
6.2	An illustration of the VAE model, along with its constituent parts. . . . .	147
6.3	The proposed device representation learning scheme within a sizing problem, where the latent variables are shown in red. <b>a)</b> The original variable space of the sizing problem is changed into a transformed one, by mapping sets of variables that belong to devices into continuous latent ones. <b>b)</b> During the optimization, a query from the optimization algorithm is transformed back to the original variable space using the predictor networks, prior to simulation. . . . .	149
6.4	A depiction of the proposed architecture. 1D vectors of inductors' Quality Factor and Inductance frequency behavior are inputs to the 1D convolutional filters of the architecture. The filter sizes are shown as well. The predictor FCNN gets as input the latent representation of the inductor's frequency characteristics and yields its geometric sizes. . . . .	152
6.5	A reconstruction of inductor's inductance across the frequency range of interest. . . . .	154
6.6	500 samples of inductance curves sampled from the generative model. . . . .	154
6.7	Slices of the 3D latent space across the xy, yz and xz planes. The three geometric variables for the spiral inductors are shown. . . . .	155
6.8	The Inductively Degenrated LNA considered as a case study. . . . .	157
6.10	The wideband, noise canceling LNA [128] considered in this subsection. . . . .	160
6.9	The LNA's CV for the discussed formulations. The blue are indicates the confidence region of $\pm\text{std}$ , while the purple lines are the curves of each repetition. . . . .	161
6.11	The evolution of the LNA's FoM during the automatic sizing, using the four discussed formulations. The blue are indicates the confidence region of $\pm\text{std}$ , while the purple lines are the curves of each repetition. . . . .	165



# List of Tables

3.1	Evolution and Problem Solving Equivalence . . . . .	57
3.2	VCOs variable ranges . . . . .	67
3.3	NCM Specifications . . . . .	73
3.4	NCM Variable Ranges . . . . .	74
3.5	NCM Example Solution . . . . .	75
3.6	Wideband LNA Specifications . . . . .	75
3.7	Wideband LNA Variable Ranges . . . . .	76
3.8	Wideband LNA Example Solution . . . . .	77
4.1	GP Training and sampling runtimes for Rosenbrock Function . . . . .	94
4.2	Three Stage Amplifier Specifications . . . . .	96
4.3	Optimization Results for the Three Stage Amplifier . . . . .	97
4.4	High Linearity LNA Specifications . . . . .	98
4.5	Optimization Results for the High Linearity LNA . . . . .	98
5.1	HV Results (mean $\pm$ std) on Benchmark Functions . . . . .	117
5.2	Runtime Comparison (s) . . . . .	117
5.3	Specifications for Three Stage Amplifier - Two Objectives . . . . .	125
5.4	Three Stage Amplifier HV Results - Two Objectives . . . . .	126
5.5	Three Stage Amplifier Cover Matrix - Two Objectives . . . . .	127
5.6	Specifications for Three Stage Amplifier - Three Objectives . . . . .	128
5.7	Three Stage Amplifier HV Results - Three Objectives . . . . .	128
5.8	Three Stage Amplifier Cover Matrix - Three Objectives . . . . .	129
5.9	Specifications for Four Stage Amplifier - Two Objectives . . . . .	131
5.10	Four Stage Amplifier HV Results - Two Objectives . . . . .	131
5.11	Four Stage Amplifier Cover Matrix - Two Objectives . . . . .	133
5.12	Specifications for Four Stage Amplifier - Three Objectives . . . . .	133
5.13	Four Stage Amplifier HV Results - Three Objectives . . . . .	134
5.14	Four Stage Amplifier Cover Matrix - Three Objectives . . . . .	134
5.15	Specifications for Low Noise Amplifier . . . . .	136
5.16	Low Noise Amplifier HV results . . . . .	137
5.17	Low Noise Amplifier Cover Matrix . . . . .	137

6.1	LNA Variable Ranges . . . . .	158
6.2	LNA Sizing Results . . . . .	159
6.3	Wideband LNA Variable Ranges . . . . .	163
6.4	Wideband LNA Sizing Results . . . . .	164

# List of Algorithms

1	BO Algorithm . . . . .	85
2	Inducing Point Initial Location Selection . . . . .	94
3	LoCoMOBO Acquisition Function . . . . .	113
4	LoCoMOBO Algorithm . . . . .	116





# Abbreviations

AE	Autoencoder
ANN	Artificial Neural Network
API	Application Process Interface
BBMM	BlackBox Matrix-Matrix Inference
BO	Bayesian Optimization
CIM	Compute-in-Memory
CMA	Covariance Matric Adaptation
CV	Constraint Violation
DE	Differential Evolution
EA	Evolutionary Algorithm
EC	Evolutionary Computation
EDA	Electronic Design Automation
EI	Expected Improvement
EIC	Expected Improvement with Constraints
ELBO	Evidence Lower Bound
EM	Expectation Maximization
FCNN	Fully Connected Neural Network
FoM	Figure of Merit
GA	Genetic Algorithm
GP	Gaussian Process
HLS	High Level Synthesis
HV	Hypervolume
HVI	Hypervolume Improvement
IC	Integrated Circuit
LCB	Lower Confidence Bound
LNA	Low Noise Amplifier
LOVE	Lanczos Variance Estimates
MC	Monte Carlo
ML	Machine Learning
MO	Multi Objective
MOP	Multi-Objective Optimization Problem
MSE	Mean Squared Error
NCM	Nested Current Mirror
NSGA	Non-dominated Sorting Genetic Algorithm

PCA	Principal Component Analysis
PDK	Process Development Kit
PES	Predictive Entropy Search
PF	Pareto Front
PI	Probability of Improvement
PS	Pareto Set
PVT	Process, Voltage, Temperature
RF	Radio Frequency
ReLU	Rectified Linear Unit
RFF	Random Fourier Features
RTL	Register Level Transfer
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TS	Thompson Sampling
VAE	Variational Autoencoder
VCO	Voltage Controlled Oscillator
VLSI	Vaery Very Large Scale Integration

# Chapter 1

## Introduction

Since the fabrication of the first integrated circuit in the 1960s [1], the semiconductor industry has seen tremendous advances and has effectively altered the way people interact and behave in their everyday lives. By providing the ability to integrate logic circuits in small pieces of semiconductors, it became the enabling factor for modern computing systems which dominate our world.

The driving force for the semiconductor's advancements has been the continuous miniaturization of integrated circuits; device sizes are constantly being reduced, effectively leading to faster, less power-hungry and more compact Integrated Circuits (ICs). Besides these rather obvious benefits, device miniaturization has been the key factor for the flourishing of the semiconductor market. While moving to newer, smaller devices for manufacturing requires more sophisticated manufacturing processes that in turn raise the cost for IC development, the ability to embed multiple logic functions in the same chip area not only evens out the cost but produces further financial gains. Besides this 'more functionality for the same price' [2] benefit, miniaturizing devices in chips has led to more reliable systems, by reducing the number of solder points and dispensing with many discrete devices.

There are two types of circuitry embedded in an IC design nowadays; analog circuits and digital ones. Digital circuits operate on discrete-valued signals, which are most of the times binary. In this case, voltage signals can be interpreted as having one of two distinct values, a logic low '0' and a logic high '1'. To distinguish between these two values, all nodes should converge to a voltage value that belongs in the range of logic low or logic high. On the other hand, analog circuitry operates on continuous valued signals. Analog circuits' role in ICs is mainly to implement an interface between the digital domain and the external 'real-world', as well as to perform signal processing tasks. With technologies such as Complementary Metal-Oxide Semiconductor (CMOS) and BiCMOS [3], which enable the implementation of both digital and analog functionalities on the same chip, modern ICs often combine analog and digital operations. The idea behind the combination of both domains in the same IC stems from the need to further miniaturize electronics systems, as described previously.

To further leverage the capabilities of CMOS technologies, the semiconductor industry has seen a trend to integrate complete systems in a single chip. These Systems-on-

a-Chip (SoCs) include of course analog, digital and even radio-frequency (RF) circuit blocks and have been in the forefront of the modern electronics market for more than a decade. In this context, digital circuitry has replaced analog blocks in operations such as signal processing, in order to reduce the non-ideal effects rising from the operation of transistors in the continuous regime. However, analog circuit will always be indispensable to perform some typical functions such as sensor interfaces, voltage references, analog-to-digital converters, etc.

Manufacturing these integrated circuits is tedious undertaking and requires well studied, established procedures. These are often referred to as technology or process nodes and are characterized by the minimum size of a device that can be manufactured using them. Typically, the smallest length of a unipolar transistor is used for this purpose. Transistor miniaturization does involve arbitrary scaling factors. Rather, newer technology nodes are scaled by a factor of  $1/\sqrt{2}$  in comparison to their predecessors. This transistor shrinking has been approximately predicted by Gordon Moore, whose famous heuristic stated the number of devices in ICs would double every two years [4]. In fact, by observing the actual shrinking history of the semiconductor industry, shown in Fig. 1.1, *Moore's Law* seems to be an accurate prediction about the state of the semiconductor manufacturing process.

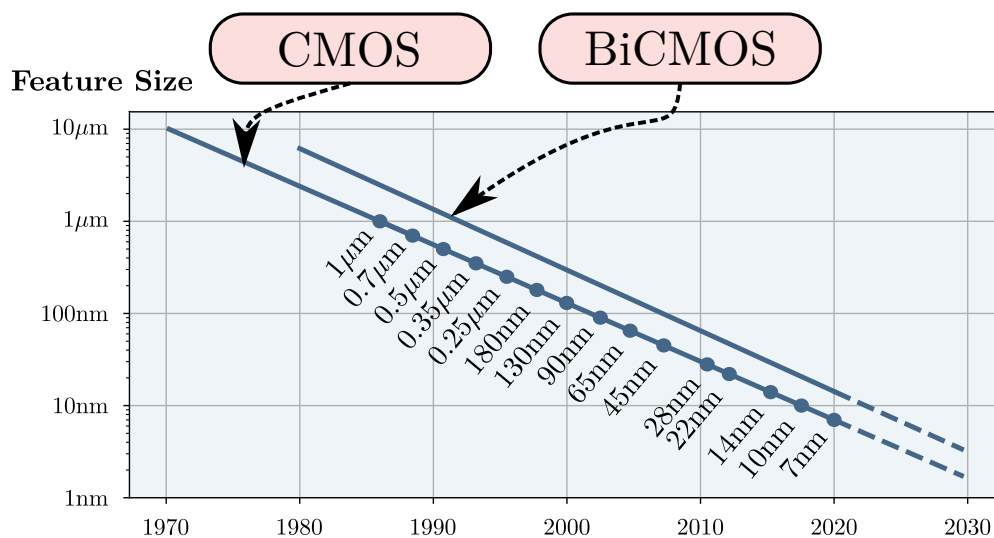


Figure 1.1: Historic evolution of the smallest manufacturing node [2].

## 1.1 Motivation

The economic and social impact of the device miniaturization described by Moore's Law is undeniable. Individuals harness the benefits of a interconnected world, with

computational devices being the enabling factor, and semiconductor investors claim considerable financial gains. However, the design of ICs in miniaturized technologies is becoming increasingly difficult, with designer's productivity constantly diminishing.

Miniaturization, however, has had more impact on the work of analog designers, compared to digital ones. In practice, although digital circuitry dominates the area of modern SoCs, analog blocks tend to be the design bottleneck, both in terms of time and budget. The fundamental reason behind this issue is the nature of the circuits themselves; digital circuitry that operates on signals with two levels and large tolerance levels are more robust to higher-order effects [5], compared to analog designs that must take into account the physics of each fabrication process and which requires expert handcraft skills.

Besides the aforementioned fundamental difference, analog design proves more cumbersome than digital since its design space is virtually infinite. In fact, there is a large variety of analog circuit topologies that may have the same functionalities and the choice of which one to design is a complex decision that depends on the exact application. In addition, the sizing of individual devices involves selecting values from a continuous range and the resulting performance metrics, which are closely correlated to the device sizes, are much more complicated both to reason with and to determine a beneficial compromise among. For instance, typical Operational Amplifiers have up to ten continuous-valued specification metrics [2], whereas digital standard cells have only two, timing and power. A further factor that makes analog design harder is hierarchy. Digital logic has established means of abstraction, meaning that low-level logic can be accurately represented by behavioral models, which is not the case of analog circuits. In fact, low-level properties such as voltages, currents and impedances must all be taken into account when designing an relatively large analog system. These depend highly on the manufacturing process that will be eventually used, thus preventing the reuse of existing designs in new technology nodes. Therefore, analog circuit technology node retargeting may require considerable man-hours, even though the core functionality remains the same.

Layout-induced effects, such as parasitics, are more prominent to newer technology nodes and must be taken into account when designing analog circuits. In addition, designing using newer technology nodes requires the evaluation of each design's yield, which is affected by process, temperature and voltage variations. In the case of large systems that implement both analog and digital functionalities, the analog design is even more difficult since it must take into account system-level interactions, like crosstalk effects. All these render the analog design an iterative procedure, where the steps of topology selections, sizing, layout design and verification take place in turns. In contrast, digital design has effectively become an 'one-pass' procedure, where once the logic is

implemented using a register-transfer level (RTL) representation and some design scripts are configured, the whole design can be implemented automatically. This means that a different variation of the design can occur by only providing an updated RTL description.

The automation of digital design is achieved by leveraging dedicated software tools. These *Electronic Design Automation* (EDA) tools were introduced in the 1980s to help designers cope with the exponential growth of device density and relieve them from the manual design. Nowadays, EDA tools provide digital designers the ability to define logic circuits in high-level abstraction using High-Level Synthesis (HLS) and to focus on the actual design by performing repetitive tasks in a structured, formalized manner. By doing so, the productivity of designers, measured in devices per IC and man-years for design has increased exponentially over the decades. In fact, recent chips that hold tens of billions of transistors [2, 6] would not be possible without the use of custom EDA-based design flows.

The case for analog EDA software, however, is not the same. As stated before, an analog circuit's performance is affected by a number of interleaved factors, ranging from manufacturing process properties, to topology selection and physical design. This complex interaction between different levels of abstraction in analog design does not favour formalized automation procedures. Manual analog design, therefore, has traditionally been the only choice to implement analog functionalities. Designers make use of standalone tools like circuit simulators, layout and verification tools as well as some software design suites that combine some basic functionalities. However, this kind of design flow fails to meet the requirements for modern, complex systems, since it does not allow neither for first-time-right designs nor for meeting time-to-market-constraints in general. A graphical illustration of this analog design gap is shown in Fig. 1.2, where the black line denotes the increase in transistor density in analog blocks over the years and the blue line depicts the designer's productivity, as defined previously. In fact, analog design has become the bottleneck in system design; even though analog blocks take up only 20% of the total area of a modern SoC [7], they require more effort and iterations to design.

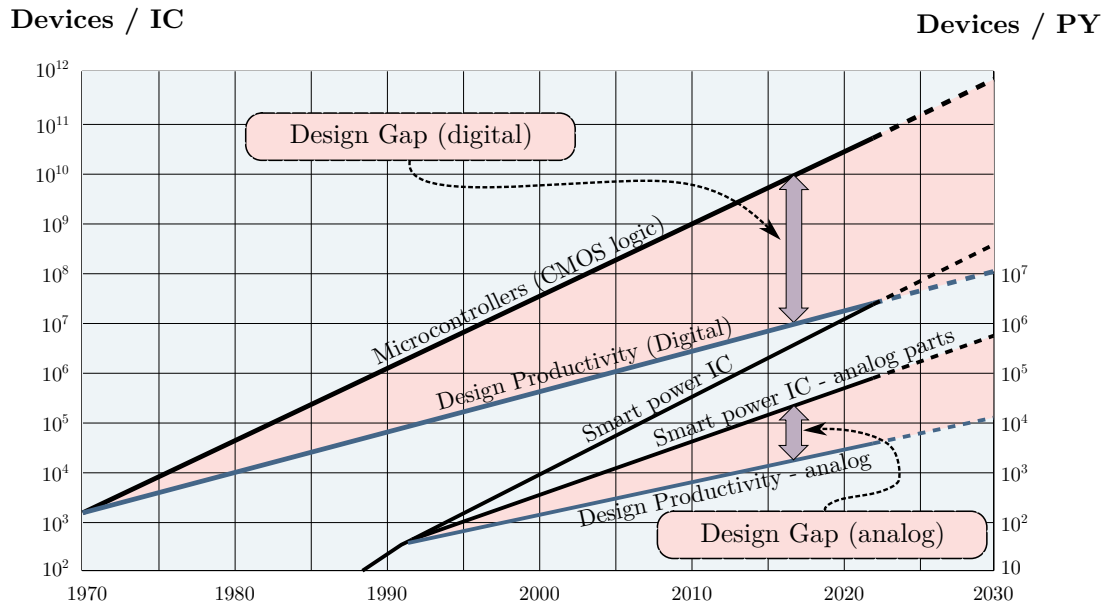


Figure 1.2: Graphical illustration of the analog design gap [2].

Despite the fact that analog design is cumbersome, it is also crucial for next-generation computing systems. Based on the 2020 International Roadmap for devices and systems [8], the *More-than-Moore* initiative aims to develop techniques for the increased integration of even more non-digital functionalities into the future SoCs. By incorporating sensory blocks, power management and RF communications on the chips, analog IC design will become even more important in the foreseeable future. In addition, two recent trends in computing require substantial effort in analog circuitry: the low-power Internet-of-Things (IoT) applications, requiring power-gated designs, sensor interfaces and subthreshold operation and the Compute-in-Memory (CIM) one [8]. CIM targets Artificial-Intelligence applications by implementing dot product and other non-linear functionalities by analog circuitry to reduce power-costs and increase area efficiency. CIM falls under the umbrella of the Analog-AI trend in unconventional computing design.

Given the aforementioned productivity gap in analog circuit design, and taking into account that analog circuits will be in greater demand in the near future, it is reasonable to conclude that steps must be taken to facilitate the job of analog designers. A step towards this can be the development of new, dedicated EDA tools that assist in analog design by providing insights on circuits and automating parts of the overall design procedure.

## 1.2 Thesis Contributions

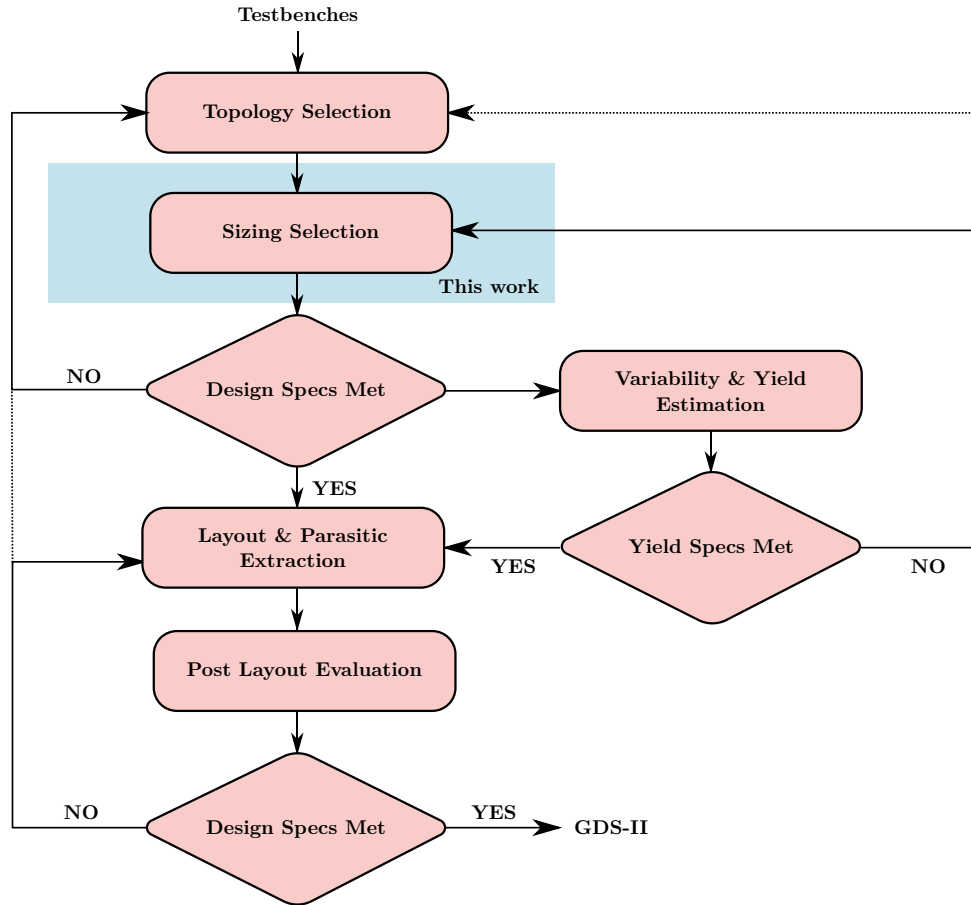


Figure 1.3: A sample analog IC design flow. This work focuses on device sizing selection.

An illustrative example of a typical analog IC design flow is shown in Fig. 1.3. It is shown that there are multiple steps and conditions that a design needs to fulfill in order to proceed to tape-out. In this thesis, we try to facilitate the whole procedure by attacking the problem of device *sizing selection*, which is highlighted in Fig. 1.3. This problem involves determining the sizes of each individual device for a particular topology, given a set of predefined performance metrics constraints. In particular, there are a number of research questions that we address:

- With circuits that have completely different applications, topologies and performance specifications, is there a universal manner in which their device sizes can be inferred in an structured, (semi)-automatic manner?
- Can we develop principled methods for exploring the competing trade-offs of different topologies and reason about their attainable performance?



- With the recent developments in computational intelligence and Machine Learning (ML), is there a way to develop algorithms that render the aforementioned procedures accurate and fast enough for the designer's to use them?

This thesis addresses the above questions and the challenges for automated EDA-powered design in general by a series of works:

- Study, application and expansion of black-box simulation based approaches, including Evolutionary Computation (EC) ones, for sizing both analog and RF circuits in the block level. We compare EC algorithms against each other and develop custom operators to enhance mixed-variable black-box optimization, as many analog-design problems can be cast as such.
- Development of ML-based Single-Objective optimization algorithms, requiring few evaluations to reach feasible, optimal solutions. By utilizing the framework of Bayesian Optimization (BO), one can optimize expensive-to-evaluate black-box functions. We develop SO BO variants that are scalable with the input variable space's dimensionality and exploit kernel approximations to decrease the complexity that BO entails when many query points are sampled. This combination of attributes is a novel approach to BO.
- Development of new Multi-Objective, low-budget optimization algorithms, for design space exploration and trade-off reasoning. By using a local-based scheme and a new activation function that works with multiple objectives, a new, multi-objective Bayesian Optimization variant is proposed. We also present an approach to having multiple query points at the same time, by building analytic approximations to Gaussian Process samples. A deep analysis of the proposed method suggests that it finds wider pareto fronts and more feasible solutions compared to other state-of-the-art black box algorithms.
- Incorporation of Process, Voltage and Temperature variations in the context of automatic sizing, for robustness. We propose an optimization formulation with which circuits under test can be optimized under PVT constraints. The proposed approach successfully yields robust circuits under tight constraints.
- Using Deep-Learning Techniques for device surrogate modeling and mixed-variable sizing. We present a Deep-Learning scheme in which data driven representations of devices can be derived. These latent representations are continuous valued ones and allow us to define optimization problems whose input space is the union of device sizes and latent vectors, thereby allowing for BO in mixed-variable spaces.

- The development of a user-friendly API for procedural optimization and simulation execution. We developed an API to allow for procedural simulation, optimization, design space exploration, device modeling and circuit process retargeting in pure Python. All of the described approaches can be reproduced using this framework and an off-the-shelf simulator.

### 1.3 Thesis Outline

The thesis is organized as follows. First, Chapter 2 provides a thorough literature overview of the analog sizing problem, provides description on different strategies for the automation of circuit design and comments on the advantages and disadvantages of the state-of-the-art. Chapter 3 presents the formulation of the analog circuit sizing problem as a simulation-based and black-box optimization one. Using this formulation, and by establishing the background of the EC-based black box optimization algorithms, this chapter proceeds to illustrate the use of simulation-based optimization on two LC-VCO topologies, and comments on the resulting design spaces that were found. Furthermore, Chapter 3 illustrates two proposed methods for enhancing the performance of the EC-based algorithms, in cases where constraints apply and mixed-variable search spaces exist.

Chapter 4, on the other hand, illustrates the use of low-budget optimization algorithms to tackle the large runtimes of circuit simulations. The reader is reminded of the basic mathematical background of Gaussian Processes and introduced to Bayesian Optimization. In addition, a new methodology to scale Bayesian Optimization to large variable spaces is discussed, through the use of trust-regions, which restrict the modeling of the loss landscape in certain parts only. The methodology is illustrated on two real-world circuits, and provides better runtime and results compared to other SO alternatives.

Chapter 5 extends the work of the previous one to MOPs. A new Multi-Objective Bayesian Optimization is introduced, which uses trust-regions to scale to large variable spaces. In addition, the use of RFF and the hypervolume metric are discussed in the context of the proposed algorithm's acquisition function. A thorough comparison of the algorithm's performance against baseline algorithms and EC-based ones, in real-world circuits, considering both nominal and PVT variations.

Chapter 6 describes a novel, Deep-Learning approach to extend BO to mixed-variable spaces, in the context of analog circuit sizing. The proposed approach consists of using Deep Generative models to capture the functional behavior of devices that are parametrized by both discrete and continuous variables. The methodology is demon-

strated on two device models, in a popular PDK, and example automatic sizings that incorporate this procedure are also given.

Finally, Chapter 7 concludes the thesis with a general summary of its contributions and a presentation of open problems, paving the way for future work.



# Chapter 2

## Related Work

The last decades, many attempts to tackle the automation of analog and RF circuit sizing and process migration have been proposed in the literature. These approaches vary in terms of both algorithmic aspects and formulation of the sizing problem. Commercial EDA vendors have also proposed tools offering optimization-based sizing functionalities [9, 10], but fail to deliver automation and acceptable results. Based on their properties, the approaches for automatic device sizing can be categorized into two groups, namely knowledge-based ones and optimization-based ones. The following sections discuss both groups of studies and compare their pros and cons.

### 2.1 Knowledge-Based Approaches

The defining property of knowledge-based approaches is their attempt to incorporate some sort of information about the circuit and existing expert methodologies to size circuits into procedural sizing frameworks. The prior information can be either analytic equations that relate device sizes to the circuit's performance metrics, heuristic rules for sizing selection in steps, or device interaction modeling using the circuit's netlist graph. Some of the first approaches for analog circuit sizing include [11, 12], which require user intervention to define a design plan, i.e. a breakdown of the overall sizing procedure into smaller and simpler tasks. Once these rules are decided, the procedure can be applied to provide both a topology and an initial rough-cut sizing. These procedures expect as inputs both design specifications and fabrication process information. Although they are intuitive and easy-to-grasp for the designers, these approaches are limited by their dependence on predefined building-block libraries that guide the design procedure, and may not include sub-blocks of interest for a specific design.

A particularly popular approach has been the use of graph structures to encode topological constraints in the sizing procedure. In [13, 14] graph structures are used to perform symbolic analysis on circuit topologies. Given small-signal models of the devices used, analytical expressions that relate circuit performances and device sizes are computed, accounting for terms that cancel-out. These approaches for symbolic analysis are later used to automatically derive cancellation-free analytic equations to optimally

size analog circuits in works such as [15, 16].

Another perspective in graph-based circuit sizing is given in [17, 18, 19, 20, 21], where device dependency graphs are generated from netlists and user-input and encode the sizing procedure to be executed. However, the most interesting aspect of these works is the formulation of the design process as an operating point driven one. This practically means that a search engine, whether it is an optimization algorithm or heuristic rules, operates on the space of circuits' electrical parameters such as voltages and currents and not on device sizes. The inverse mapping from these variables to actual geometric sizes is done by using biasing operators, multivariate functions built for every compact model utilized in the schematic under test.

The operating point formulation of analog sizing is more intuitive, since designers often reason about circuit's performance with respect to voltages and currents. In [22], it is suggested that this approach could lead to significant sizing performance gains, since it results in significant reduction of the search space. More recent works such as [23, 24] include operating point driven constrained optimization procedures, by incorporating the small-signal properties of transconductance-to-current ratio, i.e.  $gm/I_d$  and the Inversion Coefficient one. Both of these properties indicate the level of inversion of transistor devices in a process-independent manner.

The works of Murman and Jespers [25, 26] are based on the operating point driven methodology and utilize the  $gm/I_d$  property to guide analog block sizing. Their approach involves computing look-up tables of the compact models of a particular process and using them in user-defined sizing procedures. The user writes procedures in high-level programming languages to define analytical equations that describe the block's performance in terms of its electrical variables and/or the devices' sizes. The unknown variables are computed using the regression models and inverse mapping based on the look-up tables. The aforementioned works have inspired a number of efforts to include both look-up tables and the  $gm/I_d$  metric in circuit sizing, such as [27, 28, 29, 30, 31, 32, 33].

Although the previously described methods for analog circuit sizing may be user-friendly and provide the designer with insights for the procedure, they have the following disadvantages:

- They require analytical expressions for circuits' performances. Some simple topologies may be easy to derive expressions for, but complex performance metrics such as timing jitter, phase noise and third-order intercept point are not amenable to closed form expressions. Besides, the use of newer process nodes results in high-order effects that are not easy for designers to account for when writing down equations.

- The use of operating point driven methods does not take into account large signal behavior of compact models. This seriously restricts their application in most of the analog circuits considered nowadays.
- They require libraries of compact models and/or subcircuits, which need to be updated when moving to newer process nodes.

## 2.2 Optimization-Based Approaches

The second category of automatic circuit sizing approaches is the optimization-based one. These methods cast the sizing procedure as a minimization problem, where the devices of the circuit under test are updated until the imposed specifications are met. They are composed of two basic ingredients; an optimization algorithm and an evaluation engine. As shown in Fig. 2.1, an iterative procedure that loops between the search process for potentially promising solutions and their evaluation is the core of this approach. Here, we distinguish between the ones that use simulators and the ones that use models for evaluations.

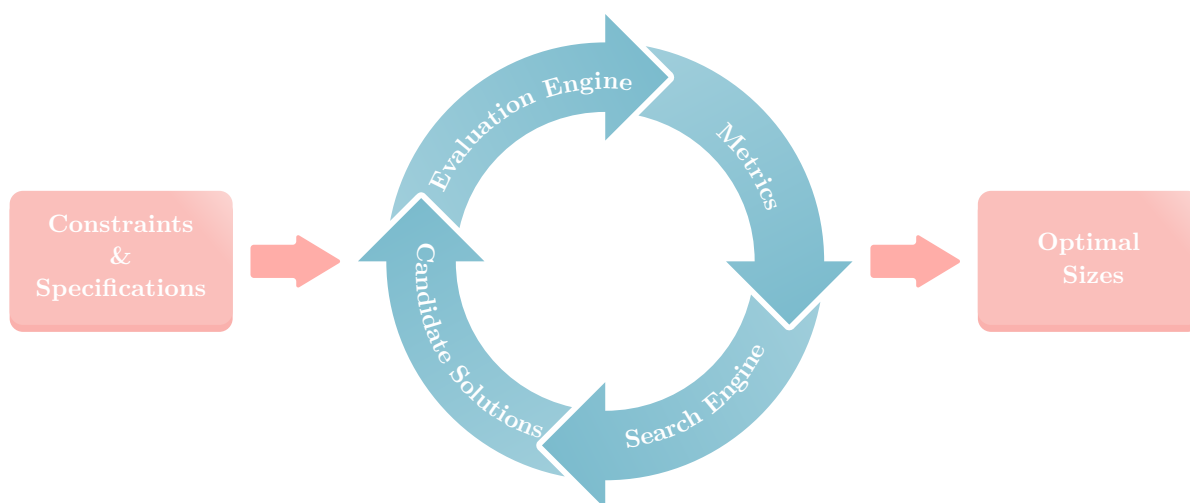


Figure 2.1: An illustration of the optimization-based approach to analog circuit sizing.

### 2.2.1 Model-Based Sizing

The use of predictive models for circuit sizing involves two parameters that need to be taken into account; the kind of the model itself and the way this model is used to guide the optimization. The most popular approach involves statistical models that relate the devices' geometries to the circuits' performance metrics. These models are used as surrogates to an actual simulator and are evaluated to provide the optimizer

with a feedback about its current best. These models are trained using simulation data either prior the optimization or on the fly.

In the early 2000s, the seminal work of M. Hershenson [34] proposed a geometric-programming optimization scheme to derive component values for operational amplifiers. The method required the use of analytic expressions for fitness and constraint functions in the form of posynomials [34] and guarantees global optimal solution. Despite the novelty of this work, a limiting factor for its wider adoption was the need for analytic posynomial functions. Therefore, a number of works were proposed later [35, 36, 37, 38] that suggest building posynomial-type models out of simulation data and following, using geometric programming to derive optimal solutions.

More recently, the use of Artificial Neural Networks (ANNs) [39] for the surrogatization of circuits' performance metrics has become the most popular approach. For instance, [40] uses ANNs to approximate locally the performance space of a particular circuit. This model is incorporated on a two-level optimization scheme where a Genetic Algorithm (GA) is used to search globally for promising regions of the design space and a local search scheme uses the ANN to find a local minimum of certain regions. In [41], an ANN trained on basic analog sub-blocks using simulation data predicts device sizes, given the required specifications. The work in [42] proposes to train ANNs during the optimization procedure. These models are then used to replace the simulator for evaluation, resulting in significant runtime gains. Besides ANNs, several ML-based prediction models are used in the context of optimization based sizing. These include Support-Vector-Machines (SVMs) [43], sparse regression models [44], and kriging models [45], which are essentially Gaussian Processes.

The aforementioned model-based techniques share a common goal; the replacement of time-consuming simulator evaluation using a fast-to-evaluate surrogate model. There are approaches which use predictive models, however, in tandem with simulations. For instance, in [46], a classifier model is trained on the fly and incorporated within an Evolutionary Algorithm (EA) to better determine which candidate solutions are more promising and shall be simulated. Similarly, in [47], an ANN model is trained and incorporated as the infill criterion of a Differential Evolution (DE) algorithm [48], in order to improve the algorithm's performance. The GASPAD method [49] uses kriging models in a similar fashion as well.

The contribution of the model-based approaches is twofold; the replacement of time-consuming simulators with fast surrogates and the incorporation of knowledge specific to particular circuits in a representation that is later used to enhance the optimizer's search. However, there are considerable drawbacks as well. First, usually there is no a-priori knowledge about the complexity of the underlying functions that are being



approximated by the models. This practically leads to two cases; either the models are over-parametrized, leading to overfitting the training simulation data [50], or they are under-parametrized, which means that they will be unable to generalize to unseen inputs [50]. In both cases the model's predictions may be inaccurate and lead to negative effects on the optimizer's performance.

In addition, training and using models can be time-consuming sometimes. In the case of Gaussian Process for instance, their cubic computational complexity [51] can induce training times that may exceed the simulation time. One more interesting note about model-based approaches is the following; the need for accurate models necessitates an abundance of data, which in turn require many simulations. This, however, is in contradiction with their advertised small-data nature. Training models involves optimization procedures and simulation data, which otherwise would have been used for rough optimization based approaches (see next subsection). There is essentially no guarantee that these approaches provide any kind of shortcut towards the optimal solution, which is backed by the no-free-lunch theorem [52].

### 2.2.2 Simulation-Based

This subsection discusses the works that address circuit sizing using simulation data. This approach requires an interface to a commercial simulator, which assesses the performance of potential solutions and returns a score to the optimization algorithm, essentially forming a *simulation-in-the-loop* procedure. One of the first approaches to attack analog circuit sizing with a simulator was [53], that allowed designers to tune the circuit's parameters, provided an initial design. MAELSTROM [54] proposed the use of a simulator and a simulated-annealing algorithm to automatically size circuits, without any requirement for initial sizes. The use of a stochastic optimizer is due to its robustness on nonlinear, non convex cost functions, such as the ones in analog circuit sizing.

In the literature, the use of stochastic algorithms for optimization has been the most popular choice, mainly due to the ease of implementation and their empirical effectiveness. In this context approaches from the field of EC like [7, 55, 56, 57, 58] use a population of initial, randomly selected potential solutions, which are referred to as individuals, and based on their assessment by the simulator proceed to create new solutions in a stochastic manner, by using mutation, crossover and selection operations. Different optimization algorithms are produced by the combination of different operators. For instance, in [7], a GA algorithm is utilized employing heuristics in its mutation operation and a step-size control policy that reduces the variance of the newer potential solutions as the procedure evolves. In [55], the optimization procedure involves a competitive co-evolutionary differential evolution and incorporates a feasibility rule to account for

constraints, instead of using penalty functions.

Variation-aware sizing has also been addressed within simulation-based approaches. FUZYE [58] uses a GA in parallel with a clustering technique to avoid running full Monte Carlo (MC) simulations on all candidate solutions at each iteration. By running a single MC simulation for the candidate solution at the center of each cluster and computing membership degrees for all solutions and clusters, each solution is assigned an approximate yield score. A homotopy type formulation of the sizing problem is proposed in [57], where the evaluation procedure is decomposed into multiple steps; at the lowest level, fast-to-evaluate metrics such as the ones that depend on DC simulations are computed. The complexity of the evaluation is increased in later stages, where candidate solutions are cut-off as soon as they fail to meet a particular specification. At the highest level, process, voltage and temperature variations are included in the simulation testbench, thereby reducing the effective number of corner simulations for the initial pool of candidate solutions.

Interesting attacks on optimal sizing of RF circuits have been proposed in [59, 60, 61, 62], where Multi-Objective and Many-Objective optimization procedures have been utilized. In particular, [59] tackles the design space exploration of a class B/C Hybrid-Mode Voltage Controlled Oscillator (VCO), accounting for process, voltage and temperature variations. The Non-dominated Sorting Genetic Algorithm (NSGA-II) [63] is utilized to find non-dominated solutions over a 108-dimensional performance space. The resulting solutions can be visualized and the designers can reason about competing trade-offs and select a device sizing that best fits their application. Similarly, [60, 61] attack the automatic design space exploration of VCOs in a 65nm process, whereas [62] explores the design space of CMOS Low Noise Amplifiers (LNAs) for 5G communications. By employing this Many Objective approach, the designer is relieved from having to make a compromise between competing trade-offs a-priori, when defining the optimization formulation.

The most important shortcoming of the discussed simulation-based approaches is their computation burden. Since they employ population-based stochastic algorithms, they typically require many evaluations to reach feasible and satisfying solutions. Taking into account that some analyses for analog circuits may be time consuming themselves, the overall computational overhead is often unbearable. In addition, an obvious limitation of these approaches is the fact that the knowledge accumulated in the process of the optimization, which could be used to better guide the search process, is not used at all.

To alleviate the aforementioned drawbacks of populations-based approaches, a new trend in analog circuit sizing involves using low-budget optimization algorithms, i.e.

algorithms that typically require fewer evaluations to reach feasible and optimal solutions. Bayesian Optimization (BO) [64] is such an algorithm. It builds models of the performance space using GPs but does not use them to approximate performance values. Instead, the GPs are used in an auxiliary optimization task, where the point of the search space with the maximum utility in the search process is found. The utility metric is defined by means of an acquisition function. The maximum utility point is chosen as the next point for evaluation, and the whole procedure is an iterative one.

Recently, there have been applications of vanilla BO and some variants to analog circuit and integrated system sizing [65, 66, 67, 68, 69]. In particular, WEIBO in [68] uses an Expected Improvement (EI) [64] acquisition function, which is transformed so as to account for constraints. By multiplying the EI acquisition function with the outputs of a model that provides the probability of feasibility of each point in the search space, this approach is shown to reach feasible solutions. In [66], a Multi-Objective BO variant for design space exploration is proposed, where the Lower Confidence Bound (LCB) of each objective are used to define an optimization problem, which is solved using NSGA-II. Taking into the cubic complexity of GP training, [67] proposes a BO variant where the GP's kernel is a learnable neural network. This way, the complexity for GP training is reduced to  $\mathcal{O}(N)$ .

The above BO approaches share one common drawback; they are sequential in nature, which means that at each iteration they evaluate a single point. This leads to excessive runtime, since it necessitates that each GP should be trained for every single of the evaluations. Recent works [70, 69] address this issue by defining batched acquisitions functions. For instance, the MACE algorithm [69] uses multiple vanilla acquisition functions, such as EI and LCB, and defines a multi-objective auxiliary optimization task, which results in a pareto set. The points in the pareto set are simulated in batch mode, i.e. in parallel or using sweep analysis, before moving on to GP model training.

## 2.3 Other Approaches

Besides the methodologies discussed in the context of optimization-based sizing and knowledge-based one, there is a number of works that are more exotic to be classified in either one. For instance, in [71], an ANN is trained to repurpose previously acquired data from optimized circuits to new ones, which may be designed in a different process, or using different parameters such as loads, supplies, etc. The model, thus, is trained to produce device sizing without user intervention or simulation executions.

A separate approach involves mining causal relationships from simulation data [72, 73, 74, 75]. In these works, a framework of knowledge representation is built and it is

parametrized by data in an effort to replicate cognitive human activities. These methods use both the topology structure and simulation data derived from sensitivity analysis to derive the performance trade-offs. They are both used to reason about the sequence of individual device sizing, when traversing the solution space.

# Chapter 3

## Evolutionary Algorithm-based Sizing

In this chapter the automatic sizing and trade-off exploration of analog circuits is discussed from the perspective of the population based EA approach. The formulation of the sizing problem as an optimization one is given and the application of EAs on a set of analog and RF circuits is discussed.

The structure of this chapter is as follows; first, an introduction to the basic concept of Evolutionary Computation (EC), which encompasses EAs, is given, and the common operations that take place within the frameworks of EAs are discussed. Following, a Single-Objective (SO) formulation of the sizing problem is introduced, and it is used to define and solve two different circuits. Last, a Multi-Objective (MO) formulation is also introduced and solved using EAs, in order to provide performance trade-off information of various circuits.

### 3.1 Evolutionary Computation Preliminaries

This section provides the motivation and the basic operations of EC. A brief historical background is given, some of EC's more popular algorithms applied for optimization are discussed and common variants are also presented.

The field of EC applies natural principles within the framework of AI to achieve some sort of automated problem solving, such as optimization. Although its basic concepts have been introduced back in the 1940s by Alan Turing [76], it gained much of its popularity in the 1990s by its application to real-world problems, using contemporary computing [77]. In the literature one may find various titles for nature-inspired algorithms, like genetic algorithms (GAs), evolutionary algorithms (EAs), etc, which all belong to the broader field of EC and follow its basic principles, which will be discussed following.

EC draws inspiration from both Darwin's theory of Evolution and genetics. From a high-level perspective, two key assumptions are made; there is an environment which may hold up to a certain number of entities, or *individuals* and every individual aims to

reproduce. In addition, each individual has their own characteristics, called phenotypes, that determine its ability to adapt to the environment and reproduce. Within the concept of EC, this adaptation capability is quantified by the *fitness* of each individual. A natural consequence is that individuals struggle for reproduction and survival, since their total number, i.e. their *population* is limited. Individuals that are not well-adapted, i.e. their phenotypes are not favourable, do not survive. This process is called survival of the fittest.

Besides the competition between individuals for survival, the second driving force of the evolutionary process is the variation of the phenotypes. This involves small, random variations in the phenotypes of the individuals during the reproduction process. Therefore, the evolutionary process relies on the survival of the fittest and the phenotypic variations so as to yield individuals that adapt better to their environment, and at the same time maintain diversity of phenotypes throughout their population.

In a low-level perspective, EC draws inspiration from genetics to mimic the process of reproduction. The main principle that EC borrows from genetics is that an individual's phenotype is completely described by another hidden set of characteristics, called *genotype*. The genotype is a set of *genes*, which can take values from a prescribed range. The 'values' of the genes constitute the individual's genotype and consequently define its phenotype, i.e. its chances of survival and reproduction. In reality, all of the genes in a living organism are arranged in *chromosomes*, which in higher forms of life come in pairs. The chromosomes of life forms that are reproduced in pairs are a combination of paternal and maternal ones, a process which in the context of EC is referred as *crossover*. In practice, this allows genetic information to pass to the offspring, affecting their phenotype and their chances of survival. However, the offspring differ from their parents due to the mutation, i.e. random variations that take place during the reproduction.

EC-based optimization algorithms attempt to incorporate the previously discussed mechanisms in a stochastic trial-and-error scheme, in order to approximate the solution of optimization problems. Given an optimization problem to be solved in iterations, a trial-and-error approach requires the creation of candidate solutions, which will be evaluated to derive their goodness. A correspondance between this scheme and the evolutionary process that EC tries to mimic holds and it is demonstrated in Table 3.1, where components of each process are mapped to their duals.

To emulate the evolutionary process, EC-based algorithms must maintain a population of individuals, which both holds the candidate solutions of an optimization problem and serves in the selection process. The competition of individuals to survive the selection results in an increase in the fitness of the population, i.e. the quality of the candidate solutions. A typical EC-based optimization algorithm involves an initial-

Table 3.1: Evolution and Problem Solving Equivalence

<b>Evolution</b>		<b>Stochastic Problem Solving</b>
Environment	$\longleftrightarrow$	Problem
Individual	$\longleftrightarrow$	Candidate Solution
Fitness	$\longleftrightarrow$	Quality

ization step, where the first generation of individuals is created in random. The fitness of these individuals is measured, for instance by evaluating a black-box function's output on their genotypes. Based on their fitnesses, pairs of individuals are selected to reproduce. The offspring resulting from the reproduction are produced by two distinct operations; the recombination, which involves the combination of the genotypes of the parents, and the mutation which alters the genotype of the offspring. The pool of offspring candidate solutions are evaluated and they compete with the parents for survival in the next generation. This iterative process is depicted in Fig. 3.1, and it terminates when a predefined criterion is met.

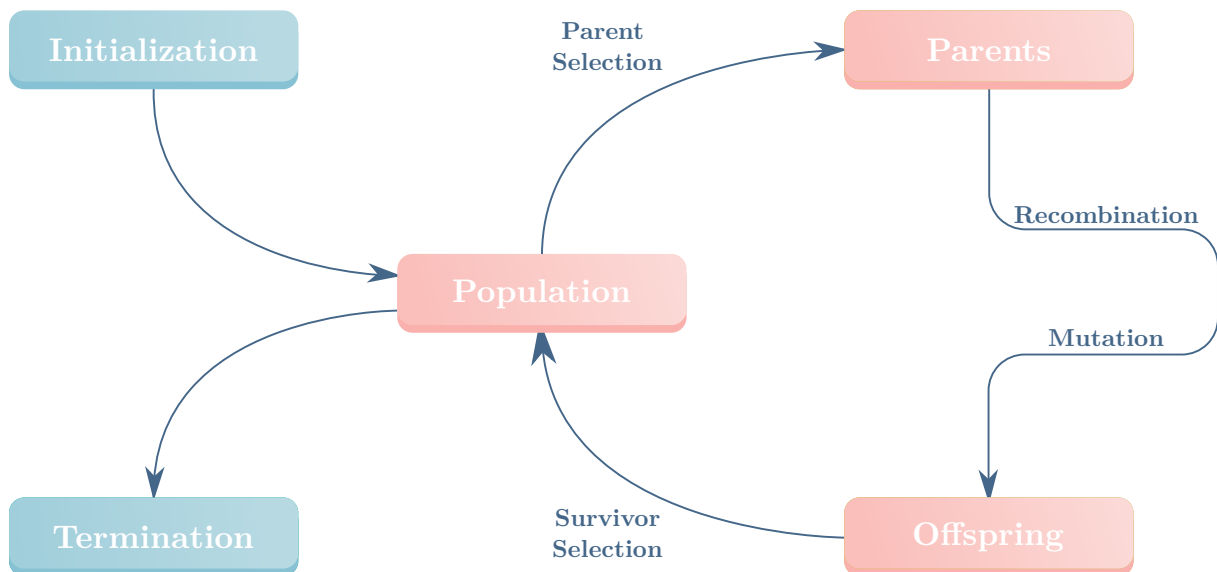


Figure 3.1: A flowchart depicting the general operation of EC-based optimization algorithms. Figure adapted from [76]

The factors that lead to the improvement of the fitness across generations ( iterations) are the selection mechanism of the EC-based algorithm, as well as the variation process ( recombination and mutation). Although one can alter the way these processes work, they will always involve some kind of stochasticity. This renders the whole algorithm a stochastic optimization approach. In the following subsections, we will discuss the specific details of three EC-based algorithms used in this work, namely Genetic

---

Algorithms, Differential Evolution and Covariance Matrix Adaptation.

### 3.1.1 Genetic Algorithms

The Genetic Algorithm (GA) is the most popular optimization algorithm within the context of EC. It follows the exact same procedure as shown in Fig. 3.1, with its mutation, crossover and selection operators being inspired from natural evolution. Given a hyperparameter for the population count,  $M$ , and assuming that the search space is  $\mathbb{R}^N$ , the initialization procedure typically consists of sampling from a  $N$ -dimensional Gaussian or a Uniform distribution in  $\mathbb{R}^N$ . The resulting vectors  $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M]$  constitute the initial population. In this step, the most important aspect is the good coverage of the whole search space. To achieve this, some modern techniques resort to quasi-random sequences [78] that produce finer uniform partitions of the design space, compared to other approaches.

As stated previously, environmental pressure is the main force for attaining better solutions through the use of EAs, and it is realized with the selection mechanism. In the context of GAs, there exist numerous selection mechanisms, however the most important one is the roulette selection operator. This assigns a probability of survival to each individual  $\mathbf{x}_i$ , according to its fitness score. All individual probabilities are weighted so as to add up to one and then a random sampling takes place to determine the individuals that survive the selection. By doing so, all of the individuals in the population have a non-zero probability of surviving, thereby ensuring (implicitly) the diversity of the features in the population.

Emulating the natural process of fertilization, where maternal and paternal chromosomes are combined to produce the chromosome of the offspring, the GA defines pairs of surviving individuals to mate. Considering a case where individuals  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are selected to mate, their chromosomes, i.e. the entries of the vectors, are swapped to produce the new offspring vector. Although there are several variations to this process, the single crossover point one is the most widespread. It involves sampling a random integer  $c_p$  from the range  $[1, N]$  and setting the entries of the offspring vector  $\mathbf{x}_{new}$  equal to the first  $c_p$  ones from  $\mathbf{x}_i$ , while the rest of them are set equal to the last  $M - c_p$  ones from  $\mathbf{x}_j$ . This process is illustrated in Fig. 3.2.



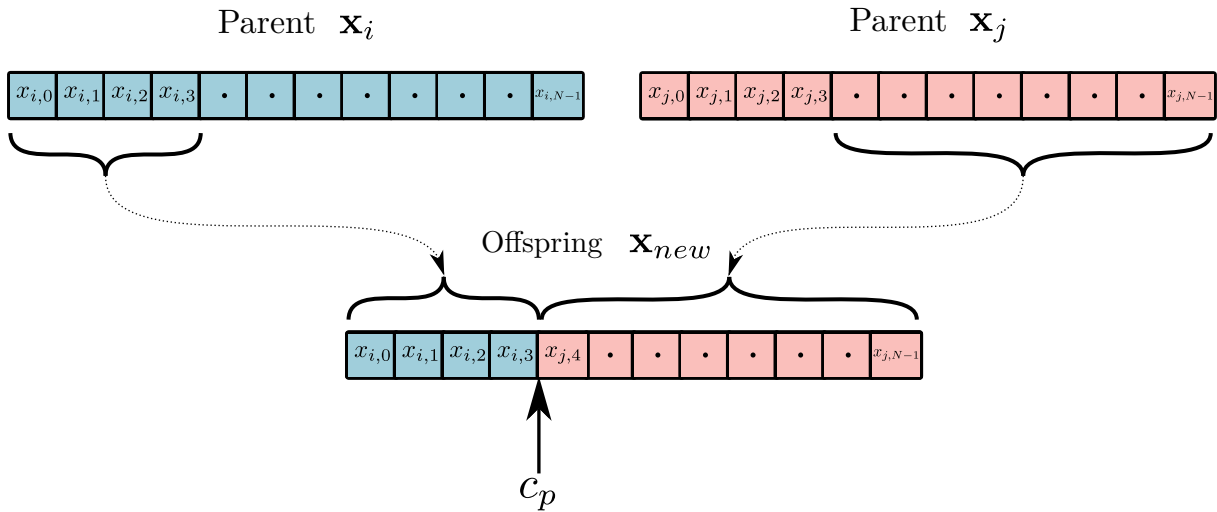


Figure 3.2: A depiction of the single-point crossover operation in GAs.

The last operation within the GA is the mutation one; after the crossover, the entries of the offspring vector are perturbed randomly to emulate the random fluctuations that take place in the fertilization process. Typically, this process is implemented by adding a multivariate vector sampled from an isotropic Gaussian distribution with given mean vector and variances.

### 3.1.2 Differential Evolution

Differential Evolution (DE) [79] is another popular algorithm from the family of EC. Although DE follows the same principles as most EC-based algorithms, having mutation, recombination, crossover and selection operators, it does not mimic any natural phenomenon. In practice, at a particular iteration of DE, each offspring individual is produced by applying a crossover operation between a single parent individual and a mutant vector. Mutant vectors are produced by a mutation operation, where a weighted difference of a number of individuals is computed and added to a particular individual vector of the population.

More formally, let us consider an optimization problem and let us assume that its search space is  $\mathbb{R}^N$ . The initial population of  $M$  individuals are generated at random and are denoted as  $X = [\mathbf{x}_i]_{i=1}^M$ , where each individual vector has  $N$  entries such that  $\mathbf{x}_i = [x_{i,0}, x_{i,1}, \dots, x_{i,N-1}]$ . For each individual in the population, its associated mutant vector  $\mathbf{v}_i = [v_{i,0}, v_{i,1}, \dots, v_{i,N-1}]$  is created by following two steps:

- Select  $k$  individuals from the current population  $[\mathbf{d}_i]_{i=1}^k$  at random and compute their pair-wise differences
- Add a scaled version of the differences to a base vector  $\mathbf{d}_{base}$ .

In the typical case where  $k = 2$ , the mutant vector is given by

$$\mathbf{v}_i = \mathbf{d}_{base} + F \cdot (\mathbf{d}_1 - \mathbf{d}_2), \quad (3.1)$$

where  $F$  is the scaling factor and it is a hyperparameter of the DE algorithm. The selection of  $\mathbf{d}_{base}$  is also a parameter of the whole procedure and there exist many variants for the differential operation based on it. The most popular strategy, which is followed in the experiments in this work and it is referred to as the DE/rand/1 variant, involves selecting the base vector in random from the current population.

Following the creation of the mutant vectors, a recombination procedure takes place where a trial vector  $\mathbf{v}_i = [v_{i,0}, v_{i,1}, \dots, v_{i,N-1}]$  is produced by crossing-over features from the individual  $\mathbf{x}_i$  and its associated mutant vector  $\mathbf{v}_i$ , such that

$$v_{i,j} = \begin{cases} v_{i,j} & \text{if } \text{rand}(i, j) \leq CR \\ x_{i,j} & \text{otherwise.} \end{cases} \quad (3.2)$$

Function  $\text{rand}(i, j)$  yields samples from a uniform distribution in the range  $[0, 1]$  and  $CR$  is a scalar hyperparameter that controls the crossover probability. The resulting trial vector  $\mathbf{v}_i$  is evaluated and competes with the individual  $\mathbf{x}_i$  based on their fitness values, to determine which one survives the next generation.

### 3.1.3 Covariance Matrix Adaptation

A particular subset of EC-based algorithms targeting optimization problems is the family of Evolutionary Strategies. They typically work in continuous valued search spaces and use a sampling procedure to generate candidate solutions, i.e. offspring in the EC framework. Starting from an initial population, the Evolutionary Strategies typically select the fittest  $\mu$  individuals, which are used to sample the search space and produce  $\lambda$  offspring candidates, where  $\mu < \lambda$ . The resulting  $\lambda$  individuals are then compared to determine the top  $\mu$  individuals to survive the generation and to be used eventually for the generation of the next offspring. This sampling scheme is adopted by the Covariance Matrix Adaptation (CMA) algorithm, which was first proposed in [80] and has become the most popular approach for SO problems among the Evolutionary Strategies.

Considering a SO minimization problem with  $N$  variables and denoting its search space as  $\mathbb{S}$ , CMA works by sampling offspring from a parametrized multivariate Gaussian distribution  $\mathcal{N}(\mathbf{m}_i, C_i)$  over  $\mathbb{S}$ , where  $i$  is the index of the iteration. Therefore, at the  $i$ -th iteration  $\lambda$  individuals are sampled from the aforementioned distribution. These individuals are then compared to find the fittest  $\mu$ , which are used in a weighted sample

average form to derive the  $\mathbf{m}_{i+1}$  mean vector. This serves as a recombination operation, where old individuals determine the features of the future offspring. At each iteration, the covariance matrix of the Gaussian distribution is updated by accounting its previous state, the evolution path, i.e. the path that the mean vector  $\mathbf{m}$  has followed throughout the iterations and the most fit offspring of the current one. In addition, the update procedure of the CMA depends on the step size control variable  $\sigma$ , which controls the convergence rate of the covariance matrix. A thorough explanation of how the CMA update works, as well as its mathematical foundations can be found in [81].

## 3.2 Single-Objective Sizing

In this section we will consider the case of circuit sizing using EC-based algorithms by formulating it as a Single Objective (SO) optimization problem. Consider a given topology, with the geometric variables of its devices (MOS devices, capacitors, etc) arranged in a  $(1, d)$  vector,  $\mathbf{x}$ . In the SO case, only a single circuit performance metric is accounted for optimization. We may view this metric as a black-box function  $f : \mathbb{R} \rightarrow \mathbb{R}$ , which we wish to minimize. In addition, to restrict the circuit in desired performance regions, one may indirectly account for  $l$  other performance metrics in the form constraints. These can also be viewed as black-box functions and here we use the notation  $[g_i(\mathbf{x})]_{i=1}^l$  to denote them. Last, the search for each and every geometric variable of the circuit must be restricted within specified ranges, in order to account both for manufacturability and for the step of the layout. Taking all the above into consideration, the mathematical formulation of the sizing problem as a SO optimization problem is defined as:

$$\begin{aligned} \min \quad & f(\mathbf{x}), \quad \mathbf{x} = [x_1, x_2, \dots, x_d] \\ \text{s.t.} \quad & g_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, l \\ & L_i \leq x_i \leq U_i, \quad i = 1, \dots, d \end{aligned} \tag{3.3}$$

where  $L_i$  and  $U_i$  are the lower and upper bounds of the  $i$ -th variable in the design vector  $\mathbf{x}$  and the space  $S = \prod_{i=1}^d [L_i, U_i]$  is the search domain of the problem.

## 3.3 Multi-Objective Sizing

Let us now consider a more general formulation of the optimization problem for analog circuit sizing. Considering once more that a subset of the studied performance metrics must be restricted within pre-specified values, the optimization problem is a

constrained one and can be defined as:

$$\begin{aligned}
\min \quad & F(\mathbf{x}), \quad \mathbf{x} = [x_1, x_2, \dots, x_d] \\
\text{s.t.} \quad & g_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, l \\
& L_i \leq x_i \leq U_i, \quad i = 1, \dots, d
\end{aligned} \tag{3.4}$$

In comparison to the initial SO formulation in Eq. 3.3, this case substitutes the scalar-output function  $f(\mathbf{x})$  with a vector of  $m > 1$  scalar-output functions  $F(\mathbf{x}) = [f_i(\mathbf{x})]_{i=1}^m$ . The simultaneous minimization of all function components of  $F$  is the goal of Eq. 5.1, and it is the way Multi-Objective Optimization problems (MOPs) are formulated.

Typically, in the case of MOPs, the function components of  $F$  are conflicting, i.e. the improvement in one component leads to the deterioration in another. Therefore, minimizing  $F$  results in multiple solutions that constitute a *Pareto Set* (PS). For a point  $\mathbf{x}^*$  to be a pareto optimum, i.e. to belong in the PS, every variation of  $\mathbf{x}^*$  further minimizing one of the  $m$  objective functions  $f_i(\mathbf{x}^*)$  must also deteriorate another one. We say that a feasible solution  $\mathbf{x}_1$  dominates another feasible solution  $\mathbf{x}_2$ , if  $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)$  for  $i = 1, \dots, m$  and there exists  $m' \in [1, \dots, m]$  such that  $f_{m'}(\mathbf{x}_1) < f_{m'}(\mathbf{x}_2)$ . Consequently,  $\mathbf{x}_1$  belongs to the PS if it is feasible and it is not pareto dominated by any other  $\mathbf{x} \in \mathbb{S}$ . We also use the term *Pareto Front* (PF) to denote the image of PS via function  $F$ , i.e.,  $\text{PF} = \{F(\mathbf{x}) | \mathbf{x} \in \text{PS}\}$ .

In an arbitrary finite subset  $P = \{\mathbf{x}_i\}_{i=1}^N$  of  $\mathbb{S}$ , multiple levels of pareto dominance (subsets of  $P$ ) may be defined. The first level is the subset of  $P$  including all vectors that are non-dominated by any other vector in  $P$ , and only those. The first level is an approximation of PS of the MOP. The second level is defined by removing the first level (approximate PS) from  $P$ , and, keeping the non-dominated solutions from this set. The process is repeated until all samples in  $P$  are assigned to a dominance level. In the context of analog circuit sizing, a PF provides designers with a model of the conflicting relationships of circuit performance aspects, aiding the device sizing.

### 3.4 Constraint Handling

In the context of EC-based algorithms, the selection and recombination operators do not offer any functionality to handle constrained optimization problems. In this case, the goal of the optimizer must be twofold; to drive the search to the parameter space region where the constraints are satisfied (feasible region), and then perform an elitist approach to find the optimal solutions. For completeness, let us define the *degree of*

constraint violation of each individual as

$$G(\mathbf{x}) = \sum_{j=1}^L \max [0, g_j(\mathbf{x})]. \quad (3.5)$$

A straightforward modification of the EC-algorithms to account for constraints is to introduce penalty terms, which will be added to fitness function values of each individual. Typically, these terms are derived as weighted averages of the degree of constraint violation of each constraint function, i.e. a weighted version of the term in Eq. 3.5. The selection of the weights, however, is left to end-user and may lead to biasing towards some constraint satisfaction on top of the others.

A preference based scheme (feasibility rule) provides unbiased operation and has recently gained attention [49]. It compares pairs of individuals as follows:

1. Feasible candidate solutions are preferred than infeasible ones,
2. Amongst feasible solutions, the ones with better fitness function are preferred and,
3. Amongst infeasible solutions, the ones with the least degree of constraint violation are preferred.

### 3.5 Application in Design Space Exploration

In this section, an application of the popular Multi-Objective variant of GA, named Non-dominated Sorting Genetic Algorithm (NSGA-II) [63] is demonstrated for the design space exploration of two LC-VCO topologies. The use of the automatic, optimization-based approach for the assessment of the attainable performance ranges of the two topologies provides an easy way to compare them quantitatively, without the use of empirical data or closed-form equations.

The motivation for selecting VCO topologies for design space exploration stems from their difficulty to design; in practice, VCO design includes finding a compromise between many competing specifications. Besides Phase Noise and power consumption, metrics such as tuning range and voltage swing must be carefully considered. In addition, as the supply voltages grow smaller, optimal LC-tank design to achieve a high quality factor  $Q_{tank}$  is becoming difficult to accomplish. In particular, the use of integrated coils that provide with relatively low  $Q_{tank}$  brings about the degradation of Phase Noise or increased power consumption [82].

For example, in cases where low power consumption is the most important specification for a particular application, transistors are biased with relatively small currents.

However, this strategy can increase the parasitic elements of MOS devices, thereby degrading the circuit's Phase Noise performance. On the other hand, in cases where low Phase noise is the most important design aspect, one should look to achieve high output voltage swings. This, however, results in higher power consumption. Therefore, finding a balance between the specifications of a LC-VCO topology would require extensive design space exploration, which can be achieved through the proposed black-box optimization approach.

In the following, we discuss briefly the two topologies considered and present the optimization formulation and results.

### 3.5.1 LC-VCO topologies

In this application we address the automatic sizing of two different LC-VCO topologies. These are a nMOS cross-coupled and a complementary nMOS-pMOS cross-coupled LC-VCO and are shown in Figures 3.3 and 3.4 respectively. Both topologies consist of two sub-circuits; an LC tank that determines the oscillation frequency  $\omega_0$  and an active sub-circuit that provides the necessary negative conductivity to compensate for the LC tank losses. In the circuit of Fig. 3.3, the nMOS pair is responsible for the negative conductivity, whereas in the one shown in Fig. 3.4, the complementary nMOS-pMOS pair compensate for the tank losses.

Denoting as  $C_{tank}$  and  $g_{tank}$  the capacitance and conductivity of the VCO tank, respectively, two fundamental equations describe the VCO operation. It is noted that the varactor's capacitance, the parasitic capacitances of the active elements and the capacitive load are included in  $C_{tank}$ . The frequency of oscillation  $\omega_0$  is given by [83]

$$\omega_0 = \frac{1}{\sqrt{L_{tank}C_{tank}}} \quad (3.6)$$

and describes the frequency of the output differential signal at nodes  $V_{outp}$ ,  $V_{outn}$ . To achieve oscillation, it must hold [83]

$$g_{active} \geq a \cdot g_{tank}, \quad (3.7)$$

where  $a \in [1.5, 3]$  is a safety margin factor securing the start-up condition. The term  $g_{active}$  represents the conductivity of the active sub-circuit. In the case of the nMOS-only LC-VCO, it holds  $g_{active} = g_{mn}/2$ , whereas in the case of the complementary one  $g_{active}$  is equal to  $g_{mn}/2 + g_{mp}/2$ . Similarly, the conductance of the LC-tank,  $g_{tank}$ , is computed

using the inductor and varactor conductances and it is given by

$$g_{tank} = \begin{cases} g_L + g_{var} + \frac{g_{ds,n}}{2}, & \text{nMOS} \\ g_L + g_{var} + \frac{g_{ds,p}}{2} + \frac{g_{ds,n}}{2}, & \text{nMOS-pMOS} \end{cases}.$$

An important characteristic for the VCO is its output signal's spectral purity near the frequency  $\omega_0$ . Phase Noise is a measure for this quantity. By defining a frequency shift  $\Delta\omega$  around  $\omega_0$ , one can determine the Phase Noise through [84]

$$\mathcal{L}\{\Delta\omega\} = 10 \log \left[ \frac{1}{4\Delta^2\omega} \cdot \frac{L_{tank}^2(\omega_0)^4}{V_{tank}^2} \cdot 2K_B T \left( g_L + g_{var} + \gamma \cdot g_{d0} \right) \right] \quad (3.8)$$

where  $g_{d0}$  is the transconductance of drain when  $V_{ds} = 0$ ,  $K_B$  is the Boltzmann constant,  $T$  the temperature at Kelvin and  $\gamma$  is the excess noise factor.

Both of the topologies are complemented using a notch filter [85] to enhance their Phase Noise performance. The objective is to reduce the noise components stemming from the tail current, by cutting-off their second harmonic. In both circuits,  $L_{filter}$  and  $C_{filter}$  are connected in parallel at the drain of the current source and they are tuned to resonate at  $2\omega_0$ . In addition, a capacitor  $C_T$  is added in parallel with the current source to short high frequency noise components [85].

This modification was implemented in both circuits with a small difference in nMOS VCO, where top biasing is used to lower the common mode voltage at the output node to  $V_{DD}/2$  and improve the varactor's tuning range. In addition, another pair of  $L_{filter}$ ,  $C_{filter}$  was added to the sources of the nMOS transistors. This is because it provides with high impedance at the source of the cross-coupled transistors and restricts them from loading the tank when in triode region [86].

### 3.5.2 LC-VCO Optimization

In this section we present the optimization procedure for both LC-VCO topologies, with the desired  $\omega_0$  equal to  $2\pi \cdot 5\text{GHz}$ . We consider the general case where the Phase Noise and power consumption of the VCOs are equally important and compare them by comparing their attainable performances, derived using multi-objective optimization. Based on the formulation of Eq. 5.1,  $F$  is a vector of 2 conflicting functions  $F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})]$ , that correspond to the Phase Noise and power dissipation of the circuit.

To account for the output signal oscillation frequency of the VCOs, we use a single constraint function that must fulfill  $g_1(\mathbf{x}) \leq 0$  and corresponds to the deviation of the output signal's frequency from the desired  $\omega_0$ . In particular,  $g_1(\mathbf{x}) = |\omega_0 - \omega_{osc,output}| - \epsilon$ , where  $\epsilon \geq 0$  is a relaxation term that depends on the desired  $\omega_0$  and  $\omega_{osc,output}$  represents

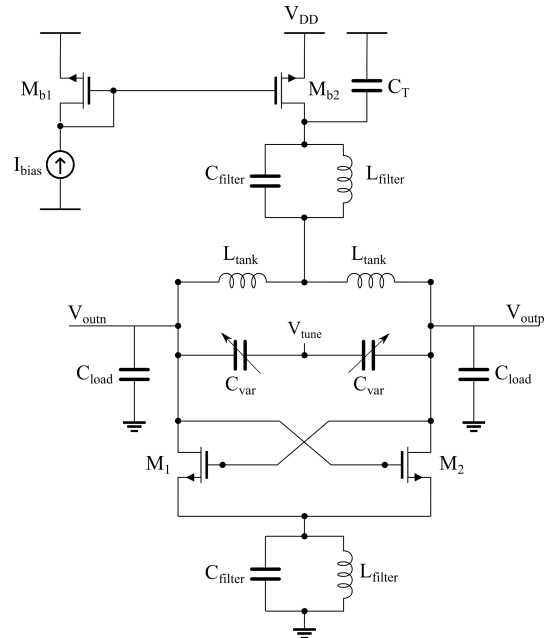


Figure 3.3: nMOS cross-coupled LC-VCO

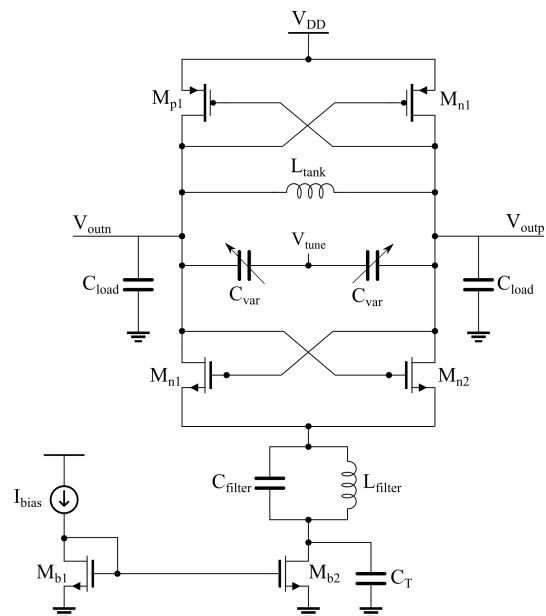


Figure 3.4: Complementary nMOS-pMOS cross-coupled LC-VCO



Table 3.2: VCOs variable ranges

Design Variable	Units	Range
$W_{nmos}$	um	[2, 80]
$L_{nmos}$	nm	[100, 200]
$I_{bias}$	mA	[1, 10]
$W_{tail}$	um	[2, 80]
$L_{tail}$	nm	[100, 200]
$C_T$	pF	[0.5, 6]
$C_{filter}$	pF	[0.15, 1]
$L_{filter}$ Inner Radius	um	[1, 80]
$L_{tank}$ Inner Radius	um	[10, 60]
$L_{filter}$ Num of turns	–	[1, 5]
$L_{tank}$ Num of turns	–	[1, 5]
$C_{var}$ Num of fingers	–	[2, 32]
$C_{var}$ Num of groups	–	[1, 5]

the actual output signal frequency of the parametrized VCO schematic.

All of the aforementioned circuit performance metrics are calculated by an in-house software tool and the commercial simulator Cadence Spectre, which processes the outputs of periodic steady state and DC analyses, in batch mode. This is very important in our case, since the employed NSGA-II is a population-based approach, i.e. it requires many simulations per iteration to converge to optimal solutions.

Both circuit topologies are designed using a general-purpose flavour TSMC 90nm PDK. For each one of them, a single testbench is used to acquire Phase Noise, oscillation frequency and power dissipation outputs. The allowable ranges for the design variables, which are identical for both topologies, are given for reproduction purposes in Table 5.3.

The hyperparameters of the optimization algorithm are given below: The population count is 100 and the algorithm terminates when 150 generations are completed. Since we deal with both integer-valued and continuous variables, the mutation scheme of the algorithm works as follows; a polynomial mutation is used first to change the value of the variables by a non-integer value. Then, the integer valued variables are rounded to the closest integer-value. On an 8 core machine, the optimization took approximately 1.5 hours to complete for each circuit. The resulting PFs are shown in Fig. 3.5.

As expected, higher power consumption provides better Phase Noise performance for both circuits. By comparing the PFs, it is seen that the complementary cross-coupled

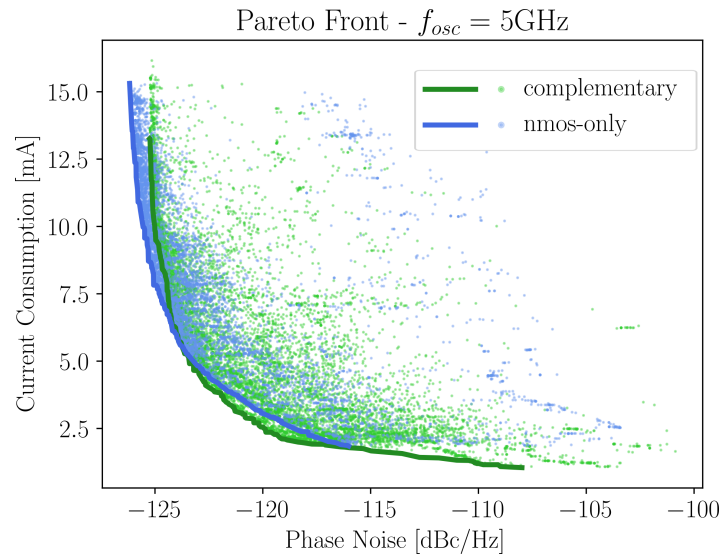


Figure 3.5: Acquired Pareto Fronts for both LC-VCO topologies, with  $f_{osc} = 5\text{GHz}$

topology achieves better Phase Noise and power consumption trade-off for relatively low biasing currents. When higher current consumption is acceptable, the nMOS-only topology is preferable. In fact, nMOS-only topologies provide with larger output voltage swings compared to the complementary ones, leading to better Phase Noise performance, when enough current is available [87]. Using Fig. 3.5, we are able to determine 5mA as the minimum current threshold for using nMOS-only topology.

The above experiment is repeated once more, for the case of  $\omega_0 = 2\pi \cdot 2.4\text{GHz}$ . The NSGA-II hyperparameters are the same as before, and the allowable parameter ranges remain as given in Table 5.3. The optimization took 1.5 hours to complete and the resulting PFs are given in Fig. 3.6.

In this case, the difference between the two topologies is more evident, with the complementary one being even more preferable. Qualitatively, this can be explained as follows; in lower oscillation frequencies, the inductance and capacitance of the tank for a given biasing current need to increase. This requires higher negative impedance from the active sub-circuit to achieve oscillation, which is more easily attainable from the complementary topology. Quantitatively, using the optimization results from Fig. 3.6, the complementary topology is preferred for current consumption below 9mA. It is worth noting that the above results hold for the PDK used in this study, and the actual PFs may differ from one technological node to another.

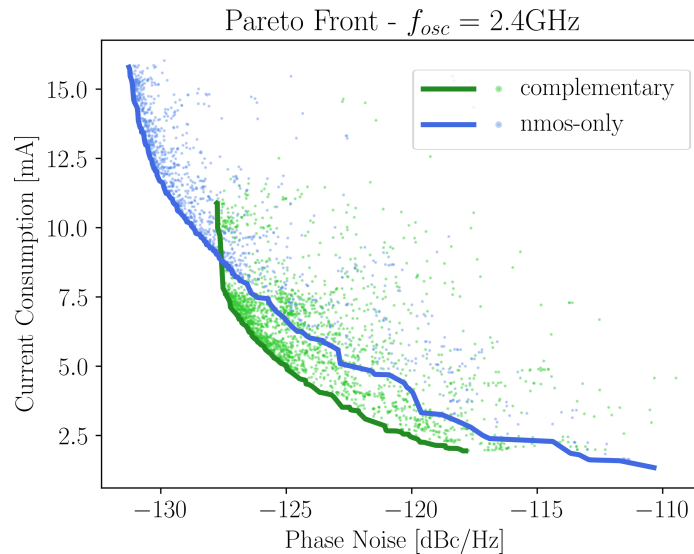


Figure 3.6: Acquired Pareto Fronts for both LC-VCO topologies, with  $f_{osc} = 2.4\text{GHz}$

## 3.6 Proposed Multi-Objective Variant

In this section a modification of the NSGA-II algorithm is proposed to address the mixed parameter space of circuits and to provide a framework that produces competitive results when constraints apply. To address the constraint handling part, we note that the main approach, which is motivated by the feasibility rule [49], does not make use of the infeasible space fitness function information. Here we employ a selection scheme within NSGA-II to mitigate this shortcoming. Our approach can handle integer as well as continuous search spaces, exploit infeasible fitness function information and optimize for multiple objectives. In the following subsections, we discuss the proposed modification in detail and provide example circuit optimization experiments to showcase their merit.

### 3.6.1 Handling Integer Parameters

The trivial approach to addressing mixed parameter spaces in the mutation operations of EC algorithms is to treat discrete parameters as continuous-valued ones. Following mutation, the parameters that need to be discrete are transformed back to the nearest integer value. Though simple, this approach is flawed; in the case that a particular discrete variable is restricted to small ranges at a particular stage of the optimization, it remains unchanged for the remainder of the procedure and therefore hinders exploration[88]. To address this problem, we adopted a quantization scheme: vector  $\mathbf{x}$

is divided into a continuous and an integer part,

$$\begin{aligned}\mathbf{x}_c &= [x_{c0}, x_{c1}, \dots] \\ \mathbf{x}_i &= [x_{i0}, x_{i1}, \dots]\end{aligned}\tag{3.9}$$

where  $x_{cj}$  and  $x_{ij}$  are the  $j$ -th continuous and integer parameters respectively. For the vector containing continuous variables, the bounded polynomial mutation is applied as in [63], while for the integer variables, a Poisson distribution  $P(\lambda)$  is assumed. New values for  $x_{ij}$  are sampled from  $P(\lambda)$ , where  $\lambda$  is equal to  $x_{ij}$ . This process retains the stochasticity of the original mutation operation. For instance, in Fig. 3.7, the probability mass function of the poisson distribution for two different values of  $\lambda$  is plotted, showcasing the possible transitions in the value of a variable under the proposed scheme. While the greater mass of probability is centered at the proximity of  $\lambda$ , the chances of mutating towards a different variable are considerable. This suggests that the problem of stagnation is, in theory, alleviated.

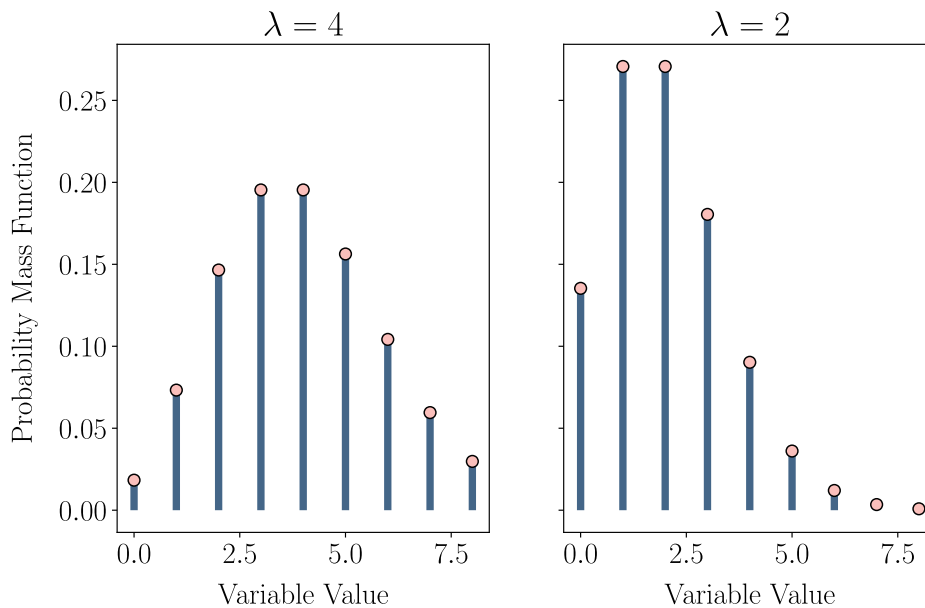


Figure 3.7: The probability mass function of two sample Poisson distributions with  $\lambda = 2$  and  $\lambda = 4$ .

### 3.6.2 Constraint Handling

The feasibility rule scheme is the most popular approach for constraint handling in black-box optimization algorithms and it is incorporated in the EC-algorithm's selection process to drive the population to feasible regions. A limitation of this approach,

however, is that by favouring constraint satisfaction more than fitness function minimization, it leads to convergence to feasible, but not optimum parameter regions[88]. In practice, this problem is exacerbated in situations where the feasibility regions are disjoint, or the optimum solutions lie close to the feasibility boundary.

Since trade-offs are the essence of circuit sizing, optimal solutions may lie close to the infeasible spaces in analog design spaces. Therefore, driven by [89], we introduce a mechanism to the NSGA-II algorithm that executes in parallel with the feasibility rule and makes use of infeasible fitness functions. In particular, at each iteration (generation), the offspring are sorted according to their fitness using the non-dominated ranking algorithm. Those candidate vectors that do not survive to the next generation and are on the first pareto level, are stored in an archive. Then, the individuals in the archive and the surviving population are sorted by their degree of constraint violation. The next step is to select  $k$  vectors with the minimum (maximum) constraint violation in the archive (population), a total of  $2k$  vectors. These are sorted once more using the non-dominating procedure, accounting only for their fitness functions. The  $k$  best individuals are placed on the population and the rest are discarded. As a rule of thumb, we choose  $k$  to be equal to  $1/20$  of the total population.

To showcase the usefulness of the proposed scheme, an example is illustrated in Fig.3.8, where a 2D Rastrigin function is limited to be feasible only in the regions inside the curves, i.e.

$$\begin{aligned} \min f(\mathbf{x}) &= \sum_{i=1}^2 x_i^2 - 10\cos(2\pi x_i) + 10, \\ \text{subject to: } & 3(x_1 + 7)^2 + x_2^2 \leq 0.3 \\ & (x_1 + 8)^2 + (x_2 - 3)^2 \leq 2. \end{aligned} \tag{3.10}$$

In this case, a single-objective GA is used to find the minimum of this constrained problem. Two cases are considered; one using the feasibility rule and another using the proposed selection scheme. The one using the feasibility rule concentrates its search in the large feasible region, whereas the GA incorporating the constraint-handling method described above, for single-objective this time, manages to reach both feasible regions, finding the true global minimum of the problem.

### 3.6.3 Example Applications

In this subsection we compare the proposed MO optimizer in two design-exploration tasks that deal with an amplifier and a Low Noise Amplifier (LNA) circuit. Similar to the previous section, we use an in-house tool for result parsing, simulation and testbench

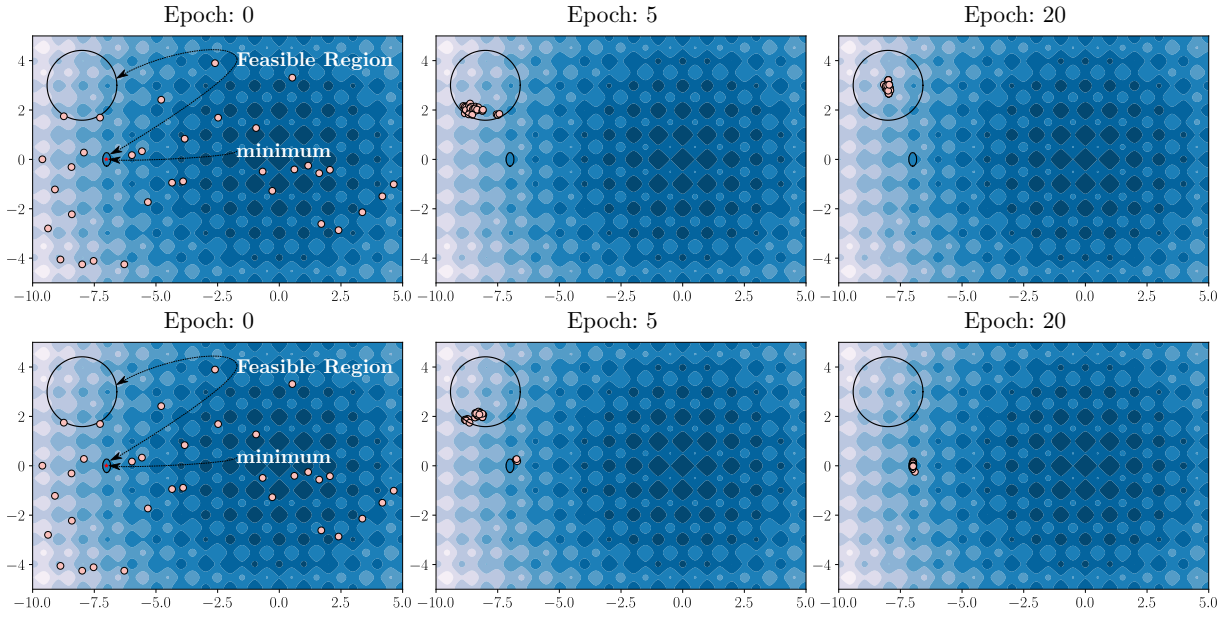


Figure 3.8: Top row: Optimization run for a multimodal constrained function, using the feasibility rule. Bottom row: Optimization run using the constrained handling method proposed. Initial candidate vectors are identical.

parameter update and execute simulations in Cadence Spectre in batch mode.

## Nested-Current-Mirror amplifier

Here we demonstrate our sizing strategy on the single stage Nested-Current-Mirror amplifier (NCM) [90], shown in Fig. 3.9. This topology aims to address display applications, where large capacitive loads need to be handled. The NCM is a single stage amplifier topology that uses the load capacitance for compensation and promises to enhance DC gain, bandwidth and Slew Rate performances compared to standard single stage topologies. The ratios of the current mirrors employed in this topology are key parameters for sizing. These are integer numbers and are addressed accordingly within our proposed approach.

For this experiment, we seek to determine the mirror ratios (K1-K6), the unit transistors dimensions and voltage  $V_b$ . Three types of unit transistors are considered, one for nmos devices, one for pmos devices and one for Mb<sup>1</sup>. A TSMC 90nm process is used to design the amplifier.

For comparison, we follow the original implementation of the topology with 15nF load capacitor,  $V_{DD} = 1.2V$  and set the design constraints equal to the ones stated in [90].

<sup>1</sup>The transistors sizes are defined by parameters K1-K6 and the unit transistors, i.e. M2 width is (K2+K3) times the unit pmos width.

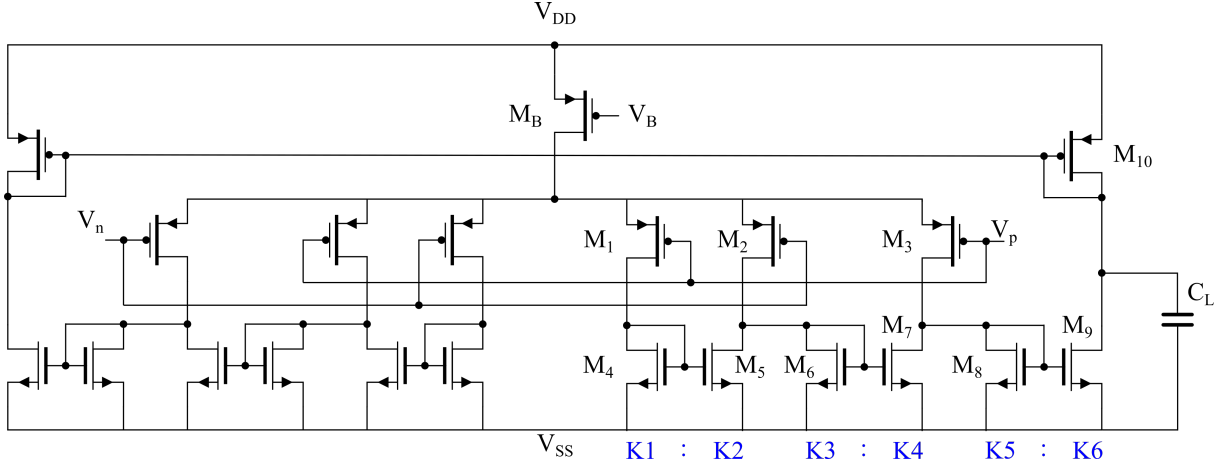


Figure 3.9: Nested Current Mirror amplifier proposed in [90]. Half-circuit instance names are shown, since the circuit is symmetric.

Table 3.3: NCM Specifications

Performance	Description	Specification
PM	Phase Margin	$\geq 90^\circ$
GM	Gain Margin	$\geq 60\text{dB}$
$A_0$	DC Gain	$\geq 72\text{dB}$
$N_{10k}$	Noise @ 10kHz	$\leq 100\text{nV}/\sqrt{\text{Hz}}$
$SR_{avg}$	Slew Rate (average rise and fall)	maximize
Pdc	Total power dissipation	minimize

For trade-off exploration, we optimize for high slew rate and low power consumption. The specifications (objectives and constraints) are given in Table 3.3, while the variable ranges are given in Table 3.4.

Fig. 3.10 shows the pareto fronts resulting from two NSGA-II optimization runs, one using the feasibility rule and the other using the new constraint handling method. Both experiments use the same set of hyperparameters, with population size and maximum generations set to 200 and 300 respectively, and the mixed-integer mutation scheme proposed. The plot suggests that we are able to size the circuit with better slew rate and power trade-off compared to the original implementation. Also, the proposed constraint handling is able to provide slightly better and denser pareto fronts. The patches in the pareto front of the feasibility rule can be explained as follows: Traversing an infeasible part of the design space to reach non-dominated solutions is easier for the proposed algorithm.

The performance of the two methods is assessed quantitatively using the Hyper-

Table 3.4: NCM Variable Ranges

Design Variable	Description	Constraint
$W_i$	MOS Width	[1 – 30] $\mu\text{m}$
$L_i$	MOS Gate Length	[0.2 – 1] $\mu\text{m}$
$K_i$	Mirror Parameters	[1 – 5]
$V_b$	Biasing Voltage	[0.7 – 1.1]V

volume indicator (HV) [91]. This indicator provides a measure of the region which is dominated by each pareto front and bounded by a reference point, therefore higher HV values are better. Using the same reference point for both pareto fronts, the HV value for the proposed method is  $28.5 \cdot 10^4$  whereas for the feasibility rule  $18.34 \cdot 10^4$ .

The optimization took approximately 20 minutes. Table 3.5 provides the sizes for an example solution marked on the pareto front, with 1.2 $\mu\text{W}$  power dissipation and the same slew rate as the original implementation (250 V/s).

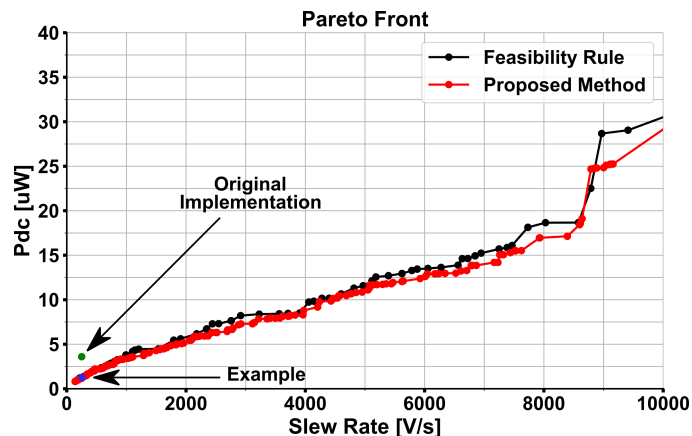


Figure 3.10: Pareto fronts for the NSGA-II algorithm with the proposed constraint handling method, and the typical NSGA-II with feasibility rule.

### 3.6.4 Inductorless Wideband LNA

An inductorless, wideband LNA, shown in Fig. 3.11[92], is sized in this example. This topology adopts active shunt feedback to achieve wideband operation.

We use the same TSMC 90nm process, with 1.2V supply voltage for both the main and the feedback amplifier. The capacitive load is 50fF and the buffer is considered lossless. In the same manner as in [92], transistor lengths are set to the lowest acceptable value by the process, i.e. 100nm. The design specifications are set equal to the ones shown in the original implementation, with the exception of higher bandwidth, and they are



Table 3.5: NCM Example Solution

Variable	Size
$W_{pmos,nmos}$	1u
$L_{pmos,nmos}$	1u
$K1$	2
$K2$	2
$K3$	1
$K4$	5
$K5$	1
$K6$	5
$V_b$	0.98V
$W_{bias}$	6.1u
$L_{bias}$	1u

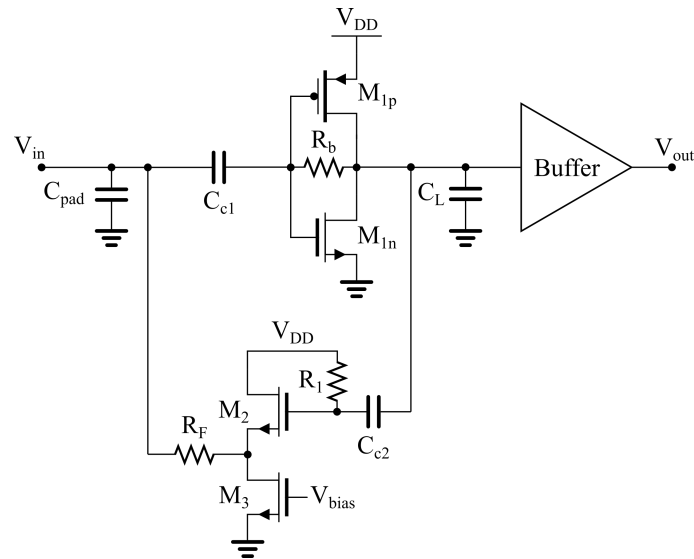


Figure 3.11: Inductorless, wideband low noise amplifier proposed in [92].

Table 3.6: Wideband LNA Specifications

Performance	Description	Specification
$S_{11}$	Input Matching (entire bandwidth)	$\leq -10\text{dB}$
IIP3	Third-order intercept point @ 2GHz	$\geq 8\text{dBm}$
$A_v$	Voltage Gain	$\geq 18\text{dB}$
$BW$	$-3\text{dB}$ Bandwidth	$\geq 3\text{GHz}$
$NF$	Noise Figure @ 2GHz	minimize
Pdc	Total power dissipation	minimize

Table 3.7: Wideband LNA Variable Ranges

Design Variable	Description	Constraint
$W_i$	MOS Width	[1 – 100] $\mu\text{m}$
$R_F$	Feedback Resistor	[1 – 10K] $\Omega$
$R_b$	Self-Biasing Resistor	[1 – 20]K $\Omega$
$R_1$	Biasing Resistor	[1 – 20]K $\Omega$
$C_{ci}$	Coupling capacitors	[0.5 – 10]pF
$V_{BIAS}$	$M_3$ biasing	[0.3 – 0.8]V

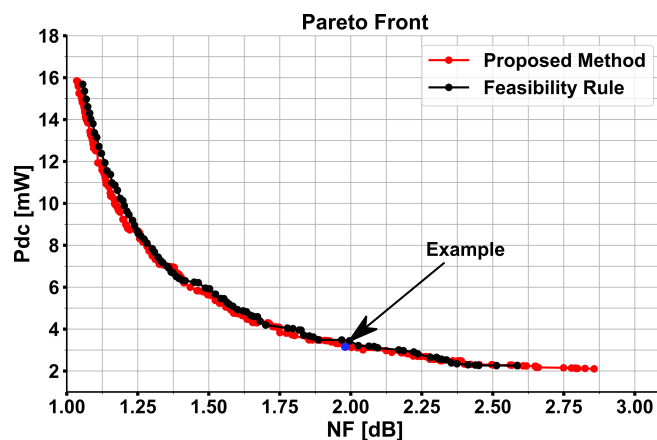


Figure 3.12: Pareto fronts for the wideband LNA experiment.

shown in Table 6.4. In addition, the variables considered in the optimization and their allowable ranges are given in Table 3.7.

The optimization goal is to determine the trade-off between power consumption and Noise Figure. The population count is set to 150 and the maximum generations to 200. The resulting pareto fronts using the NSGA-II with feasibility rule and with the proposed constraint handling method are shown in Fig. 3.12.

The plot suggests that the proposed method finds wider pareto fronts than the feasibility rule. The HV value for the feasibility rule is 26.28 and for the proposed method 100.62, indicating that the proposed method provides more uniform and widespread solutions. Both experiments took approximately 25 minutes.

Repeating the above experiment 5 times, we calculate the Figure-of-Merit (FoM) for wideband LNAs[92]. The mean FoM for the pareto front of the proposed methodology is 41.2dB, while for the one resulting from feasibility rule and NSGA-II is 39.6dB. We note that the achieved FoM is increased with comparison to the original implementation. The sizes of an example design marked on the pareto front are shown in Table 3.8.

Table 3.8: Wideband LNA Example Solution

<b>Variable</b>	<b>Size</b>
$W_{1p}$	$12 \times 1.05\mu$
$W_{1n}$	$90 \times 1\mu$
$W_2$	$10 \times 1.05\mu$
$W_3$	$3 \times 1.2\mu$
$R_1$	$15.6\text{K}\Omega$
$R_b$	$9\text{K}\Omega$
$R_F$	$340\ \Omega$
$C_{c1}$	$10\text{pF}$
$C_{c2}$	$5\text{pF}$
$C_{c3}$	$900\text{fF}$
$V_{BIAS}$	$0.55\text{V}$

### 3.7 Summary & Concluding Remarks

An analysis of the EC-based concepts for black-box optimization, along with the definition of the sizing procedure and the design exploration one as optimization problems, was given in this chapter. The use of EC-based algorithms for exploring the attainable performance of real-world circuits was also demonstrated, by using two LC-VCO as examples. In addition, to enhance the performance of EC-based algorithms for optimization, a variation of the commonly used mutation operation was proposed that uses a Poisson distribution to alter the chromosomes of the algorithm's individuals. This methodology, in combination with an introduced selection mechanism that handles constraints and takes into account the fitness of individuals that are not feasible, was applied on an operational amplifier targeting display applications and an inductorless wideband LNA, demonstrating better trade-off exploration capabilities.



# Chapter 4

## Sample-Efficient Single-Objective Sizing

Automatic sizing of complex circuits with high-dimensional variable spaces and tight constraints requires many evaluations to yield satisfactory results, in the case of EC-based approaches. In reality, circuit simulation may take from some seconds to even hours to complete, which renders the simulation-based optimization using population algorithms impractical. A useful EDA tool should offer designers the opportunity to size and explore the performance space of a circuit in as much less time as possible. Drawing motivation from the above, a sample-efficient black-box optimization approach for SO circuit sizing is proposed in this chapter. It relies on the concept of Bayesian Optimization (BO), and takes advantage of local Gaussian Process (GP) models to enhance both BO's scalability and optimization results. To better highlight the advantages of the proposed method, background information on GPs and BO are also provided.

### 4.1 Background

#### 4.1.1 Gaussian Processes

Within the context of supervised learning, the construction of models that approximate unknown, black-box functions from measurement data holds a special place. The choice of model represents our assumptions about the function to be approximated and the goal is to determine its parameters such that the prediction performance is optimized. Often, given a set of data, the nature of the selected model poses a restriction to its prediction capabilities. For instance, in the case of Neural Networks, the choice of architecture determines partly the resulting performance, even after the optimization of its internal parameters. This family of models, that are adapted to the data by determining a set of predefined in number parameters, are referred to as parametric models. Another family of models that can be used for the same task are the non-parametric ones. In this case, the structure of the function to be modeled is not assumed to be within the capacity of a family of functions with a finite set of parameters. Instead, the non-parametric models assume parameters that are flexible in number and grow

as the amount of input data grows. Intuitively, one may view the infinite-dimensional parameter vector of a non-parametric model as a function.

Gaussian Processes (GPs) [51] belong to the family of non-parametric models. They build a distribution over the function space and perform Bayesian inference to approximate the unknown black-box function. In fact, GPs are a highly studied topic in the literature with applications in diverse fields of study. Their popularity stems from the fact that 1) they rely on very few *hyperparameters* to approximate the given dataset, which are determined in a systematic manner, 2) they yield uncertainty estimates about their predictions and 3) they use closed-form expressions for all operations required to perform inference.

Let us now discuss the GP models in mathematical terms. Consider a dataset  $\mathcal{D}$  comprised of  $n$   $d$ -dimensional parameter vectors  $X = \{\mathbf{x}_i\}_{i=1}^n$  and a corresponding 1-dimensional observation value set  $\mathbf{y} = \{y_i\}_{i=1}^n$ , derived by measuring an unknown black-box function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$ . In the general case, we assume that the observations are corrupted by an uncorrelated additive noise source such that it holds

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad i = 1, \dots, n, \quad (4.1)$$

where  $\epsilon_i$  is the noise component in the  $i$ -th measurement and function  $f$  is the true process that must be modelled.

In the black-box setting, we say that  $f$  in Eq. 4.1 is modeled by a GP  $\tilde{f}$  such that

$$y_i = \tilde{f}(\mathbf{x}_i) + \epsilon_i, \quad i = 1, \dots, n. \quad (4.2)$$

Here, the random noise is considered to follow a zero mean Gaussian distribution such that  $\epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$ , where  $\sigma_n$  is the standard deviation.

The values of  $\tilde{f}$  at any point  $\mathbf{x}^* \in \mathbb{R}^d$  are 1D random variables that follow Gaussian distributions. In practice, the distribution  $\tilde{f}(\mathbf{x}^*)$  serves as an uncertainty estimate for the GP model's prediction. The aforementioned property leads to the conclusion that a GP can be considered as a stochastic process with infinitely many random variables, which together form a probability distribution over the unknown function. For every positive integer  $n$ , any  $(n \times 1)$  vector  $\mathbf{f} = [\tilde{f}(\mathbf{x}_i)]_{i=1}^n$ , which is comprised out of a subset of these random variables, follows a Multivariate Gaussian distribution, i.e.

$$\mathbf{f} = [\tilde{f}(\mathbf{x}_1), \dots, \tilde{f}(\mathbf{x}_n)]^T \sim \mathcal{N}(\boldsymbol{\mu}, K). \quad (4.3)$$

Here, vector  $\boldsymbol{\mu}$  is the GP mean, defined by a mean function  $m : \mathbb{R}^d \rightarrow \mathbb{R}$ , and  $K$  is the covariance matrix, constructed by a kernel function  $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ , such that

$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ . Considering two distinct points  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , the covariance between values  $\tilde{f}(\mathbf{x}_i)$  and  $\tilde{f}(\mathbf{x}_j)$  is given by

$$k(\mathbf{x}_i, \mathbf{x}_j) = \text{Cov}[\tilde{f}(\mathbf{x}_i), \tilde{f}(\mathbf{x}_j)] \quad (4.4)$$

and the mean of the random variable  $\tilde{f}(\mathbf{x}_i)$  is given by

$$m(\mathbf{x}) = \mathbb{E}[\tilde{f}(\mathbf{x}_i)]. \quad (4.5)$$

Functions  $m(\mathbf{x})$  and  $k(\mathbf{x}, \mathbf{x}')$  define uniquely the GP model and constitute design choices. The mean function imposes a bias to the GP's predictions and it can be used to describe our prior beliefs about the unknown black-box function. Often, in cases when no prior information about  $f$  is available, the mean function is set to a constant  $\mu$ . In these cases, the expressive capabilities of the GP model are determined solely by the kernel function  $k$ . The kernel function is a measure of similarity between the GP outputs of any two input points and determines the behavior of the output model, such as for instance periodicity, smoothness, etc.

Kernel functions are constructed such that the resulting covariance matrix  $K$  is positive semi-definite, for any modeling problem. In practice, the most widespread kernel functions are stationary, which means that their values  $k(\mathbf{x}, \mathbf{x}')$  depend only on the difference of the input points,  $\boldsymbol{\tau} = \mathbf{x} - \mathbf{x}'$ . The fact that sample correlations depend on the square of  $\boldsymbol{\tau}$  in many kernel functions enforces closeby points to have similar outputs. While this property may seem intuitive, it also leads to a critical limitation of the GP models; the Euclidean distance between different points in the variable space becomes virtually uninformative in high-dimensional spaces [93], rendering them susceptible to the curse of dimensionality.

Popular kernel functions include the squared exponential and Matérn kernel families [51]. The former is given by

$$k_{se}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left[-\frac{1}{2\lambda^2} \|\mathbf{x} - \mathbf{x}'\|_2^2\right], \quad (4.6)$$

where the hyperparameters  $\sigma^2$  and  $\lambda$  are called the variance and the lengthscale of the process. A property of the squared exponential kernel is that it produces infinitely differentiable functions. In fact, this kernel is the standard for GP regression tasks, mainly due to its simplicity. The the Matérn kernel family, on the other hand, is expressed by

$$k_{matern}(\mathbf{x}, \mathbf{x}') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}r\right)^\nu K_\nu\left(\sqrt{2\nu}r\right), \quad (4.7)$$

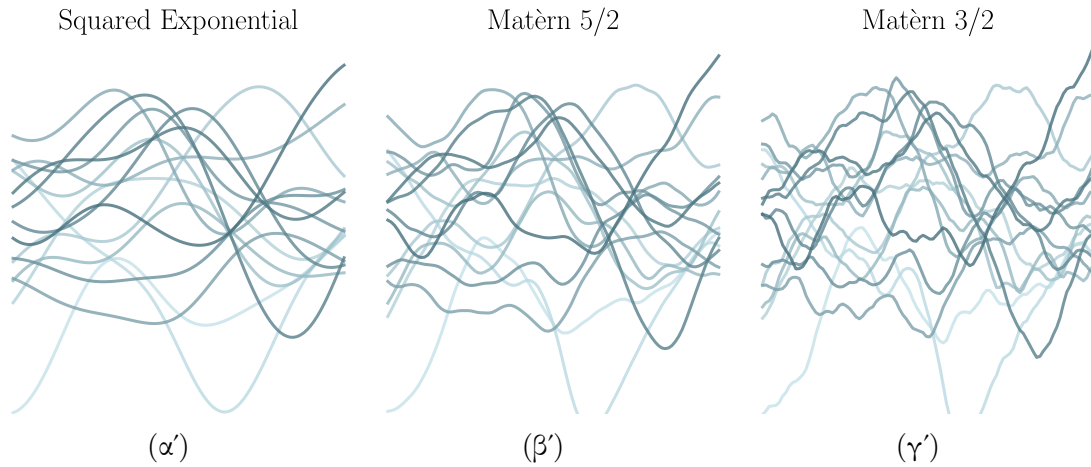


Figure 4.1: Sampled functions from different GP priors.

where  $r$  is given by

$$r = \left( \sum_{k=1}^d \frac{(x_k - x'_k)^2}{\lambda_k^2} \right)^{1/2}. \quad (4.8)$$

Here, the parameters  $\lambda_k$  are the lengthscales of the kernel and correspond to the degree of variation of the GP with respect to the dimensions of the input point, and functions  $K_\nu$  and  $\Gamma$  are the modified Bessel function and the gamma function respectively. The important parameter is  $\nu$ , which takes only positive values. For  $\nu \rightarrow \infty$ , the kernel in Eq. 4.7 is identical to the squared exponential kernel. To relax the smoothness property of the squared exponential kernel, one may choose smaller values for  $\nu$ , with the 5/2 being the most popular. The Matèrn 5/2 provides with twice differentiable functions and its equation, derived from 4.7 is given by

$$k_{\nu=5/2}(\mathbf{x}, \mathbf{x}') = \sigma^2 \left( 1 + \sqrt{5}r + \frac{5}{3}r^2 \right) e^{-\sqrt{5}r}. \quad (4.9)$$

Fig. 4.1 depicts sampled functions from three distinct GP models, each one having zero mean function and Squared Exponential, Matèrn 5/2 and Matèrn 3/2 kernels. It is obvious that the Matèrn kernels provide less smooth functions, and that the  $\nu$  parameter controls the smoothness.

Having established the kernel and mean functions of the GP model, one must adapt the model to the observed dataset  $\mathcal{D}$ . This involves learning the kernel's hyperparameters, i.e. lengthscales, variance, which are grouped into a vector  $\boldsymbol{\theta}$ . To do so, one needs to maximize the marginal likelihood of the observations in  $\mathcal{D}$ , i.e. determine the values of the hyperparameters such that the observations  $\mathbf{y}$  become likely, under the GP.

Arranging the noise-corrupted measurements in vector  $\mathbf{y}$ , and taking into account



the additive Gaussian noise in Eq. 4.2, this vector follows a Gaussian distribution with mean and covariance

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, K + \sigma^2 \mathbb{I}). \quad (4.10)$$

The marginal likelihood is therefore given by

$$p_{\boldsymbol{\theta}}(\mathbf{y}) = (2\pi)^{-n/2} |K + \sigma^2 \mathbb{I}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^T (K + \sigma^2)^{-1} (\mathbf{y} - \boldsymbol{\mu})}. \quad (4.11)$$

The learning process for a GP consists of maximizing Eq. 4.11. Typically, this optimization problem is formulated as a minimization one, where the negative log-marginal likelihood is minimized:

$$L(\boldsymbol{\theta}) = \frac{1}{2} \mathbf{y}^T K^{-1} \mathbf{y} + \frac{1}{2} \log(|K + \sigma^2 \mathbb{I}|) + \frac{n}{2} \log(2\pi). \quad (4.12)$$

This expression is used to learn parameters  $\boldsymbol{\theta}$ , using off-the-shelf optimizers such as the Limited-memory BFGS [94]. More recently, stochastic gradient descent algorithms have been utilized along with backpropagation to learn GP kernel hyperparameters in a more efficient way [95]. To do so, they need to compute the derivatives of the log-marginal likelihood with respect to each hyperparameter  $\theta_i$ , which are given by

$$\frac{dL}{d\theta_i} = \frac{1}{2} \text{Tr} \left( K^{-1} \frac{dK}{d\theta_i} \right) - \frac{1}{2} \mathbf{y}^T K^{-1} \frac{dK}{d\theta_i} K^{-1} \mathbf{y}. \quad (4.13)$$

After adapting the GP model to the data, one can make predictions about a point  $\mathbf{x}^*$  that does not belong in  $\mathcal{D}$ . This is achieved by utilizing the GP's predictive distribution. In order to define this distribution, let us first consider the joint distribution of  $\tilde{f}(\mathbf{x}^*)$  and  $\mathbf{y}$ , which is a Gaussian one

$$\begin{bmatrix} \mathbf{y} \\ \tilde{f}(\mathbf{x}^*) \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix}, \begin{bmatrix} K + \sigma^2 \mathbb{I} & \mathbf{k} \\ \mathbf{k}^T & k(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix} \right), \quad (4.14)$$

where  $\mathbf{k}^T$  is a  $(1 \times n)$  vector with values  $k(\mathbf{x}_i, \mathbf{x}^*)$  for  $i = 1, \dots, n$  and  $K$  is the  $n \times n$  kernel matrix. By using the conditional properties of the Gaussian distribution [96], one can obtain the predictive distribution  $p(\tilde{f}(\mathbf{x}^*))$  of the GP model as a Gaussian one with mean and variance given by

$$\begin{aligned} \mu_{\tilde{f}|\mathbf{y}}(\mathbf{x}^*) &= \boldsymbol{\mu} + \mathbf{k}^T (K + \sigma^2 \mathbb{I})^{-1} (\mathbf{y} - \boldsymbol{\mu}) \\ \sigma_{\tilde{f}|\mathbf{y}}^2(\mathbf{x}^*) &= k(\mathbf{x}, \mathbf{x}) - \mathbf{k}^T (K + \sigma^2 \mathbb{I})^{-1} \mathbf{k} \end{aligned} \quad (4.15)$$

The predictive distribution expresses the GP model's predictions after having been

adapted to the dataset  $\mathcal{D}$ . By using the Gaussian distribution defined in Eq. 4.15, one can produce 1D distributions which serve as predictions, for pointwise inputs  $\mathbf{x}$ . Fig. 4.2 demonstrates an example GP regression on a sinewave function, where the GP's predictive distribution is utilized to demonstrate the pointwise 1D Gaussian distributions in the range  $[0, 1.6]$ . The mean of each pointwise distribution is shown in dark blue, the confidence bounds  $\mu \pm \sigma$ ,  $\mu \pm 2\sigma$ ,  $\mu \pm 3\sigma$  in light blue and the actual sinewave in red.

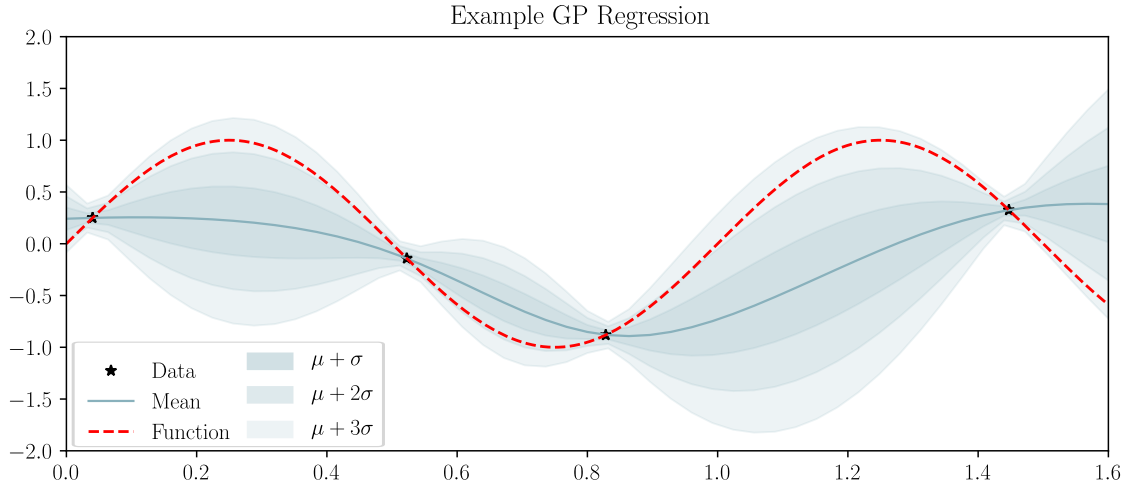


Figure 4.2: A depiction of GP regression to a set of sinewave measurements. The measurements are shown as black stars, the actual sinewave is shown in red and the GP mean and confidence bounds are shown in blue.

It is important to note that, although the predictive distribution in Eq. 4.15 is given for the pointwise input  $\mathbf{x}^*$ , one can define and use a multivariate predictive distribution as well. Considering a set of  $k > 1$  unseen points  $\mathbf{x}_1, \dots, \mathbf{x}_k$ , the predictive distribution  $p\left(\left[\tilde{f}(\mathbf{x}_1), \dots, \tilde{f}(\mathbf{x}_k)\right]\right)$  has a covariance matrix

$$\text{Cov}(\mathbf{x}^*, \mathbf{x}^{*'}) = k(\mathbf{x}^*, \mathbf{x}^{*'}) - \mathbf{k}_{\mathbf{X}, \mathbf{x}^*}^T (K + \sigma^2 \mathbb{I})^{-1} \mathbf{k}_{\mathbf{X}, \mathbf{x}^{*'}}, \quad (4.16)$$

where the  $(1 \times n)$  vector  $\mathbf{k}_{\mathbf{X}, \mathbf{x}^*}^T = [k(\mathbf{x}_i, \mathbf{x}^*)]_{i=1}^n$ , while the mean vector remains the same as in Eq. 4.15. Sampling from this joint predictive distribution results in a  $k$ -dimensional output vector  $\tilde{\mathbf{f}}$  and it is typically done by sampling a vector  $\zeta$  from a  $k$ -dimensional unit-variance, isotropic and zero mean Gaussian distribution and then computing

$$\tilde{\mathbf{f}} = \boldsymbol{\mu}_{\tilde{f}|\mathbf{y}} + A\boldsymbol{\zeta}, \quad (4.17)$$

where  $\boldsymbol{\mu}_{\tilde{f}|\mathbf{y}}$  is the mean vector of the joint predictive distribution and  $A$  the matrix square root of its covariance matrix  $\tilde{K} = [\text{Cov}(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^k$ , such that  $\tilde{K} = AA^T$ .

Despite the fact that GPs offer good approximation capabilities and yield uncertainty

estimates about their predictions, an important limitation is their computational performance. In practice, the derivation of the predictive distribution in Eq. 4.15 requires the inversion of the  $n \times n$  matrix  $K + \sigma^2 \mathbb{I}$ . This is done using Cholesky decomposition, which requires  $\mathcal{O}(n^3)$  time. Typically, the inverse of this matrix is stored in the memory and the prediction requires matrix-vector products, which require  $\mathcal{O}(n^2)$  time each. At test time, the predictive mean and variance require  $\mathcal{O}(n)$  and  $\mathcal{O}(n^2)$  time respectively, which in the case of multiple ( $k > 1$ ) points becomes  $\mathcal{O}(k \cdot n^2)$ . The sampling procedure in Eq. 4.17 also requires a Cholesky decomposition of the covariance matrix of the joint predictive distribution, which has  $\mathcal{O}(k^3)$  time complexity.

### 4.1.2 Bayesian Optimization

Bayesian Optimization (BO) [64] is a sample efficient method to solve global optimization problems, particularly aiming expensive-to-evaluate cost functions. In the unconstrained regime, a real valued, unknown function  $f$  is provided, and BO learns a fast to evaluate surrogate model from past evaluations. It selects next query points for evaluation sequentially, by balancing exploration and exploitation to find the global optimum.

---

#### Algorithm 1: BO Algorithm

---

**Input** : Initial samples  $N_{init}$ , number of iterations  $T_{max}$ , variable space  $\mathbb{S}$

**Output**: Global minimum  $\mathbf{x}_{best}$

Create randomly a set  $\mathbf{X}$  of  $N_{init}$  initial samples from  $\mathbb{S}$

Evaluate  $\mathbf{X}$  to acquire observations  $\mathbf{y}$

**for**  $i = 1, \dots, T_{max}$  **do**

Adjust GP models using Eq. 4.12

$\mathbf{x}^* \leftarrow \operatorname{argmax}_{\mathbf{x} \in \mathbb{S}} \alpha(\mathbf{x}, \mathbf{y})$

Evaluate  $\mathbf{x}^*$  to acquire  $y^*$

Update archive  $\mathbf{X} \leftarrow \mathbf{X} \cup \{\mathbf{x}^*\}$ ,  $\mathbf{y} \leftarrow \mathbf{y} \cup \{y^*\}$

Find  $\mathbf{x}_{best}$  from  $\mathbf{X}, \mathbf{y}$

---

The BO framework consists of two main components; the probabilistic surrogate model that aims to approximate  $f$  and an *acquisition function*  $a : \mathbb{S} \rightarrow \mathbb{R}$  that provides a score of utility for evaluating a candidate query point, based on the probabilistic model. In most approaches, the surrogate model is a GP, which is trained and used for predictive point evaluations as discussed in the previous subsection.

BO works in iterations and its pseudocode is given in Algorithm 1. Starting from an initial set of evaluations, BO incrementally builds a GP model based on historic data, and selects a new query point, as the one that optimizes the acquisition function. This is an auxiliary optimization problem, but since the acquisition function is fast to evaluate,

off-the-shelf optimization methods such as CMA-ES [64] can be used. After the selection of the query point  $\mathbf{x}^*$ , the black-box function is evaluated to acquire the corresponding measurement value  $y^*$ . The data pair  $\{\mathbf{x}^*, y^*\}$  is appended to the historic data and BO proceeds with training the models once again.

Acquisition functions take as inputs the trained GP model and produce estimates about the goodness of each point in the search space. To do so, they rely on the predictive distributions of Eq. 4.15 and Eq. 4.16 and implement a procedure to translate these distributions to a scalar-valued measure. Typically, they are designed such that their optimization yields the best candidate point for evaluation, at a particular BO iteration. Since in the context of black-box optimization there is no universally accepted ‘best’ way to explore the search space, there exist several acquisition functions and their choice is left to the engineer. Despite their different approaches, they all try to balance the exploration (i.e. evaluation of points in regions with large variance) and the exploitation (i.e. the evaluation of points closeby to the current best) trade-off. In the case of GPs, the acquisition functions are formulated in terms of the predictive distribution, requiring no further evaluations of the black-box function to be optimized. This renders their evaluation, and consequently their optimization, fast, in comparison the unknown black-box function. In the following we discuss the most popular acquisition functions for unconstrained BO.

- **Lower Confidence Bound (LCB)**: The LCB acquisition function defines the goodness of each point in the search space as a lower bound provided by the pointwise predictive distribution of the GP model. It favours points that have low predictive means and high predictive variances. Mathematically, it is formulated as:

$$\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x} \in \mathbf{S}} \left\{ \mu_{\tilde{f}|\mathbf{y}}(\mathbf{x}) - \beta^{1/2} \cdot \sigma_{\tilde{f}|\mathbf{y}}(\mathbf{x}) \right\}, \quad (4.18)$$

where  $\beta$  is a (positive) hyperparameter and functions  $\mu_{\tilde{f}|\mathbf{y}}(\mathbf{x})$ ,  $\sigma_{\tilde{f}|\mathbf{y}}(\mathbf{x})$  are the predictive means and variances of the GP model, as given in Eq. 4.15.

- **Probability of Improvement (PI)**: PI defines the goodness of each point as the probability of that point delivering better objective function value compared to a given threshold  $\xi$ , which is typically the current best solution. In the case of GP models, this probability can be expressed analytically and the resulting query point is given by

$$\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x} \in \mathbf{S}} p\left(\tilde{f}(\mathbf{x}) \geq \xi\right) = \operatorname{argmax}_{\mathbf{x} \in \mathbf{S}} \Phi\left(\frac{\mu_{\tilde{f}|\mathbf{y}}(\mathbf{x}) - \xi}{\sigma_{\tilde{f}|\mathbf{y}}(\mathbf{x})}\right), \quad (4.19)$$

where  $\Phi(\mathbf{x})$  is the Gaussian Cumulative Distribution Function (CDF).

- **Expected Improvement (EI):** This acquisition function measures the expected improvement upon a predefined threshold  $\xi$ , which each point in the search space may provide. To do so, it relies on the improvement function

$$I(\mathbf{x}) = \max \left\{ 0, \tilde{f}(\mathbf{x}) - \xi \right\}. \quad (4.20)$$

In the case of GPs, the resulting query point from the EI acquisition function is the one that maximizes the expectation of  $I(\mathbf{x})$ :

$$\begin{aligned} \mathbf{x}^* &= \operatorname{argmax}_{\mathbf{x} \in \mathbb{S}} \{ \mathbb{E} [I(\mathbf{x})] \} \\ &= \operatorname{argmax}_{\mathbf{x} \in \mathbb{S}} \left\{ z(\mathbf{x}) \cdot \Phi(z(\mathbf{x})) + \sigma_{\tilde{f}|\mathbf{y}}(\mathbf{x}) \cdot \phi(z(\mathbf{x})) \right\}, \end{aligned} \quad (4.21)$$

where  $\Phi(\cdot)$  is the Gaussian CDF,  $\phi(\cdot)$  is the Gaussian PDF and  $z(\mathbf{x})$  is equal to

$$z(\mathbf{x}) = \frac{\mu_{\tilde{f}|\mathbf{y}}(\mathbf{x}) - \xi}{\sigma_{\tilde{f}|\mathbf{y}}(\mathbf{x})}.$$

- **Predictive Entropy Search (PES):** The PES acquisition function seeks to maximize the information gain with respect to the global optimizer of the black-box function,  $\mathbf{x}_*$ . Considering  $\mathbf{x}_*$  as a random variable, PES approximately minimizes its entropy after a new observation  $\{\mathbf{x}, y\}$ . The resulting query point must satisfy

$$\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x} \in \mathbb{S}} \left\{ \mathbb{H} [p(\mathbf{x}_*|\mathbf{y})] - \mathbb{E}_{p(y|\mathbf{x})} [\mathbb{H} [p(\mathbf{x}_*|\mathbf{y} \cup y)]] \right\}. \quad (4.22)$$

Function  $\mathbb{H}[p(\mathbf{x})] = - \int p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x}$  is the differential entropy. There is no analytical expression for this acquisition function. To use it, [97] rewrites it using the Mutual Information and generates samples for  $p(\mathbf{x}_*|\mathbf{y})$  by sampling from the joint predictive distribution of the GP models over a finite set of points and computing their maximizer, in a monte-carlo fashion.

- **Thompson Sampling (TS):** TS is a randomized selection strategy addressing the exploration-exploitation tradeoff by drawing random samples from the posterior distribution of the GP models and selecting a query vector by optimizing on these samples. TS draws samples from the joint predictive distribution of the GP models over a large number of input points, to provide with a vector of values from each sample function. Intuitively, TS draws a function from the GP model and optimizes it to provide a single query point. However, there is no way to yield

exact, analytical expressions for the samples of GP models. Therefore, TS usually samples the values of the functions at a predefined, finite-length vector of input points, and determines the query point by finding the best value among them. This operation can be expressed as

$$\begin{aligned} \mathbf{x}^* &= \operatorname{argmax}_{\mathbf{x} \in \mathbf{S}} \hat{f}(\mathbf{x}) \\ &\text{where } \hat{f} \sim \tilde{f}. \end{aligned} \tag{4.23}$$

An example optimization of an 1D Ackley test function using BO and the LCB acquisition function is given in 4.3. Starting from an initial sampling of  $n = 3$  points, which are shown as black stars, the optimization of the acquisition function yields points that minimize the lower confidence bound. At each iteration, the location of the query point is marked with a pink vertical line. The true function is shown in black. After having evaluated  $n = 12$  points in total, BO has approximated the optimum very well.

### 4.1.3 BO & Constraint Handling

Thus far we have discussed the concept of BO and the acquisition functions from the scope of SO unconstrained problems. Although vanilla BO targets such problems, there are cases where black-box constraints exist within the formulation of the optimization problem, such as in the case of analog circuit sizing, and there exist BO variants that address them. In this subsection we discuss methodologies for constrained SO BO.

- **Penalty methods:** Penalty methods constitute the simplest approach to handle constrained optimization problems. Typically, a constraint violation measure is computed for each constraint function and a weighted average is added to the objective function values. Therefore, the exact same unconstrained BO algorithm can be used, and the constraint satisfaction is accounted implicitly by favouring objective function values that are less than others. In some cases, the objective function values for unconstrained query points are set to infinity, so as to diminish the chances of selecting unconstrained solutions in the future.

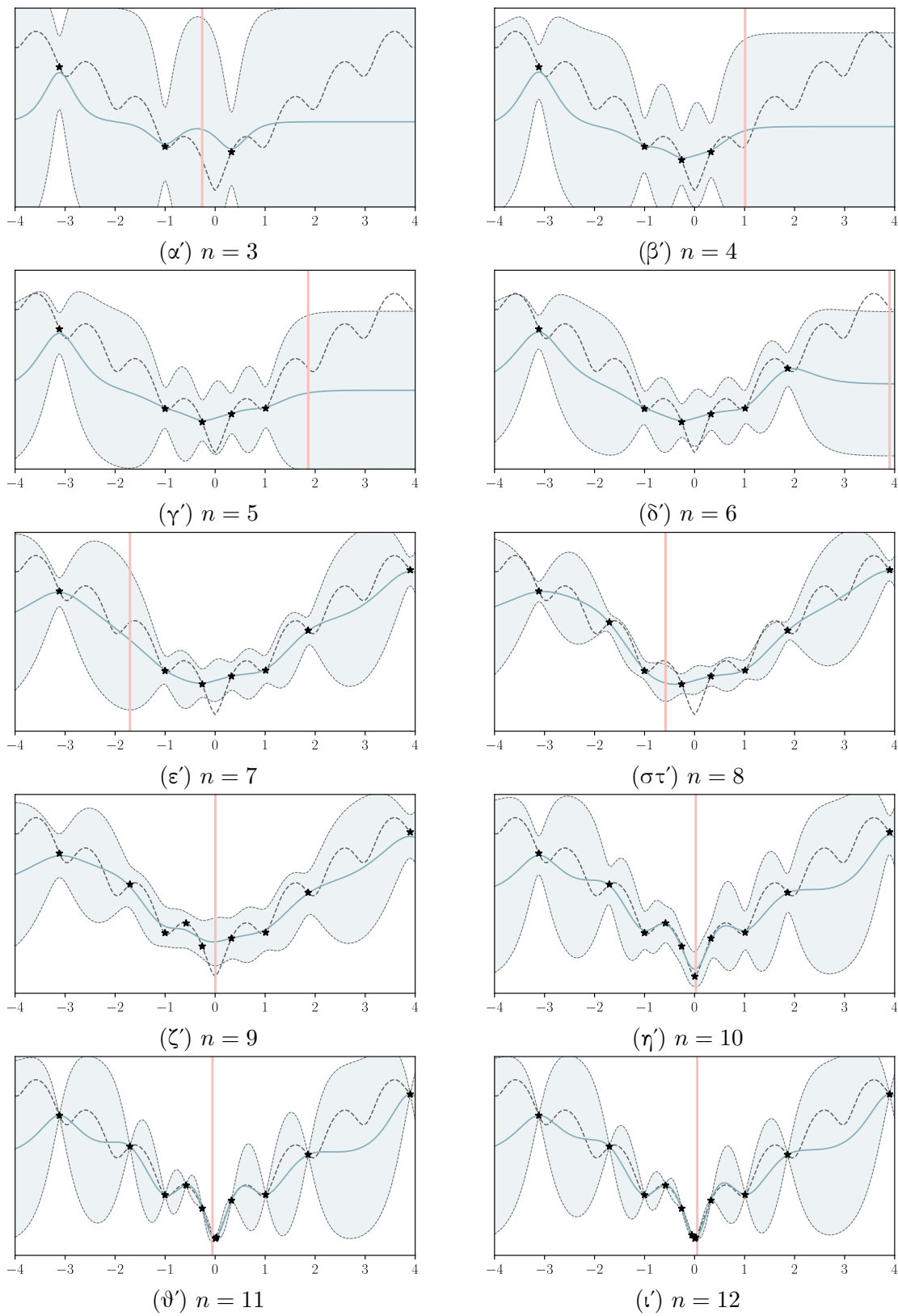


Figure 4.3: Example BO execution using the LCB acquisition function on a 1D Ackley function.

- **Expected Improvement with Constraints (EIC):** In this case, a modification of the EI acquisition function is used. The same concept with the improvement over a predefined value  $\xi$  applies, but this time it is weighted by the probability of feasibility of each point. To estimate this probability, one must use separate GP models that are trained on the constraint function data. If we denote as  $[\tilde{c}_i(\mathbf{x})]_{i=1}^l$  the GP models that approximate each constraint function, then the query point at each iteration is given by

$$\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x} \in \mathcal{S}} \left\{ \left( z(\mathbf{x}) \cdot \Phi(z(\mathbf{x})) + \sigma_{\tilde{f}|\mathbf{y}}(\mathbf{x}) \cdot \phi(z(\mathbf{x})) \right) \cdot \prod_{i=1}^l p(\tilde{c}_i(\mathbf{x}) \geq 0) \right\}. \quad (4.24)$$

A similar approach would be to use a single GP model to approach the whole constraint violation of each point, i.e. the sum of individual constraint violations.

- **Predictive Entropy Search with Constraints (PESC):** Similar to the PES acquisition function, PESC uses the notion of entropy and aims to maximize the information gain about the location of the global maximizer,  $\mathbf{x}_*$ . To account for constraints, the expectation function in Eq. 4.22 is taken over the whole vector  $\hat{\mathbf{y}}$ , which results by concatenating the values of all functions (constraints and objective). Again, PESC does not come in close form, and its maximization requires a series of computationally unstable operations [98].

## 4.2 Proposed Approach

In this section, the proposed approach for local BO based analog circuit sizing is presented, along with implementation details for computational efficiency.

### 4.2.1 Local Bayesian Optimization

Local based approaches for global optimization are extensively studied in the context of EAs. A surrogate within a restricted (trust) region is trained and used to suggest query points. A similar approach is proposed in [78], where a sequential model building optimization algorithm uses GP models inside a trust region to model the objective function, and defines an acquisition function to select future query points. This BO variant alleviates the problem of heterogeneity of objective functions, since query points are selected based on only the local dynamics of the problem at hand.

To achieve global optimization using local-based GP models, however, multiple trust regions are employed in parallel. In the case of  $k$  trust regions,  $k \times (1 + l)$  GP models



(a single objective and  $l$  constraints) must be maintained and trained throughout the procedure. Starting from an initial sampling of the design space using Latin Hypercube Sampling, each trust region is assigned a number of observations to build and initial set of GP models. While they are allowed to overlap, they maintain separate historic data archives used for training their respective GP models.

Each trust region is a hyper-rectangle and its center is chosen to be the maximum utility point in their respective historic data archive, meaning either minimum objective function value or constraint violation, if no feasible solution is yet to be found. The centre, as well as the length of each trust region are updated in each iteration. In particular, the length of each hyper-rectangle is denoted by  $\mathcal{L}$  and is updated according to the trust region progress. To quantify this progress, we borrow an approach from EAs [99]. Given  $p_i$  as the current centre of the  $i$ -th hyper-rectangle and  $q_{i,\text{best}}$  as the best query point probed in it in a particular iteration, quantities

$$I_{\text{opt}} = \frac{f(p_i) - f(q_{i,\text{best}})}{\hat{f}(p_i) - \hat{f}(q_{i,\text{best}})}, I_{\text{inf}} = \frac{CV(p_i) - CV(q_{i,\text{best}})}{\hat{CV}(p_i) - \hat{CV}(q_{i,\text{best}})} \quad (4.25)$$

define the optimization progress, where  $(\hat{\phantom{x}})$  denotes values predicted using GP models and  $CV$  is the constraint violation function. Parameters,  $c_1$ ,  $c_2$  and the boundary values for  $\mathcal{L}$ ,  $\mathcal{L}_{\text{max}}$  and  $\mathcal{L}_{\text{min}}$  control its adaptation, with  $0 < c_1 < 1 < c_2$ . The indicator for the  $i$ -th trust region is given by

$$\mathcal{I}_i = \begin{cases} I_{\text{opt}} & \text{for } p_i, q_{i,\text{best}} \text{ feasible} \\ I_{\text{inf}} & \text{for } p_i, q_{i,\text{best}} \text{ infeasible} \\ c_1 & \text{for } p_i \text{ feasible, } q_{i,\text{best}} \text{ infeasible} \\ c_2 & \text{for } p_i \text{ infeasible, } q_{i,\text{best}} \text{ feasible} \end{cases}. \quad (4.26)$$

At the end of each iteration,  $\mathcal{L}_i$  is updated by following rule:

$$\mathcal{L}_i = \begin{cases} \max(c_1 \times \mathcal{L}_i, \mathcal{L}_{\text{min}}) & \text{for } \mathcal{I}_i \leq c_1 \\ \min(c_2 \times \mathcal{L}_i, \mathcal{L}_{\text{max}}) & \text{for } \mathcal{I}_i \geq c_2 \\ \mathcal{L}_i & \text{otherwise} \end{cases}.$$

The algorithm operates in a transformed variable space, where the upper and lower bounds for each variable lie within the unit cube, therefore it holds  $\mathcal{L}_{\text{max}} < 2$  and  $\mathcal{L}_{\text{min}} > 0$ .

To select the next query points for evaluation, a modified Thomson Sampling [64] acquisition function is employed. Consider the case when a single trust region is used.

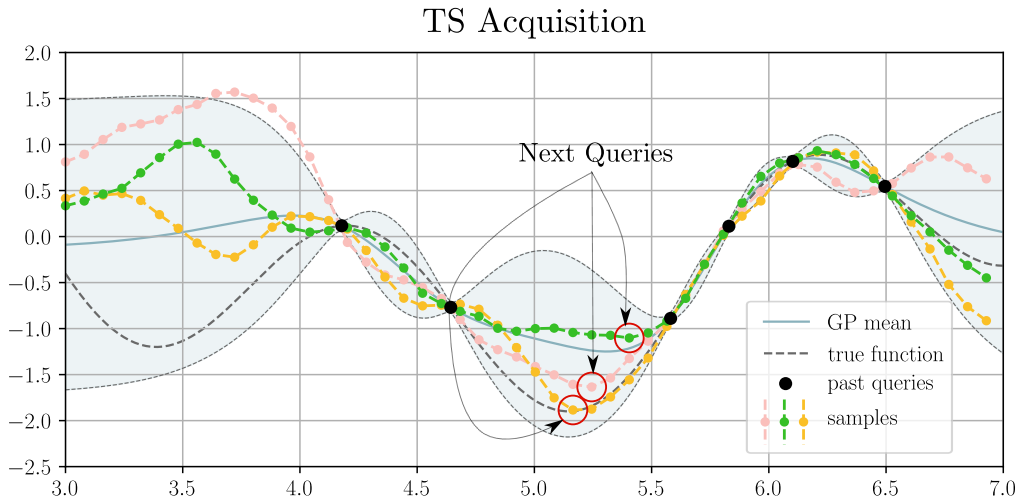


Figure 4.4: A GP’s pointwise predictions on a test function (mean and 95% confidence bounds shown), along with 3 samples from a joint predictive distribution. Past queries are shown as black dots, while next queries are marked in red circles.

Thomson sampling uses a Sobol sequence [78] to select  $r$  candidate query points  $\{\mathbf{x}_i\}_{i=1}^r$  residing in the trust region. For each one of the  $l + 1$  GP models employed, a sample is taken from their respective joint posterior distributions. This results in vectors  $[\hat{f}(\mathbf{x}_i), \hat{g}_1(\mathbf{x}_i), \dots, \hat{g}_l(\mathbf{x}_i)]$  used to compute  $[\hat{f}(\mathbf{x}_i), \hat{C}V(\mathbf{x}_i)]$  for every  $\mathbf{x}_i$  produced by the Sobol sequence. The  $\mathbf{x}_i$  of maximum utility is chosen using the feasibility rule.

In the case of more than one trust regions, the sampling procedure is repeated for each one of them, and the maximum utility point is selected from the pool of all candidate query points and all trust regions.

The above procedure extends naturally to batched query point selection; from the joint GP posterior on all candidate points, one can have multiple samples. Sequentially, the aforementioned selection scheme picks the maximum utility point, corresponding to a single posterior sample, and makes sure not to pick the same candidate point again. A demonstration (no constraints apply) is given in Fig. 4.4, where 3 samples are drawn from the GP posterior, and the next batch of query points are given as the minimum of each sample.

## 4.2.2 Scaling Up Gaussian Processes

In practice, GP training and prediction become intractable for large data sets [100]. This has motivated the research for approximate GP inference. Perhaps the most widespread approach includes the *inducing point* methods [51, 100, 101], which employ a set of  $m \ll n$  inputs  $Z = \{\mathbf{z}_1, \dots, \mathbf{z}_m\}$  to form a Nystrom approximation [102] of the covari-

ance matrix,

$$K \approx K_{Nystrom} = K_{xz}K_{zz}^{-1}K_{zx}, \quad (4.27)$$

where  $K_{xz} = K_{zx}^T$  is a  $(n \times m)$  covariance matrix between training and inducing points, evaluated using the exact kernel  $k$ . By adding a diagonal correction term,

$$K \approx K_{Nystrom} + \Lambda,$$

where  $\Lambda_{(i,i)} = k(\mathbf{x}_i, \mathbf{x}_i) - k_{zx_i}K_{zz}^{-1}k_{zx_i}$ , the approximate covariance matrix is ensured to be full rank [103]. Using this approximation, GP inference time complexity is reduced to  $\mathcal{O}(nm^2)$ . A comprehensive review of inducing point methods is provided in [100].

The selection of inducing points determines the predictive performance of the approximate GP. A naive approach would be to select a subset of the training samples to construct the covariance matrix either in random, or by means of an expensive combinatorial optimization [51]. A more elegant approach regards the inducing point locations as additional  $(m \times d)$  GP hyperparameters, and provides solutions during the gradient based likelihood optimization of Eq. (4.12). We use the latter approach and modify it to fit the local-based BO scheme better.

In the context of local-based BO, trust region sizes and locations alter during the optimization, resulting in GP models trained with archived samples that do not reside in the current trust region. The reader is reminded that the acquisition function is restricted to select query points strictly within the trust regions. Thus, one should select inducing point locations that provide higher predictive accuracies within the trust regions. It has been noted that GPs with sparse kernels provide good accuracy in regions where the inducing points are densely concentrated [101]. We therefore restrict the inducing points to lie within each trust region.

Using the Adam [95] optimizer for gradient based optimization, we empirically found that the optimized locations of inducing points depend on two factors; the initial locations used and the optimizer’s learning rate. We therefore impose the starting locations to be within or in proximity of the trust regions, and select their learning rate to be an order of magnitude less than that of the kernel function hyperparameters. Algorithm 2 explains the initial location selection for the inducing points.

While approximate kernels provide a remedy to the sample budget bottleneck of GPs, further performance gains can be achieved in terms of matrix computations. Inference and training for GPs require the evaluation of terms  $K^{-1}\mathbf{y}$ ,  $\log|K|$  and  $\text{Tr}\left(K^{-1}\frac{dK}{d\theta_i}\right)$  in expressions (4.15), (4.12) and (4.13). The straightforward approach for this is Cholesky decomposition, which scales cubically with sample size [51]. The Blackbox Matrix-

**Algorithm 2:** Inducing Point Initial Location Selection

---

**Input:**  $\mathcal{L}_i$  (trust region length),  $X$  (trust region archive),  $p_i$  (trust region center),  $m$  (inducing point count)  
**Output:**  $loc$  (inducing point locations)  
 $A \leftarrow X \in \text{Hyper-rectangle}(\mathcal{L}_i, p_i)$  // Subset of training vectors lying inside the TR hyper-rectangle;  
**if**  $len(A) > m$  **then**  
  |  $loc \leftarrow \text{KMeans}(A, m \text{ clusters})$  //  $m$  centroids;  
**else**  
  |  $loc \leftarrow \arg \min_{i=1, \dots, m} \|p_i - X\|$  //  $m$  training vectors closest to  $p_i$ ;  
**end**  
**Return**  $loc$

---

Table 4.1: GP Training and sampling runtimes for Rosenbrock Function

Operation	Exact GP		SGP (m=200)		SGP (m=100)	
	CPU	GPU	CPU	GPU	CPU	GPU
Training (s)	18.54	5.26	10.53	5.08	7.9	4.79
Sampling (s)	27.29	0.33	23.06	0.257	17.07	0.23
Sampling (LOVE)(s)	1.59	0.17	0.41	0.153	0.34	0.158

Matrix multiplication method [95] is a recently proposed alternative that reduces the asymptotic complexity of GP inference to  $\mathcal{O}(n^2)$ . This is a modified conjugate gradient algorithm that allows for GPU acceleration. The software package `GPYtorch` implements this method and it is used in our implementation. Moreover, to scale GP sampling, which is required in the acquisition function of the proposed BO, we use the LanczOs Variance Estimates (LOVE) [104] method.

The gains of the aforementioned practices are demonstrated empirically using a toy experiment; one exact and 2 sparse GP models approximate a 20D Rosenbrock function. For a training dataset of 3000 samples, and 5000 candidate points to jointly sample from, the times required for training and sampling are given in Table 4.1. For all GP models, 100 gradient descent iterations were used. Both LOVE and GPU acceleration provide a combined speedup of  $\times 8$  for exact GPs and  $\times 6$  for approximate ones.

### 4.3 Circuit Design Applications

To test the performance of the proposed approach on large-scale problems, we perform experiments on two circuits, each one having more than 20 parameters. For comparison, we use the DE algorithm with the feasibility rule for constrained optimization [105], and the BO variant WEIBO [68] proposed for automated analog design. For each

experiment, a set of trial and error attempts were made to find the most suitable hyper-parameters for DE. All algorithms are implemented in Python, and all GP regression models were set using `GPYtorch`, using the same set of kernel and mean functions. For the proposed method, the number of inducing points is  $m = 150$ . The optimization runs are repeated 5 times to account for random effects. An in-house tool was used to automate simulations and result processing from Cadence Spectre. All circuits were implemented using a TSMC 90nm PDK, on a 8 core machine with a Quadro P5000 GPU.

### 4.3.1 Three-Stage Amplifier

A Three-Stage Amplifier shown in Fig. 4.5 [106] is sized in this example. This circuit was originally implemented in a .35um process, offering good driveability for large capacitive loads.

For this circuit, the search variables include transistor widths, the lengths for the devices in each stage of the amplifier and for the biasing transistors, resistors  $R_1$ ,  $R_2$ ,  $R_z$ , capacitors  $C_m$  and  $C_z$  and two biasing currents, based on the original sizing flow [106]. Note that symmetry restrictions are taken into account, resulting in a 23-dimensional optimization problem. We use a 2V power supply and a 15nF capacitive load.

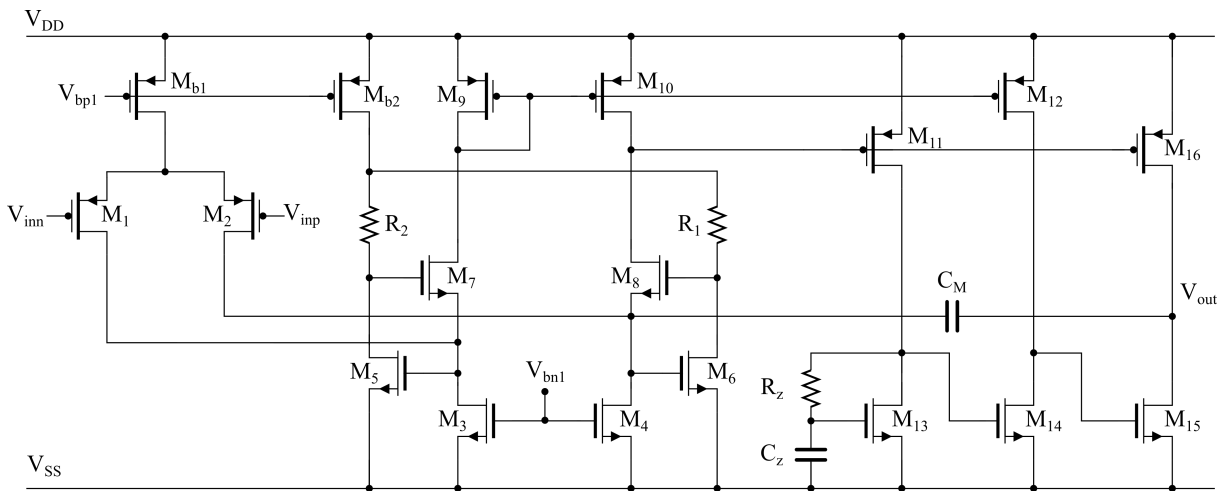


Figure 4.5: Three stage amplifier proposed in [106].

To make fair comparisons, each algorithm is restricted to search within the same variable space, which consists of the minimum and maximum ranges for device sizes imposed by the PDK, while resistors are restricted to be less than 200k $\Omega$ , capacitors less than 2pF and biasing currents less than 15uA. For specifications, we use the ones provided in the original implementation in the case of 15nF load capacitor, which are

Table 4.2: Three Stage Amplifier Specifications

Performance	Description	Specification
Load $C_L$ (nF)	Capacitive Load	15
PM ( $^\circ$ )	Phase Margin	$\geq 52.3$
DC Gain (dB)	Voltage Gain	$\geq 100$
GM (dB)	Gain Margin	$\geq 18.1$
Average SR (V/ $\mu$ s)	Average Slew Rate	$\geq 0.22$
UGF (MHz)	Unity Gain Frequency	$\geq 0.95$
$P_{dc}$ ( $\mu$ W)	Total Power @ 2V $V_{dd}$	minimize

shown in Table 4.2. The sizing goal is to minimize the total power dissipation,  $P_{dc}$ . DE population and generations are 100, the maximum number of evaluations for the other two approaches is 1500 and the initial sampling size is 100.

To examine the effect of the trust-region count, two separate experiments were executed with a single and three trust regions, all of which have a batch size of 15. The sizing results are shown in Table 5.2, where the best solution refers to lowest  $P_{dc}$  acquired with constraints met (averages and standard deviations shown). The proposed method consistently outperforms WEIBO and DE in terms of acquired solutions, with DE finding feasible solutions only 3 out of 5 times. Using 3 trust regions provides slightly better results, with an overhead on runtime, as training and sampling may occur for multiple GP models in certain iterations.

Special notice must be given on the runtimes; circuit evaluations are done in parallel, using 8 instances of the simulator each one running batched simulations. For this example, which includes two separate testbenches (small signal analysis, slew rate measurement), the averaged runtimes over 5 evaluations for a batch of 100, 15 and a single simulation are 16.3sec, 3.8sec and 1.3sec respectively. Therefore, batched parallel evaluation, provides significant runtime gains compared to sequential evaluations, which is the main reason for WEIBO to be so slow. Besides this, GPU acceleration provides an overall speedup of  $\times 2.8$  for the proposed method (over total runtime), and achieves a  $\times 40$  overall speedup compared to WEIBO.

### 4.3.2 High Linearity LNA

An LNA shown in Fig. 4.6 [107] is sized in this example. This circuit uses a complementary transconductance input stage and a set of auxiliary devices to achieve good linearity. Power supply is set to 1.2V. Following an initial manual design, the specifications for this circuit are based on the original implementation, with the exception of the working frequency, which is 2.4GHz instead of 1GHz, and they are given in Table

Table 4.3: Optimization Results for the Three Stage Amplifier

Method	Best Solution	Time	Success
DE	$131.63 \pm 3.94 \mu\text{W}$	$27.7 \pm 0.2\text{min}$	3/5
WEIBO	$114.98 \pm 0.96 \mu\text{W}$	$350.3 \pm 1.94\text{min}$	5/5
TR-1	$112.42 \pm 0.7 \mu\text{W}$	$24.33 \pm 0.31 \text{ min}$	5/5
TR-3	$112.50 \pm 0.95 \mu\text{W}$	$27.25 \pm 0.24 \text{ min}$	5/5
TR-1 (GPU)	$111.76 \pm 1.19 \mu\text{W}$	$8.71 \pm 0.39 \text{ min}$	5/5
TR-3 (GPU)	$110.54 \pm 1.33 \mu\text{W}$	$9.43 \pm 0.41 \text{ min}$	5/5

4.4. Maximization of the third-order intercept point (IIP3) is the optimization goal.

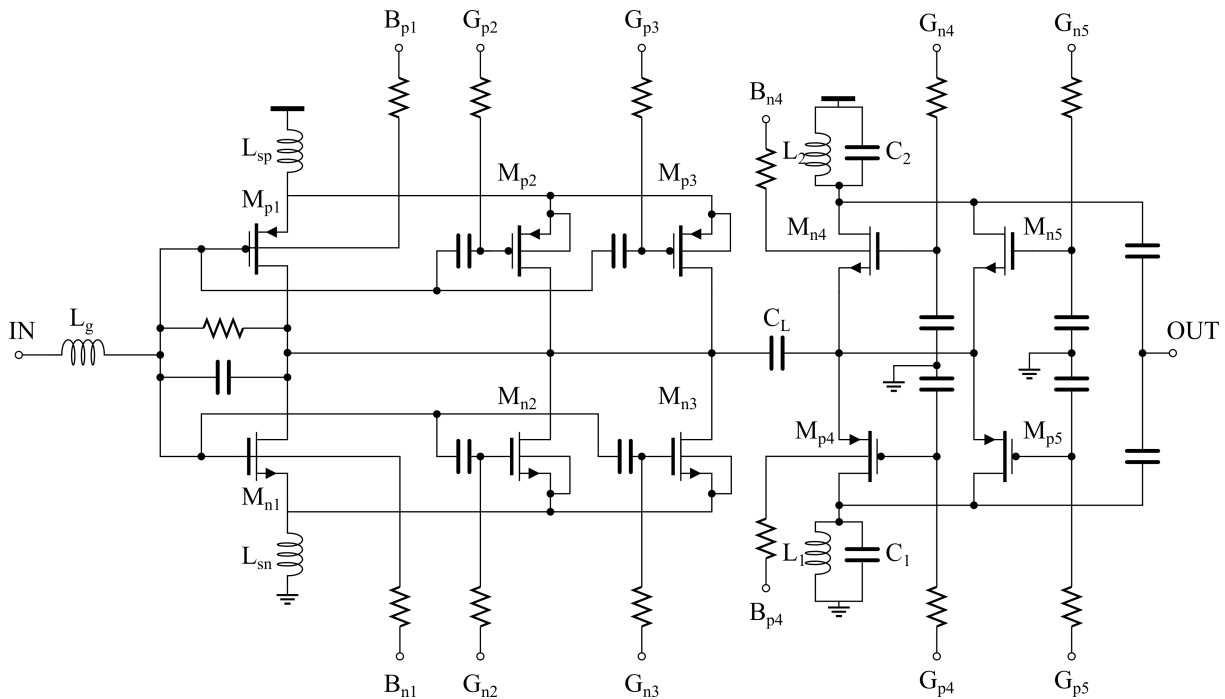


Figure 4.6: High Linearity LNA proposed in [107].

Design variables include device widths, capacitances, inductances, resistances and bias voltages. Two separate variables correspond to pmos and nmos device lengths. This amounts to a total of 33 design variables. Their ranges are the maximum and minimum allowed by the PDK for device geometries, while the others are set by an initial rough design.

The batch size of the proposed approach is 15 and its simulation budget is 2000, same with WEIBO. Both methods sample initially a batch of 100 candidate vectors. The population and generations count for DE are 100 and 200, respectively. The experimental results are shown in Table 4.5. In this demanding problem, the proposed approach clearly outperforms the other methods in terms of feasible IIP3 outcomes. While WEIBO is

Table 4.4: High Linearity LNA Specifications

Performance	Description	Specification
$S_{11}$ (dB)	Input Matching @ 2.4GHz	$\leq -10$
$S_{22}$ (dB)	Output Return Loss @ 2.4GHz	$\leq -10$
Gain (dB)	Voltage Gain @ 2.4GHz	$\geq 15$
NF (dB)	Noise Figure	$\leq 1$
$P_{dc}$ (mW)	Power dissipation	$\leq 50$
IIP3 (dBm)	Third-order intercept point	maximize

Table 4.5: Optimization Results for the High Linearity LNA

Method	Best Solution	Time	Success
DE	3.12 dBm	58min	1/5
WEIBO	$10.87 \pm 1.63$ dBm	$623 \pm 4.2$ min	5/5
TR-1	$19.81 \pm 0.97$ dBm	$37 \pm 1.3$ min	5/5
TR-3	$21.41 \pm 0.44$ dBm	$41 \pm 1.9$ min	5/5
TR-1(GPU)	$19.06 \pm 0.69$ dBm	$14.78 \pm 0.9$ min	5/5
TR-3(GPU)	$22.34 \pm 0.553$ dBm	$15.97 \pm 1.1$ min	5/5

able to find feasible solutions all times, it provides close to 10dBm less IIP3 compared to the proposed approach, within the same sample budget. This highlights the performance advantage of the proposed method in high-dimensional problems, especially against DE which finds feasible solutions only once.

In terms of runtime, execution and results processing for a batch of 100, 15 and a single simulation take 17.2, 4.9 and 1.4 seconds respectively. Therefore, batched parallel simulation favours both the proposed approach and DE against WEIBO again. GPU acceleration provides remarkable optimization speedup once more; compared to the cpu implementation, it is  $\times 2.6$  faster and provides a total runtime speedup of  $\times 42$  compared to WEIBO, for the same number of simulations.

In this experiment, employing multiple trust regions leads to better results. This can be explained as follows. Multiple trust regions traverse the variable space finding multiple paths into feasible sub-regions. By exploring different parts of the feasible variable space, the search for global optimum becomes more efficient. This however, comes with a cost in runtime, since more GP models need to be trained and sampled to provide query points.



## 4.4 Summary & Concluding Remarks

In this chapter, an efficient, low-budget, SO black-box optimization algorithm based on BO was presented. To address the case where analog circuits are parametrized by relatively large variable spaces, and their simulation is time-consuming, the proposed algorithm uses GP models to model the loss landscape and guide the optimization more efficiently, compared to population-based algorithms. The scalability of the proposed approach is due to the employed trust-region scheme, where the GP models are restricted in hyper-rectangles that do not span the entire variable space. In addition, the selection of candidate points for evaluation is based on a new acquisition function, which is able to provide multiple suggestions at each iteration. The proposed algorithm is used in combination with the in-house optimization tool that interfaces Cadence Spectre and applied on two real world circuits, in nominal conditions. In comparison with the DE population-based algorithm, and the BO-based WEIBO, which was proposed to handle circuit sizing problems in the constrained regime, the proposed method proves favourable both in terms of runtime ( $\times 42$  runtime gain compared to WEIBO for the same number of simulations) and in terms of final performance and statistical robustness.



# Chapter 5

## Sample Efficient Design Space Exploration

As analog and RF circuit design reduces to exploration of competing specifications, an efficient automated sizing framework should be able to determine optimal trade-offs while satisfying constraints. This trade-off exploration [83] process provides insights for the selection of device sizes and can be addressed by defining and solving multi-objective optimization problems (MOPs). However, methods that address constrained MOPs in a sample-efficient manner have not been thoroughly studied in the context of analog ICs. Although approaches for sample-efficient sizing of analog circuits have been proposed in the literature, they fail to include constraints in their formulations, do not handle the curse of dimensionality in any way or do not allow for parallelizable simulations [108].

Motivated by the above, a new multi-objective BO algorithm that handles constraints, LoCoMOBO, is proposed in this chapter, to address the analog and RF IC sizing problem in a sample-efficient manner. Similar to the strategy proposed in the SO setting, LoCoMOBO utilizes a local based approach that maintains separate GP models in promising sub-regions of the search space, therefore enhancing BO efficiency. This boosts the predictive capabilities of GP models in small regions of the search space and reduces extensive exploration. In contrast to classical BO and other variants, LoCoMOBO is able to provide multiple query points, therefore it enables the use of parallel simulations to speed up the optimization process. A modified Thompson Sampling [109] acquisition function is used, that ranks query points based on their constraint violation degree and their contributing Hypervolume [110]. In addition, instead of sampling from predefined quantizations of the search space, analytic samples are created and used within the proposed acquisition function. The theory for their derivation and the advantages that their use entails are also discussed. To address the challenge of computational complexity regarding training times for GP models, GPU acceleration and a batch method to simultaneously train multiple GP models is also used within LoCoMOBO.

LoCoMOBO is the first low-budget MO algorithm that accounts for the curse of dimensionality, by using trust regions. Experimental results using benchmark functions and three real-world circuits suggest that LoCoMOBO provides better trade-off infor-

mation and reduces the total runtime of the optimization compared to state-of-the-art approaches.

The chapter is organized as follows: First, the local based approach for MOPs is discussed from a high-level perspective. The procedure for creating analytic samples from GP posteriors is discussed next. Having established the above, the acquisition function of the proposed algorithm is analyzed and it tested on benchmarks functions. Finally, the algorithm is applied to three different analog circuits, considering two-objectives and three-objectives formulations with constraints, as well as PVT variations.

## 5.1 BO Local Based Approach

Before proceeding, the reader is reminded about the MOP formulation, for notational purposes. Here, analog and RF circuit sizing is cast as a MOP:

$$\begin{aligned} \min \quad & F(\mathbf{x}), \quad \mathbf{x} = [x_1, x_2, \dots, x_d] \\ \text{s.t.} \quad & g_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, l \\ & L_i \leq x_i \leq U_i, \quad i = 1, \dots, d \end{aligned} \quad (5.1)$$

where  $F$  is vector of  $m$  scalar-valued functions,  $d$  is the number of the variables,  $\mathbb{S} = \prod_{i=1}^d [L_i, U_i]$  is the variable space,  $l$  the number of constraint functions.

Expanding BO to multiple objectives and constrained high-dimensional problems are two critical challenges for an efficient low-budget MOP optimizer. In high-dimensional search spaces, GP models result in high predictive uncertainties, thereby encouraging exploration on top of exploitation of promising subregions of the design space [78]. In addition, query point selection is not straightforward in the multi-objective setting, since it must take into account pareto dominance and diversity. To address the aforementioned issues, we need to resort to new model-based approaches that exploit better the promising regions and define corresponding acquisition functions.

The family of local-based optimization algorithms is an approach towards this direction. Their purpose is to search locally for an improvement, starting from an initial estimate  $\mathbf{x}_c$  of the point of global optimum (or in this case a PS). Typically in such cases, an approximation  $\hat{F}$  of the objective function  $F$  is used to determine future query points according to

$$\mathbf{x}^* = \operatorname{argmin} \hat{F}(\mathbf{x}), \quad \|\mathbf{x} - \mathbf{x}_c\| \leq R. \quad (5.2)$$

Here  $\mathbf{x}_c$  is the current best solution and  $R$  is a trust radius, restricting the next query point to be near the current best. If the expensive evaluation of  $F(\mathbf{x}^*)$  leads to a better solution, the current best is updated. In our case,  $\hat{F}$  relates indirectly to the actual prob-

lem defined in Eq. 5.1, since it provides a metric upon the multi-objective optimization progress (see subsection 5.1.2).

Based on the aforementioned general scheme for local-based optimization, an SO BO was proposed in Chapter 4, where query point selection is restricted within a hypercube centered at  $\mathbf{x}_c$ , with its edge denoted by  $\mathcal{L}$ . GP models are trained using a dataset (archive) consisting of past objective function evaluations. By setting the maximum utility point in the archive as  $\mathbf{x}_c$ , this approach restricts excessive exploration of the search space. This trust region approach is adapted here for multi-objective BO. By using  $N_{TR} \geq 1$  trust regions in parallel, each one with their separate center and past archive, multiple paths towards the optimal sub-regions of the variable space are followed and global search is enhanced. This is particularly helpful in the case of multi-modal functions.

To extend the trust region approach to the constrained multi-objective case, one should define a measure for the *utility of the evaluated points*. This is imperative in order to select the trust region center  $\mathbf{x}_c$ , but it is not straightforward, since typically multiple pareto optimal solutions exist. Starting from the requirement for well spread pareto optimal solutions in the objective space, the trust region center is selected as follows. In each iteration, every trust region has a local PF and PS, namely  $PF_i$  and  $PS_i$ , that are determined among the samples from their respective archive  $\mathcal{D}_i$  that reside in the trust region's hypercube. The global archive  $\mathcal{D} = \bigcup_i [\mathcal{D}_i]$  is used to determine the global PF and PS. The selection of the  $i$ -th trust region center is done based on the following rule:

1. if no feasible samples exist in  $\mathcal{D}_i$ ,  $\mathbf{x}_{c,i}$  is the sample in  $\mathcal{D}_i$  with the minimum constraint-violation.
2. if feasible samples exist in  $\mathcal{D}_i$  and  $A = PS_i \cap PS \neq \emptyset$ ,  $\mathbf{x}_{c,i}$  is the the maximum crowding distance sample in  $A$
3. if feasible samples exist in  $\mathcal{D}_i$  and  $A = PS_i \cap PS = \emptyset$ ,  $\mathbf{x}_{c,i}$  is the sample from  $PS_i$  being in the topmost global dominance level and having the maximum crowding distance.

The crowding distance metric [63] quantifies the diversity of each solution in the objective space. It is used to compare solutions that are on the same dominance level. Since the crowding distance of the samples that lie on the edges of the PF is infinite, we use the maximum finite values to determine the trust region centers.

The optimizer operates on normalized search space  $\mathbb{S}_n = [0, 1]^d$ , and query points are transformed back to the original search space prior to evaluation. This facilitates GP

training and helps define trust region lengths  $\mathcal{L}_i$  in the normalized space, irrespective of the variable domain boundaries. For the  $i$ -th trust region, its hyperrectangle occupies a space  $\mathbb{S}_n^{(i)} = \{\mathbf{x} | \mathbf{x}_{L,i} \leq \mathbf{x} \leq \mathbf{x}_{U,i}\}$  with  $\mathbf{x}_{L,i} = \max(0, \mathbf{x}_{c,i} - \mathcal{L}_i)$  and  $\mathbf{x}_{U,i} = \min(1, \mathbf{x}_{c,i} + \mathcal{L}_i)$ .

Each space  $\mathbb{S}_n^{(i)}$  is updated according to the rate with which the optimizer finds new maximum utility points  $\mathbf{x}_{c,i}$ . In particular, a simple method is used that counts the successes and failures of the optimizer to find better solutions. After a user-specified number of successes (failures) for the  $i$ -th trust region,  $\mathcal{L}_i$  is increased (decreased) by a user-specified factor  $\rho > 1$ , i.e.

$$\mathcal{L}_i := \begin{cases} \rho \cdot \mathcal{L}_i & \text{for consecutive successes} \\ 1/\rho \cdot \mathcal{L}_i & \text{for consecutive failures} \end{cases} \quad (5.3)$$

Therefore, in cases where the surrogate model's predictions lead to improvements upon the optimization goal, the trust region edge increases, otherwise it shrinks to restrict the search closer to the current best point. For the rest of this work,  $\rho$  is considered constant and equal to 1.2 to allow for smooth transition between trust-region sizes.

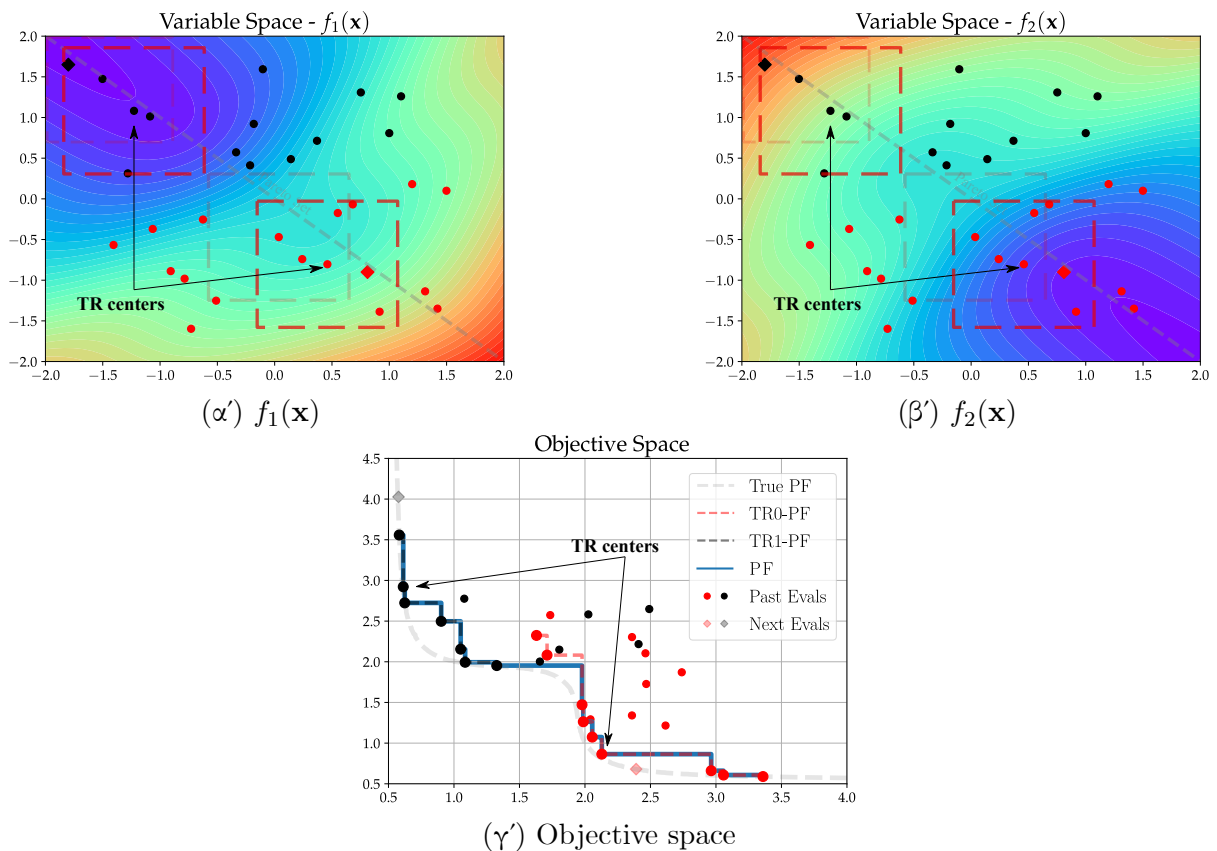


Figure 5.1: A demonstration of the local based approach for minimizing  $F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})]$  in a 2D space.

Fig. 5.1 shows an example of the local based approach, using two trust regions and a function  $F(\mathbf{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ , over a bounded space  $[-2, 2]^2$ . In this example no constraints apply. After an initial sampling of the variable space, the samples (shown as dots) are clustered to 2 trust regions, using k-Means clustering, with the black ones corresponding to the upper trust region. In each trust region (shown in red dotted boxes) in subfigures (a) and (b), a new solution (marked as a diamond) is selected and evaluated. The trust region centers are chosen so as to facilitate the diversification of the global PF solutions, shown in the objective space plot in (c). Once the query points are evaluated, the trust regions move to new locations shown in light-red dotted boxes. Subfigure (c) also shows the true PF, the empirical-approximate PF and the local PFs of each trust region. The location of the true PS is shown in light grey, the next query points as diamonds and the updated trust regions locations are shown as light-red boxes, in (a) and (b).

### 5.1.1 Sampling Functions from GP posteriors

In Chapter Chapter 4, an introduction to the theory of GPs was given, describing them as distributions over functions. This description of GPs, which is referred to as function space view [51], provides intuition regarding the shape of the unknown function to be approximated from a probabilistic perspective. However, there exists an alternative approach to describing GPs, which is referred to as weight space view [51] and it is based on Bayesian linear regression [96]. In this subsection, the weight space view of GPs is discussed and used to explain how analytic functions can be sampled from GP posteriors. These sampled functions are utilized within the proposed LoCoMOBO framework.

Initially, assume a linear model  $f$  with a set  $n$  observations  $\{X, \mathbf{y}\}$ , where  $\mathbf{y} = [y_1, \dots, y_n]$  and  $X = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T$ , with  $\mathbf{x} \in \mathbb{R}^d$ . The observations are corrupted by additive noise such that:

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{w}, \quad y = f(\mathbf{x}) + \epsilon. \quad (5.4)$$

Here, the additive noise is drawn independently from a Gaussian distribution, i.e.  $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ . The likelihood of the observations, i.e. the probability density function of the observations given the parameters  $\mathbf{w}$ , assuming that the observations are independent, is given by

$$p(\mathbf{y}|X, \mathbf{w}) = \prod_{i=1}^n p(y_i|\mathbf{x}_i, \mathbf{w}) \quad (5.5)$$

$$= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{(y_i - \mathbf{x}_i^T \mathbf{w})^2}{2\sigma_n^2}\right) \quad (5.6)$$

$$= \mathcal{N}(X^T \mathbf{w}, \sigma_n^2 \mathbf{I}). \quad (5.7)$$

In addition, assume that the weights  $\mathbf{w}$  follow a zero-mean Gaussian prior distribution, with covariance matrix  $\Sigma_p$ .

Given the data  $\{X, \mathbf{y}\}$ , the posterior distribution of the weights of the linear model is computed using the Bayes' rule

$$p(\mathbf{w}|X, \mathbf{y}) = \frac{p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|X)}, \quad (5.8)$$

where  $p(\mathbf{w})$  is the prior distribution of the weights and the denominator is the marginal likelihood, which is computed by marginalizing away the weights from the likelihood, i.e.

$$p(\mathbf{y}|X) = \int p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w})d\mathbf{w}. \quad (5.9)$$

Mathematical derivations [51] lead to the following relation for the posterior distribution

$$p(\mathbf{w}|X, \mathbf{y}) \sim \mathcal{N}\left(\frac{1}{\sigma_n^2}A^{-1}X\mathbf{y}, A^{-1}\right), \quad (5.10)$$

where  $A = \frac{1}{\sigma_n^2}XX^T + \Sigma_p^{-1}$ . In the Bayesian setting, the output of the linear model in Eq. 5.4 is a predictive distribution. Based on the assumptions about Gaussianity, the predictive distribution  $f(\mathbf{x}^*)$  for a query point  $\mathbf{x}^*$  is

$$p(f(\mathbf{x}^*)|\mathbf{x}^*, X, \mathbf{y}) = \int p(f(\mathbf{x}^*)|\mathbf{x}^*, \mathbf{w})p(\mathbf{w}|X, \mathbf{y})d\mathbf{w} \quad (5.11)$$

$$= \mathcal{N}\left(\frac{1}{\sigma_n^2}\mathbf{x}^{*T}A^{-1}X\mathbf{y}, \mathbf{x}^{*T}A^{-1}\mathbf{x}^*\right). \quad (5.12)$$

In order to increase the expressiveness of the linear model, one can project the  $d$ -dimensional inputs to a  $m$ -dimensional space, with  $m > d$ , using a feature function  $\phi: \mathbb{R}^d \rightarrow \mathbb{R}^m$ , which is essentially a vector of basis functions

$$\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \dots, \phi_m(\mathbf{x})]^T,$$

where  $\phi_i: \mathbb{R}^d \rightarrow \mathbb{R}$ , for  $i = 1, \dots, m$ . By applying linear regression on the augmented space, the model can capture complex response surfaces. In this case, the new model is given by

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w}, \quad (5.13)$$

and the number of weights is the same with the number of basis functions  $m$ . By substituting the inputs matrix  $X$  with the  $n \times m$  matrix  $\Phi = \Phi(X)$ , i.e. the concatenation of the  $(1 \times m)$  outputs of  $\phi(\mathbf{x})$  for all input vectors, as well as  $\mathbf{x}^*$  with  $\phi(\mathbf{x}^*)$ , the predictive



distribution of Eq. 5.12 becomes

$$p(f(\mathbf{x}^*)|\mathbf{x}^*, X, \mathbf{y}) \sim \mathcal{N}(\mathbf{k}(\mathbf{x}^*)^T(K + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}, k^{**} - \mathbf{k}(\mathbf{x}^*)^T(K + \sigma_n^2\mathbf{I})^{-1}\mathbf{k}(\mathbf{x}^*)), \quad (5.14)$$

where  $K = \Phi^T \Sigma_p \Phi$ ,  $\mathbf{k}(\mathbf{x}^*) = \Phi \Sigma_p \phi(\mathbf{x}^*)$  and  $k^{**} = \phi(\mathbf{x}^*)^T \Sigma_p \phi(\mathbf{x}^*)$ .

The interesting part of Eq. 5.14 is that the vector-valued basis function values are used within inner products of the form  $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \Sigma_p \phi(\mathbf{x}')$  inside quantities  $K$ ,  $\mathbf{k}(\mathbf{x}^*)$  and  $k^{**}$ . Since  $\Sigma_p$  is positive definite, it has a square root and therefore one can define  $\psi(\mathbf{x}) = \Sigma_p^{1/2} \phi(\mathbf{x})$ , which leads to  $k(\mathbf{x}, \mathbf{x}') = \psi(\mathbf{x})\psi(\mathbf{x}')^T$ , i.e. function  $k$  is equivalent to a dot product.

By replacing the inner products in Eq. 5.14 with function  $k$ , which is the kernel function, it is possible to avoid computing the features (basis functions) explicitly and focus on their inner product values. This approach is called the kernel trick [51]. A closer look at Eq. 5.14 reveals that the predictive distribution of the kernelized linear model is the same as the one of a zero mean GP. Thus, by using the kernel trick and selecting an appropriate kernel function, one can turn the Bayesian linear model of Eq. 5.4 to a GP one.

Having established the weight space view of GPs, and the fact that GP regression can be viewed as a kernelized version of the Bayesian linear model, we proceed with the sampling procedure. Let us consider a shift-invariant GP kernel  $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}')$ . Kernels such as the radial basis and the Matérn ones are shift-invariant. According to Bochner's theorem [111], a continuous function of the form  $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}') = k(\boldsymbol{\tau})$  is positive definite if and only if  $k(\boldsymbol{\tau})$  is the Fourier transform of a non-negative measure. Thus, any positive-definite measure  $k(\boldsymbol{\tau})$  can be expressed as the inverse Fourier transform of a probability distribution  $p(\boldsymbol{\omega})$ . In practice, the spectral density  $s(\boldsymbol{\omega})$  of  $k(\boldsymbol{\tau})$  must be properly scaled by a constant  $\alpha$  to correspond to a probability distribution. The kernel function can be written as the inverse Fourier transform

$$\begin{aligned} k(\mathbf{x}, \mathbf{x}') &= \int_{\mathbb{R}^d} e^{j\boldsymbol{\omega}^T(\mathbf{x}-\mathbf{x}')} s(\boldsymbol{\omega}) d\boldsymbol{\omega} \\ &= \alpha \cdot \int_{\mathbb{R}^d} e^{j\boldsymbol{\omega}^T(\mathbf{x}-\mathbf{x}')} p(\boldsymbol{\omega}) d\boldsymbol{\omega} \\ &= \int_{\mathbb{R}^d} \phi(\mathbf{x}; \boldsymbol{\omega}) \overline{\phi(\mathbf{x}'; \boldsymbol{\omega})} p(\boldsymbol{\omega}) d\boldsymbol{\omega} \\ &= \mathbb{E}_{p_{\boldsymbol{\omega}}} \left[ \phi(\mathbf{x}; \boldsymbol{\omega}) \overline{\phi(\mathbf{x}'; \boldsymbol{\omega})} \right], \end{aligned} \quad (5.15)$$

where  $\phi(\mathbf{x}; \boldsymbol{\omega}) = \sqrt{\alpha} e^{j\boldsymbol{\omega}^T \mathbf{x}}$ . Since the normalized density function  $p$  is symmetric with

respect to the origin, the kernel function is real and  $e^{j\boldsymbol{\omega}^T(\mathbf{x}-\mathbf{x}')}$  can be replaced by  $\cos[\boldsymbol{\omega}^T(\mathbf{x}-\mathbf{x}')].$

Motivated by the above, we consider a parametrized feature function  $\phi(\mathbf{x}; \boldsymbol{\omega}, b) = \sqrt{2\alpha}\cos(\boldsymbol{\omega}^T \mathbf{x} + b)$ , with  $b \sim q$ , where  $q = \mathcal{U}([0, 2\pi])$  is the uniform distribution in  $[0, 2\pi]$ . In this case it holds that

$$\begin{aligned} & \mathbb{E}_{p,q} \left[ \phi(\mathbf{x}; \boldsymbol{\omega}, b) \overline{\phi(\mathbf{x}'; \boldsymbol{\omega}, b)} \right] \\ &= 2\alpha \mathbb{E}_p \left[ \mathbb{E}_q \left[ \cos(\boldsymbol{\omega}^T \mathbf{x} + b) \cos(\boldsymbol{\omega}^T \mathbf{x}' + b) \right] \right] \\ &= \alpha \mathbb{E}_p \left[ \mathbb{E}_q \left[ \cos(\boldsymbol{\omega}^T (\mathbf{x} - \mathbf{x}')) + \cos(\boldsymbol{\omega}^T (\mathbf{x} + \mathbf{x}') + 2b) \right] \right] \\ &= \alpha \mathbb{E}_p \left[ \cos(\boldsymbol{\omega}^T (\mathbf{x} - \mathbf{x}')) \right] \end{aligned} \quad (5.16)$$

and therefore, from (5.15),  $\phi(\mathbf{x}; \boldsymbol{\omega}, b)$  can be used to express the kernel function as

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}_{p,q} \left[ \phi(\mathbf{x}; \boldsymbol{\omega}, b) \overline{\phi(\mathbf{x}'; \boldsymbol{\omega}, b)} \right]. \quad (5.17)$$

To approximate the expectation in (5.17),  $M$  distinct feature functions are utilized in a Monte Carlo fashion, by sampling  $M$  vectors  $[\boldsymbol{\omega}_i]_{i=1}^M$  according to the normalized spectral density  $p$  and  $M$  scalars  $[b_i]_{i=1}^M$  according to  $q$ . In the case of the employed Matèrn kernel, the spectral density  $p$  is a  $d$ -dimensional T-distribution, i.e.,  $p(\boldsymbol{\omega}) \sim T(0, \Lambda, 5/2)$ , where  $\Lambda$  is the diagonal matrix of the kernel's lengthscales [51]. Let us consider a column vector  $\boldsymbol{\phi}$ , which is built by concatenating the aforementioned  $M$  feature functions, scaled by  $\sqrt{M}$ , i.e.

$$\boldsymbol{\phi}(\mathbf{x}) = \begin{bmatrix} \frac{1}{\sqrt{M}} \phi(\mathbf{x}, \boldsymbol{\omega}_1) \\ \frac{1}{\sqrt{M}} \phi(\mathbf{x}, \boldsymbol{\omega}_2) \\ \vdots \\ \frac{1}{\sqrt{M}} \phi(\mathbf{x}, \boldsymbol{\omega}_M) \end{bmatrix} = \begin{bmatrix} \frac{2\alpha}{\sqrt{M}} \cos(\boldsymbol{\omega}_1^T \mathbf{x} + b_1) \\ \frac{2\alpha}{\sqrt{M}} \cos(\boldsymbol{\omega}_2^T \mathbf{x} + b_2) \\ \vdots \\ \frac{2\alpha}{\sqrt{M}} \cos(\boldsymbol{\omega}_M^T \mathbf{x} + b_M) \end{bmatrix}. \quad (5.18)$$

The inner product  $\boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{x}')$  results in the scalar  $(1/M) \sum_{i=1}^M \phi(\mathbf{x}; \boldsymbol{\omega}_i, b_i) \phi(\mathbf{x}'; \boldsymbol{\omega}_i, b_i)$ , which is a sample average approximation to the quantities in (5.17). Figure 5.2 provides an illustration of the above procedure, which is termed Random Fourier Features (RFF)[97, 111]. A Matèrn 5/2 kernel is approximated with different numbers of samples  $M$ , and functions  $k(\tau) = k(x - x')$  and  $\tilde{k}(\tau) = \boldsymbol{\phi}(x)^T \boldsymbol{\phi}(x')$  are plotted, along with the scaled components resulting from the inner product of the column vectors  $\boldsymbol{\phi}$ .

The above provide a finite-dimensional approximation of the GP's kernel function. One can use this approximation to sample from a GP as

$$\tilde{f}(\mathbf{x}) \approx \boldsymbol{\phi}(\mathbf{x})^T \mathbf{w}, \quad (5.19)$$

where  $\mathbf{w}$  is the weight vector of the kernelized linear model, as in Eq. 5.13. In case of posterior sampling,  $\mathbf{w}$  is sampled from the posterior distribution of Eq. 5.10, which is a Gaussian distribution with mean and covariance matrix

$$\begin{aligned}\mu_{w|D} &= (\Phi\Phi^T + \sigma_n^2\mathbb{I})^{-1} \Phi\mathbf{y} \\ \sigma_{w|D}^2 &= (\Phi\Phi^T + \sigma_n^2\mathbb{I})^{-1} \sigma_n^2.\end{aligned}\tag{5.20}$$

Note that in this case, the covariance matrix of prior distribution is considered to be the unit matrix  $\mathbb{I}$ .

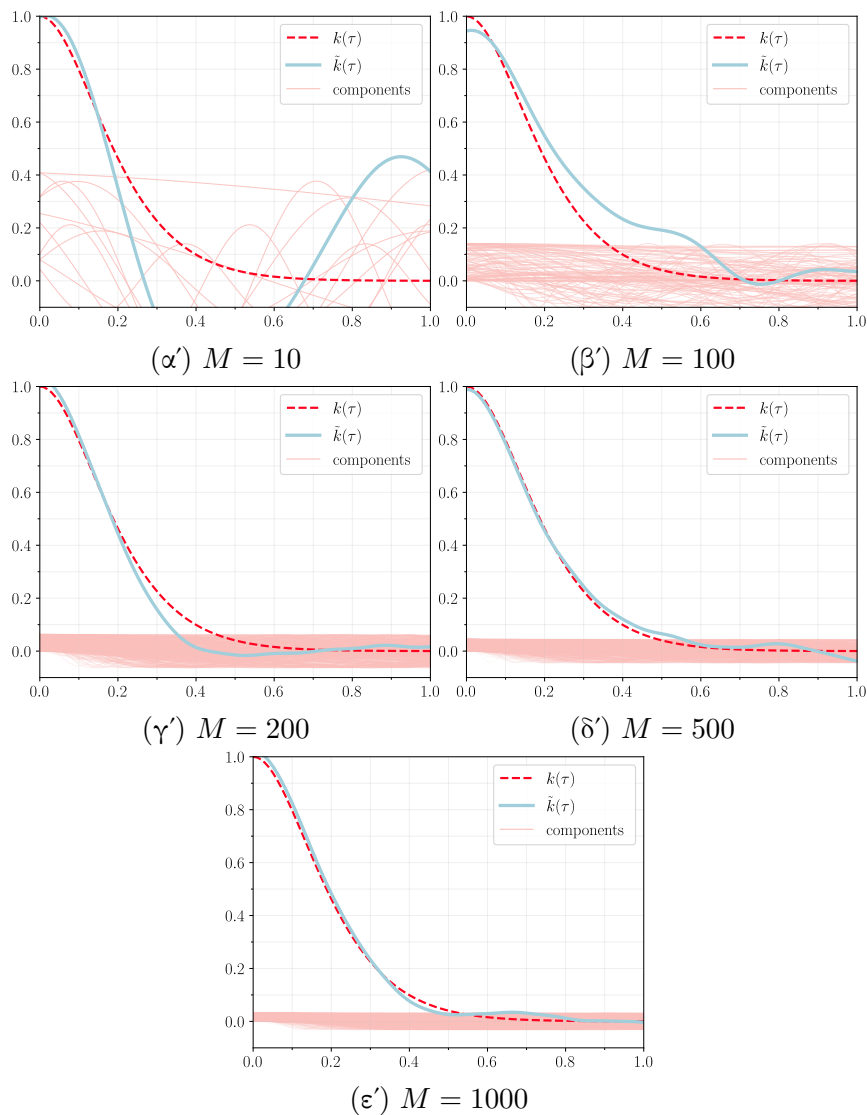


Figure 5.2: Matérn kernel approximation using RFF samples. The  $x$  axes depict the quantity  $\tau = x - x'$ .

In this subsection, the reader was reminded that GPs can be viewed as a Bayesian

linear model, where the linear operator is substituted with a set of basis functions. In practice, since these appear only within inner products in the predictive distributions, one can define a kernel function as the outcome of their inner product. Typically used kernels correspond to infinitely-many basis functions. By using Bochner’s theorem, an approximation of the kernel is derived analytically and it is used within the Bayesian linear model to derive approximate and analytic samples from the GP posterior. A reasonable question that may arise at this point is whether this approximate analytic sampling procedure yields benefits for the proposed BO framework. This question is addressed in the next subsection.

### 5.1.2 Acquisition Function

To address the constrained multiobjective selection, LoCoMOBO uses a composite selection scheme where both feasibility criteria and the pareto optimality are taken into account. The proposed acquisition function is based on Thompson Sampling (TS) [64]. Recent works highlight the suitability of TS acquisition functions for parallelizing BO [109]. By drawing  $N_S > 1$  samples from the GP posteriors, and finding a single maximum utility point for each one of them, one can have multiple query points to evaluate in parallel [109]. This is useful in cases where time-consuming function evaluations can be run in parallel, such as in our case, since parallel function evaluations provide with more information in the same time-frame.

Sampling functions from GP models, however, would require computing the joint predictive distribution over infinitely many points, which is not possible. A common practice is to find the maximizer of a discretization over the search domain [78], which is the approach discussed in the previous chapter. Although this is a simple approach and works in many cases, it is not suitable for high-dimensional problems, where exponentially many discrete points need to be used to achieve a good coverage. To circumvent this limitation, we sample GP models by building an analytic approximation to the samples, using the RFF procedure discussed in the previous subsection.

Utilizing (5.20), *multiple* GP sample approximations (RFF samples) can be drawn within TS. These are used to select query points in the constrained multi-objective setting, where query points are restricted to lie in small regions in the variable space.

**Unconstrained Problem:** Assume for now that there are no constraints and a single trust region is used, with the space occupied by its hypercube denoted by  $\mathbb{S}_n^{(1)}$ . For each of the trained GP models that approximate the objective functions, one can draw a sample using RFF. This results in a set of fast-to-evaluate functions  $\tilde{F} = [\tilde{f}_1(\cdot) \dots, \tilde{f}_m(\cdot)]$ ,

which are used in a cheap auxiliary optimization problem,

$$\min \tilde{F}(\mathbf{x}) = [\tilde{f}_1(\mathbf{x}), \dots, \tilde{f}_m(\mathbf{x})], \quad \mathbf{x} \in \mathbb{S}_n^{(1)}. \quad (5.21)$$

This problem is solved using NSGA-II [63]. The resulting PS points are query point candidates and are denoted as  $X_{cand}$ .

Selecting a query point in  $X_{cand}$  requires a metric (measure) of the *utility* of the points in  $X_{cand}$ . In our case, where multiple objectives apply, the Hypervolume Indicator (HV) [110] is an appropriate measure. Following the definition in [110], for a point set  $\mathcal{P}$  and a reference vector  $\mathbf{r} = [r_1, \dots, r_m] \in \mathbb{R}^m$ , the HV of  $\mathcal{P}$  is the  $m$ -dimensional Lebesgue measure  $\lambda_m$  of the region that is dominated by  $\mathcal{P}$  and bounded by  $\mathbf{r}$ , .i.e

$$\text{HV}(\mathcal{P}, \mathbf{r}) = \lambda_m\left(\bigcup_{\mathbf{p} \in \mathcal{P}} [\mathbf{p}, \mathbf{r}]\right). \quad (5.22)$$

Moreover, HV is used to define the *Hypervolume Improvement* (HVI) measure of the PF, used to evaluate new samples. Given a new set of points  $\mathcal{Y}$ , it is

$$\text{HVI}(\mathcal{P}, \mathcal{Y}, \mathbf{r}) = \text{HV}(\mathcal{P} \cup \mathcal{Y}, \mathbf{r}) - \text{HV}(\mathcal{P}, \mathbf{r}). \quad (5.23)$$

We select the single query point in  $X_{cand}$  as one with the largest HVI when added to the current PF, i.e.

$$\mathbf{x}^* = \underset{\mathbf{x} \in X_{cand}}{\text{argmax}} \text{HVI}(\mathcal{P}, \tilde{F}(\mathbf{x}), \mathbf{r}). \quad (5.24)$$

Note that the values of the sampled functions are used to evaluate HVI. Also, the reference vector  $\mathbf{r}$  is specified either using domain knowledge, or using past evaluations in  $\mathcal{D}$ , as

$$\mathbf{r} = \left[ \max_{\mathbf{x} \in \mathcal{D}}(f_1(\mathbf{x})), \dots, \max_{\mathbf{x} \in \mathcal{D}}(f_m(\mathbf{x})) \right]. \quad (5.25)$$

The definition in Eq. 5.25 assumes a minimization problem as in Eq. 5.1. If the problem is a maximization one, then max function should be replaced by min in Eq. 5.25. An illustration of the HV concept is given in Fig. 5.3.

**Constrained Problem:** The above procedure extends to constrained problems. In our approach, each constraint function is modeled separately by a GP model. Sampling from all of the employed GP models using RFF results in an additional set of constraint functions  $[\tilde{g}_j(\cdot)]_{j=1}^l$ . Now NSGA-II is equipped with the feasibility rule [112] to account

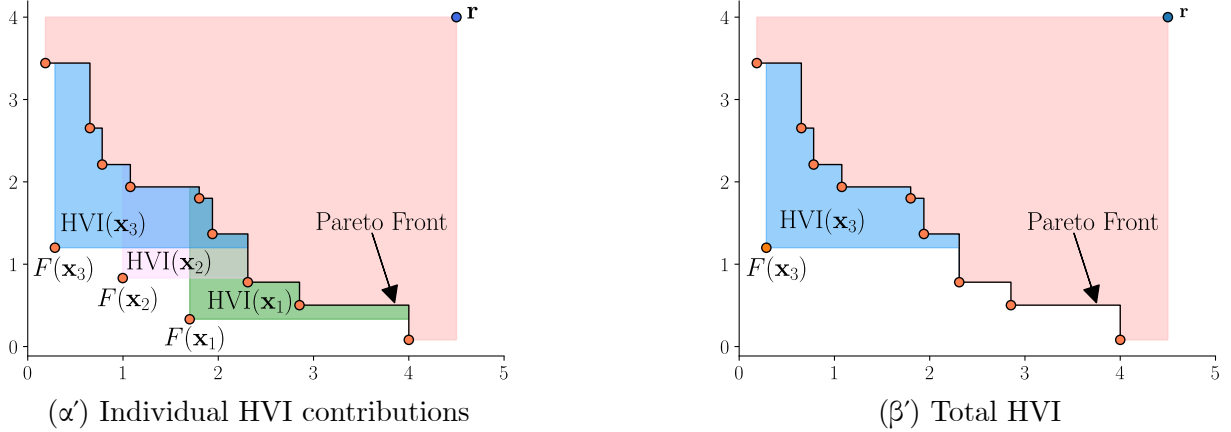


Figure 5.3: Illustration of the Hypervolume and Hypervolume Improvement concepts. Minimization is assumed here. The light-orange area is the dominated HV. Out of three candidates,  $\mathbf{x}_3$  provides the largest HVI.

for constraints and the cheap optimization problem is formulated as

$$\begin{aligned} \min \quad & \tilde{F}(\mathbf{x}) = [\tilde{f}_1(\mathbf{x}) \dots, \tilde{f}_m(\mathbf{x})] \\ \text{s.t.} \quad & \tilde{g}_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, l, \quad \mathbf{x} \in \mathbb{S}_n^{(1)}. \end{aligned} \quad (5.26)$$

To select a single query point, we distinguish two cases: 1) If no feasible pareto optimal solution is found, the set of the candidate points for evaluation,  $X_{cand}$ , is formed by the individuals of the final NSGA-II generation, and, the query point is selected according to minimize the constraint violation, i.e.

$$\mathbf{x}^* = \underset{\mathbf{x} \in X_{cand}}{\operatorname{argmin}} \tilde{C}\tilde{V}(\mathbf{x}). \quad (5.27)$$

This drives the optimization towards regions of the search space with low constraint violation. Note that the notation  $\tilde{C}\tilde{V}$  is used to highlight the fact that the constraint violation function is applied on the sampled function values. 2) If feasible pareto optimal solutions exist, the query point is selected among them, based on their respective HVI contributions.

Multiple query points can be selected by optimizing on multiple GP posterior samples. In this case,  $N_S$  batches of sampled objective and constraint functions  $[\tilde{F}^{(i)}]_{i=1}^{N_S}$  and  $[\tilde{g}_1^{(i)}, \dots, \tilde{g}_l^{(i)}]_{i=1}^{N_S}$  are used in  $N_S$  separate auxiliary optimization problems as in Eq. 5.26, resulting in  $N_S$  batches of candidate points  $[X_{cand}^{(i)}]_{i=1}^{N_S}$ . The main challenge is to determine a set of  $N_S$  distinct candidate vectors, jointly resulting in maximum HVI. This is a non-trivial task since computing the joint HVI scales exponentially with  $N_S$  [110].

Instead, we adopt an iterative greedy approach where no joint HVI calculation is performed. Starting from the current PF HV, the candidate point with maximum HVI contribution from the first batch of sampled functions is selected and added to the PF. The next query point will be selected among the auxiliary optimization results of the next batch of sampled functions using the augmented PF from the previous selection. After  $N_S$  iterations, a total of  $N_S$  query points are selected for evaluation.

When  $N_{TR} > 1$  trust regions are employed, each one holds  $N_S$  batches of sampled functions and auxiliary optimization results that reside in their separate hypercube. Similarly to the above procedure,  $N_S$  query points are selected greedily from  $N_S$  augmented candidate pools  $[X_{cand}^{(i)}]_{i=1}^{N_S}$ , each one corresponding to one RFF sample. In this case, however, each  $X_{cand}^{(i)}$  is the union of candidate points of all trust regions, based on their  $i$ -th sample, i.e.  $X_{cand}^{(1)}$  holds the results from the first sample of all trust regions and so on.

Algorithm 3 summarizes the selection scheme for the proposed LoCoMOBO algorithm.

---

**Algorithm 3: LoCoMOBO Acquisition Function**


---

**Input** : batch size  $N_S$ ; trust region count  $N_{TR}$ ; GP models  $[f_1^{(i)}, \dots, f_m^{(i)}]_{i=1}^{N_{TR}}$ ,  $[g_1^{(i)}, \dots, g_l^{(i)}]_{i=1}^{N_{TR}}$ ; data  $\mathcal{D}$ , spaces  $\{\mathbb{S}_n^{(i)}\}_{i=1}^{N_{TR}}$ , reference vector  $\mathbf{r}$

**Output**: query points  $\mathbf{x}_i^*$ ,  $i = 1, \dots, N_S$

Compute PF  $\mathcal{P}$  from  $\mathcal{D}$

$X_{cand}^{(i)} \leftarrow \emptyset \quad \forall i \in \{1, \dots, N_S\}$

**for**  $i = 1, \dots, N_{TR}$  **do**

**for**  $j = 1, \dots, N_S$  **do**

        Sample  $\tilde{f}_r \sim f_r^{(i)}$  using RFF  $\forall r \in \{1, \dots, m\}$

        Sample  $\tilde{g}_r \sim g_r^{(i)}$  using RFF  $\forall r \in \{1, \dots, l\}$

$X_{cand} \leftarrow$  candidates from problem in (5.26) on  $\mathbb{S}_n^i$

$X_{cand}^{(j)} \leftarrow X_{cand}^{(j)} \cup X_{cand}$

**for**  $i = 1, \dots, N_S$  **do**

$A \leftarrow \{\mathbf{x} \in X_{cand}^{(i)} \mid \tilde{C}\tilde{V}(\mathbf{x}) = 0\}$

**if**  $A \neq \emptyset$  **then**

$\mathbf{x}_i^* \leftarrow$  Maximum HVI( $\mathcal{P}$ ,  $\tilde{F}(\mathbf{x})$ ,  $\mathbf{r}$ ) among  $A$

$\mathcal{P} \leftarrow \mathcal{P} \cup \tilde{F}(\mathbf{x}_i^*)$

**else**

$\mathbf{x}_i^* \leftarrow \operatorname{argmin}_{\mathbf{x} \in X_{cand}^{(i)}} \tilde{C}\tilde{V}(\mathbf{x})$

---

The proposed acquisition function, therefore, depends on the sampling of analytic functions in order to utilize an off-the-shelf optimizer on them. In comparison to the naive sampling of the variable space, this approach is intuitively better, since it sub-

stitutes the discrete quantization of the search space with continuous functions, which can provide queries everywhere in their domain. To highlight the advantage of the RFF compared to the quantization procedure, we provide empirical evidence on a relatively high-dimensional 30D MOP. We use the unconstrained *zdt1* MOP with two objectives, and utilize the proposed acquisition function with 100, 300, 1000 and 2000 RFF features. To include the quantization procedure as well, we consider the same selection strategy, i.e. using the hypervolume and the trust regions, but instead of using the pareto set resulting from the problem of Eq. 5.21, we use the same Sobol sampling procedure as in the previous chapter, with 100, 300, 1000 and 2000 samples per query point. For both cases, the initial sample count is 20, the total sample count 120 and  $N_S = 5$ . The experiments are repeated 10 times to account for fluctuations.

Fig. 5.4 shows the results of the experiment, in terms of the resulting Hypervolume. The reference point in this case is  $[0.99, 5.9]$ . It is seen that the RFF experiments yield better values for HV, compared to the ones using the quantization procedure. In addition, Fig. 5.5 depicts all of the PFs resulting from this experiment. The ones corresponding to quantization are clustered upwards, meaning that they provide worse trade-off compared to the RFF ones, since this is a minimization problem. The empirical results, therefore, align with the theoretical expectation that RFF sampling would be beneficial for LoCoMOBO.

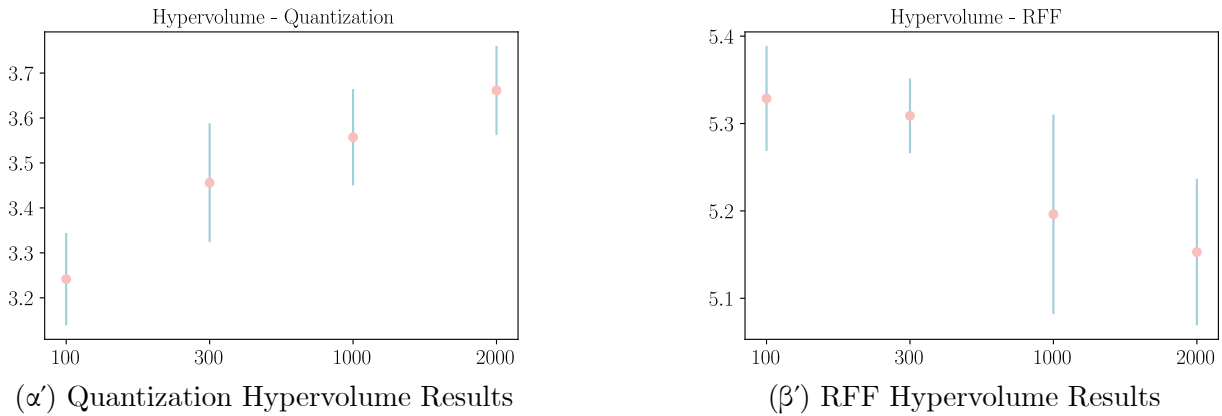


Figure 5.4: Illustration of the mean and standard deviations of the resulting Hypervolume metrics on the *zdt1* experiment.

### 5.1.3 Implementation

Here we provide implementation instructions for LoCoMOBO and relevant comments. The probabilistic nature of GP models is useful for fast convergence to optimal solutions, but it comes at a high computational cost. In practice, GP inference and



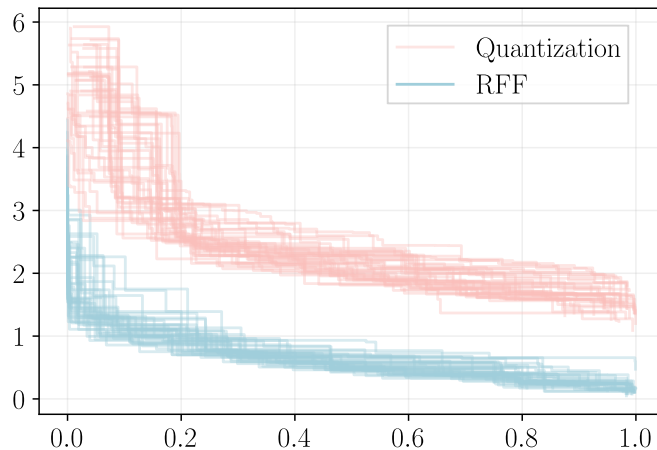


Figure 5.5: The PFs from the zdt1 experiment. The RFF results in better trade-offs.

training scale cubically with the number of samples used for training [51]. Therefore, as the optimization process progresses and more query points are evaluated, the time spent on GP model training increases. Another important aspect concerns problems with many objective and/or constraint functions; training separate GP models for each one of them can be time-consuming and even surpass the time needed for expensive exact function evaluation.

To this end, we leverage recent advances in GP inference by using Black-box Matrix-Matrix multiplication (BBMM) [95]. This uses a highly parallelized routine for matrix-matrix multiplications to perform all computations necessary for GP inference. GPyTorch provides a framework for BBMM based GP training and was used for the development of the GP models. By using tensorial representation for GP kernel matrices, GPyTorch allows for the simultaneous training of multiple GP models with considerable speed enhancements [95]. Furthermore, GPyTorch enables the use of GPUs for fast inference. In our case, models associated with a single trust region are trained simultaneously.

#### 5.1.4 Summary

The complete flow of LoCoMOBO is shown in Algorithm 4. The maximum number of iterations is computed by the maximum number of evaluations and the batch size  $N_S$ . It is worth mentioning that the GP models of a trust region are trained only when new query points coming from the same trust region are selected and evaluated.

To demonstrate the benefits of LoCoMOBO, we consider a set of benchmark functions for constrained MOPs with varying dimensions. OSY is a 6D problem having 2 objectives and 6 constraints, MW2 has 15 variables, 2 objectives and 1 constraint and C2DTLZ2 has 12 variables, 3 objectives and one constraint. For performance evaluation we use the HV metric of the resulting PF, with reference points  $[0, 100]$ ,  $[1.5, 1.5]$  and

**Algorithm 4:** LoCoMOBO Algorithm

---

**Input** : trust region count  $N_{TR}$ ; batch size  $N_S$ , initial samples  $N_0$ , maximum iterations  $T_{max}$ , initial trust region lengths

**Output:** PF, PS

Evaluate objectives and constraints at  $N_0$  random points

Initialize  $\mathcal{D}_i \forall i \in \{1, \dots, N_{TR}\}$  with all initial samples

k-Means clustering of initial samples and select  $\mathbf{x}_{c,i} \forall i \in \{1, N_{TR}\}$  as the  $i$ -th centroid

**for**  $i = 1, \dots, T_{max}$  **do**

- Select  $N_S$  query points using the acquisition function
- Evaluate objectives and constraints at selected points
- for**  $j = 1 \dots, N_{TR}$  **do**
- if**  $j$ -th trust region evaluated query points **then**
- Update  $\mathcal{D}_j$  and trust region center  $\mathbf{x}_{c,j}$
- Train GP models of  $j$ -th trust region
- Update  $\mathcal{L}_j$

---

[1.1, 1.1, 1.1] respectively.

To highlight the advantages of the local-based approach, we introduce two baseline methods for comparison: 1) Baseline-1 uses the proposed acquisition function with a single trust region that spans the entire variable space. This is a special case of the proposed algorithm and can be considered as a constraint handling TSEMO [113] variant. 2) Baseline-2 uses a single trust region that also spans the entire variable space, and selects query points based on the PF of LCB acquisition functions on each objective, such as in [66]. To account for constraints, the pointwise values of each LCB function are weighted by their probability of feasibility [68]. The most diverse candidate solutions are selected as query points from the resulting PF. For the proposed approach, we include a single-trust-region (LoCoMOBO  $N_{TR} = 1$ ) and a two-trust-region (LoCoMOBO  $N_{TR} = 2$ ) approach, both of which use local trust regions, unlike Baseline-1. The batch size for the above methods is set to  $N_S = 5$ . WEIBO [68], MESMOC [114], NSGA-II and NSGA-III [115] are also included for comparisons. For all BO-related methods the initial samples were set to  $2(d + 1)$  where  $d$  is the dimension of each problem, and the maximum evaluations were set to 200 for OSY, 500 for C2DTLZ2 and 900 for the demanding MW2. Maximum evaluations for NSGA-II and NSGA-III are set to 3 times the aforementioned limits, and the number of reference directions is 24.

Table 5.1 shows the results of the experiments. LoCoMOBO outperforms the other methods in the given evaluation budgets. For the case of OSY, it is evident that a single trust region is enough to produce acceptable results. This can be verified by the performance of global Baseline-1. For the other two functions, the two-trust-region LoCoMOBO performs better than the single trust-region one, while the rest of the methods

Table 5.1: HV Results (mean  $\pm$  std) on Benchmark Functions

	OSY	MW2	C2DTLZ2
LoCoMOBO $N_{TR}=1$	<b>21339 <math>\pm</math> 345</b>	1.34 $\pm$ 0.11	0.49 $\pm$ 0.02
LoCoMOBO $N_{TR}=2$	20956 $\pm$ 616	<b>1.46 <math>\pm</math> 0.12</b>	<b>0.54 <math>\pm</math> 0.06</b>
WEIBO	10343 $\pm$ 1207	NA	0.283 $\pm$ 0.12
MESMOC	11394 $\pm$ 977	NA	0.184 $\pm$ 0.02
Baseline-1	21056 $\pm$ 522	0.92 $\pm$ 0.09	0.46 $\pm$ 0.03
Baseline-2	20097 $\pm$ 609	0.81 $\pm$ 0.12	0.43 $\pm$ 0.05
NSGA-II	11508 $\pm$ 3105	1.12 $\pm$ 0.22	0.39 $\pm$ 0.11
NSGA-III	12213 $\pm$ 1793	1.22 $\pm$ 0.15	0.45 $\pm$ 0.03

Table 5.2: Runtime Comparison (s)

		OSY	MW2	C2DTLZ2
	LoCoMOBO $N_{TR}=1$	57	531	184
GPU	LoCoMOBO $N_{TR}=2$	73	706	251
	WEIBO	944	4671	1832
	LoCoMOBO $N_{TR}=1$	65	1887	345
CPU	LoCoMOBO $N_{TR}=2$	77	2217	393
	WEIBO	1124	15933	3519

either fail to find feasible points or result in much worse PFs. This is demonstrated in the case of MW2, where the feasible region occupies little portion of the design space, showing that the proposed method can handle effectively constrained problems. Multiple trust-region approach proves to be effective in high-dimensional problems, where pareto optimal solutions may lie away from each other or in disconnected regions.

To provide a quantitative measure of the speed gains of GPU acceleration, Table 5.2 shows the average runtimes of the above experiments for the batched proposed LoCoMOBO and the sequential WEIBO. It is shown that a slight increase in runtime is induced by including additional trust regions, which is expected as more GP models need to be trained. As expected, the batched approach is faster than the sequential WEIBO, since GP model training is done fewer times for the same sample budget. In the 15-dimensional MW2 problem, where the sample budget is larger compared to the other experiments, the inclusion of GPU acceleration provides a speed-up of  $\times 3.5$  in the optimization procedure.

By optimizing the OSY function using different batch size  $N_S$ , we demonstrate the effect of the batch size parameter on the LoCoMOBO's performance. Fig. 5.6 shows the results of this experiment, highlighting that there is no profound difference in the

performance of LoCoMOBO with respect to different batch sizes. In this case, we used a single trust region.

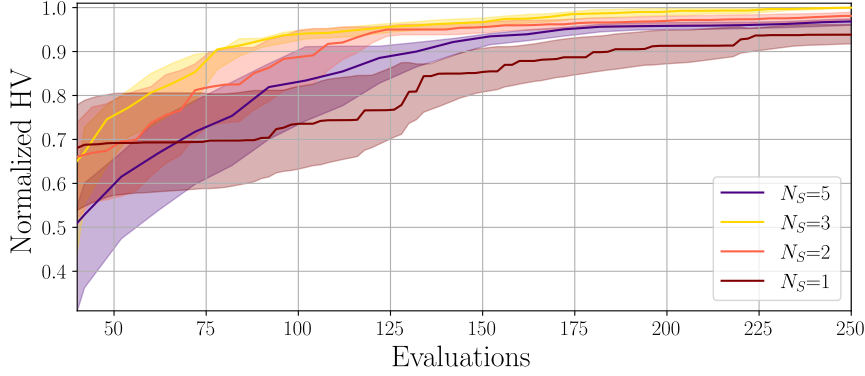


Figure 5.6: The evolution of HV attained by the proposed approach using different batch sizes. The values in the  $y$  axis are normalized to the largest HV value attained.

### 5.1.5 MOP Example

Here, for completeness, the operation of the proposed LoCoMOBO algorithm is illustrated on a simple, non-constrained MOP:

$$\begin{aligned} \min \quad & F(\mathbf{x}), \quad \mathbf{x} = [x_1, x_2] \\ \text{s.t.} \quad & -2 \leq x_i \leq 2, \quad i = 1, 2. \end{aligned} \quad (5.28)$$

where it holds that

$$\begin{aligned} f_1(x_1, x_2) &= \frac{1}{2} \cdot (\sqrt{1 + (x_1 + x_2)^2} + \sqrt{1 + (x_1 - x_2)^2} + (x_1 - x_2)) + d, \\ f_2(x_1, x_2) &= \frac{1}{2} \cdot (\sqrt{1 + (x_1 + x_2)^2} + \sqrt{1 + (x_1 - x_2)^2} - (x_1 - x_2)) + d, \\ d &= 0.8 \cdot e^{-(x_1 + x_2)^2}. \end{aligned}$$

A single trust region is considered, and the number of samples is set to  $N_S = 2$ . The number of initial points is 5 and the number of maximum evaluations is 45. The initial trust region length is 0.3, and the number of RFF features is 300.

Figures 5.7, 5.8, 5.9, 5.10 demonstrate the variable and the objective spaces of the examined MOP, with additional information regarding the query points' locations, trust region edge size and center, candidate pareto fronts, candidate pareto sets, and the approximation of the optimal pareto set and front that the algorithm determines in

each step. Each row represents an optimization step (i.e. an iteration), with  $f_1(x_1, x_2)$  being the leftmost plot,  $f_2(x_1, x_2)$  the center one and the objective space being the rightmost one. Each of the first two plots show the optimal pareto set as a dashed green line, the locations of the points that constitute the current approximation of the optimal pareto set as black dots, the trust region center as a yellow cross and the trust region as a red-colored square. In addition, the locations of the candidate pareto sets that result from optimizing Eq. 5.26, for one of the two RFF samples considered ( $N_S = 2$ ), are given as pink and brown dots. The locations of the query points, which are selected from the aforementioned candidate pareto sets, are also given as cyan-colored dots.

The rightmost plot of each row depicts the optimal pareto front as a green dotted line. Also, the candidate pareto fronts resulting from Eq. 5.26 for each RFF sample are given as yellow and red dots. The current approximation of the optimal pareto front is given with black color, and the candidate values of the selected query points are purple squares. Finally, cyan-colored diamonds represent the actual, evaluated values of the query points.

By examining the figures, one can make the following remarks: The actual values of the black-box function deviate from the predictions of the GP model's sample functions. This is evident especially in the first five iterations of the procedure, however, it is expected. GP models cannot produce exact approximations about the underlying functions using few evaluations, and, in the context of LoCoMOBO, they are used to suggest promising locations instead of to serve as surrogates. Another remark is about the size of the trust region; since in this case the algorithm keeps on finding new solutions to add to its approximate pareto front, the trust regions are enlarged to allow for better exploration of the variable space. The initial location and size of the trust region allow for finding query points that correspond to the lower-right part of the optimal pareto front (Steps 1 in Fig. 5.7 $\alpha'$  and 2 in Fig. 5.7 $\beta'$ ). However, the size adjustment and the relocation of the trust region enables the algorithm to find pareto optimal solutions from the whole range of the pareto set, in less than 15 steps (Step 14 in Fig. 5.9 $\delta'$ ).

The usefulness of the trust region approach in reducing excessive exploration can be witnessed in the cases where the candidate pareto sets lie at the borders of the trust region. For instance, in Step 13, shown in Fig. 5.9 $\gamma'$ , candidate points from the brown-colored sample are located far away from the trust region center, and at the border of the trust region. Had there not been that limitation, a query point could have been selected at the low-left part of the variable space, which is far away from the optimal pareto set location.

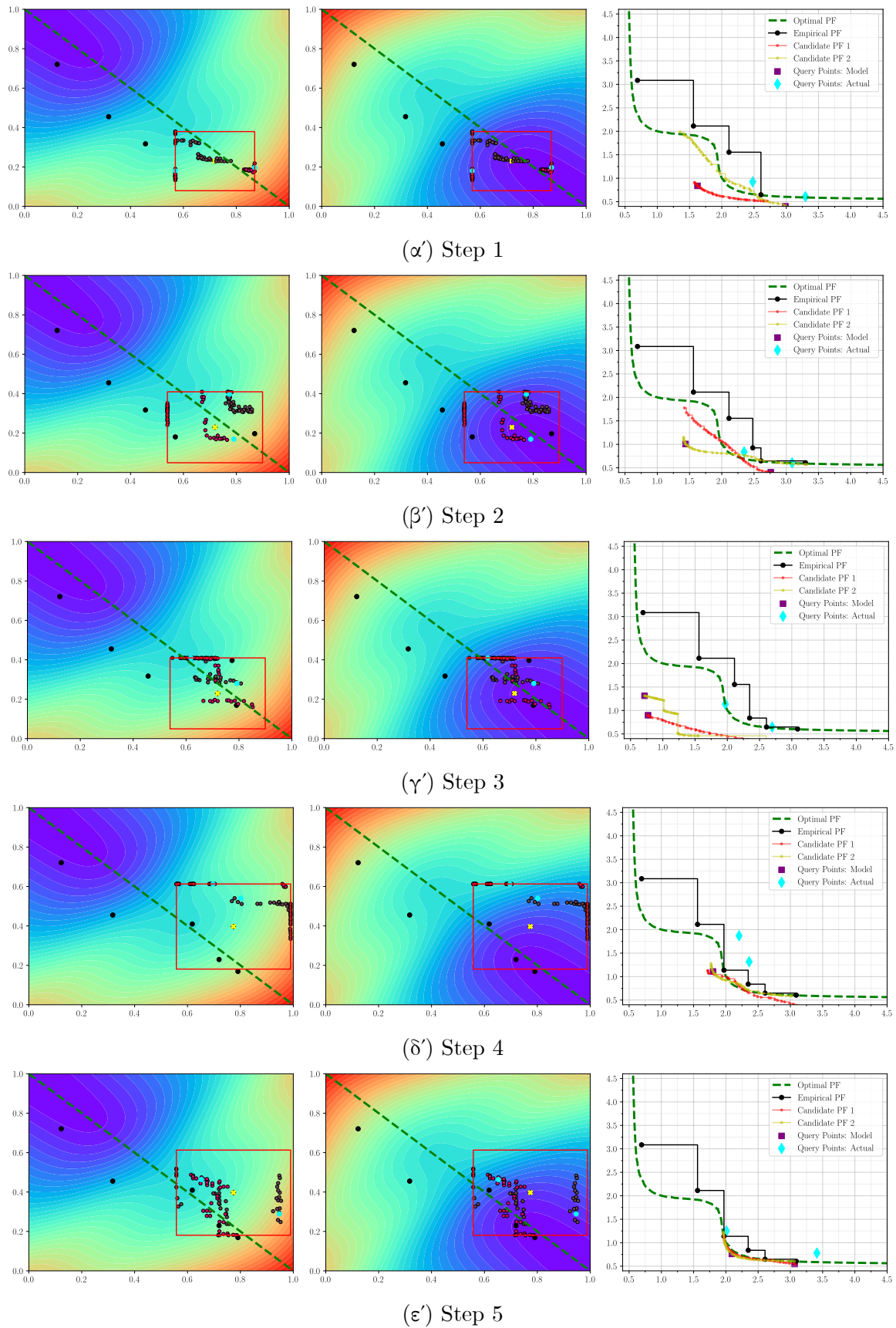


Figure 5.7: Illustration of the operation of LoCoMOBO for optimizing the MOP of Eq. 5.28.



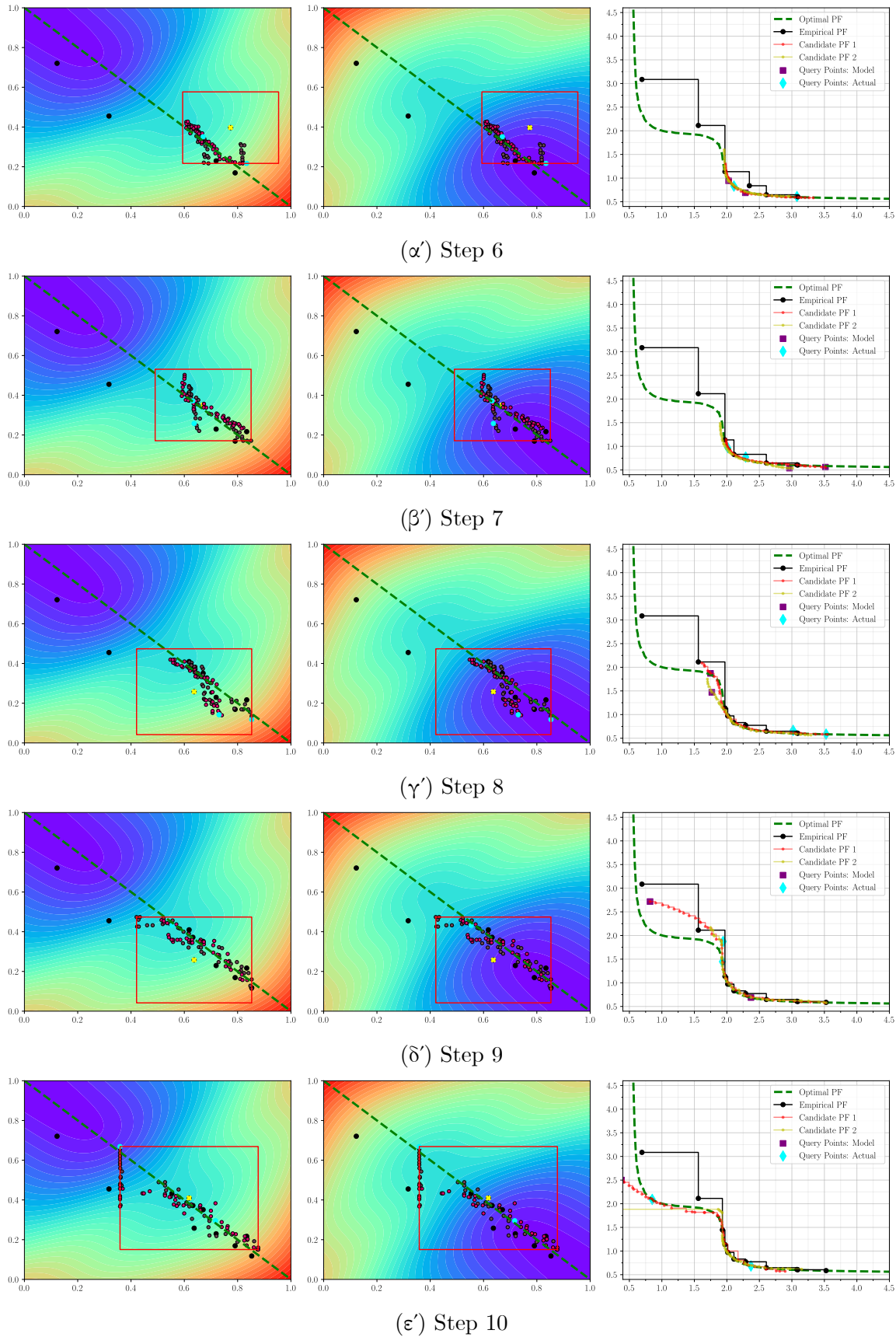


Figure 5.8: Illustration of the operation of LoCoMOBO for optimizing the MOP of Eq. 5.28.

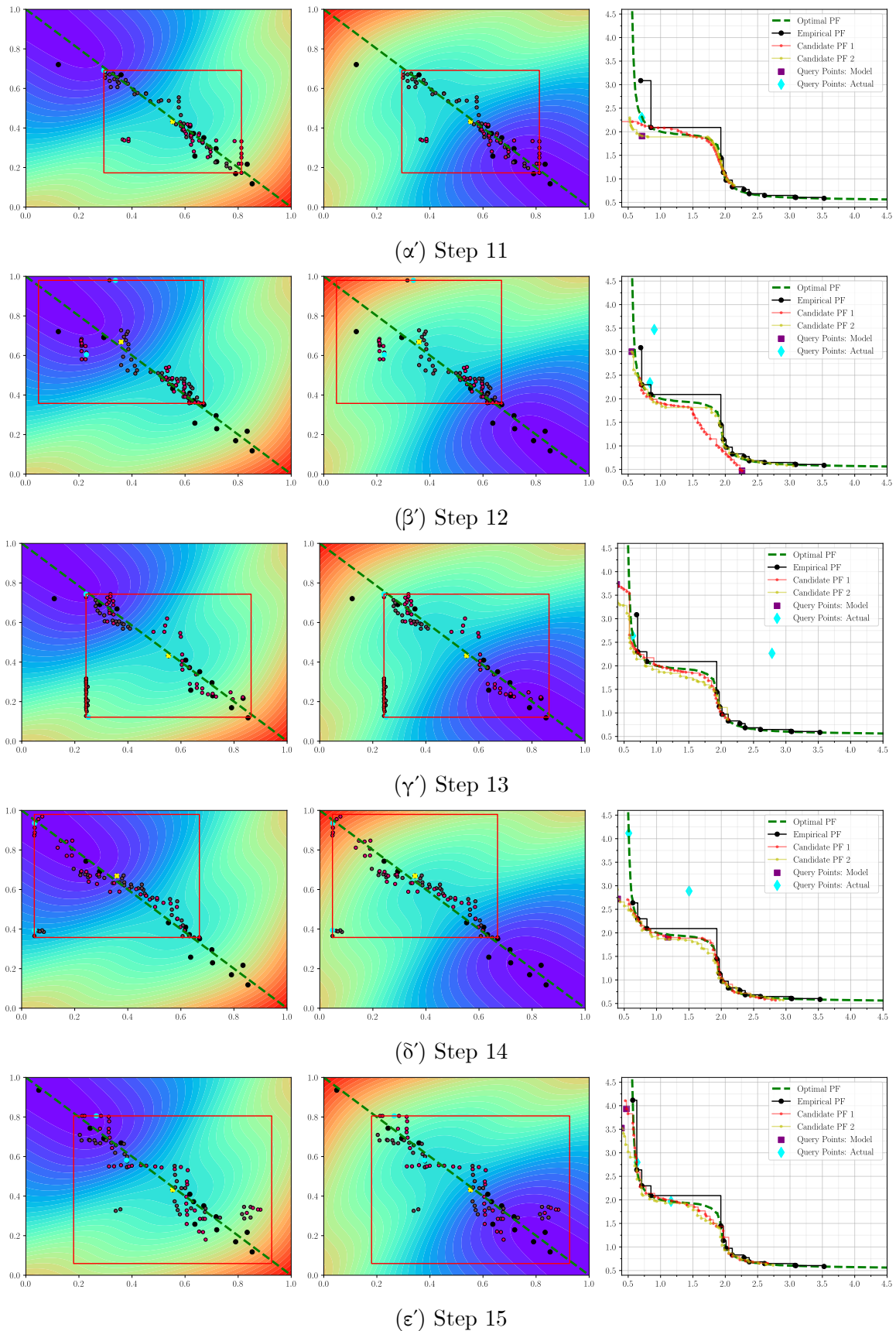


Figure 5.9: Illustration of the operation of LoCoMOBO for optimizing the MOP of Eq. 5.28.



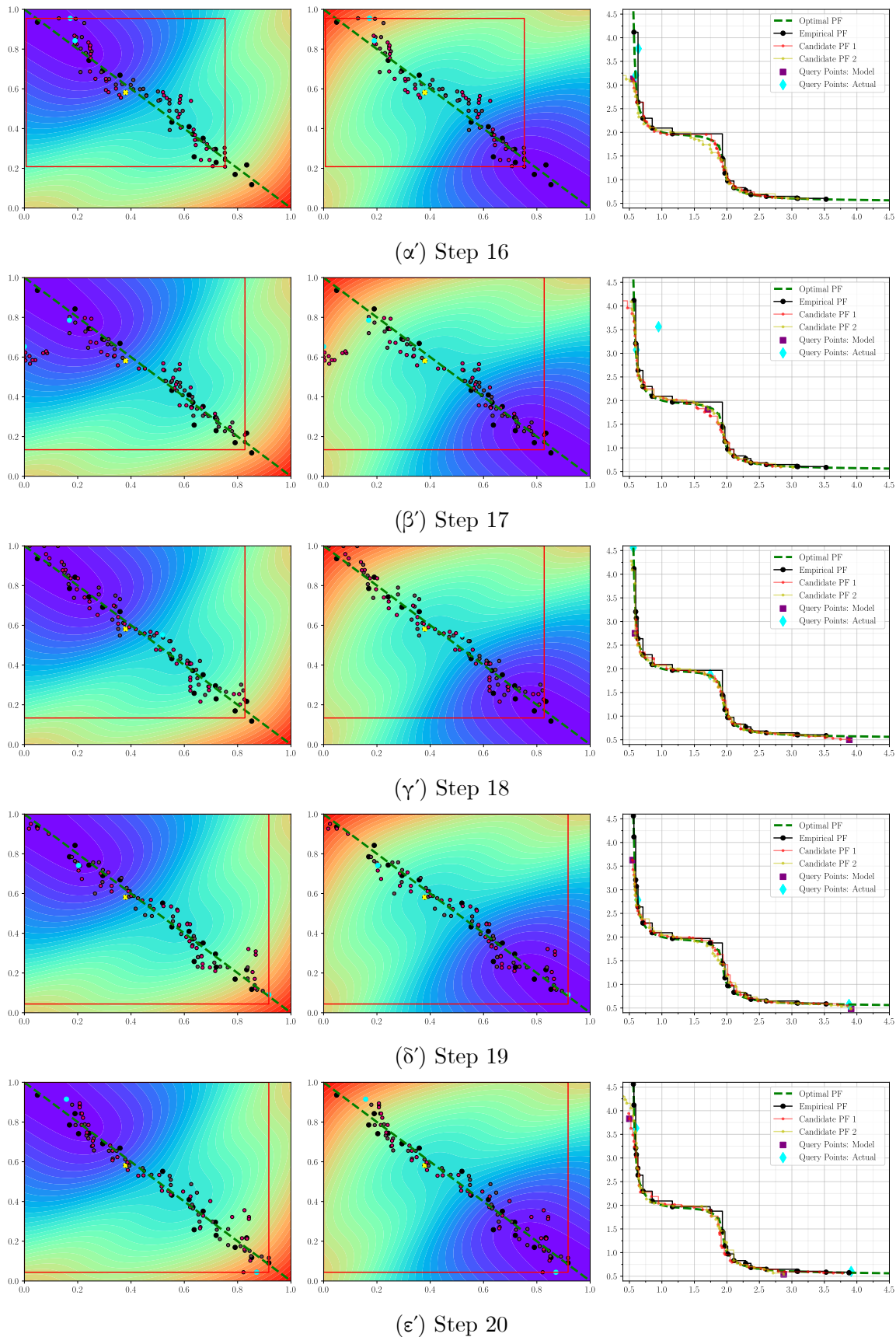


Figure 5.10: Illustration of the operation of LoCoMOBO for optimizing the MOP of Eq. 5.28.

## 5.2 Circuit Design Applications

In this section, LoCoMOBO is applied to two real-world CMOS amplifiers selected from the literature. To highlight the gains of LoCoMOBO, we consider two optimization problem formulations for sizing each circuit, a two-objective and a three-objective one. Both circuits are designed in Cadence Virtuoso and Spectre is used for simulation. A TSMC 90nm PDK is used. All of the algorithms discussed are implemented in Python and the experiments were executed on a Linux workstation using an 8 core CPU and a P5000 GPU.

### 5.2.1 Three Stage Amplifier

A three stage amplifier [106] shown in Fig. 5.11 is sized in this subsection. It includes a wideband current buffer and an active left-half-plane zero [106] to increase its driveability to large capacitive loads, without sacrificing its bandwidth.

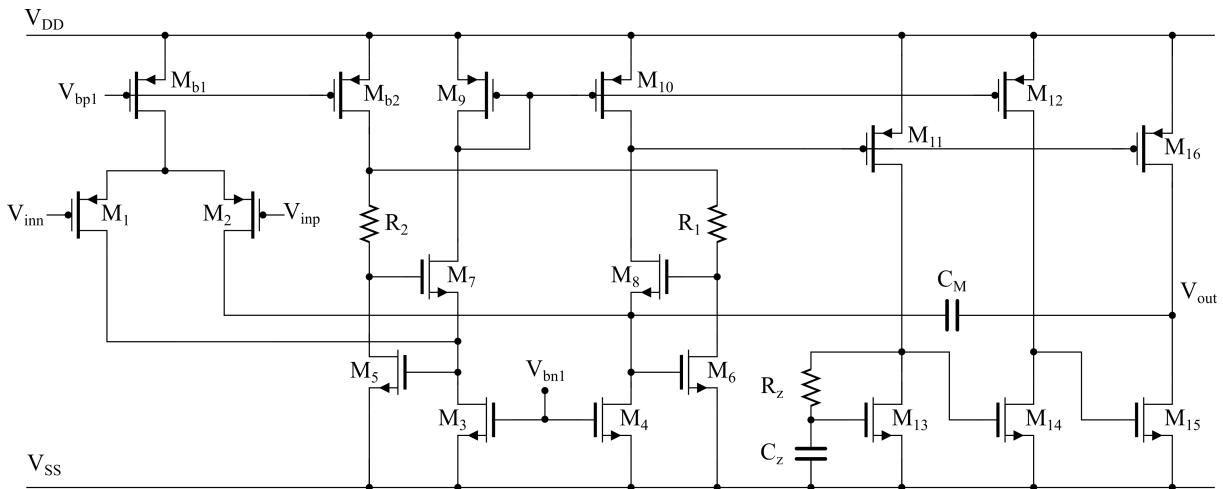


Figure 5.11: Three stage amplifier proposed in [106].

The amplifier consists of 18 transistors, the halves of the current mirrors generating voltages  $V_{bn1}$ ,  $V_{bp1}$  (not shown in Fig. 5.11), three resistors and two capacitors. Based on the circuit's topology and taking into account symmetry constraints, a total of 23 independent variables are used for sizing. These include 13 transistor widths, four transistor lengths, two biasing currents and  $R_1$ ,  $R_z$ ,  $C_z$  and  $C_M$ . Assuming no prior knowledge of the optimal parameter values, the ranges of the transistors' lengths and widths are restricted to be less than  $1.5\mu\text{m}$  and  $100\mu\text{m}$  respectively, while their lower bounds are equal to the minimum acceptable by the PDK. The allowable ranges for capacitors, resistors and biasing currents are  $[0.5, 6]\text{pF}$ ,  $[0.1, 300]\text{k}\Omega$  and  $[0.4, 6]\mu\text{A}$  respectively. To

obtain the circuit’s performance metrics, two testbenches are used, one for transient simulation and slew rate measurements, and the other for AC and DC analysis.

## Two-objective formulation

For the two-objective case, we seek to determine the optimal trade-off between power consumption and DC voltage gain. The remaining specifications, based on the original implementation in [106] with capacitive load of  $C_L = 15\text{nF}$ , constitute the optimization constraints and are given in detail in Table 5.3.

Table 5.3: Specifications for Three Stage Amplifier - Two Objectives

Performance Metrics	Specifications [106]
DCGain	maximize
$P_{dc}$	minimize
$SR_{avg}$	$\geq 0.22V/\mu s$
UGF	$\geq 0.95\text{MHz}$
PM	$\geq 52.3^\circ$
GM	$\geq 18\text{dB}$
$C_L$	$= 15\text{nF}$
$V_{DD}$	$= 2\text{V}$

For LoCoMOBO, two cases with a single and two trust regions are considered. All BO-related methods discussed in the previous section are applied for comparison. The maximum number of simulations is 1300 and the initial sample count is 150. Two batch sizes  $N_S = 5$  and  $N_S = 10$  are used for LoCoMOBO and the two baseline methods. In this experiment, the population and generation count for NSGA-II is 50 and 80 respectively. All experiments are repeated 10 times to account for random fluctuations and the results in terms of PF HV are given in Table 5.4.

We used the worst performance values from the PFs of all algorithms and runs to determine the reference point, in a similar manner as in Eq. 5.25, and utilized total biasing current instead of power consumption as indicator. This resulted in  $\mathbf{r} = [81.9, 321/V_{DD}] = [81.9, 160.5]$ . Except for MESMOC, WEIBO and NSGA-II, all algorithms take advantage of batched simulation execution. The GP models of LoCoMOBO and the baseline methods were trained using both GPU acceleration and batch GP training, and WEIBO models were trained using only GPU acceleration.

In the case of  $N_S = 5$  and  $N_{TR} = 1$ , LoCoMOBO provides a runtime gain of  $\times 10.9$  and  $\times 19.6$  compared to WEIBO and MESMOC respectively. For  $N_S = 10$ , the aforementioned speedup is by  $\times 23$  and  $\times 42$  respectively. The two-trust-region optimization

Table 5.4: Three Stage Amplifier HV Results - Two Objectives

		HV			Runtime (min)
		Mean	Best	Worst	Mean
$N_S=5$	LoCoMOBO $N_{TR}=1$	2147	2243	1928	60
	LoCoMOBO $N_{TR}=2$	2165	2206	1958	70
	Baseline-1	1481	1589	1385	61
	Baseline-2	1597	1755	1298	59
$N_S=10$	LoCoMOBO $N_{TR}=1$	2014	2158	<b>1984</b>	28
	LoCoMOBO $N_{TR}=2$	<b>2187</b>	<b>2284</b>	1975	32
	Baseline-1	1512	1754	1414	29
	Baseline-2	1661	1994	1387	28
	WEIBO	1262	1420	1136	654
	MESMOC	926	1003	859	1171
	NSGA-II	1126	1285	921	241
NSGA-III	1325	1456	1207	259	

takes  $\times 1.2$  more time to complete compared to the single-trust-region case, due to the training of additional GP models.

As shown in Table 5.4, in terms of dominated HV, it is evident that LoCoMOBO finds better solutions within the simulation budget restrictions. The two-trust-region approach provides on average the best PFs in the case of  $N_S = 10$ . LoCoMOBO consistently surpasses the examined algorithms considering different batch sizes and trust-region count.

During the experiments, we noticed that the main difficulty when sizing this topology is that the optimum solutions lie close to the infeasible region. This is the reason why the entropy-based MESMOC results in lesser HV than the rest of the methods, since it found only a small amount of feasible solutions. An interesting note is that the batched methods provide better results compared to the sequential ones. The PFs resulting from a single run per algorithm are given for demonstration in Fig. 5.12.

To demonstrate in a qualitative manner the results of the experiments, we provide a cover matrix [68] in Table 5.5. This is a matrix where its  $(i, j)$  entry represents the portion of PF solutions from the  $j$ -th algorithm that are dominated by the PF solutions of the  $i$ -th one, providing an additional metric for comparison between two different PFs. We used the PFs from the  $N_S = 10$  case to compute the cover matrix. It is seen that the PFs from LoCoMOBO dominate most of the pareto solutions resulting from the rest of the examined approaches. The results from the two-trust region optimization dominate 44% of the ones from the single trust region optimization. Among the rest

Table 5.5: Three Stage Amplifier Cover Matrix - Two Objectives

	LoCoMOBO $N_{TR}=1$	LoCoMOBO $N_{TR}=2$	WEIBO	MESMOC	Baseline-1	Baseline-2	NSGA-II	NSGA-III
LoCoMOBO $N_{TR}=1$	-	0.25	1	1	1	0.04	1	1
LoCoMOBO $N_{TR}=2$	0.44	-	1	1	1	0.08	1	1
WEIBO	0	0	-	0.75	0	0	0.32	0
MESMOC	0	0	0	-	0	0	0	0
Baseline-1	0	0	0.81	0.75	-	0	1	0.37
Baseline-2	0.23	0.17	0.63	0.75	0.78	-	1	0.63
NSGA-II	0	0	0.25	0.5	0	0	-	0
NSGA-III	0	0	0.67	0.5	0.05	0	0.32	-

of the methods, Baseline-2 provide better PFs qualitatively. It provides a dense set of PF solutions, which explains the fact that its PF is not dominated by the any other approach, but it does not yield diverse solutions like LoCoMOBO. For the single-trust-region LoCoMOBO PF of Fig. 5.12, the average, maximum and minimum typical FOM<sub>S</sub> [106] of the PF solutions are 105, 113, 99 (MHz·pF/μW). The same metrics for NSGA-II are 96, 104 and 85.

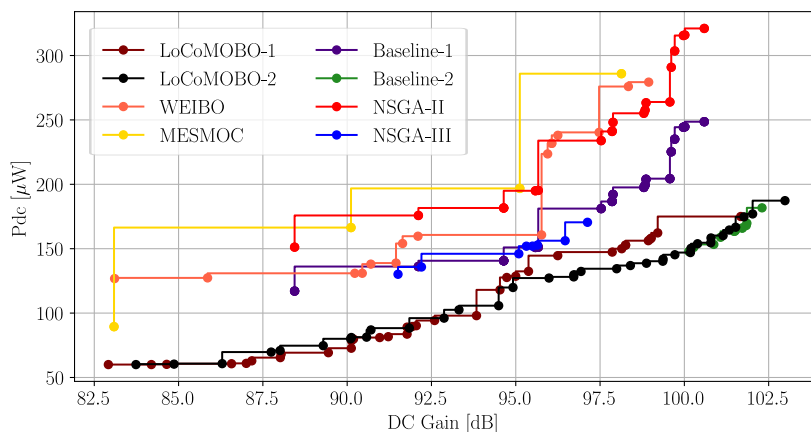


Figure 5.12: Pareto Fronts for the Three Stage Amplifier. Bottom-right values result in better DC Gain vs Pdc trade-off.

### Three-objective formulation

We now proceed with the formulation of the three-objective optimization problem. The design space remains the same, and the optimization goals and constraints are given in Table 5.6. In this case, the simulation budget for BO methods is increased to 1600

Table 5.6: Specifications for Three Stage Amplifier - Three Objectives

Performance Metrics	Specifications [106]
DCGain	maximize
$P_{dc}$	minimize
$SR_{avg}$	maximize
UGF	$\geq 0.95\text{MHz}$
PM	$\geq 52.3^\circ$
GM	$\geq 18\text{dB}$
$C_L$	$= 15\text{nF}$
$V_{DD}$	$= 2\text{V}$

Table 5.7: Three Stage Amplifier HV Results - Three Objectives

		HV			Runtime (min)
		Mean	Best	Worst	Mean
$N_S=5$	LoCoMOBO $N_{TR}=1$	7983	8104	7475	73
	LoCoMOBO $N_{TR}=2$	<b>8034</b>	<b>8164</b>	<b>7606</b>	80
	Baseline-1	5646	5881	5354	73
	Baseline-2	6046	6573	5319	71
$N_S=10$	LoCoMOBO $N_{TR}=1$	7752	8127	7254	42
	LoCoMOBO $N_{TR}=2$	7821	8095	7485	45
	Baseline-1	5366	5576	5256	41
	Baseline-2	6458	6808	5985	41
	WEIBO	4270	4786	3864	825
	MESMOC	4897	5576	4668	1528
	NSGA-II	2809	3315	2633	308
NSGA-III	4348	4779	3954	321	

and the initial sample count 150. For NSGA-II, 50 individuals and 100 generations are used. Again, all experiments were executed 10 times to account for random effects.

The results in terms of final PF HV are given in Table 5.7, with  $\mathbf{r} = [80.3, 473, 0.05]$ . LoCoMOBO outperforms the other algorithms in this case as well. By using two trust regions, we are able to get slightly better results. In this case LoCoMOBO performs better when using a smaller batch size. Among the rest of the methods, Baseline-2 results in the best solutions. The fact that LoCoMOBO consistently outperforms the global Baseline-1 highlights the advantage of the local-based approach. In terms of runtime comparison, the proposed approach is  $\times 20$  and  $\times 36$  faster compared to the sequential WEIBO and MESMOC, respectively, for  $N_S = 10$ .

The PFs resulting from the three objectives formulation ( $N_S = 10$ ) are compared

qualitatively in the cover matrix of Table 5.8.

Table 5.8: Three Stage Amplifier Cover Matrix - Three Objectives

	LoCoMOBO $N_{TR}=1$	LoCoMOBO $N_{TR}=2$	WEIBO	MESMOC	Baseline-1	Baseline-2	NSGA-II	NSGA-III
LoCoMOBO $N_{TR}=1$	-	0.21	0.87	0.85	0.86	0.79	1	0.79
LoCoMOBO $N_{TR}=2$	0.28	-	1	0.9	0.96	0.97	1	1
WEIBO	0.01	0	-	0.1	0.03	0.08	0	0.1
MESMOC	0.02	0.02	0.38	-	0.12	0.28	0.63	0.21
Baseline-1	0.02	0.01	0.66	0.54	-	0.1	0.84	0.52
Baseline-2	0.11	0.06	1	0.4	0.7	-	0.95	1
NSGA-II	0	0	0.02	0.02	0	0	-	0
NSGA-III	0	0	0.11	0.08	0.03	0	0.83	-

Here, it is seen that LoCoMOBO dominates most of the solutions of the other approaches, as expected from the HV results. In this problem, most of the algorithms found feasible solutions from the initial samples, therefore the ratio of feasible to infeasible space is quite large. This explains the fact that some approaches performed better in the three-objective case compared to the two-objective one. For demonstration purposes, an example PF resulting from LoCoMOBO ( $N_{TR} = 2$ ,  $N_S = 10$ ) is shown in Fig. 5.13.

Three Stage Amplifier PF

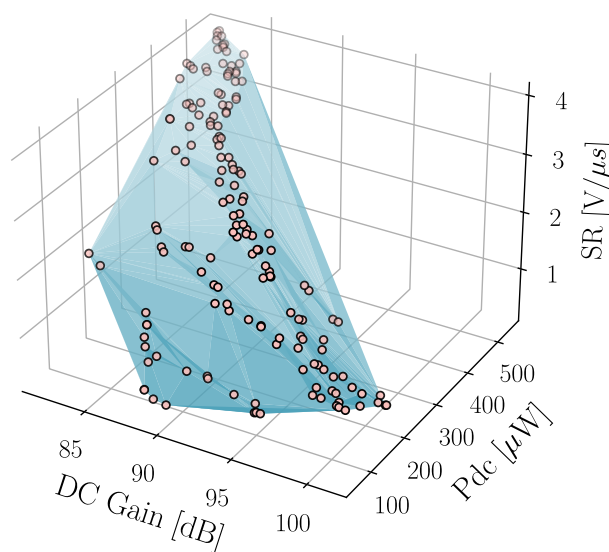


Figure 5.13: A pareto front for the Three Stage Amplifier.

## 5.2.2 Four Stage Amplifier

A Four-Stage Amplifier [116] shown in Fig. 5.14 is examined in this experiment. It can drive large capacitive loads by employing an active zero sub-circuit, a slew-rate enhancer sub-circuit and four gain stages. Similarly to the previous case, we consider two optimization formulations, one with two and one with three objectives respectively.

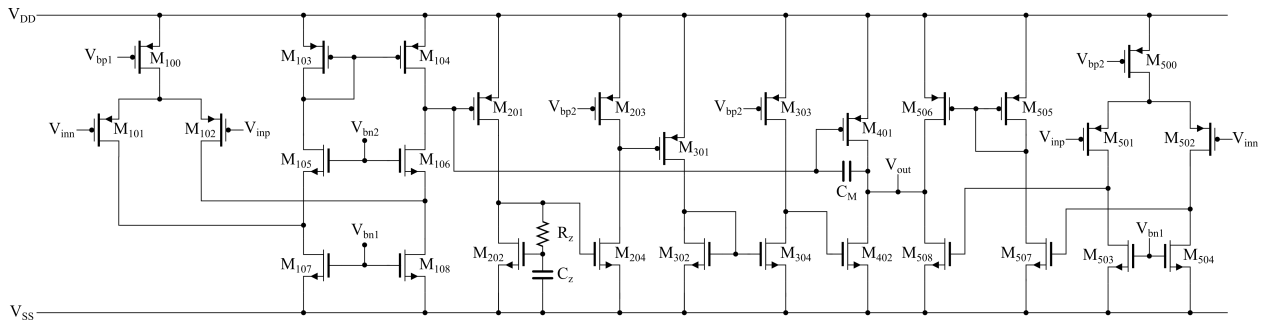


Figure 5.14: Four stage amplifier proposed in [116].

This circuit consists of the core amplifier shown in Fig. 5.14 and a biasing circuit that is responsible for voltages  $V_{bn1}$ ,  $V_{bn2}$ ,  $V_{bp1}$ ,  $V_{bp2}$  [116] and it is demonstrated in Fig. 5.15. In total, 35 transistors, 2 capacitors, a single resistor and a current source are employed. We use two testbenches, one for the slew rate and one for AC and DC analysis. The testbenches are parametrized by 43 parameters, including 20 transistor widths, 19 transistor lengths, a bias current and  $C_Z$ ,  $C_M$  and  $R_Z$ . The variable ranges are determined as follows. Transistor lengths and widths are again restricted to be less than  $1.5\mu\text{m}$  and  $100\mu\text{m}$  respectively with the lower bounds set to the minimum acceptable values of the PDK. The biasing current range is  $[0.5, 10]\mu\text{A}$  and the ranges of the resistor and the capacitors are the same as in the previous example.

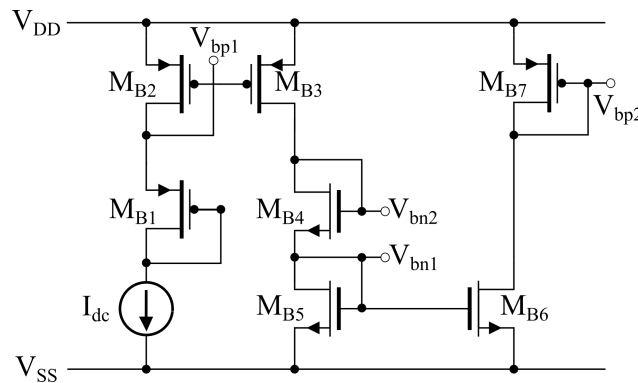


Figure 5.15: Biasing sub-circuit for the considered four stage amplifier.



Table 5.9: Specifications for Four Stage Amplifier - Two Objectives

Performance Metrics	Specifications [116]
DCGain	maximize
$P_{dc}$	minimize
$SR_{avg}$	$\geq 0.14V/\mu s$
UGF	$\geq 1.18\text{MHz}$
PM	$\geq 48.1^\circ$
GM	$\geq 8.26\text{dB}$
$V_{DD}$	$= 1.2\text{V}$
$C_L$	$= 12\text{nF}$

Table 5.10: Four Stage Amplifier HV Results - Two Objectives

		HV			Runtime (min)
		Mean	Best	Worst	Mean
$N_S=5$	LoCoMOBO $N_{TR}=1$	15204	15866	14788	67
	LoCoMOBO $N_{TR}=2$	17299	17425	<b>16897</b>	81
	Baseline-1	14019	14565	13843	65
	Baseline-2	13462	14187	12892	64
$N_S=10$	LoCoMOBO $N_{TR}=1$	15515	15725	15212	31
	LoCoMOBO $N_{TR}=2$	<b>17416</b>	<b>17854</b>	16680	44
	Baseline-1	14418	14810	13982	30
	Baseline-2	15014	15794	14424	32
	WEIBO	11943	12384	11205	754
	MESMOC	11702	11918	11399	1333
	NSGA-II	10942	12075	9547	305
NSGA-III	11639	1285	921	312	

## Two-objective formulation

The objectives and constraints for the two-objective formulation are based on the original implementation in the case of  $C_L = 12\text{nF}$  and they are given in Table 5.9. For the BO-related methods, the total simulation budget is 1300, with 150 initial sampling simulations. For NSGA-II, the population count is 60 and the number of generations 80.

In terms of HV, the optimization results are given in Table 5.10. These were computed after 10 repetitions of each experiment, with the reference point  $\mathbf{r} = [81, 645]$  determined using the worst PF values among all algorithms and executions. The gains from the two-trust-region LoCoMOBO are more evident in this higher-dimensional case. Among all methods, the two-trust-region LoCoMOBO with  $N_S = 10$  delivers the best

PF with a slight overhead in runtime ( $\times 1.4$  slower) compared to the single-trust-region LoCoMOBO.

In this problem, the single-trust-region LoCoMOBO is  $\times 24$  and  $\times 43$  faster than the sequential WEIBO and MESMOC when using  $N_S = 10$ . Therefore, LoCoMOBO provides better PFs within shorter time limits compared to the other algorithms. The PFs resulting from the optimization procedures ( $N_S = 10$ ) are shown in Fig. 5.16. LoCoMOBO  $N_{TR} = 1$  has average, maximum and minimum  $FOM_S$  105, 166, 64 (MHz·pF/ $\mu$ W) in its PF of Fig. 5.16. For comparison, NSGA-II provides with 85, 125 and 49.

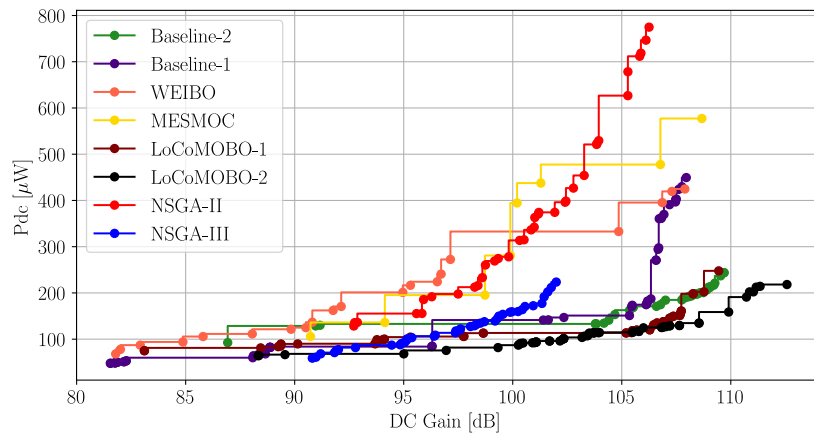


Figure 5.16: Pareto Fronts for the Four Stage Amplifier. Bottom-right values result in better DC Gain vs Pdc trade-off.

We compare the results ( $N_S = 10$ ) in terms of pareto dominance using the cover matrix shown in Table 5.11. In this case, where the variable space is larger compared to the previous example, the two-trust region LoCoMOBO dominates 88% of the PF solutions from the single trust-region one. Both proposed approaches dominate large portions of the PFs from the rest of the methods. Baseline-2 again proves favourable in comparison to Baseline-1, WEIBO, MESMOC, NSGA-II and NSGA-III.

### Three-objective formulation

The variable space remains the same in the three-objective formulation and the objectives and constraints are given in Table 5.12. Here, the total simulation budget for the BO-related methods is increased to 1600, with 150 initial sampling simulations. The NSGA-II population count and maximum generations are 60 and 100 respectively. The results in terms of final PF HV are given in Table 5.13, with  $\mathbf{r} = [75, 497, 0.027]$ .

Table 5.11: Four Stage Amplifier Cover Matrix - Two Objectives

	LoCoMOBO $N_{TR}=1$	LoCoMOBO $N_{TR}=2$	WEIBO	MESMOC	Baseline-1	Baseline-2	NSGA-II	NSGA-III
LoCoMOBO $N_{TR}=1$	-	0.12	0.95	1	0.56	0.43	1	0.68
LoCoMOBO $N_{TR}=2$	0.88	-	1	1	0.72	1	1	0.95
WEIBO	0	0	-	0.38	0.02	0	0.53	0.05
MESMOC	0	0	0.55	-	0	0	0.38	0
Baseline-1	0.28	0	0.95	0.87	-	0.1	1	0.53
Baseline-2	0.06	0	0.7	0.88	0.54	-	0.97	0.43
NSGA-II	0	0	0.4	0.25	0	0.05	-	0
NSGA-III	0.16	0.05	0.65	0.75	0.17	0.08	0.63	-

Table 5.12: Specifications for Four Stage Amplifier - Three Objectives

Performance Metrics	Specifications [116]
DCGain	maximize
$P_{dc}$	minimize
$SR_{avg}$	maximize
UGF	$\geq 1.18\text{MHz}$
PM	$\geq 48.1^\circ$
GM	$\geq 8.26\text{dB}$
$V_{DD}$	$= 1.2\text{V}$
$C_L$	$= 12\text{nF}$

The proposed approach provides higher HV values compared to those of all the other approaches. The two-trust-region LoCoMOBO again proves favorable compared to the single-trust-region one, but with higher output variance. This is highlighted by the fact that the single-trust-region LoCoMOBO provides the least-worst solution among all algorithms. We noticed that all algorithms managed to find feasible solutions with relative ease, which means that this problem is not highly constrained. This may explain the fact that WEIBO and MESMOC provide better solutions compared to their two-objective performances, but they do not manage to come close to the HV values attained by LoCoMOBO. Selecting and evaluating  $N_S = 5$  query points proves favourable in this case. We argue that this is due to the multi-modal loss landscape of this problem; GP models need to be trained more often before providing sampled functions to the acquisition function.

Table 5.13: Four Stage Amplifier HV Results - Three Objectives

			HV			Runtime (min)
			Mean	Best	Worst	Mean
$N_S=5$	LoCoMOBO	$N_{TR}=1$	4721	4895	<b>4588</b>	79
	LoCoMOBO	$N_{TR}=2$	<b>4745</b>	<b>4954</b>	4512	93
	Baseline-1		3392	3604	3184	80
	Baseline-2		3561	3813	3197	77
$N_S=10$	LoCoMOBO	$N_{TR}=1$	4321	4502	4150	47
	LoCoMOBO	$N_{TR}=2$	4707	4925	4571	63
	Baseline-1		3276	3539	3095	49
	Baseline-2		3762	3953	3413	47
	WEIBO		3728	4026	3376	951
	MESMOC		3621	3892	3450	1685
	NSGA-II		2700	3106	2503	383
NSGA-III		3185	2912	3397	396	

In terms of runtime, the single-trust-region LoCoMOBO with  $N_S = 5$  provides an overall speedup of  $\times 12$  and  $\times 21$  compared to WEIBO and MESMOC respectively. These quantities become  $\times 20$  and  $\times 35$  when the batch size is  $N_S = 10$ . A qualitative comparison of the resulting PFs from the tested methods ( $N_S = 10$ ) is given in the cover matrix of Table 5.14.

Table 5.14: Four Stage Amplifier Cover Matrix - Three Objectives

	LoCoMOBO $N_{TR}=1$	LoCoMOBO $N_{TR}=2$	WEIBO	MESMOC	Baseline-1	Baseline-2	NSGA-II	NSGA-III
LoCoMOBO $N_{TR}=1$	-	0.33	0.53	0.69	0.61	0.42	0.75	0.58
LoCoMOBO $N_{TR}=2$	0.57	-	0.64	0.77	0.81	0.68	0.91	0.76
WEIBO	0.21	0.02	-	0.15	0.26	0.24	0.36	0.06
MESMOC	0.07	0.03	0.2	-	0.25	0.22	0.32	0.07
Baseline-1	0.1	0.07	0.25	0.6	-	0.1	0.53	0.28
Baseline-2	0.17	0.02	0.5	0.18	0.31	-	0.67	0.48
NSGA-II	0.01	0	0.07	0.04	0.06	0.02	-	0
NSGA-III	0.08	0.05	0.17	0.24	0.13	0.11	0.25	-

The PFs of LoCoMOBO dominate a large portion of the PFs of the other algorithms. In addition, the two-trust-region case seems to provide with wider PFs in comparison to

the single-trust-region one. While it provides a relatively large PF HV difference, it does not dominate a large portion of the single-trust-region PF. The HV difference, therefore, can be attributed to more diverse PF. An example PF for this circuit ( $N_{TR} = 2$ ,  $N_S = 10$ ) is given in Fig. 5.17.

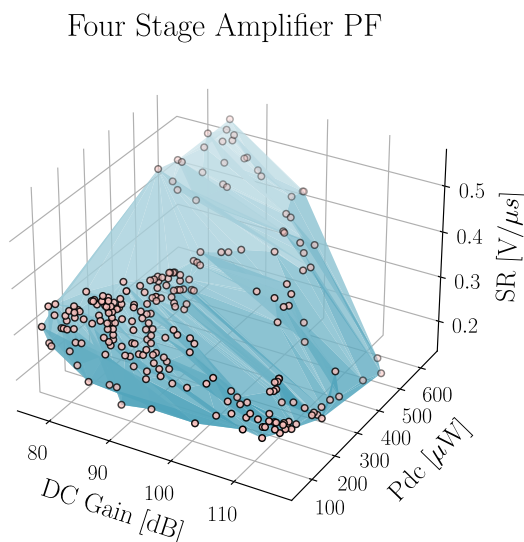


Figure 5.17: A pareto front for the Four Stage Amplifier.

### 5.2.3 Low Noise Amplifier

A Low Noise Amplifier is sized in this subsection. The circuit is shown in Fig. 5.18 and it is an inductively degenerated common-source cascode topology.

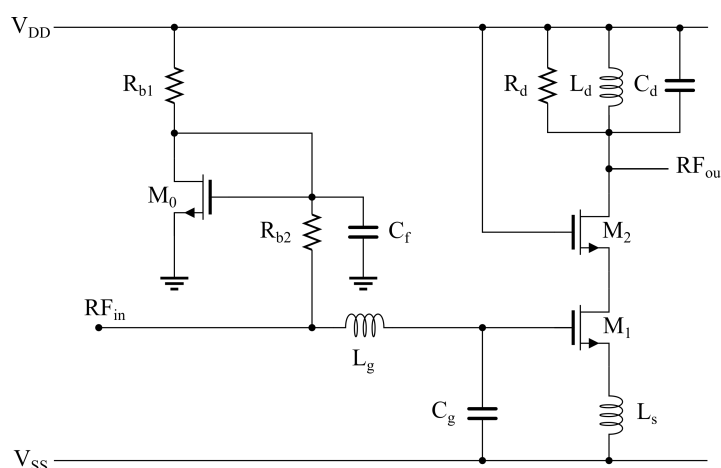


Figure 5.18: Low Noise Amplifier examined in this subsection.

The LNA consists of three nMOS transistors (one current source), three inductors, three capacitors and two resistors. All of the employed inductors are spiral ones and

Table 5.15: Specifications for Low Noise Amplifier

Performance Metrics	Testbenches	Specifications
$S_{21}@2.4\text{GHz}$	nominal	maximize
$P_{dc}$	nominal	minimize
$S_{21}@2.4\text{GHz}$	nominal	$\geq 20\text{dB}$
$S_{11}@2.4\text{GHz}$	all	$\leq -8\text{dB}$
$S_{22}@2.4\text{GHz}$	all	$\leq -8\text{dB}$
IIP3	all	$\geq -5\text{dBm}$
NF @2.4GHz	all	$\leq 2.5\text{dB}$

capacitors  $C_g$ ,  $C_d$ ,  $C_f$  are Metal-Oxide-Metal (MOM) ones. Devices  $R_{b1}$ ,  $R_{b2}$  are p-diffusion resistors, while  $R_d$  is fixed to  $100\Omega$ . In total, there are 21 independent variables and their ranges are as follows. Inductor widths are chosen between the available values by the PDK ( $3\mu\text{m}, 6\mu\text{m}, 9\mu\text{m}, 15\mu\text{m}$ ), while the number of turns ranges between 0.5 and 5 with a step of 0.25 and the inner radius between  $10\mu\text{m}$  and  $100\mu\text{m}$ . Transistor length and width ranges are  $[100, 240]\text{nm}$  and  $[5, 250]\mu\text{m}$ . The total length of the resistors ranges between  $1\mu\text{m}$  and  $30\mu\text{m}$ , while their width is fixed to  $2\mu\text{m}$ . For the capacitors, the fingers width is fixed to  $140\text{nm}$  and the number of horizontal and vertical fingers range between 1 and 100.

This problem involves both continuous and integer-valued variables. To handle this variable space, we adopt the methodology proposed in [117] and extend LoCoMOBO to handle mixed-integer variables. To this end, prior to the evaluation of any sampled function via RFF when solving Eq. 5.26, the input vector  $\mathbf{x}$  is replaced by a transformed vector  $T(\mathbf{x})$ , where  $T(\cdot)$  rounds the input entries that correspond to integer variables to the closest integer.

In this experiment we consider optimizing for power dissipation and Gain ( $S_{21}$ ) with 2.4GHz operating frequency. To account for PVT variations, we consider 45 separate testbenches, combining 5 different corner model-files ( typical, SS, SF, FS, FF), three operating temperatures (-50, 27 and 125 Celsius) and three supply voltages (typical, typical $\times 1.1$ , typical $\times 0.9$ ). The specifications for the LNA, which include optimization objectives and constraints, are given in detail in Table 5.15. In summary, the constraints are chosen to ensure that a 20dB Gain is achieved in typical conditions, while linearity, input matching and noise performances at all corners meet the chosen thresholds.

In this example we compare the proposed LoCoMOBO approach using a single and two trust regions against Baseline-1, NSGA-II and NSGA-III. The batch size for BO approaches is  $N_S = 5$ . WEIBO and MESMOC were not proposed to handle problems with mixed integer and discrete variables and Baseline-2 cannot be extended to do

Table 5.16: Low Noise Amplifier HV results

		HV			Runtime (min)
		Mean	Best	Worst	Mean
$N_S=5$	LoCoMOBO $N_{TR}=1$	44.2	45.3	40.8	144
	LoCoMOBO $N_{TR}=2$	<b>45.1</b>	<b>46.2</b>	<b>43.6</b>	152
	Baseline-1	39.1	42.2	35.8	139
	NSGA-II	22.9	27.2	18.1	1586
	NSGA-III	27.1	35.4	21.8	1590

so, thus they are not included in this test-case. For LoCoMOBO and Baseline-1, the initial samples are 150 and the maximum number of evaluations 1300. For NSGA-II and NSGA-III 50 individuals and 80 generations were used, with 24 reference directions. Experiments are repeated five times to account for random fluctuations and the results in terms of HV are given in Table 5.16, where  $\mathbf{r}=[20.06, 30]$ , computed from the minimum Gain and maximum DC current encountered from all of the algorithms.

It seen that LoCoMOBO outperforms the population-based NSGA-II and NSGA-III in this example as well. The global Baseline-1 performs better than the population-based algorithms but it does not reach the performance of the proposed approach using a single or two trust regions. A cover matrix for this example is shown in Table 5.17. Here the results from the HV-based comparison are validated, demonstrating the effectiveness of the proposed approach. For demonstration purposes, the resulting PFs from a single run of each algorithm are shown in Fig. 5.19. For the case of LoCoMOBO with a single trust region, the average, maximum and minimum LNA FOM [118] from the PF solutions are 17485, 21060, 13069  $W^{-1}$ . For comparison, the same metrics for NSGA-II are 13884, 16233 and 9881.

Table 5.17: Low Noise Amplifier Cover Matrix

	LoCoMOBO $N_{TR}=1$	LoCoMOBO $N_{TR}=2$	Baseline-1	NSGA-II	NSGA-III
LoCoMOBO $N_{TR}=1$	-	0.1	0.69	1	1
LoCoMOBO $N_{TR}=2$	0.38	-	0.69	1	1
Baseline-1	0.04	0.06	-	0.71	0.69
NSGA-II	0	0	0	-	0.16
NSGA-III	0	0	0	0.84	-

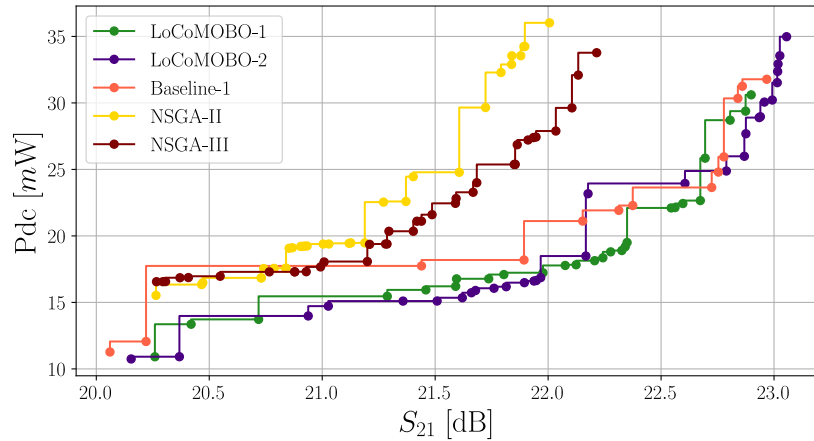


Figure 5.19: Pareto Fronts for the Low Noise Amplifier. Bottom-right values result in better  $S_{21}$  vs  $P_{dc}$  trade-off.

To highlight the effect of including PVT variations in the optimization procedure, we proceed with a comparison between variation-aware and nominal optimization results. The above experiment is repeated once more, but this time accounting only for nominal conditions, therefore we utilize a single testbench. The variable ranges as well as the algorithmic hyperparameters remain the same and the optimization constraints are changed accordingly.

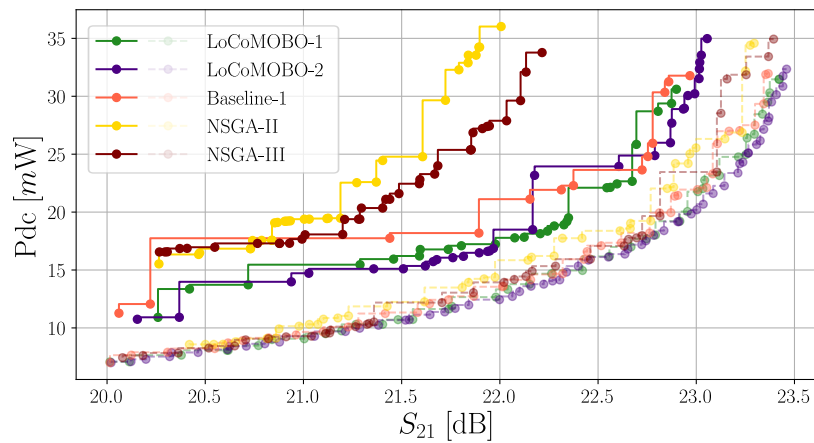


Figure 5.20: A comparison between the Pareto Fronts derived accounting for PVT variations and only for nominal conditions. Nominal sizing results are depicted using light dotted lines and PVT-aware results using solid lines.

In Fig. 5.20, the PFs from the nominal optimization and the ones resulting from the corner based one are superimposed. Two major conclusions can be drawn from this figure: 1) LoCoMOBO outperforms the other algorithms considering both nominal conditions and PVT variations and 2) the PFs from the variation-aware sizing provide worse  $S_{21}$  versus  $P_{DC}$  trade-off, which is expected due to the additional constraints that



ought to be satisfied. This in turn implies that the nominal PFs do not satisfy the constraints of Table 5.15 in all corners.

### 5.3 Summary & Concluding Remarks

This chapter presented a novel BO approach for MOPs in high-dimensional search spaces. The proposed LoCoMOBO framework makes use of the following tools:

- Local Trust Regions, that reduce the excessive exploration associated with high-predictive uncertainties of GP models in high-dimensional spaces. The framework for local BO permits the use of multiple trust regions in parallel, which are adapted according to the rate in which better solutions are found. Continuous improvements upon the objective result in the enlargement of the trust region.
- Random Fourier Features, which is a method to sample approximate but analytic functions from the posterior distributions of GP models. The availability of analytic functions permits the use of auxiliary optimization algorithms and boosts of the algorithm's efficiency, in comparison to quantization techniques.
- A greedy, hypervolume-based acquisition function, which is based on TS and the sampled RFF-functions to select *multiple* query points at each iteration. The selection accounts for constraints and it is able to distinguish between candidate solutions based on their pareto optimality.

The algorithm was extensively tested both using benchmark MOPs and real-world circuit design space exploration applications. In comparison to population based approaches, LoCoMOBO is able to provide better empirical Pareto fronts with only a fraction of the evaluations. In comparison with other BO approaches, LoCoMOBO is able to deliver both runtime gains, mainly due to its parallel nature, and due to its ability to handle high dimensional and constrained MOPs. LoCoMOBO was also applied to a variation-aware design exploration problem, where it outperformed other baselines in a highly constrained MOP, where 45 testbenches were used per simulation.



# Chapter 6

## Device Representation Learning

### 6.1 Motivation

In the previous chapters, we have seen the advantages that BO offers for analog circuit sizing and performance space exploration. BO’s core functionality revolves around modeling the loss landscape using GPs [51] as surrogate models. GPs, however, operate only on continuous spaces [51] and subsequently hinder the application of BO in problems that include discrete variables. In practice, many real-world circuits include devices such as integrated inductors, whose geometry is parametrized by both discrete and continuous variables. A partial remedy to this situation is allowing GPs to operate in a continuous space and round to the closest integer the values that are selected as query points. This, however, may lead to systematic biasing towards some discrete values on top of others. An alternative approach, where no brute-force rounding takes place is much needed.

Recently, the topic of *latent space optimization* has emerged in the field of ML as a promising approach to solving optimization problems that are hard to formulate, either because of the large dimensionality of the search space or due to the representation of the input one. This approach involves a latent variable generative model  $G : \mathcal{Z} \rightarrow \mathcal{X}$  that maps from vectors from a space  $\mathcal{Z}$  to the actual search space of the optimization  $\mathcal{X}$ . By choosing  $\mathcal{Z}$  to be of low dimensionality and continuous and using it as the search space, one can map the initial optimization problem into a new, low-dimensional and continuous one.

Latent space optimization has been proven very useful in the cases of complex input spaces, such as graph-based ones or string-based ones. For instance, in [119, 120], continuous representations of molecules, which are represented in a string-based format called SMILES [119] are derived. The space on which the representations reside is used as an auxiliary search space for optimization algorithms and chemical property predictor models. In [121] a generative model is used to learn latent codes of simulated robot trajectories stemming from various controllers, which are then used to define an optimization objective targeting constrained controller design. Further works in latent space optimization include [122, 123].

By using a generative model to map the mixed-variable geometry space of devices into a continuous one, we define a space on which the GPs can operate efficiently, rendering BO feasible. Here, the sample correlations given by the GP’s kernel function and the GP’s mean function operate on the latent space  $\mathcal{Z}$ , i.e. a GP is defined as  $f(\mathbf{z}) \sim \mathcal{GP}(m(\mathbf{z}), k(\mathbf{z}, \mathbf{z}'))$ . In practice, since only a subset of the initial parameters will accept discrete-values, we can exchange only these variables with latent codes.

Motivated by the above, in this chapter we present an approach for learning continuous, data-driven representations of integrated devices. We use a Variational Autoencoder [124] as the generative model for learning the representations and, to ensure that the latent space is well structured, we embed a label-guiding supervised network that maps latent codes to device geometric sizes. More precisely, instead of training the generative network to map geometric sizes to latent codes, we embed domain-specific simulation data to  $\mathcal{Z}$ . The latent codes are used as inputs in a predictor neural net, which is trained simultaneously with the generative model to predict device sizes. During optimization, the search is conducted on the latent space, and the geometric features are acquired by using a predictor model that maps latent codes to actual geometries.

## 6.2 Latent Space Optimization

### 6.2.1 Autoencoders

In the following, the concept of Variational Autoencoders (VAEs), the deep generative model which is used to create continuous-valued representations of device models, is discussed. In order to better understand the concept of VAEs, first we will introduce the simpler, non-generative model of Autoencoders (AEs).

An AE is a particular type of neural network that aims to reproduce its input in its output [39]. Its usefulness stems from the fact that it has an internal layer  $\mathbf{h}$  which serves as a *code* or *representation* of the input vector  $\mathbf{x} \in \mathbb{R}^N$ . By being trained to learn the identity function, an AE often results in useful representations for input data, which may be used in downstream tasks. AEs consist of two separate neural network blocks, namely:

- An **Encoder**  $\mathbf{h} = f_{\theta}(\mathbf{x})$ , which is tasked to learn the representation  $\mathbf{h} \in \mathbb{R}^M$  of the input vector  $\mathbf{x}$ , and,
- a **Decoder**  $\mathbf{r} = g_{\phi}(\mathbf{h})$ , which maps the encoded representation of the Encoder  $\mathbf{h}$  back to the original space  $\mathbf{r} \in \mathbb{R}^N$ .

Both Encoder and Decoder networks are parametrized by variable sets  $\theta, \phi$ .

AEs are trained in an unsupervised manner, i.e. they do not require the existence of labels to map the inputs to. Since their goal is to minimize the distance between their inputs and outputs, their objective function is the sum of squared differences between the inputs  $\mathbf{x}$  and the reconstructions  $\mathbf{r}$ . Given a dataset of  $k$  real-valued input vectors  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k)}$ , the loss is given by

$$\mathcal{L}_{AE} = \frac{1}{k} \sum_{i=1}^k (\mathbf{x}^{(i)} - f_{\theta}(g_{\phi}(\mathbf{x}^{(i)})))^2, \quad (6.1)$$

where parameter vectors  $\phi$  and  $\theta$  are learned simultaneously by minimizing  $\mathcal{L}_{AE}$ .

As already mentioned, the most important aspect of AEs is the useful representations that are created by the encoder network. The most prominent approach to construct useful representations is to enforce that their dimensionality is lower than the input's one, i.e.  $M < N$ . In this case, the AE is *undercomplete* [39] and it is forced to learn salient and sparse features from the provided inputs [39]. In essence, using the loss term in Eq. 6.1 and linear nets as decoders, an undercomplete Autoencoder learns the same principal subspace with that of a Principal Component Analysis (PCA) [96]. The incorporation of nonlinearities in the AE's nets generalizes PCA and leads to a more powerful, data-driven nonlinear dimensionality reduction approach[39].

## 6.2.2 Variational Autoencoders

VAEs are generative models, i.e. provided a dataset  $\mathcal{D} = \{\mathbf{x}_i \in \mathbb{X}\}_{i=1}^N$  they can produce synthetic samples that follow approximately the distribution of the inputs  $\{\mathbf{x}_i\}_{i=1}^N$ . Structurally, they can be thought of as a probabilistic extension of AEs, since they rely on an encoder-decoder architecture. Before moving on to the implementation of the VAE networks, we will discuss its foundations from a probabilistic viewpoint.

VAEs fall into the category of latent variable models [96]. These assume that, the -potentially complex- distribution of  $\mathcal{D}$  which we wish to approximate, can be better explained using some *hidden* or *latent* variables  $\mathcal{Z} = \{\mathbf{z}_i\}_{i=1}^N$ . By defining a joint distribution over the observed variables in  $\mathbf{x}$  and the unobserved latent variables  $\mathbf{z}$ , one can acquire the distribution of the observable ones  $p_{\theta}(\mathbf{x})$  by marginalization:

$$p_{\theta}(\mathbf{x}) = \int p_{\theta}(\mathbf{x}, \mathbf{z}) d\mathbf{z}. \quad (6.2)$$

Note that the distributions in the above expression stem from a flexible parametric model  $p_{\theta}$ , with parameters  $\theta$ . Therefore,  $p_{\theta}(\mathbf{x})$  can be expressed using simpler distributions as components.

The relation between the observable and the latent variables of the VAE framework can be expressed using the notion of directed graphical models [96]. Directed graphical models are acyclic graphs which express the dependencies of random variables using nodes and links. The joint distribution of all the variables of directed graphical models is given by

$$p_{\theta}(\mathbf{x}_1, \dots, \mathbf{x}_N) = \prod_{i=1}^N p_{\theta}(\mathbf{x}_i | T(\mathbf{x}_i)), \quad (6.3)$$

with  $T(\mathbf{x}_i)$  being the parent variables, i.e. previous nodes of  $\mathbf{x}_i$ . Therefore, variables with no parents have unconditional distributions whereas the rest of the random variables are conditioned on their parents. The VAE latent variable model follows the same principle by defining the joint distribution of the observable and latent variables as

$$p_{\theta}(\mathbf{x}, \mathbf{z}) = p_{\theta}(\mathbf{x} | \mathbf{z}) p_{\theta}(\mathbf{z}). \quad (6.4)$$

Here,  $p_{\theta}(\mathbf{z})$  is the prior distribution and controls the behavior of the latent variables. The model's likelihood  $p_{\theta}(\mathbf{x} | \mathbf{z})$  describes the mapping from latent variables to observable ones. In addition,  $p_{\theta}(\mathbf{z} | \mathbf{x})$  is termed the posterior distribution of the model.

Given the above, the generational procedure of the VAE consists of two steps:

1. Sampling a latent variable vector  $\mathbf{z}_i$  from the prior distribution  $p(\mathbf{z})$ , and,
2. using the likelihood  $p_{\theta}(\mathbf{x} | \mathbf{z} = \mathbf{z}_i)$  to generate an observable vector.

The inference process of a latent variable model involves determining the latent variable value given an input data point  $\mathbf{x}$ , and it is formulated by the posterior distribution  $p_{\theta}(\mathbf{z} | \mathbf{x})$ .

Using maximum likelihood estimation to learn the above model, i.e. determine the model's parameters  $\theta_{opt}$ , necessitates the maximization of the likelihood of each and every data point in  $\mathcal{D}$ , which is formulated as an optimization problem:

$$\theta_{opt} = \operatorname{argmax}_{\theta} \prod_{i=1}^N p_{\theta}(\mathbf{x}_i). \quad (6.5)$$

This expression can be simplified by using the sum of the log probabilities. Typically, Eq. 6.5 cannot be solved analytically and gradient based methods, such as Stochastic Gradient Descent (SGD) [125] are used. This, however, requires the evaluation and differentiation of the marginal likelihood  $p_{\theta}(\mathbf{x})$  with respect to  $\theta$ . In the case of simple nonlinear likelihood functions  $p_{\theta}(\mathbf{x} | \mathbf{z})$ , such as neural nets, the integral in Eq. 6.2 becomes intractable and it is not amenable to differentiation [126]. Instead of brute-force



Figure 6.1: Directed graphical models for (a) the approximate posterior  $q_\phi(\mathbf{z}|\mathbf{x})$  and (b) the probabilistic decoder  $p_\theta(\mathbf{x}|\mathbf{z})$ . The coloured node indicates the observable random variable and the nodes outside the box are model parameters.

optimization, one can resort to the Expectation-Maximization (EM) algorithm [96] to find a maximum likelihood estimation of the model parameters. The EM algorithm assumes an initial parameter vector  $\theta_{init}$  and proceeds by iterating over two steps; the expectation one where the posterior  $p_{\theta_{init}}(\mathbf{z}|\mathbf{x})$  is evaluated and the maximization one where an updated set of parameters  $\theta_{new}$  is computed by maximizing the expectation of the likelihood  $p_{\theta_{init}}(\mathbf{z}, \mathbf{x})$  over the previously computed posterior. Considering Bayes rule [96], it holds that

$$p_\theta(\mathbf{z}|\mathbf{x}) = \frac{p_\theta(\mathbf{x}, \mathbf{z})}{p_\theta(\mathbf{x})}, \quad (6.6)$$

and given the fact that the joint distribution is easy to evaluate, the posterior  $p_\theta(\mathbf{z}|\mathbf{x})$  is also intractable [126]. This effectively renders the EM algorithm useless for the case of intractable functions as well.

To circumvent the intractability of the posterior  $p_\theta(\mathbf{z}|\mathbf{x})$ , VAEs build a parametric approximation  $q_\phi(\mathbf{z}|\mathbf{x})$ . This approximate posterior is also called encoder or recognition model and its parameters  $\phi$  are called variational parameters. The term variational refers to the approximation of a distribution by optimizing a parametric model. In the VAEs' case,  $q_\phi(\mathbf{z}|\mathbf{x})$  is parametrized by deep neural nets and the variational parameters are the weights and the biases of the networks. Therefore, the generational and the inference process of the VAE can be expressed by the graphical models shown in Fig. 6.1. The most prominent parametrization of the approximate posterior using a neural network  $f: \mathbb{R}^N \rightarrow \mathbb{R}^{2 \times M}$  is formulated as follows:

$$\begin{aligned} (\boldsymbol{\mu}, \log(\boldsymbol{\sigma})) &= f_\phi(\mathbf{x}), \\ q_\phi(\mathbf{z}|\mathbf{x}) &= \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma})). \end{aligned} \quad (6.7)$$

To define a cost function for training the latent variable model, we proceed with

analyzing the log-likelihood of the data as

$$\begin{aligned}
\log p_{\theta}(\mathbf{x}) &= \int q_{\phi}(\mathbf{z}|\mathbf{x}) \log p_{\theta}(\mathbf{x}) d\mathbf{z} \\
&= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left[ \log p_{\theta}(\mathbf{x}) \right] \\
&= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left[ \log \left[ \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{p_{\theta}(\mathbf{z}|\mathbf{x})} \right] \right] \\
&= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left[ \log \left[ \frac{p_{\theta}(\mathbf{x}, \mathbf{z}) q_{\phi}(\mathbf{z}|\mathbf{x})}{p_{\theta}(\mathbf{z}|\mathbf{x}) q_{\phi}(\mathbf{z}|\mathbf{x})} \right] \right] \\
&= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left[ \log \left[ \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right] + \log \left[ \frac{q_{\phi}(\mathbf{z}|\mathbf{x})}{p_{\theta}(\mathbf{z}|\mathbf{x})} \right] \right] \\
&= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left[ \log \left[ \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right] \right] + \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left[ \log \left[ \frac{q_{\phi}(\mathbf{z}|\mathbf{x})}{p_{\theta}(\mathbf{z}|\mathbf{x})} \right] \right] \\
&= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left[ \log \left[ \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right] \right] + D_{KL} \left( q_{\phi}(\mathbf{z}|\mathbf{x}) \parallel p_{\theta}(\mathbf{z}|\mathbf{x}) \right).
\end{aligned} \tag{6.8}$$

The second term in the right hand side of the Eq. 6.8 is the Kullback–Leibler (KL) divergence between the approximate and the true posterior. It is used as a measure for the similarity between the aforementioned distributions and it is non-negative, with values closer to zero indicating greater similarity. The first term in the right hand side of Eq. 6.8 is called the evidence lower bound (ELBO) [126]. Since KL divergence is non-negative, ELBO is always lower than the log-likelihood of the dataset and it serves as a lower bound for  $\log p_{\theta}(\mathbf{x})$ .

By solving for ELBO in Eq. 6.8, it is seen that its maximization induces both the maximization of the log-likelihood  $\log p_{\theta}(\mathbf{x})$  ( and therefore maximization of the marginal likelihood), as well as the minimization of the KL divergence between the approximate and the true posterior. Therefore, maximizing ELBO with respect to both  $\theta, \phi$  results both in greater reconstruction likelihood, i.e. greater generative capabilities, and in learning a better approximation to the true posterior  $p_{\theta}(\mathbf{z}|\mathbf{x})$ . Given that  $p_{\theta}(\mathbf{x}, \mathbf{z}) = p_{\theta}(\mathbf{x}|\mathbf{z})p_{\theta}(\mathbf{z})$  and using the logarithm’s properties, the expression for ELBO in Eq. 6.8 can be written as:

$$\mathcal{L}_{VAE}(\phi, \theta) = \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left[ \log p_{\theta}(\mathbf{x}|\mathbf{z}) \right] - D_{KL} \left( q_{\phi}(\mathbf{z}|\mathbf{x}) \parallel p_{\theta}(\mathbf{z}) \right). \tag{6.9}$$

Parameters  $\theta$  and  $\phi$  are used to parametrize the posterior and likelihood functions. These, in turn, are implemented using neural nets. As already mentioned, the approximate posterior  $q_{\phi}(\mathbf{z}|\mathbf{x})$  is implemented by an encoder net, which takes as inputs vectors



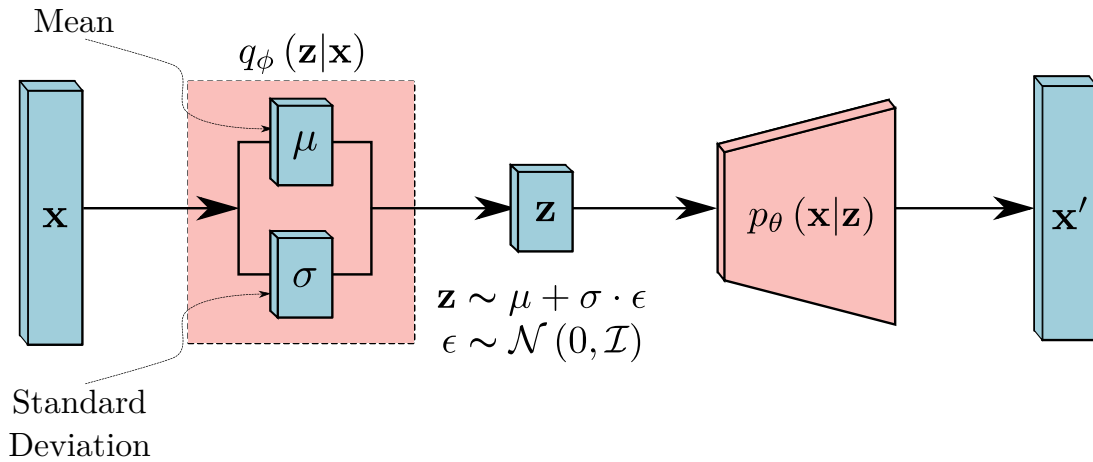


Figure 6.2: An illustration of the VAE model, along with its constituent parts.

from  $\mathcal{D}$  and provides their latent representation, whereas the likelihood  $p_\theta(\mathbf{x}|\mathbf{z})$  is implemented via a decoder neural net, with parameters  $\phi$ . A depiction of the encoder-decoder VAE architecture is given in Fig. 6.2. Since neural nets are deterministic models, the approximate posterior  $q_\phi(\mathbf{z}|\mathbf{x})$  is implemented as in Eq. 6.7, i.e. as a multivariate Gaussian distribution with diagonal covariance matrix with learned mean and covariance functions. Likelihood  $p_\theta(\mathbf{x}|\mathbf{z})$ , on the other hand, is implemented as a parametrized Bernoulli or diagonal-covariance Gaussian, based on whether the data in  $\mathcal{D}$  are binary or real-valued. The most popular choice for the prior is an isotropic, unit-variance multivariate Gaussian distribution with zero mean, i.e.  $p_\theta(\mathbf{z}) = \mathcal{N}(0, \mathbb{I}_M)$ . This prior results in an analytic expression for the KL divergence term [124] in Eq. 6.9 as

$$D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})) = \frac{1}{2} \sum_{i=1}^M \left[ \mu_i^2 + \sigma_i^2 - \log(\sigma_i^2) - 1 \right], \quad (6.10)$$

with  $\boldsymbol{\mu} = [\mu_i]_{i=1}^M$  and  $\boldsymbol{\sigma} = [\sigma_i]_{i=1}^M$  being the outputs of the encoder network.

### 6.2.3 Proposed Approach

Based on the concept of latent space optimization, we use a data-driven approach to capture the data distribution of integrated devices' geometry, and produce a continuous representation out of it. Although the simpler AE model does produce a compressed representation of the input dataset, VAEs are preferable for the following reason; the loss function of AEs includes solely the reconstruction loss term, thereby allowing the encoder to map input vectors in arbitrary positions in the latent space. This almost always results in empty regions in the latent space, where input vectors' latent codes are clustered and the space between them is empty. In the case where a latent code that

resides in these empty spaces is queried, by an optimization algorithm for instance, the reconstruction may not be a valid one, i.e. it will not correspond to input data ranges, etc.

On the other side, VAEs, with the additional KL divergence term in their loss function, are trained both to produce samples from the input distribution and to enforce a constraint in the latent space; since the posterior distribution is enforced to resemble a zero mean, unit variance isotropic gaussian one, most of the latent codes will be placed in proximity to the origin. In this case, a logical choice for latent variable ranges in an optimization setting is the hypercube  $[-3, 3]^M$  with  $M$  being the dimensionality of the latent space. This translates to a 3-sigma interval, where 97.7% of all samples from a unit variance gaussian distribution reside, i.e. it includes almost all latent codes produced by the dataset.

In this setting, the structure of the VAE’s latent space is of utter importance. In fact, the VAE must learn to map input vectors that yield similar outputs closeby to the latent space, such that the optimization algorithm’s exploration is not deteriorated. In the case of analog circuit sizing optimization problems, learning a VAE model directly from input (device sizes) and output (performance metrics) data is not practical. This is because the model will not be able to generalize to new, unseen topologies and processes. Also, the process of learning such a model involves simulations that could otherwise be used to optimize the circuit in the first place. In contrast, our approach to latent space optimization involves learning multiple VAE models, each one for a particular PDK model of an integrated device. Thus, devices that are parametrized by discrete variables can be represented by continuous valued ones, under a trained VAE model. These continuous representations, then, can be used in the optimization setting, replacing the original discrete variables, and rendering the problem continuous. An illustration of the proposed procedure is shown in Fig. 6.3.

A reasonable question that may arise is what data are fed to the VAE, in order to produce a latent space. Since our goal is to learn continuous representations of integrated devices, and, provided that the VAE learns a manifold of the input data in such a manner that ‘similar’ inputs are mapped to ‘close-by’ latent codes, we should use data that enable such similarity comparison to take place. An i.i.d. sampling of the structured combinatorial design space of a device can be used define similarity via the Euclidean distance, but this may not be useful in real-life cases, where a slight modification of a device’s geometry yields completely different behavior. For this reason, we choose to use simulation data, tailored to each specific device. We argue that by using current-voltage characteristics or frequency responses, over some predefined ranges, functional similarity between integrated devices can be implicitly captured by the VAE model.

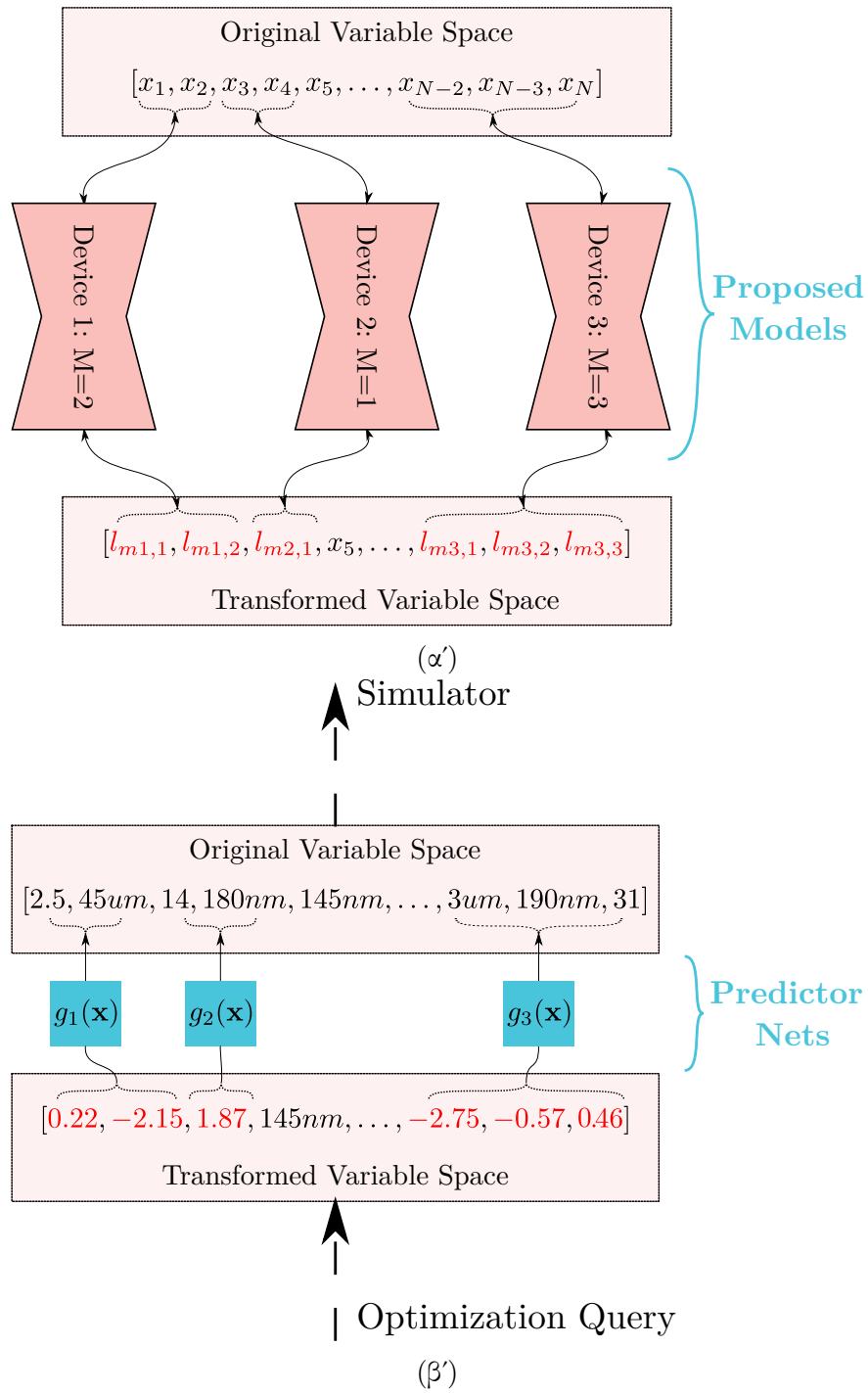


Figure 6.3: The proposed device representation learning scheme within a sizing problem, where the latent variables are shown in red. **a)** The original variable space of the sizing problem is changed into a transformed one, by mapping sets of variables that belong to devices into continuous latent ones. **b)** During the optimization, a query from the optimization algorithm is transformed back to the original variable space using the predictor networks, prior to simulation.

In the following section, for example, we use frequency response data to learn a latent space representation for spiral inductors.

Since we choose to learn a representation of an integrated device using simulation data, we must incorporate the geometric characteristics of the devices in the model as well, otherwise there will be no way to embed them in an optimization procedure. To this end, besides the unsupervised VAE model, our approach also uses a predictor model  $g : \mathbb{Z} \rightarrow \mathbb{R}$ , which maps latent codes to actual geometries. The predictor model is trained simultaneously with the VAE and its parameters are learned so as to minimize the deviation between the actual and predicted geometries. Therefore, one can use the latent space in an optimization formulation and by passing latent code queries to the predictor, they can have valid geometries to pass to the simulator. This operation is illustrated graphically in Fig. 6.3. The inclusion of the predictor network has an additional effect on the latent space structure; it enforces an ordering of the latent codes, such that they have clear gradients with respect to geometrical characteristics.

The overall class of the proposed models for device representation learning are therefore, composed of a VAE part and a predictor network. The simultaneous training of these parts requires the definition of a composite loss function, as well. By denoting the VAE loss in Eq. 6.9 as  $\mathcal{L}_{VAE}(\phi, \theta)$ , and introducing the supervised loss of the predictor as  $\mathcal{L}_{sup}(\phi, \lambda)$ , where  $\lambda$  are the parameters of the predictor network, the overall loss function is their summation, i.e.

$$\mathcal{L}_{overall} = \mathcal{L}_{VAE}(\phi, \theta) + \mathcal{L}_{sup}(\phi, \lambda). \quad (6.11)$$

The supervised loss induces changes in the encoder network, and its actual formulation depends on the device characteristics.

In real applications, only a subset of the circuit variables  $[x_1, x_2, \dots, x_d]$  are discrete, or belong to the class of devices for which we learn continuous representations. This results in the fact that a new search space will be used in the optimization, where both original, continuous-valued geometric parameters and latent variables reside. Without loss of generality, let us assume that only variable  $x_1$  is discrete and that the rest of the variables reside in a space  $\mathbb{S}$ . Let us also assume that the latent variable has  $M = 1$  dimension. Then, the new search space for the optimization is defined as  $\mathbb{D} = \mathbb{S} \times [-3, 3]$ , where  $[-3, 3]$  is the space where the latent variable that substitutes  $x_1$  resides. As the optimizer queries points from  $\mathcal{D}$ , these are processed prior to evaluation to determine whether they are latent ones, associated with a particular model. In this case, the single latent variable is identified and passed through the predictor model of the associated composite model, which yields a valid geometry for the simulator.

### 6.2.4 Applications on Integrated Devices

In this subsection we apply the continuous representation learning technique to integrated spiral inductors and rotative metal capacitors. We use a TSMC 90nm process and the 'spiral std' model for 3-terminal octagonal, symmetric spiral inductors, while the 'crtmom' model is used for the metal capacitor.

#### Spiral Inductor

The geometry of the spiral octagonal inductor considered in this case is parametrized by 3 variables, namely inner radius, inductor width and number of turns<sup>1</sup>. Based on the PDK's acceptable inputs, the inner radius variable is continuous in the range  $[15, 90]\mu\text{m}$ , whereas the inductor width one takes values from the set  $\{3, 6, 9, 15\}\mu\text{m}$  and the number of turns is a discrete variable with quarter-turn multiples, in the range  $[0.5, 5.25]$ . Although this parametrization suits the case of the PDK-provided model, it is important to state that the proposed concept applies to other inductors' models or geometric parametrizations.

In this case, we use frequency response data to train the proposed model. We consider that functional similarity between different inductors can be inferred through 1D vectors of inductance  $L(f)$  and quality factor  $Q(f)$ , over a wide frequency range. To obtain these data, we first define a grid on which we conduct parametric sp analyses, based on the geometric variable ranges described previously. This results in 6000 inductor geometries, which are simulated on a predefined set of 250 frequencies, in the range  $[0.1, 100]\text{GHz}$ . The inductors are simulated in single-ended fashion and their frequency responses are acquired by the impedance parameters as [127]

$$L(f) = \frac{\text{imag}(Z_{11})}{2\pi \cdot f}, \quad Q(f) = \frac{\text{imag}(Z_{11})}{\text{real}(Z_{11})}. \quad (6.12)$$

Therefore, both inductance and quality factor features for each geometry are 1D vectors with 250 entries.

The overall employed architecture for the derivation of continuous inductor representations is shown in Fig. 6.4. The VAE part of this model is implemented using 1D convolutional filters, and the inputs to this model are tensors of size  $B \times 2 \times 250$ , where  $B$  is the batch size. Both the encoder and the decoder have 3 convolutional layers with kernel size 4, stride 2 and padding 1, while the filter size is shown in the Fig. 6.4, and make use of the ReLU non-linearity. Two linear layers are used before and after the projection to the latent space, which has  $M = 3$  dimensions. It is important to state

<sup>1</sup>The spacing between inductor turns is considered to be a fixed value.

that the convolutional architecture is preferred, in order to take advantage of the spatial correlations between the entries of the 1D vector inputs.

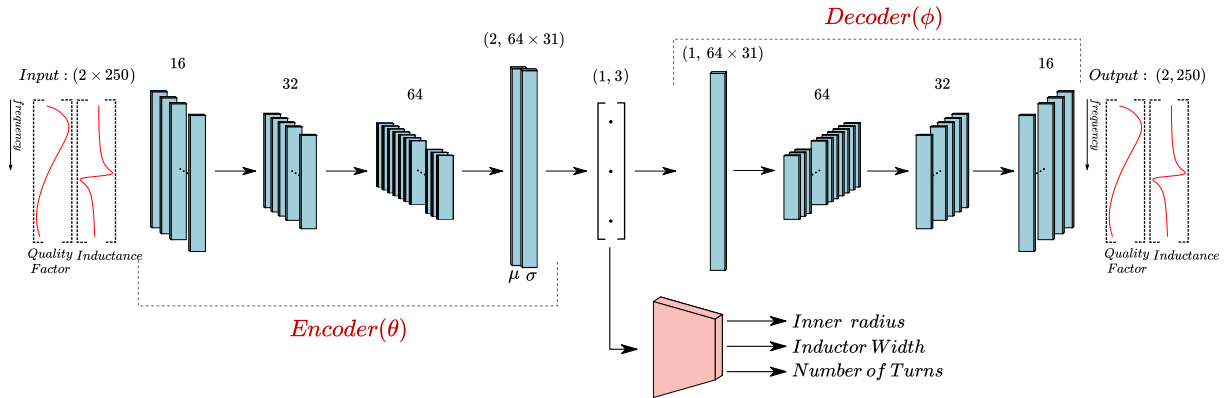


Figure 6.4: A depiction of the proposed architecture. 1D vectors of inductors’ Quality Factor and Inductance frequency behavior are inputs to the 1D convolutional filters of the architecture. The filter sizes are shown as well. The predictor FCNN gets as input the latent representation of the inductor’s frequency characteristics and yields its geometric sizes.

The predictor part of the model is a Fully Connected Neural Network (FCNN) which takes as inputs the sampled points from the approximate posterior distribution and maps them to the actual geometric characteristics. The FCNN, shown in pink color in Fig. 6.4, has 3 layers in total, which are linear with 3, 50 and 25 neurons and uses ReLU activations. The 25 outputs of the predictor network are utilized as follows. For the number of turns and the inductor width, we use a classification approach and define 20 and 4 distinct outputs, each one of them corresponding to a particular valid geometric value. A single output from the FCNN corresponds to the inductor’s inner radius, since it is a continuous-valued variable and its approximation is handled by regression.

In order to train the composite model, we need to define its supervised loss as in Eq. 6.11. Let us denote as  $\tilde{x}_i$ , with  $i = 1, \dots, 25$  the outputs of the predictor net. Outputs  $[\tilde{x}_i]_{i=1}^{20}$  correspond to the number of turns, outputs  $[\tilde{x}_i]_{i=21}^{24}$  to the inductor’s width and output  $\tilde{x}_{25}$  to the inner radius. For a particular geometry, let us also denote as  $\hat{\mathbf{y}} = [y_1, \dots, y_{25}]$  the ground truth results, where out of the first 20 items all of them are zero, with the exception of the corresponding index of the ground truth number of turns. The same applies for the next 4 items, and the last one is a continuous variable. Therefore, ground truth vector  $\hat{\mathbf{y}}$  has always 22 zeros, two ones and a real-valued item. The loss function that penalizes deviations from actual geometries is comprised out of three individual losses,  $L_{NoT}$  for the number of turns,  $L_{IW}$  for the inductor’s width and

$L_{IR}$  for the inner radius. These are computed via

$$\begin{aligned} L_{NoT} &= - \sum_{i=1}^{20} y_i \cdot \log \left[ \frac{\exp(x_i)}{\sum_{j=1}^{20} \exp(x_j)} \right], \\ L_{IW} &= - \sum_{i=21}^{24} y_i \cdot \log \left[ \frac{\exp(x_i)}{\sum_{j=21}^{24} \exp(x_j)} \right], \\ L_{IR} &= (x_{25} - y_{25})^2. \end{aligned} \tag{6.13}$$

We use the squared euclidean loss for the inner radius, and a cross-entropy loss for the two discrete variables. In the batched training, these losses are reduced to scalars by taking their means over the batched inputs. The total supervised loss of the architecture is given, therefore, by

$$\mathcal{L}_{sup} = L_{NoT} + L_{IW} + L_{IR}. \tag{6.14}$$

The loss for the VAE component of the model in Fig. 6.4 can be defined analytically, since we enforce that both the prior  $p(\mathbf{z})$  and the posterior  $q_{\phi}(\mathbf{z}|\mathbf{x})$  are Gaussian distributions. The KL-divergence term of Eq. 6.9 is given in Eq. 6.10, while the first of Eq. 6.9 is the reconstruction loss which in our case is defined as the mean-squared distance of the reconstruction  $\tilde{\mathbf{x}}$  from the ground truth vectors  $\hat{\mathbf{x}}$ , i.e.

$$L_{VAE} = \|\tilde{\mathbf{x}} - \hat{\mathbf{x}}\|^2 - \frac{1}{2} \sum_{i=1}^M \left[ \mu_i^2 + \sigma_i^2 - \log(\sigma_i^2) - 1 \right]. \tag{6.15}$$

The overall loss of the composite model which is used in the training procedure is derived using equations 6.13, 6.14, 6.15 and 6.11.

The model was trained using the Adam optimizer for 1000 epochs and a 80%-20% training-test split. After training, the test data were mapped to their latent representations and predictor's accuracy for number of turns and width is 94% and 96% respectively. For the inner radius, the MSE score is 0.11, where the values are normalized in the  $[0, 1]$  range. Fig. 6.5 depicts a reconstructed inductance curve, along with the predicted geometry for a real spiral inductor using the proposed scheme. Besides the original inductance curve its reconstruction by the model, the latent code and the predicted geometry produced by the model are also given in the plot. We can make a few observations from Fig. 6.5: The reconstruction of input curves is sufficiently good, since it captures the resonance frequencies. Also, the predicted geometry is close to the actual, real world one. In addition, the latent code produced is well within the search boundaries within the latent space, which will be later used in an optimization procedure.

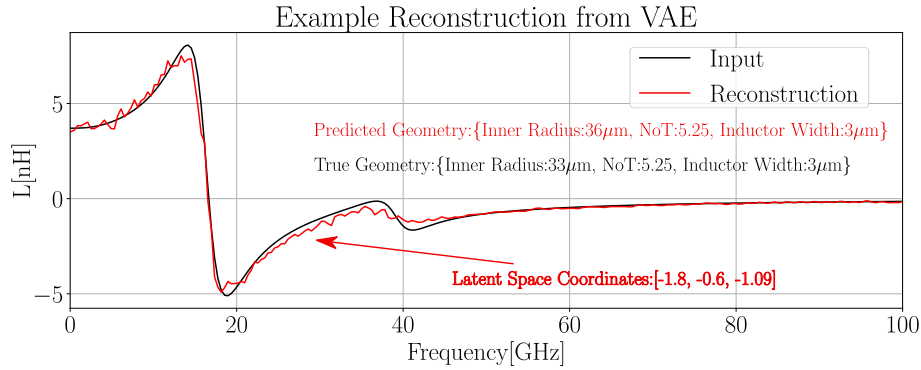


Figure 6.5: A reconstruction of inductor’s inductance across the frequency range of interest.

To further investigate the results of the trained model, we proceed to sample inductance curves from its VAE part. This is done by sampling from the isotropic, unit variance, zero mean gaussian distribution  $p(\mathbf{z})$  and passing the sample from the decoder network. The 500 descaled random samples from the model are shown in Fig. 6.6. It is seen that the curves follow a particular trajectory, which resembles the resonant characteristic of the actual inductor’s curves.

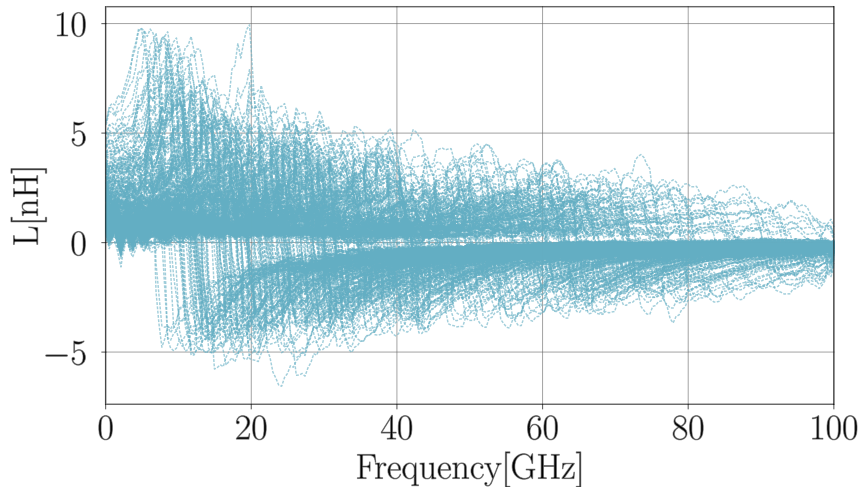


Figure 6.6: 500 samples of inductance curves sampled from the generative model.

To study the structure of the latent space, we consider the following experiment. We use the proposed model to map every single inductor frequency response that was acquired by simulations to the three-dimensional latent space. To visualize the latent space, we depict three cross sections, using the planes  $x = 0$ ,  $y = 0$  and  $z = 0$ , assuming  $\mathbf{z} = [x, y, z]$ . At each cross section, every single point is overlaid with the values of the corresponding inductor’s number of turns, inner radius and inductor width, in Fig. 6.7. By observing each plot, we reach the conclusion that there are clear boundaries between different values for the number of turns and the inductor width, and there is



a clear distinction between large and small inner radius in Fig. 6.7. In practice, this behavior is highly desirable since it ensures that no abrupt jumps between closely latent codes take place, when searching locally in the latent space. This smoothness in the latent space, therefore, ensures that the optimization algorithm can assume that nearby vectors are related functionally, as is the case with BO and the GP models which assign the correlation between different points based on their Euclidean distance.

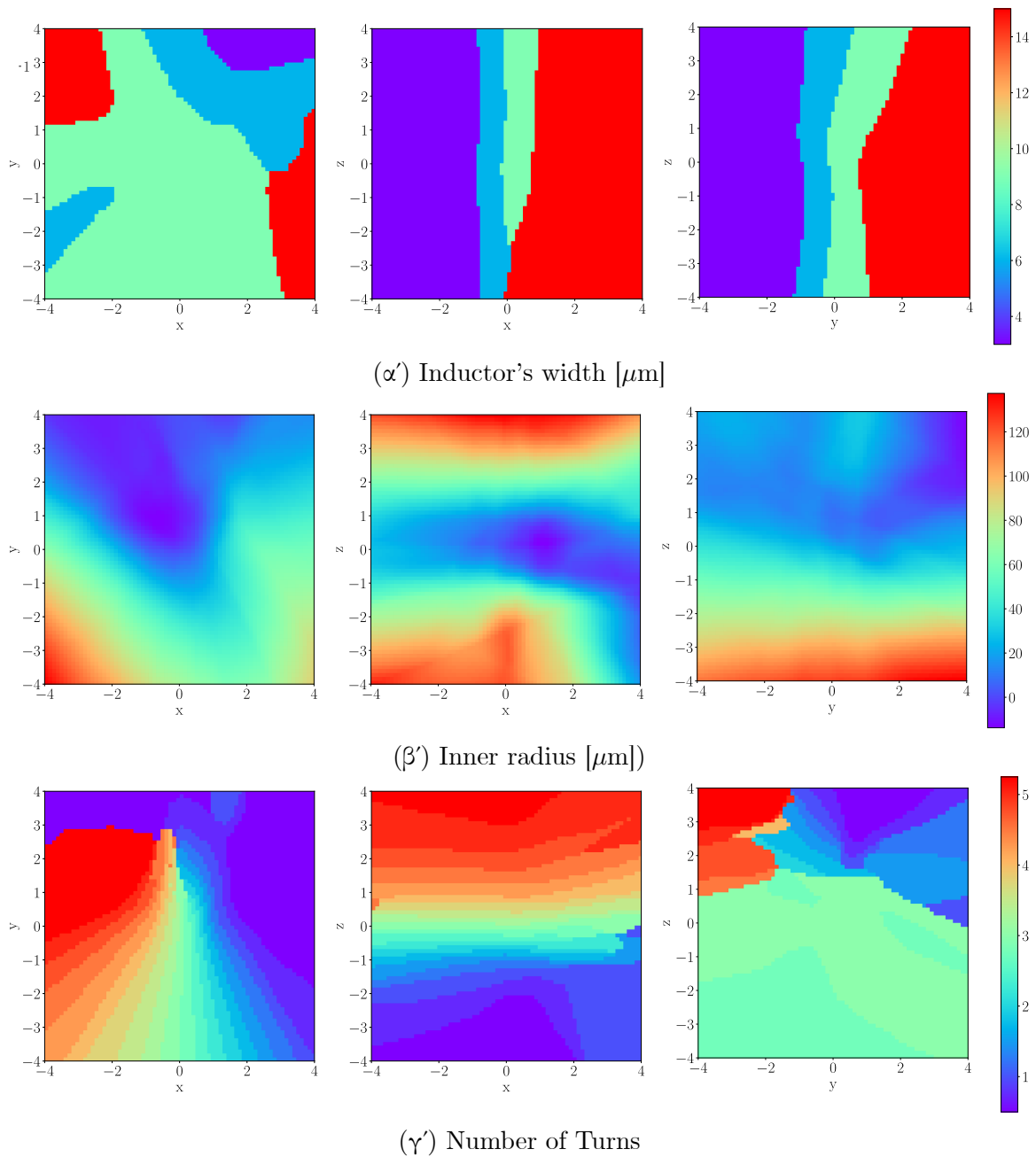


Figure 6.7: Slices of the 3D latent space across the xy, yz and xz planes. The three geometric variables for the spiral inductors are shown.

## Metal Capacitor

In this case, the proposed model is used to derive continuous representations for a TSMC 90nm 'crtmom' rotative metal capacitor. We consider a parametrization with 3 variables, namely fingers space, number of vertical and number of horizontal fingers. The width of the fingers is considered fixed, since it contributes relatively little to the frequency behavior of the device. The fingers spacing variable is continuous in the range [140, 180]nm, whereas the rest of the variables take integer values in the range [6, 200].

Similarly to the previous case of spiral inductors, we gather a dataset of 3000 frequency responses in the range of [0.1, 330]GHz. The data are obtained as  $S_{11}$  responses from a parametric sp analysis. Instead of following the convolutional approach as in the spiral inductor case, in this case we consider a simpler approach is sufficient. We proceed to select 30 frequencies in the aforementioned range and keep the real and imaginary parts of the  $S_{11}$  responses only for them. The VAE model then consists of fully connected encoder and decoder networks, with its input being the concatenated imaginary and real part of each frequency response, i.e. a vector of 60 length. Both the encoder and decoders have three layers with 200, 400 and 600 neurons, with ReLU activations. The chosen latent space dimensionality is  $M = 3$  and the predictor network is a three layer FCNN with 50, 50 and three neurons and ReLU activations.

To train the composite metal capacitor model, we define its supervised loss  $\mathcal{L}_{sup}$  as the summation of three individual losses,  $L_{HF}$  for horizontal fingers,  $L_{VF}$  for vertical fingers and  $L_{fspace}$  for finger spacing. All of them are mean squared losses, defined in a similar way as  $L_{IR}$  in Eq. 6.13, and the overall loss is the summation of all of them and the VAE loss. The model is trained for 1000 epochs, using the Adam optimizer. After training, the test data were mapped to their latent representations through the encoder part of the trained VAE and the predictor's mean squared error for all three outputs was 0.13, with the labels being scaled to [0, 1].

## 6.3 Circuit Design Applications

In this section, the continuous representations of spiral inductors and metal capacitors are used to automatically size two LNA topologies. For both devices, we use the composite models discussed in the previous section and the same TSMC 90nm process. It is worth noting that for device models that are used more than once in a single topology, the same models are utilized, but with different latent variables for optimization.

### 6.3.1 Inductively Degenerated LNA

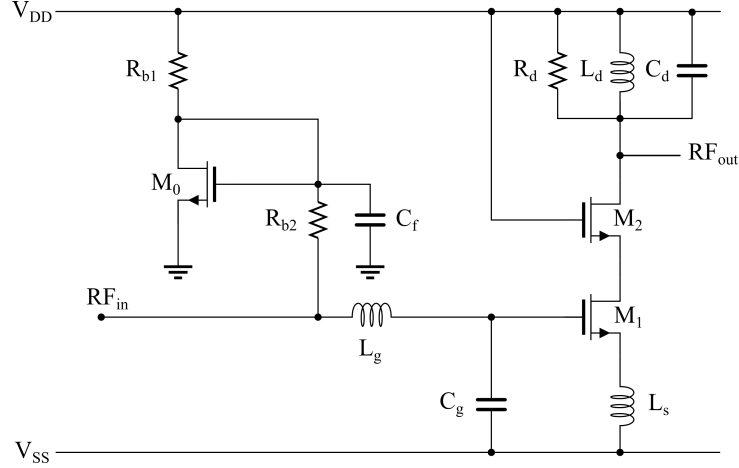


Figure 6.8: The Inductively Degenerated LNA considered as a case study.

In this example we consider the LNA topology shown in Figure 6.8. It is an inductively degenerated single-stage LNA, having three spiral inductors and two metal capacitors, namely  $C_g$  and  $C_d$ . The supply voltage is 1.2V and the operating frequency is 2.4GHz.

This circuit is parametrized by its transistor lengths and widths, the metal resistors' widths and lengths as well as the geometric parameters of the capacitors and inductors. The parameters of the inductors in the latent space formulation are substituted with 9 latent variables in total, three for each device. Similarly, the parameters of the capacitors amount to 6 latent variables in total. In total, there are 21 variables both in the latent space formulation and in the original variable space. The ranges of the variables, both in their original form and in the transformed-latent space form are given in Table 6.1.

As far as the optimization goals are concerned, we employ a single objective formulation where we wish to minimize static power consumption  $P_{dc}$  in the nominal operating conditions, while enforcing  $IP3 \geq -5\text{dBm}$ ,  $NF \leq 2.5\text{dB}$ ,  $S_{11} \leq -8\text{dB}$ ,  $S_{22} \leq -8\text{dB}$  at the operating frequency at the following corners: ss, sf, fs, and ff and at the working temperatures of  $-50$ ,  $27$  and  $125$  Celcius. In addition, for the nominal conditions we enforce  $S_{21} \geq 21\text{dB}$ . This amounts to a total of 13 testbenches and 49 constraints  $[g(\mathbf{x})]_{i=1}^{49}$ .

For comparison, we consider the following sizing methodologies:

- A SO BO algorithm, which makes use of the RFF features and the single objective acquisition function as in Chapter 4, coupled with the latent space formulation,

Table 6.1: LNA Variable Ranges

Variable	Description	Range
$W_M$	Transistor Widths	$[1, 120]\mu\text{m}$
$L_M$	Transistor Lengths	$[100, 240]\text{nm}$
$L_{R1}$	Resistor $R_{b1}$ Length	$[0.1, 30]\mu\text{m}$
$L_{R2}$	Resistor $R_{b2}$ Length	$[0.1, 30]\mu\text{m}$
$W_{R1}$	Resistor $R_{b1}$ Width	$[0.4, 10]\mu\text{m}$
$W_{R2}$	Resistor $R_{b2}$ Width	$[0.4, 10]\mu\text{m}$
$IW_{L_g}/L_{L_g1}$	Inductor $L_g$ : Width/ Latent	$[3, 15]\mu\text{m}/ [-3, 3]$
$IW_{L_s}/L_{L_s1}$	Inductor $L_s$ : Width/ Latent	$[3, 15]\mu\text{m}/ [-3, 3]$
$IW_{L_d}/L_{L_d1}$	Inductor $L_d$ : Width/ Latent	$[3, 15]\mu\text{m}/ [-3, 3]$
$IR_{L_g}L_{L_g2}$	Inductor $L_g$ : Radius/ Latent	$[15, 90]\mu\text{m}/ [-3, 3]$
$IR_{L_s}/L_{L_s2}$	Inductor $L_s$ : Radius/ Latent	$[15, 90]\mu\text{m}/ [-3, 3]$
$IR_{L_d}/L_{L_d2}$	Inductor $L_d$ : Radius/ Latent	$[15, 90]\mu\text{m}/ [-3, 3]$
$NT_{L_g}L_{L_g3}$	Inductor $L_g$ : Turns/ Latent	$[0.5, 5.25]/ [-3, 3]$
$NT_{L_s}/L_{L_s3}$	Inductor $L_s$ : Turns/ Latent	$[0.5, 5.25]/ [-3, 3]$
$NT_{L_d}/L_{L_s3}$	Inductor $L_d$ : Turns/ Latent	$[0.5, 5.25]/ [-3, 3]$
$VF_{C_g}/L_{C_g1}$	Capacitor $C_g$ : Vertical Fingers/ Latent	$[6, 200]/ [-3, 3]$
$VF_{C_d}/L_{C_d1}$	Capacitor $C_d$ : Vertical Fingers/ Latent	$[6, 200]/ [-3, 3]$
$HF_{C_g}/L_{C_g2}$	Capacitor $C_g$ : Horizontal Fingers/ Latent	$[6, 200]/ [-3, 3]$
$HF_{C_d}/L_{C_d2}$	Capacitor $C_d$ : Horizontal Fingers/ Latent	$[6, 200]/ [-3, 3]$
$fs_{C_g}/L_{C_g3}$	Capacitor $C_g$ : Fingers Spacing/ Latent	$[140, 180]\text{nm}/ [-3, 3]$
$fs_{C_d}/L_{C_d3}$	Capacitor $C_d$ : Fingers Spacing/ Latent	$[140, 180]\text{nm}/ [-3, 3]$

Table 6.2: LNA Sizing Results

Formulation	$P_{dc}$ - Best[mW]	$P_{dc}$ - Avg[mW]	$P_{dc}$ - Std[mW]	Success
BO-Latent	9.5	11.7	1.7	10/10
BO-Relaxation	9.8	13.6	2.2	10/10
GA-Latent	11.9	14.3	2.5	8/10
GA	13.8	14.7	1.6	2/10

- The same BO algorithm with the relaxation procedure for integer/discrete variables, as in Chapter 5,
- A genetic algorithm operating on the transformed variable space, making use of the proposed models of the previous section to transform individuals to the original space,
- A genetic algorithm with mixed-variable operators operating on the original variable space.

The hyperparameters of the aforementioned algorithms are chosen as follows: Both BO algorithms have 1200 total evaluations, with  $N_S = 8$  RFF samples per iteration and 150 initial samples. The genetic algorithms have 100 individuals per generation, and are allowed to search for 50 generations.

To account for random fluctuations, we repeat all of the experiments for 10 times. The results of the experiments, with respect to best attained feasible solution, average best feasible solution, standard deviation of best attained feasible solution and success rate are given in Table 6.2. It is seen that, in the provided simulation budget, both BO formulations outperform the GA ones. The vanilla GA formulation, that works on the original search space, finds feasible solutions two times only, whereas the GA operating in the transformed space, i.e. having only continuous variables, finds feasible solutions 8 out of ten times. In contrast, both BO formulations find feasible solutions in all experiments.

The efficiency of the proposed approach is underlined by the power consumption results. Among all formulations, the BO that operates in the transformed variable space yields the best solution on average (11.7mW) and the best solutions out of all formulations and executions (9.5mW). The BO with relaxation is able to find a single good solution (9.8mW), but yields on average 13.6mW of power, which is close to the result of the GA working on the transformed space. The fact that the mixed variable GA formulation has the lowest standard deviation is due to the number of feasible outcomes; only two executions result in feasible results that are taken into account in Table 6.2.

A graphical illustration of the performance of all formulations in the sizing procedure of the LNA is given in Figure 6.9, where the evolution of the Constraint Violation (CV) metric is given, against the number of evaluations. Here, it is seen that the BO that works in the transformed space requires roughly 600 evaluations to find feasible solutions. In the case of the BO with the relaxation procedure, all executions find feasible solutions at around 1100 evaluations. The mixed-variable GA's CV seems to stagnate at 4000 evaluations, whereas the transformed space formulation of GA seems to improve its CV even at the end of the experiment.

### 6.3.2 Wideband LNA

In this subsection, a wideband, noise cancelling LNA [128] shown in Figure 6.10 is sized. This topology consists of Common-Gate and a Common-Source-Common-Gate stage, along with a source follower output buffer. It works on a 1.8V supply and its operating frequency range is [2, 5]GHz.

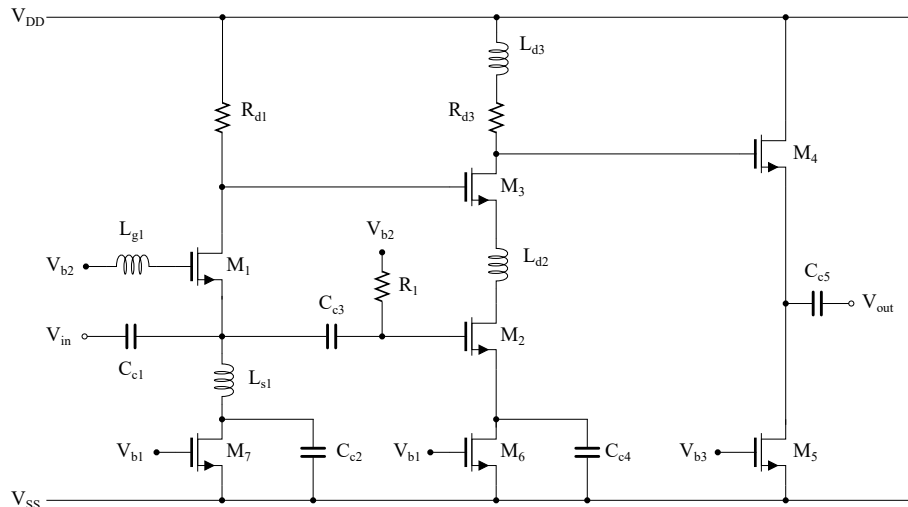


Figure 6.10: The wideband, noise canceling LNA [128] considered in this subsection.

In a similar way as in the previous example, this circuit is parametrized by its transistor widths and lengths, its metal resistors widths and lengths, and the geometric sizes of the capacitors and inductors. All of the capacitors are rotative metal ones, and the inductors are spiral octagonal ones, making the proposed models in the previous section applicable. In addition, there are three biasing voltages, namely  $V_{b1}$ ,  $V_{b2}$  and  $V_{b3}$  to be selected. Following the guidelines in the original implementation, all capacitors are chosen to be identical and the transistors share the same gate length. In total, there are 34 design variables, both in the original variable space and in the latent space

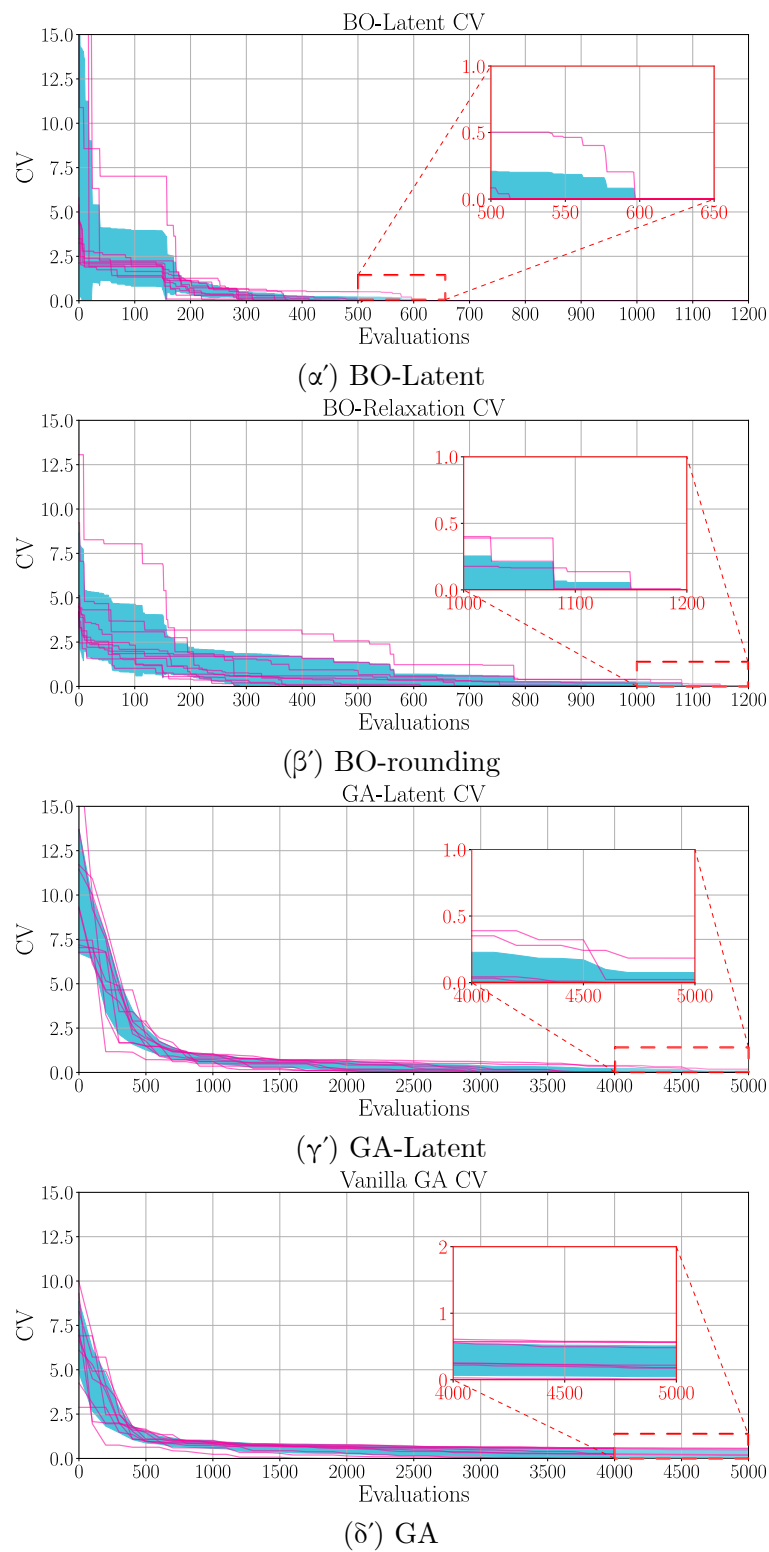


Figure 6.9: The LNA's CV for the discussed formulations. The blue are indicates the confidence region of  $\pm$ std, while the purple lines are the curves of each repetition.

formulation. The ranges of the variables, both latent ones and the original ones, are given in Table 6.3. The variable space of the optimization will be denoted as  $\mathbb{S}$ .

For automatic sizing, we consider again a single objective formulation. In order to compare the results of the sizing formulations in a principled way, we consider the following Figure of Merit (FoM) [92] for wideband LNAs and set its maximization as our optimization goal:

$$\text{FoM}(\mathbf{x}) = 20 \log_{10} \left( \frac{G_{max}(\mathbf{x}) \cdot IIP3_{max}(\mathbf{x}) \cdot BW(\mathbf{x})}{(F_{min}(\mathbf{x}) - 1) \cdot P_{dc}(\mathbf{x})} \right). \quad (6.16)$$

The term  $G_{max}$  indicates the maximum voltage gain of the LNA in V/V,  $IIP3_{max}$  is the maximum input-referred third-order intercept point in mW,  $BW$  is the LNAs bandwidth and  $F_{min}$  is the minimum noise figure in the operating frequency range, in linear units. In our formulation, we consider the bandwidth fixed to 3GHz. To ensure this, we consider the following constraint function:

$$g(\mathbf{x}) = \max_{f \in [2,5]\text{GHz}} \left[ S_{11}(\mathbf{x}, f) \right] + 10, \quad \mathbf{x} \in \mathbb{S} \quad (6.17)$$

with  $g(\mathbf{x}) \leq 0$  being the feasibility criterion. Here, function  $S_{11}(\mathbf{x}, f)$  is the  $S_{11}$  result of the simulation, in dBs. This satisfaction of this constraint ensures that the results will be matched to a  $50\Omega$  input port. Thus, the sizing of this particular LNA topology can be formulated as

$$\begin{aligned} \min \quad & \text{FoM}(\mathbf{x}), \quad \mathbf{x} \in \mathbb{S} \\ \text{s.t.} \quad & g(\mathbf{x}) \leq 0 \end{aligned} \quad (6.18)$$

For comparison, we consider again the same 4 optimization algorithm cases as in the previous subsection, with the hyperparameters of the algorithms remaining the same. The experiments were executed 10 times to account for random fluctuations.

Table 6.4 depicts the sizing results using the different formulations, with respect to the objective FoM. Among the considered cases and under the given optimization budget, the proposed BO with the latent space representations of inductors and capacitors yields in average a FoM of 63.9dBs, whereas the same BO with the relaxation of the discrete variables yields 58dBs. In addition, the latent space formulation seems to be reliable, since it yields less variance (5.5dB standard deviation) in comparison to the relaxation approach (9.3dBs).

Regarding the GA sizing formulations, both the GA that operates in the latent space and the vanilla GA succeed all of the times in finding feasible solutions. However, the GA-Latent formulation is more successful with respect to its resulting FoM, which is roughly 23dBs higher. Both GAs do not succeed in overpassing BO in the attained FoM



Table 6.3: Wideband LNA Variable Ranges

Variable	Description	Range
$L_M$	Transistor Lengths	[100, 240]nm
$W_{M1}$	$M_1$ Width	[1, 120] $\mu\text{m}$
$W_{M2}$	$M_2$ Width	[1, 120] $\mu\text{m}$
$W_{M3}$	$M_3$ Width	[1, 120] $\mu\text{m}$
$W_{M4}$	$M_4$ Width	[1, 120] $\mu\text{m}$
$W_{M5}$	$M_5$ Width	[1, 120] $\mu\text{m}$
$W_{M6}$	$M_6$ Width	[1, 120] $\mu\text{m}$
$W_{M7}$	$M_7$ Width	[1, 120] $\mu\text{m}$
$L_{Rd1}$	Resistor $R_{d1}$ Length	[0.1, 30] $\mu\text{m}$
$L_{Rd2}$	Resistor $R_{d2}$ Length	[0.1, 30] $\mu\text{m}$
$L_{Rd3}$	Resistor $R_{d3}$ Length	[0.1, 30] $\mu\text{m}$
$L_{R1}$	Resistor $R_1$ Length	[0.1, 30] $\mu\text{m}$
$W_{Rd1}$	Resistor $R_{d1}$ Width	[0.4, 10] $\mu\text{m}$
$W_{Rd2}$	Resistor $R_{d2}$ Width	[0.4, 10] $\mu\text{m}$
$W_{Rd3}$	Resistor $R_{d3}$ Width	[0.4, 10] $\mu\text{m}$
$W_{R1}$	Resistor $R_1$ Width	[0.4, 10] $\mu\text{m}$
$V_{b1}$	Volatge $V_{b1}$	[0.2, 0.5] $\mu\text{m}$
$V_{b2}$	Volatge $V_{b2}$	[0.2, 0.5] $\mu\text{m}$
$V_{b3}$	Volatge $V_{b3}$	[0.2, 0.5] $\mu\text{m}$
$IW_{L_{g1}}/L_{L_{g1}1}$	Inductor $L_{g1}$ : Width/ Latent	[3, 15] $\mu\text{m}/$ [-3, 3]
$IW_{L_{s1}}/L_{L_{s1}1}$	Inductor $L_{s1}$ : Width/ Latent	[3, 15] $\mu\text{m}/$ [-3, 3]
$IW_{L_{d2}}/L_{L_{d2}1}$	Inductor $L_{d2}$ : Width/ Latent	[3, 15] $\mu\text{m}/$ [-3, 3]
$IW_{L_{d3}}/L_{L_{d3}1}$	Inductor $L_{d3}$ : Width/ Latent	[3, 15] $\mu\text{m}/$ [-3, 3]
$IR_{L_{g1}}/L_{L_{g1}2}$	Inductor $L_{g1}$ : Radius/ Latent	[15, 90] $\mu\text{m}/$ [-3, 3]
$IR_{L_{s1}}/L_{L_{s1}2}$	Inductor $L_{s1}$ : Radius/ Latent	[15, 90] $\mu\text{m}/$ [-3, 3]
$IR_{L_{d2}}/L_{L_{d2}2}$	Inductor $L_{d2}$ : Radius/ Latent	[15, 90] $\mu\text{m}/$ [-3, 3]
$IR_{L_{d3}}/L_{L_{d3}2}$	Inductor $L_{d3}$ : Radius/ Latent	[15, 90] $\mu\text{m}/$ [-3, 3]
$NT_{L_{g1}}/L_{L_{g1}2}$	Inductor $L_{g1}$ : Turns/ Latent	[0.5, 5.25]/ [-3, 3]
$NT_{L_{s1}}/L_{L_{s1}2}$	Inductor $L_{s1}$ : Turns/ Latent	[0.5, 5.25]/ [-3, 3]
$NT_{L_{d2}}/L_{L_{d2}2}$	Inductor $L_{d2}$ : Turns/ Latent	[0.5, 5.25]/ [-3, 3]
$NT_{L_{d3}}/L_{L_{d3}2}$	Inductor $L_{d3}$ : Turns/ Latent	[0.5, 5.25]/ [-3, 3]
$VF/L_{C1}$	Capacitors: Vertical Fingers/ Latent	[6, 200]/ [-3, 3]
$HF/L_{C2}$	Capacitors: Horizontal Fingers/ Latent	[6, 200]/ [-3, 3]

Table 6.4: Wideband LNA Sizing Results

<b>Formulation</b>	<b>FoM - Best[dB]</b>	<b>FoM - Avg[dB]</b>	<b>FoM - Std[dB]</b>	<b>Success</b>
BO-Latent	71.1	63.9	5.5	10/10
BO	68.3	58.4	9.3	10/10
GA-Latent	65.9	58.2	8.5	10/10
GA	49.8	35.6	8.3	10/10

performance, both in averaged results and in terms of the best value acquired through the 10 repetitions. The above highlight the fact that the utilized BO variant is able to approximate the global optimum better, in comparison to the population based GA, under the given simulation budgets.

The fact that both BO and GA formulations that work with the continuous representations outperform their mixed-variable counterparts underlines the efficiency of our approach. This improvement for the case of BO can be attributed to the fact that there is no brute force rounding of the variables, enabling the GP models to assign correlations to points in the variable space by using their Euclidean distance. In the case of the GA, the results can be attributed to the fact that continuous-only operators work better, and the fact that the latent space of the devices is created in such a way that functionally similar devices are closeby, which assists in the exploration of the variable space. In fact, a simple perturbation of a particular discrete variable may change to a great extent the behavior of the device, such as the spiral inductor.

For a graphical comparison, Figure 6.11 demonstrates the evolution of the LNA's FoM metric for each of the 4 formulations considered, against the number of evaluations. It is seen that the GAs require roughly 800 to 1500 evaluations for reaching a feasible solution, which is seen as an abrupt deviation in the FoM metric in these plots. For the BO cases, the first feasible solutions are found below 200 evaluations. On average, the BO cases require much less evaluations to reach acceptable results, in comparison to the population-based GAs. The BO operating in the transformed variable space has the smallest variance in results, highlighted by the range of the blue region in the respective figure.

## 6.4 Summary & Concluding Remarks

In this section we described a novel approach for representation learning of integrated devices. By establishing a framework that combines i) generative models, ii) predictor models that map from latent space to valid geometric representations and iii) function-

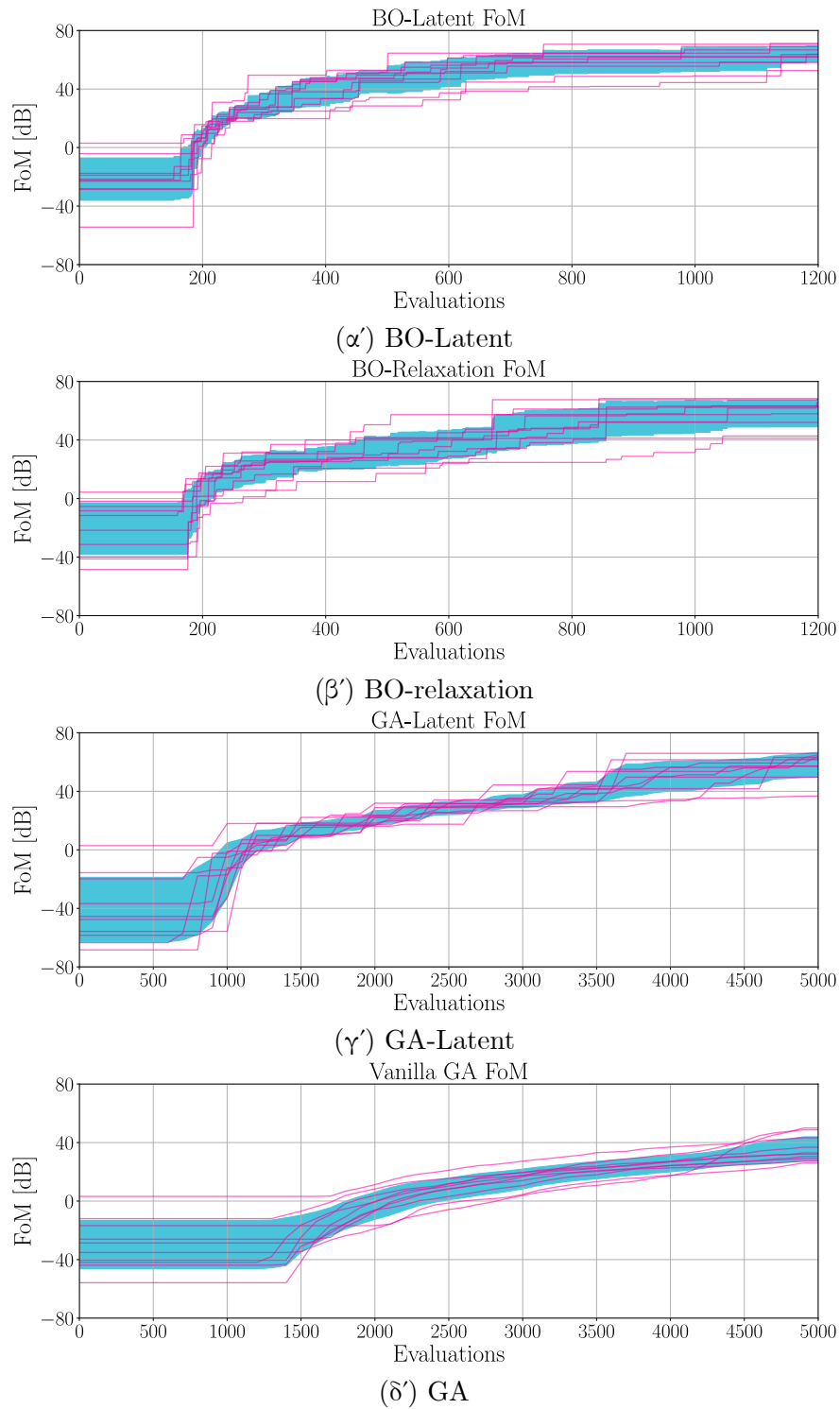


Figure 6.11: The evolution of the LNA's FoM during the automatic sizing, using the four discussed formulations. The blue are indicates the confidence region of  $\pm \text{std}$ , while the purple lines are the curves of each repetition.

related simulation data per device, we are able to build a data-driven continuous representation, which is associated with real-world devices, otherwise parametrized by variables both continuous and discrete. Thereby, we are able to transform the parametrization of a circuit that has these particular devices into a continuous-valued one, and apply BO to automatically size it. A rigorous analysis of the components of the proposed approach was presented, where both the generative capabilities and the latent space structure of the model are discussed. To test the sanity of our approach, we conducted a SO optimization on two LNAs, by substituting their spiral inductor and metal capacitor variables with the ones of the trained model. The results proved that our approach has merit.

The use-cases considered for continuous parametrization are the octagonal spiral inductors and the rotative metal capacitors, whose models are provided by a TSMC PDK. However, it should be underlined that one could possibly build such parametrization models for any device it suits them. A natural extension of this approach is to include more inductor models, more process technologies and even combine all these data into a single model, with categorical variables to distinguish between different models, etc. In addition, one could use Z-parameter data obtained via EM simulations, rather than PDK inductor models. At the same time this work was being developed, a process-agnostic representation learning approach for MOS devices was presented in [129], where a database of transistor IV curves is presented, and a statistical analysis is executed on these data. Based on this work, the authors of [130] used VAE models to learn representations of MOS devices and use these representations for downstream ML tasks. These works also highlight the potential of representation learning in the field of integrated devices.

Device modeling and continuous parametrization can have implications in other research areas as well. For instance, the problem of integrated inductor modeling has attracted the attention of many researchers, with works focusing on design using surrogate-assisted optimization [131, 132] or circuit sizing using pre-computed inductor pareto-fronts [133]. These methods are purely supervised and must conduct a search prior to producing an inductor model, i.e. they do not provide inverse maps from performance space to actual geometries. Furthermore, the field of microwave design, which relies also on simulation-driven approaches for automatic design [134, 135, 136, 137] of structures like filters could potentially benefit from a representation learning technique, along with an inverse model that maps to actual geometries. A common aspect of all the aforementioned methods is the big computational costs for using optimization-based design; EM simulations typically require much time to complete and, while methods have been proposed to relieve this issue, such as multi-fidelity modeling, a model that learns to

---

map specifications to geometries would be very helpful. Of course, the inverse mapping can be done using the proposed combination of the VAE and the predictor network, or by using some conditional generative models [138] that can generate geometries, provided the specifications as input conditions. An example application of conditional generative models in microwave design is proposed in [139].



# Chapter 7

## Conclusions & Future Directions

### 7.1 Thesis Contributions

In this dissertation, novel approaches to the task of automatic analog circuit sizing are proposed. Initially, the concept of optimization-based circuit sizing with simulations is discussed, and its advantages over other related approaches are listed. The same concept is illustrated by employing EAs for black-box optimization of real-world circuits, with results indicating that optimization-based design can help designers reason about the competing trade offs of circuit design. The concept of low-budget optimizers is then explored in the thesis, where novel SO and MO optimizers are put forward and tested on a large variety of integrated circuits, providing better results within given simulation budgets. Lastly, a method to handle devices that are parametrized by mixed type of variables is also discussed, by using a Deep-Learning technique. In summary, the contributions of this work are given below:

- The definition of a systematic approach to simulation-based sizing of analog circuits, which does not make use of hand-equations or any other information other than the circuit netlist.
- The use of low-budget, Bayesian Optimization approaches tailored to high-dimensional problems. Both of the algorithms that target SO problems and MOPs make use of the trust region concept, to relieve the optimization search from excessive exploration of the variable space. In addition, they both incorporate acquisition functions that are parallelizable, i.e. they provide multiple query points per iteration.
- The incorporation of RFF approximations to the low-budget optimization algorithms. Analytic functions are sampled from the posterior distributions of the BO GPs, and are shown to enhance the efficiency of the proposed algorithm in comparison to simple Thompson Sampling.
- A framework for device modelling and representation learning, using continuous-valued latent variables. This enables the use of BO in virtually any circuit topology

considered, but also has many potential benefits due to the ability to map device specifications to actual geometries, i.e. inverse mapping.

- The use of a programmable interface with commercial simulators, which enables designers to define custom optimization tasks, execute parametric simulations for validation or design exploration and make use of the proposed or other, custom-made algorithms to size given topologies. It offers the ability to use the previously discussed contributions of this work in an easy and systematic manner.

## 7.2 Future Work

The general goal of this thesis is the development of methods for enhancing the productivity of analog designers. In this context, one may distinguish two separate tracks for future research directions:

- *Circuit Sizing*: The task of automatic circuit sizing is handled in this work by the use of black-box optimizers. Although effective, this concept fails to take advantage of the information of the each provided topology, which is encoded in the circuit netlist itself: the type of devices and the interconnections between them can be coded into a graph structure, much like the Modified Nodal Analysis does in circuit theory. An incorporation of that kind of information into the optimizer, though the use of a learning technique, could potentially yield better results in terms of simulation budget. This approach is based on the *learning to optimize* concept [140], where an agent is learning to select query points from existing data. Such a method could potentially generalize optimization results across different topologies, reducing the computational overhead of simulations. A simple approach to its implementation would include the use of learned acquisition functions, rather than off-the-shelf ones.
- *Analog EDA*: Regarding the general task of enhancing designers' productivity, one research direction that has merit is the automation of the layout design for analog and RF sized topologies. In practice, the combination of sizing and layout design automation could result in push-button circuit design procedures. Although the layout automation task is an active research topic, its tight correlation with device sizing renders its use not so practical. A research direction that does have merit is the incorporation of sizing and layout selection into a single task, with interrelated factors being taken care of.

We should underline the fact that the methodologies proposed in this work can be used outside of the Analog EDA ecosystem. For instance, engineering design using



---

optimization based techniques is very popular with areas where intuitive design is almost impossible, like the design for microwave and antenna geometries. Low budget MO optimizers such as LoCoMOBO could be used within these research areas as well. In addition, the Device Representation Learning concept could potentially benefit the engineering design teams rapidly prototype apparatus, based on pre-existing libraries of past designs. This approach has the benefit to be used in some form of generative design procedure, where, based on an archive of past relevant data, the user provides desired specifications and obtains simulation-free suggestions. This method could be of great use especially in cases where the design spaces are not defined in terms of continuous-valued variables, but as a combination of categorical, integer or continuous variables.



# Publications

## Articles in Scientific Journals

- Konstantinos Touloupas and Paul P. Sotiriadis. “LoCoMOBO: A Local Constrained Multi-Objective Bayesian Optimization for Analog Circuit Sizing”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2021), pp. 1–1. DOI: [10.1109/TCAD.2021.3121263](https://doi.org/10.1109/TCAD.2021.3121263)
- Vassilis Alimisis, Georgios Gennis, Konstantinos Touloupas, Christos Dimas, Marios Gourdouparis, and Paul P Sotiriadis. “Gaussian Mixture Model classifier analog integrated low-power implementation with applications in fault management detection”. In: *Microelectronics Journal* (2022), p. 105510. DOI: <https://doi.org/10.1016/j.mejo.2022.105510>
- Vassilis Alimisis, Georgios Gennis, Konstantinos Touloupas, Christos Dimas, Nikolaos Uzunoglu, and Paul P. Sotiriadis. “Nanopower Integrated Gaussian Mixture Model Classifier for Epileptic Seizure Prediction”. In: *Bioengineering* 9.4 (2022). DOI: [10.3390/bioengineering9040160](https://doi.org/10.3390/bioengineering9040160)
- Konstantinos Touloupas and Paul Peter Sotiriadis. “Magnitude-only modeling for sigma-delta modulator characterization”. In: *AEU - International Journal of Electronics and Communications* 112 (2019), p. 152936. ISSN: 1434-8411. DOI: <https://doi.org/10.1016/j.aeue.2019.152936>

## Papers in Conferences

- Konstantinos Touloupas, Nikos Chouridis, and Paul P. Sotiriadis. “Local Bayesian Optimization For Analog Circuit Sizing”. In: *2021 58th ACM/IEEE Design Automation Conference (DAC)*. 2021, pp. 1237–1242. DOI: [10.1109/DAC18074.2021.9586172](https://doi.org/10.1109/DAC18074.2021.9586172)
- Konstantinos Touloupas and Paul P Sotiriadis. “Mixed-Variable Bayesian Optimization for Analog Circuit Sizing using Variational Autoencoders”. In: *2022 18th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. IEEE. 2022, pp. 1–4. DOI: [10.1109/SMACD55068.2022.9816185](https://doi.org/10.1109/SMACD55068.2022.9816185)

- Kostas Touloupas and Paul Peter Sotiriadis. “An Optimization-Based Approach for Analog Circuit Technology Migration”. In: *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*. IEEE. 2021, pp. 1–5. DOI: [10.1109/SEEDA-CECNSM53056.2021.9566219](https://doi.org/10.1109/SEEDA-CECNSM53056.2021.9566219)
- Maria-Ermioni Plagaki, Konstantinos Touloupas, and Paul P. Sotiriadis. “Multi-Objective Optimization Methods for CMOS LC-VCO Design”. In: *2021 10th International Conference on Modern Circuits and Systems Technologies (MOCASST)*. IEEE. 2021, pp. 1–4. DOI: [10.1109/MOCASST52088.2021.9493409](https://doi.org/10.1109/MOCASST52088.2021.9493409)
- Kostas Touloupas and Paul Peter Sotiriadis. “Analog and RF Circuit Constrained Optimization Using Multi-Objective Evolutionary Algorithms”. In: *2021 IEEE 12th Latin America Symposium on Circuits and System (LASCAS)*. 2021, pp. 1–4. DOI: [10.1109/LASCAS51355.2021.9459145](https://doi.org/10.1109/LASCAS51355.2021.9459145)

# References

- [1] Alan B Grebene. *Bipolar and MOS analog integrated circuit design*. John Wiley & Sons, 2002.
- [2] Jens Lienig and Juergen Scheible. *Fundamentals of layout design for electronic circuits*. Springer, 2020.
- [3] A.S. Sedra, K.C. Smith, T.C. Carusone, and V. Gaudet. *Microelectronic Circuits*. Oxford series in electrical and computer engineering. Oxford University Press, 2020. ISBN: 9780190853549.
- [4] Gordon E Moore et al. *Cramming more components onto integrated circuits*. 1965.
- [5] Andrew B Kahng, Jens Lienig, Igor L Markov, and Jin Hu. *VLSI physical design: from graph partitioning to timing closure*. Springer Science & Business Media, 2011.
- [6] Zixuan Zhang. “Analysis of the Advantages of the M1 CPU and Its Impact on the Future Development of Apple”. In: *2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*. IEEE. 2021, pp. 732–735.
- [7] Manuel Barros, Jorge Guilherme, and Nuno Horta. “Analog circuits optimization based on evolutionary computation techniques”. In: *Integration* 43.1 (2010), pp. 136–155.
- [8] Bernd Hoefflinger. “IRDS—International Roadmap for Devices and Systems, Rebooting”. In: *NANO-CHIPS 2030: On-Chip AI for an Efficient Data-Driven World* 9 (2020).
- [9] *Virtuoso Analog Design Environment GXL*, Cadence. [https://www.cadence.com/content/dam/cadence-www/global/en\\_US/documents/tools/custom-ic-analog-rf-design/virtuoso-analog-design-environment-gxl-ds.pdf](https://www.cadence.com/content/dam/cadence-www/global/en_US/documents/tools/custom-ic-analog-rf-design/virtuoso-analog-design-environment-gxl-ds.pdf). Accessed: 2022-05-19.
- [10] *MunEDA WiCkeD EDA Tools*. <https://www.muneda.com/>. Accessed: 2022-05-19.

- [11] R. Harjani, R.A. Rutenbar, and L.R. Carley. “OASYS: a framework for analog circuit synthesis”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 8.12 (1989), pp. 1247–1266. DOI: [10.1109/43.44506](https://doi.org/10.1109/43.44506).
- [12] M.G.R. Degrauwe et al. “IDAC: an interactive design tool for analog CMOS circuits”. In: *IEEE Journal of Solid-State Circuits* 22.6 (1987), pp. 1106–1116. DOI: [10.1109/JSSC.1987.1052861](https://doi.org/10.1109/JSSC.1987.1052861).
- [13] Guoyong Shi. “Graph-Pair Decision Diagram Construction for Topological Symbolic Circuit Analysis”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 32.2 (2013), pp. 275–288. DOI: [10.1109/TCAD.2012.2217963](https://doi.org/10.1109/TCAD.2012.2217963).
- [14] Yang Song and Guoyong Shi. “Hierarchical graph reduction approach to symbolic circuit analysis with data sharing and cancellation-free properties”. In: *17th Asia and South Pacific Design Automation Conference*. 2012, pp. 541–546. DOI: [10.1109/ASPDAC.2012.6165012](https://doi.org/10.1109/ASPDAC.2012.6165012).
- [15] Yanjie Gu and Guoyong Shi. “Optimal realization of switched-capacitor circuits by symbolic analysis”. In: *2015 28th IEEE International System-on-Chip Conference (SOCC)*. 2015, pp. 70–73. DOI: [10.1109/SOCC.2015.7406913](https://doi.org/10.1109/SOCC.2015.7406913).
- [16] Guoyong Shi, Jiajun Chen, Andy Tai, and Frank Lee. “A size sensitivity method for interactive CMOS circuit sizing”. In: *Analog Integrated Circuits and Signal Processing* 77.2 (2013), pp. 95–104.
- [17] R. Iskander, M.-M. Louerat, and A. Kaiser. “Hierarchical Graph-Based Sizing for Analog Cells Through Reference Transistors”. In: *2006 Ph.D. Research in Microelectronics and Electronics*. 2006, pp. 321–324. DOI: [10.1109/RME.2006.1689961](https://doi.org/10.1109/RME.2006.1689961).
- [18] Ramy Iskander, Marie-Minerve Louërat, and Andreas Kaiser. “Hierarchical sizing and biasing of analog firm intellectual properties”. In: *Integration* 46.2 (2013), pp. 172–188. ISSN: 0167-9260. DOI: <https://doi.org/10.1016/j.vlsi.2012.01.001>. URL: <https://www.sciencedirect.com/science/article/pii/S0167926012000089>.
- [19] Farakh Javid, Stéphanie Youssef, Ramy Iskander, and Marie-Minerve Louërat. “A designer-assisted analog synthesis flow”. In: *Analog/RF and Mixed-Signal Circuit Systematic Design*. Springer, 2013, pp. 123–148.

- [20] Farakh Javid, Ramy Iskander, Marie-Minerve Louërat, and François Durbin. “A structured DC analysis methodology for accurate verification of analog circuits”. In: *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2013, pp. 2662–2665. DOI: [10.1109/ISCAS.2013.6572426](https://doi.org/10.1109/ISCAS.2013.6572426).
- [21] Farakh Javid, Ramy Iskander, Marie-Minerve Louërat, and Damien Dupuis. “Analog circuits sizing using bipartite graphs”. In: *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*. 2011, pp. 1–4. DOI: [10.1109/MWSCAS.2011.6026591](https://doi.org/10.1109/MWSCAS.2011.6026591).
- [22] Ivick Guerra-Gómez, Trent McConaghy, and Esteban Tlelo-Cuautle. “Operating-point driven formulation for analog computer-aided design”. In: *Analog Integrated Circuits and Signal Processing* 74.2 (2013), pp. 345–353.
- [23] Tuotian Liao and Lihong Zhang. “Efficient Parasitic-aware gm/ID-based Hybrid Sizing Methodology for Analog and RF Integrated Circuits”. In: *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 26.2 (2020), pp. 1–31.
- [24] Florin Burcea and Helmut Graeb. “Inversion-Coefficient-Aware Yield Optimization of Analog Circuits”. In: *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. 2019, pp. 193–196. DOI: [10.1109/ICECS46596.2019.8965193](https://doi.org/10.1109/ICECS46596.2019.8965193).
- [25] Paul GA Jespers and Boris Murmann. *Systematic design of analog CMOS circuits*. Cambridge University Press, 2017.
- [26] Boris Murmann. “Systematic Design of Analog Circuits Using Pre-Computed Lookup Tables”. In: (2016).
- [27] Guoyong Shi. “Sizing of multi-stage Op Amps by combining design equations with the gm/ID method”. In: *Integration* 79 (2021), pp. 48–60.
- [28] Matthias Schweikardt and Juergen Scheible. “Improvement of Simulation-Based Analog Circuit Sizing using Design-Space Transformation”. In: *SMACD / PRIME 2021; International Conference on SMACD and 16th Conference on PRIME*. 2021, pp. 1–4.
- [29] P Prajitha. “The gm/Id Methodology based Design and Optimization of Analog Circuits”. In: *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE. 2020, pp. 504–509.

- [30] Giovanni Piccinni, Claudio Talarico, Gianfranco Avitabile, and Giuseppe Coviello. “Innovative Strategy for Mixer Design Optimization Based on gm/ID Methodology”. In: *Electronics* 8.9 (2019). ISSN: 2079-9292. DOI: [10.3390/electronics8090954](https://doi.org/10.3390/electronics8090954). URL: <https://www.mdpi.com/2079-9292/8/9/954>.
- [31] Hesham Omran, Mohamed H. Amer, and Ahmed M. Mansour. “Systematic Design of Bandgap Voltage Reference Using Precomputed Lookup Tables”. In: *IEEE Access* 7 (2019), pp. 100131–100142. DOI: [10.1109/ACCESS.2019.2930595](https://doi.org/10.1109/ACCESS.2019.2930595).
- [32] Fikre Tsigabu Gebreyohannes, Jacky Porte, Marie-Minerve Louërat, and Hassan Aboushady. “A gm/Id Methodology Based Data-Driven Search Algorithm for the Design of Multistage Multipath Feed-Forward-Compensated Amplifiers Targeting High Speed Continuous-Time  $\Sigma\Delta$ -Modulators”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.12 (2020), pp. 4311–4324. DOI: [10.1109/TCAD.2020.2966998](https://doi.org/10.1109/TCAD.2020.2966998).
- [33] Mahdi Tarkhan and Mohamad Sawan. “A Novel Current Density Based Design Approach of Low-Noise Amplifiers”. In: *IEEE Access* 10 (2022), pp. 42309–42320. DOI: [10.1109/ACCESS.2022.3168743](https://doi.org/10.1109/ACCESS.2022.3168743).
- [34] M.delM. Hershenson, S.P. Boyd, and T.H. Lee. “Optimal design of a CMOS op-amp via geometric programming”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 20.1 (2001), pp. 1–21. DOI: [10.1109/43.905671](https://doi.org/10.1109/43.905671).
- [35] Maria del Mar Hershenson. “CMOS analog circuit design via geometric programming”. In: *Proceedings of the 2004 American Control Conference*. Vol. 4. IEEE. 2004, pp. 3266–3271.
- [36] Walter Daems, Georges Gielen, and Willy Sansen. “An efficient optimization-based technique to generate posynomial performance models for analog integrated circuits”. In: *Proceedings of the 39th annual Design Automation Conference*. 2002, pp. 431–436.
- [37] Xin Li, Padmini Gopalakrishnan, Yang Xu, and T Pileggi. “Robust analog/RF circuit design with projection-based posynomial modeling”. In: *IEEE/ACM International Conference on Computer Aided Design, 2004. ICCAD-2004*. IEEE. 2004, pp. 855–862.
- [38] Tom Eeckelaert, Walter Daems, Georges Gielen, and Willy Sansen. “Generalized simulation-based posynomial model generation for analog integrated circuits”. In: *Analog Integrated Circuits and Signal Processing* 40.3 (2004), pp. 193–203.



- [39] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [40] Yaping Li, Yong Wang, Yusong Li, Ranran Zhou, and Zhaojun Lin. “An Artificial Neural Network Assisted Optimization System for Analog Design Space Exploration”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.10 (2020), pp. 2640–2653. DOI: [10.1109/TCAD.2019.2961322](https://doi.org/10.1109/TCAD.2019.2961322).
- [41] Nihan Kahraman and Tulay Yildirim. “Technology independent circuit sizing for fundamental analog circuits using artificial neural networks”. In: *2008 Ph. D. Research in Microelectronics and Electronics*. IEEE. 2008, pp. 1–4.
- [42] Gamze İslamoğlu, Tuğberk Oğulcan Çakici, Engin Afacan, and Günhan Dündar. “Artificial Neural Network Assisted Analog IC Sizing Tool”. In: *2019 16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. 2019, pp. 9–12. DOI: [10.1109/SMACD.2019.8795293](https://doi.org/10.1109/SMACD.2019.8795293).
- [43] Angelo Ciccazzo, Gianni Di Pillo, and Vittorio Latorre. “A SVM Surrogate Model-Based Method for Parametric Yield Optimization”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35.7 (2016), pp. 1224–1228. DOI: [10.1109/TCAD.2015.2501307](https://doi.org/10.1109/TCAD.2015.2501307).
- [44] 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*. 2014, pp. 1–6.
- [45] Oghenekarho Okobiah, Saraju P Mohanty, and Elias Kougianos. “Exploring Kriging for fast and accurate design optimization of nanoscale analog circuits”. In: *2014 IEEE Computer Society Annual Symposium on VLSI*. IEEE. 2014, pp. 244–247.
- [46] Kouros Hakhmaneshi, Nick Werblun, Pieter Abbeel, and Vladimir Stojanović. “BagNet: Berkeley analog generator with layout optimizer boosted with deep neural networks”. In: *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE. 2019, pp. 1–8.
- [47] 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”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41.5 (2022), pp. 1209–1221. DOI: [10.1109/TCAD.2021.3081405](https://doi.org/10.1109/TCAD.2021.3081405).

- [48] Kenneth V Price. “Differential evolution”. In: *Handbook of optimization*. Springer, 2013, pp. 187–214.
- [49] 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”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 33.2 (2014), pp. 169–182. DOI: [10.1109/TCAD.2013.2284109](https://doi.org/10.1109/TCAD.2013.2284109).
- [50] Guy S Handelman, Hong Kuan Kok, Ronil V Chandra, Amir H Razavi, Shiwei Huang, Mark Brooks, Michael J Lee, and Hamed Asadi. “Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods”. In: *American Journal of Roentgenology* 212.1 (2019), pp. 38–43.
- [51] Carl Edward Rasmussen. “Gaussian processes in machine learning”. In: *Summer school on machine learning*. Springer, 2003, pp. 63–71.
- [52] D.H. Wolpert and W.G. Macready. “No free lunch theorems for optimization”. In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 67–82. DOI: [10.1109/4235.585893](https://doi.org/10.1109/4235.585893).
- [53] W. Nye, D.C. Riley, A. Sangiovanni-Vincentelli, and A.L. Tits. “DELIGHT.SPICE: an optimization-based system for the design of integrated circuits”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 7.4 (1988), pp. 501–519. DOI: [10.1109/43.3185](https://doi.org/10.1109/43.3185).
- [54] M. Krasnicki, R. Phelps, R.A. Rutenbar, and L.R. Carley. “MAELSTROM: efficient simulation-based synthesis for custom analog cells”. In: *Proceedings 1999 Design Automation Conference (Cat. No. 99CH36361)*. 1999, pp. 945–950. DOI: [10.1109/DAC.1999.782233](https://doi.org/10.1109/DAC.1999.782233).
- [55] 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”. In: *Integration* 42.2 (2009), pp. 137–148.
- [56] Minghua Li, Dian Zhou, Sheng-Guo Wang, and Xuan Zeng. “Fmssqp: An efficient global optimization tool for the robust design of rail-to-rail op-amp”. In: *2013 IEEE 10th International Conference on ASIC*. IEEE, 2013, pp. 1–4.
- [57] Trent McConaghy and Georges G. E. Gielen. “Globally Reliable Variation-Aware Sizing of Analog Integrated Circuits via Response Surfaces and Structural Homotopy”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 28.11 (2009), pp. 1627–1640. DOI: [10.1109/TCAD.2009.2030351](https://doi.org/10.1109/TCAD.2009.2030351).

- [58] António Canelas, Ricardo Póvoa, Ricardo Martins, Nuno Lourenço, Jorge Guilherme, Joao Paulo Carvalho, and Nuno Horta. “FUZY: A Fuzzy  $c$ -Means Analog IC Yield Optimization Using Evolutionary-Based Algorithms”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.1 (2020), pp. 1–13. DOI: [10.1109/TCAD.2018.2883978](https://doi.org/10.1109/TCAD.2018.2883978).
- [59] Ricardo Martins, Nuno Lourenço, Nuno Horta, Shenke Zhong, Jun Yin, Pui In Mak, and Rui P. Martins. “Design of a 4.2-to-5.1 GHz Ultralow-Power Complementary Class-B/C Hybrid-Mode VCO in 65-nm CMOS Fully Supported by EDA Tools”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 67.11 (2020), pp. 3965–3977. DOI: [10.1109/TCSI.2020.3009857](https://doi.org/10.1109/TCSI.2020.3009857).
- [60] Ricardo Martins, Nuno Lourenço, Nuno Horta, Jun Yin, Pui-In Mak, and Rui P. Martins. “Many-Objective Sizing Optimization of a Class-C/D VCO for Ultralow-Power IoT and Ultralow-Phase-Noise Cellular Applications”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27.1 (2019), pp. 69–82. DOI: [10.1109/TVLSI.2018.2872410](https://doi.org/10.1109/TVLSI.2018.2872410).
- [61] Ricardo Martins, Nuno Lourenço, Nuno Horta, Jun Yin, Pui-In Mak, and Rui P. Martins. “Using EDA Tools to Push the Performance Boundaries of an Ultralow-Power IoT-VCO at 65nm”. In: *2019 16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. 2019, pp. 37–40. DOI: [10.1109/SMACD.2019.8795240](https://doi.org/10.1109/SMACD.2019.8795240).
- [62] Luís Mendes, João Caldinhas Vaz, Fábio Passos, Nuno Lourenço, and Ricardo Martins. “In-Depth Design Space Exploration of 26.5-to-29.5-GHz 65-nm CMOS Low-Noise Amplifiers for Low-Footprint-and-Power 5G Communications Using One-and- Two -Step Design Optimization”. In: *IEEE Access* 9 (2021), pp. 70353–70368. DOI: [10.1109/ACCESS.2021.3078240](https://doi.org/10.1109/ACCESS.2021.3078240).
- [63] Kalyanmoy Deb. “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-2”. In: *IEEE Trans. Evol. Comput.* 6.2 (2002), pp. 182–197.
- [64] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. “Taking the Human Out of the Loop: A Review of Bayesian Optimization”. In: *Proceedings of the IEEE* 104.1 (2016), pp. 148–175. DOI: [10.1109/JPROC.2015.2494218](https://doi.org/10.1109/JPROC.2015.2494218).

- [65] Hakki Mert Torun, Madhavan Swaminathan, Anto Kavungal Davis, and Mohamed Lamine Faycal Bellaredj. “A global Bayesian optimization algorithm and its application to integrated system design”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26.4 (2018), pp. 792–802.
- [66] Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. “Multi-objective bayesian optimization for analog/rf circuit synthesis”. In: *Proceedings of the 55th Annual Design Automation Conference*. 2018, pp. 1–6.
- [67] Shuhan Zhang, Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. “Bayesian optimization approach for analog circuit synthesis using neural network”. In: *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE. 2019, pp. 1463–1468.
- [68] Wenlong Lyu, Pan Xue, Fan Yang, Changhao Yan, Zhiliang Hong, Xuan Zeng, and Dian Zhou. “An efficient bayesian optimization approach for automated optimization of analog circuits”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 65.6 (2017), pp. 1954–1967.
- [69] Shuhan Zhang, Fan Yang, Changhao Yan, Dian Zhou, and Xuan Zeng. “An Efficient Batch-Constrained Bayesian Optimization Approach for Analog Circuit Synthesis via Multiobjective Acquisition Ensemble”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41.1 (2022), pp. 1–14. DOI: [10.1109/TCAD.2021.3054811](https://doi.org/10.1109/TCAD.2021.3054811).
- [70] 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*. PMLR. 2018, pp. 3306–3314.
- [71] Nuno Lourenço, Engin Afacan, Ricardo Martins, Fábio Passos, António Canelas, Ricardo Póvoa, Nuno Horta, and G Dundar. “Using polynomial regression and artificial neural networks for reusable analog ic sizing”. In: *2019 16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. IEEE. 2019, pp. 13–16.
- [72] Ahmed Abuelnasr, Mostafa Amer, Ahmed Ragab, Benoit Gosselin, and Yvon Savaria. “Causal Information Prediction for Analog Circuit Design Using Variable Selection Methods Based on Machine Learning”. In: *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2021, pp. 1–5. DOI: [10.1109/ISCAS51556.2021.9401786](https://doi.org/10.1109/ISCAS51556.2021.9401786).

- [73] Fanshu Jiao and Alex Doboli. “Causal reasoning mining approach to analog circuit verification”. In: *Integration* 55 (2016), pp. 376–383. ISSN: 0167-9260. DOI: <https://doi.org/10.1016/j.vlsi.2016.04.002>.
- [74] Fanshu Jiao, Hao Li, and Alex Doboli. “Modeling and Extraction of Causal Information in Analog Circuits”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37.10 (2018), pp. 1915–1928. DOI: [10.1109/TCAD.2017.2783359](https://doi.org/10.1109/TCAD.2017.2783359).
- [75] Hao Li, Xiaowei Liu, Fanshu Jiao, Alex Doboli, and Simona Doboli. “InnovA: A Cognitive Architecture for Computational Innovation Through Robust Divergence and Its Application for Analog Circuit Design”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37.10 (2018), pp. 1943–1956. DOI: [10.1109/TCAD.2017.2783344](https://doi.org/10.1109/TCAD.2017.2783344).
- [76] Agoston E Eiben and James E Smith. *Introduction to Evolutionary Computing*. Springer, 2015.
- [77] David B Fogel. *Evolutionary algorithms in theory and practice*. 1997.
- [78] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. “Scalable global optimization via local bayesian optimization”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 5496–5507.
- [79] Ivan Zelinka, Vaclav Snasael, and Ajith Abraham. *Handbook of optimization: from classical to modern approach*. Vol. 38. Springer Science & Business Media, 2012.
- [80] Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. “Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)”. In: *Evolutionary computation* 11.1 (2003), pp. 1–18.
- [81] Anne Auger and Nikolaus Hansen. “Tutorial CMA-ES: evolution strategies and covariance matrix adaptation”. In: *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*. 2012, pp. 827–848.
- [82] Rui Murakami, Shoichi Hara, Kenichi Okada, and Akira Matsuzawa. “Design optimization of voltage controlled oscillators in consideration of parasitic capacitance”. In: *2009 52nd IEEE International Midwest Symposium on Circuits and Systems*. IEEE. 2009, pp. 1010–1013.
- [83] Behzad Razavi and Razavi Behzad. *RF microelectronics*. Vol. 2. Prentice hall New York, 2012.

- [84] A. Hajimiri and T. H. Lee. “A general theory of phase noise in electrical oscillators”. In: *IEEE Journal of Solid-State Circuits* 33.2 (1998), pp. 179–194. DOI: [10.1109/4.658619](https://doi.org/10.1109/4.658619).
- [85] Bahram Jafari and Samad Sheikhaei. “Phase noise reduction in a CMOS LC cross coupled oscillator using a novel tail current noise second harmonic filtering technique”. In: *Microelectronics Journal* 65 (2017), pp. 21–30. ISSN: 0026-2692.
- [86] E. Hegazi, H. Sjoland, and A. A. Abidi. “A filtering technique to lower LC oscillator phase noise”. In: *IEEE Journal of Solid-State Circuits* 36.12 (2001), pp. 1921–1930.
- [87] Abhishek Arun. “Design and analysis of CMOS LC voltage controlled oscillator in 32nm SOI process.” In: (2011).
- [88] A. Maesani, G. Iacca, and D. Floreano. “Memetic Viability Evolution for Constrained Optimization”. In: *IEEE Transactions on Evolutionary Computation* 20.1 (2016), pp. 125–144.
- [89] Y. Wang, B. Wang, H. Li, and G. G. Yen. “Incorporating Objective Function Information Into the Feasibility Rule for Constrained Evolutionary Optimization”. In: *IEEE Transactions on Cybernetics* 46.12 (2016), pp. 2938–2952.
- [90] Z. Yan, P. Mak, M. Law, R. P. Martins, and F. Maloberti. “Nested-Current-Mirror Rail-to-Rail-Output Single-Stage Amplifier With Enhancements of DC Gain, GBW and Slew Rate”. In: *IEEE Journal of Solid-State Circuits* 50.10 (2015), pp. 2353–2366.
- [91] Carlos M Fonseca, Luís Paquete, and Manuel López-Ibáñez. “An improved dimension-sweep algorithm for the hypervolume indicator”. In: *2006 IEEE international conference on evolutionary computation*. IEEE, 2006, pp. 1157–1163.
- [92] M. De Souza, A. Mariano, and T. Taris. “Reconfigurable Inductorless Wideband CMOS LNA for Wireless Communications”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 64.3 (2017), pp. 675–685.
- [93] Yoshua Bengio, Olivier Delalleau, and Nicolas Roux. “The Curse of Highly Variable Functions for Local Kernel Machines”. In: *Advances in Neural Information Processing Systems*. Ed. by Y. Weiss, B. Schölkopf, and J. Platt. Vol. 18. MIT Press, 2006, pp. 107–114.
- [94] DC Liu and J Nocedal. “On the limited memory method for large scale optimization: Mathematical Programming B”. In: (1989).

- [95] Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. “Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 7576–7586.
- [96] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.
- [97] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. “Predictive Entropy Search for Efficient Global Optimization of Black-box Functions”. In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger. Vol. 27. Curran Associates, Inc., 2014, pp. 918–926.
- [98] José Miguel Hernández-Lobato, Michael Gelbart, Matthew Hoffman, Ryan Adams, and Zoubin Ghahramani. “Predictive entropy search for Bayesian optimization with unknown constraints”. In: *International conference on machine learning*. PMLR. 2015, pp. 1699–1707.
- [99] Proteek Chandan Roy, Rayan Hussein, Julian Blank, and Kalyan Deb. “Trust-Region Based Multi-objective Optimization for Low Budget Scenarios: 10th International Conference, EMO 2019, East Lansing, MI, USA, March 10-13, 2019, Proceedings”. In: Jan. 2019, pp. 373–385.
- [100] Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai. “When Gaussian process meets big data: A review of scalable GPs”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2020).
- [101] 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)*. IEEE. 2020, pp. 67–72.
- [102] Petros Drineas, Michael W Mahoney, and Nello Cristianini. “On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning.” In: *journal of machine learning research* 6.12 (2005).
- [103] Andrew Wilson and Hannes Nickisch. “Kernel interpolation for scalable structured Gaussian processes (KISS-GP)”. In: *International Conference on Machine Learning*. 2015, pp. 1775–1784.
- [104] Geoff Pleiss, Jacob Gardner, Kilian Weinberger, and Andrew Gordon Wilson. “Constant-Time Predictive Distributions for Gaussian Processes”. In: vol. 80. *Proceedings of Machine Learning Research*. PMLR, 2018, pp. 4114–4123.



- [105] B. Liu, F. V. Fernandez, and G. G. E. Gielen. “Efficient and Accurate Statistical Analog Yield Optimization and Variation-Aware Circuit Sizing Based on Computational Intelligence Techniques”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30.6 (2011), pp. 793–805. DOI: [10.1109/TCAD.2011.2106850](https://doi.org/10.1109/TCAD.2011.2106850).
- [106] Z. Yan, P. Mak, M. Law, and R. Martins. “A 0.016mm<sup>2</sup> 144μW three-stage amplifier capable of driving 1-to-15nF capacitive load with >0.95MHz GBW”. In: *2012 IEEE International Solid-State Circuits Conference*. 2012, pp. 368–370. DOI: [10.1109/ISSCC.2012.6177044](https://doi.org/10.1109/ISSCC.2012.6177044).
- [107] Bum-Kyum Kim, Donggu Im, Jaeyoung Choi, and Kwyro Lee. “A highly linear 1 GHz 1.3 dB NF CMOS low-noise amplifier with complementary transconductance linearization”. In: *IEEE Journal of Solid-State Circuits* 49.6 (2014), pp. 1286–1302.
- [108] Konstantinos Touloupas and Paul P. Sotiriadis. “LoCoMOBO: A Local Constrained Multi-Objective Bayesian Optimization for Analog Circuit Sizing”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2021), pp. 1–1. DOI: [10.1109/TCAD.2021.3121263](https://doi.org/10.1109/TCAD.2021.3121263).
- [109] Kirthivasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabas Poczos. “Parallelised Bayesian Optimisation via Thompson Sampling”. In: *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*. Ed. by Amos Storkey and Fernando Perez-Cruz. Vol. 84. Proceedings of Machine Learning Research. PMLR, 2018, pp. 133–142.
- [110] K. Shang and H. Ishibuchi. “A New Hypervolume-Based Evolutionary Algorithm for Many-Objective Optimization”. In: *IEEE Transactions on Evolutionary Computation* 24.5 (2020), pp. 839–852. DOI: [10.1109/TEVC.2020.2964705](https://doi.org/10.1109/TEVC.2020.2964705).
- [111] Ali Rahimi and Benjamin Recht. “Random features for large-scale kernel machines”. In: *Advances in neural information processing systems* 20 (2007), pp. 1177–1184.
- [112] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema. “Constraint Handling in Multiobjective Evolutionary Optimization”. In: *IEEE Transactions on Evolutionary Computation* 13.3 (2009), pp. 514–525. DOI: [10.1109/TEVC.2008.2009032](https://doi.org/10.1109/TEVC.2008.2009032).
- [113] Eric Bradford, Artur M Schweidtmann, and Alexei Lapkin. “Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm”. In: *Journal of global optimization* 71.2 (2018), pp. 407–438.



- [114] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. “Max-value entropy search for multi-objective bayesian optimization”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 7825–7835.
- [115] Kalyanmoy Deb and Himanshu Jain. “An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints”. In: *IEEE Transactions on Evolutionary Computation* 18.4 (2014), pp. 577–601. DOI: [10.1109/TEVC.2013.2281535](https://doi.org/10.1109/TEVC.2013.2281535).
- [116] S. A. Fordjour, J. Riad, and E. Sánchez-Sinencio. “A 175.2-mW 4-Stage OTA With Wide Load Range (400 pF–12 nF) Using Active Parallel Compensation”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 28.7 (2020), pp. 1621–1629. DOI: [10.1109/TVLSI.2020.2993059](https://doi.org/10.1109/TVLSI.2020.2993059).
- [117] Eduardo C. Garrido-Merchán and Daniel Hernández-Lobato. “Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes”. In: *Neurocomputing* 380 (2020), pp. 20–35. ISSN: 0925-2312.
- [118] Luís Mendes, João Caldinhas Vaz, Fábio Passos, Nuno Lourenço, and Ricardo Martins. “In-Depth Design Space Exploration of 26.5-to-29.5-GHz 65-nm CMOS Low-Noise Amplifiers for Low-Footprint-and-Power 5G Communications Using One-and- Two -Step Design Optimization”. In: *IEEE Access* 9 (2021), pp. 70353–70368. DOI: [10.1109/ACCESS.2021.3078240](https://doi.org/10.1109/ACCESS.2021.3078240).
- [119] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. “Automatic chemical design using a data-driven continuous representation of molecules”. In: *ACS central science* 4.2 (2018), pp. 268–276.
- [120] Robin Winter, Floriane Montanari, Andreas Steffen, Hans Briem, Frank Noé, and Djork-Arné Clevert. “Efficient multi-objective molecular optimization in a continuous latent space”. In: *Chemical science* 10.34 (2019), pp. 8016–8024.
- [121] Rika Antonova, Akshara Rai, Tianyu Li, and Danica Kragic. “Bayesian Optimization in Variational Latent Spaces with Dynamic Compression”. In: *Proceedings of the Conference on Robot Learning*. Vol. 100. Proceedings of Machine Learning Research. PMLR, 2020, pp. 456–465.

- [122] Austin Tripp, Erik Daxberger, and José Miguel Hernández-Lobato. “Sample-efficient optimization in the latent space of deep generative models via weighted retraining”. In: *Advances in Neural Information Processing Systems* 33 (), pp. 11259–11272.
- [123] Aryan Deshwal and Jana Doppa. “Combining Latent Space and Structured Kernels for Bayesian Optimization over Combinatorial Spaces”. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 8185–8200.
- [124] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. 2014.
- [125] Léon Bottou. “Large-scale machine learning with stochastic gradient descent”. In: *Proceedings of COMPSTAT’2010*. Springer, 2010, pp. 177–186.
- [126] Diederik P Kingma, Max Welling, et al. “An Introduction to Variational Autoencoders”. In: *Foundations and Trends<sup>®</sup> in Machine Learning* 12.4 (2019), pp. 307–392.
- [127] Maria Drakaki, Alkis A. Hatzopoulos, and Stylianos Siskos. “CMOS Inductor Performance Estimation using Z- and S-parameters”. In: *2007 IEEE International Symposium on Circuits and Systems*. 2007, pp. 2256–2259. DOI: [10.1109/ISCAS.2007.378732](https://doi.org/10.1109/ISCAS.2007.378732).
- [128] A. R. Aravindh Kumar, Bibhu Datta Sahoo, and Ashudeb Dutta. “A Wideband 2–5 GHz Noise Canceling Subthreshold Low Noise Amplifier”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 65.7 (2018), pp. 834–838. DOI: [10.1109/TCSII.2017.2719678](https://doi.org/10.1109/TCSII.2017.2719678).
- [129] Rohan Lageweg, Vladimir Stojanovic, and Kouros Hakhmaneshi. “Berkeley Open MOS dataBase (BOMB): A Dataset for Silicon Technology Representation Learning”. In: *BWRC Research Lab, Berkeley* (2021).
- [130] Arjun Mishra. “Characterizing Circuits with Deep Embeddings”. In: (2021).
- [131] F. Passos, E. Roca, R. Castro-López, and F.V. Fernández. “An inductor modeling and optimization toolbox for RF circuit design”. In: *Integration* 58 (2017), pp. 463–472. ISSN: 0167-9260. DOI: <https://doi.org/10.1016/j.vlsi.2017.01.009>. URL: <https://www.sciencedirect.com/science/article/pii/S0167926017300524>.

- [132] Thomas Guillod, Panteleimon Papamanolis, and Johann W. Kolar. “Artificial Neural Network (ANN) Based Fast and Accurate Inductor Modeling and Design”. In: *IEEE Open Journal of Power Electronics* 1 (2020), pp. 284–299. DOI: [10.1109/OJPEL.2020.3012777](https://doi.org/10.1109/OJPEL.2020.3012777).
- [133] Ricardo Martins, Nuno Lourenço, Fábio Passos, Ricardo Póvoa, António Canelas, Elisenda Roca, Rafael Castro-López, Javier Sieiro, Francisco V. Fernandez, and Nuno Horta. “Two-Step RF IC Block Synthesis With Preoptimized Inductors and Full Layout Generation In-the-Loop”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 38.6 (2019), pp. 989–1002. DOI: [10.1109/TCAD.2018.2834394](https://doi.org/10.1109/TCAD.2018.2834394).
- [134] Slawomir Koziel. “Fast simulation-driven antenna design using response-feature surrogates”. In: *International Journal of RF and Microwave Computer-Aided Engineering* 25.5 (), pp. 394–402.
- [135] Slawomir Koziel, Jie Meng, John W. Bandler, Mohamed H. Bakr, and Qingsha S. Cheng. “Accelerated Microwave Design Optimization With Tuning Space Mapping”. In: *IEEE Transactions on Microwave Theory and Techniques* 57.2 (2009), pp. 383–394. DOI: [10.1109/TMTT.2008.2011313](https://doi.org/10.1109/TMTT.2008.2011313).
- [136] Chao Zhang, Feng Feng, Venu-Madhav-Reddy Gongal-Reddy, Qi Jun Zhang, and John W. Bandler. “Cognition-Driven Formulation of Space Mapping for Equal-Ripple Optimization of Microwave Filters”. In: *IEEE Transactions on Microwave Theory and Techniques* 63.7 (2015), pp. 2154–2165. DOI: [10.1109/TMTT.2015.2431675](https://doi.org/10.1109/TMTT.2015.2431675).
- [137] José E. Rayas-Sánchez, Slawomir Koziel, and John W. Bandler. “Advanced RF and Microwave Design Optimization: A Journey and a Vision of Future Trends”. In: *IEEE Journal of Microwaves* 1.1 (2021), pp. 481–493. DOI: [10.1109/JMW.2020.3034263](https://doi.org/10.1109/JMW.2020.3034263).
- [138] Suman Ravuri and Oriol Vinyals. “Classification accuracy score for conditional generative models”. In: *Advances in neural information processing systems* 32 (2019).
- [139] Mehdi Kiani and Jalal Kiani. *Conditional Generative Adversarial Networks for Inverse Design of Multi-functional Microwave Metasurfaces*. 2021. DOI: [10.48550/ARXIV.2112.15120](https://doi.org/10.48550/ARXIV.2112.15120). URL: <https://arxiv.org/abs/2112.15120>.
- [140] Daniel Russo and Benjamin Van Roy. “Learning to optimize via information-directed sampling”. In: *Advances in Neural Information Processing Systems* 27 (2014).

- [141] Vassilis Alimisis, Georgios Gennis, Konstantinos Touloupas, Christos Dimas, Marios Gourdouparis, and Paul P Sotiriadis. “Gaussian Mixture Model classifier analog integrated low-power implementation with applications in fault management detection”. In: *Microelectronics Journal* (2022), p. 105510. DOI: <https://doi.org/10.1016/j.mejo.2022.105510>.
- [142] Vassilis Alimisis, Georgios Gennis, Konstantinos Touloupas, Christos Dimas, Nikolaos Uzunoglu, and Paul P. Sotiriadis. “Nanopower Integrated Gaussian Mixture Model Classifier for Epileptic Seizure Prediction”. In: *Bioengineering* 9.4 (2022). DOI: [10.3390/bioengineering9040160](https://doi.org/10.3390/bioengineering9040160).
- [143] Konstantinos Touloupas and Paul Peter Sotiriadis. “Magnitude-only modeling for sigma-delta modulator characterization”. In: *AEU - International Journal of Electronics and Communications* 112 (2019), p. 152936. ISSN: 1434-8411. DOI: <https://doi.org/10.1016/j.aeue.2019.152936>.
- [144] Konstantinos Touloupas, Nikos Chouridis, and Paul P. Sotiriadis. “Local Bayesian Optimization For Analog Circuit Sizing”. In: *2021 58th ACM/IEEE Design Automation Conference (DAC)*. 2021, pp. 1237–1242. DOI: [10.1109/DAC18074.2021.9586172](https://doi.org/10.1109/DAC18074.2021.9586172).
- [145] Konstantinos Touloupas and Paul P Sotiriadis. “Mixed-Variable Bayesian Optimization for Analog Circuit Sizing using Variational Autoencoders”. In: *2022 18th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. IEEE. 2022, pp. 1–4. DOI: [10.1109/SMACD55068.2022.9816185](https://doi.org/10.1109/SMACD55068.2022.9816185).
- [146] Kostas Touloupas and Paul Peter Sotiriadis. “An Optimization-Based Approach for Analog Circuit Technology Migration”. In: *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*. IEEE. 2021, pp. 1–5. DOI: [10.1109/SEEDA-CECNSM53056.2021.9566219](https://doi.org/10.1109/SEEDA-CECNSM53056.2021.9566219).
- [147] Maria-Ermioni Plagaki, Konstantinos Touloupas, and Paul P. Sotiriadis. “Multi-Objective Optimization Methods for CMOS LC-VCO Design”. In: *2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. IEEE. 2021, pp. 1–4. DOI: [10.1109/MOCAST52088.2021.9493409](https://doi.org/10.1109/MOCAST52088.2021.9493409).
- [148] Kostas Touloupas and Paul Peter Sotiriadis. “Analog and RF Circuit Constrained Optimization Using Multi-Objective Evolutionary Algorithms”. In: *2021 IEEE 12th Latin America Symposium on Circuits and System (LASCAS)*. 2021, pp. 1–4. DOI: [10.1109/LASCAS51355.2021.9459145](https://doi.org/10.1109/LASCAS51355.2021.9459145).

- 
- [149] Kostas Touloupas and Paul Peter Sotiriadis. “Mixed-Variable Bayesian Optimization for Analog Circuit Sizing through Device Representation Learning”. In: *Electronics* 11.19 (2022). ISSN: 2079-9292. DOI: [10.3390/electronics11193127](https://doi.org/10.3390/electronics11193127).



# Chapter A'

## Appendix A: Publications

### International Peer-Reviewed Journals

Konstantinos Touloupas and Paul P. Sotiriadis. “LoCoMOBO: A Local Constrained Multi-Objective Bayesian Optimization for Analog Circuit Sizing”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2021), pp. 1–1. DOI: [10.1109/TCAD.2021.3121263](https://doi.org/10.1109/TCAD.2021.3121263)

Kostas Touloupas and Paul Peter Sotiriadis. “Mixed-Variable Bayesian Optimization for Analog Circuit Sizing through Device Representation Learning”. In: *Electronics* 11.19 (2022). ISSN: 2079-9292. DOI: [10.3390/electronics11193127](https://doi.org/10.3390/electronics11193127)

Vassilis Alimisis, Georgios Gennis, Konstantinos Touloupas, Christos Dimas, Marios Gourdouparis, and Paul P Sotiriadis. “Gaussian Mixture Model classifier analog integrated low-power implementation with applications in fault management detection”. In: *Microelectronics Journal* (2022), p. 105510. DOI: <https://doi.org/10.1016/j.mejo.2022.105510>

Vassilis Alimisis, Georgios Gennis, Konstantinos Touloupas, Christos Dimas, Nikolaos Uzunoglu, and Paul P. Sotiriadis. “Nanopower Integrated Gaussian Mixture Model Classifier for Epileptic Seizure Prediction”. In: *Bioengineering* 9.4 (2022). DOI: [10.3390/bioengineering9040160](https://doi.org/10.3390/bioengineering9040160)

Konstantinos Touloupas and Paul Peter Sotiriadis. “Magnitude-only modeling for sigma-delta modulator characterization”. In: *AEU - International Journal of Electronics and Communications* 112 (2019), p. 152936. ISSN: 1434-8411. DOI: <https://doi.org/10.1016/j.aeue.2019.152936>

### International Peer-Reviewed Conferences

Konstantinos Touloupas, Nikos Chouridis, and Paul P. Sotiriadis. “Local Bayesian Optimization For Analog Circuit Sizing”. In: *2021 58th ACM/IEEE Design Automation Conference (DAC)*. 2021, pp. 1237–1242. DOI: [10.1109/DAC18074.2021.9586172](https://doi.org/10.1109/DAC18074.2021.9586172)

Konstantinos Touloupas and Paul P Sotiriadis. “Mixed-Variable Bayesian Optimization for Analog Circuit Sizing using Variational Autoencoders”. In: *2022 18th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. IEEE. 2022, pp. 1–4. DOI: [10.1109/SMACD55068.2022.9816185](https://doi.org/10.1109/SMACD55068.2022.9816185)

Kostas Touloupas and Paul Peter Sotiriadis. “An Optimization-Based Approach for Analog Circuit Technology Migration”. In: *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*. IEEE. 2021, pp. 1–5. DOI: [10.1109/SEEDA-CECNSM53056.2021.9566219](https://doi.org/10.1109/SEEDA-CECNSM53056.2021.9566219)

Maria-Ermioni Plagaki, Konstantinos Touloupas, and Paul P. Sotiriadis. “Multi-Objective Optimization Methods for CMOS LC-VCO Design”. In: *2021 10th International Conference on Modern Circuits and Systems Technologies (MOCASST)*. IEEE. 2021, pp. 1–4. DOI: [10.1109/MOCASST52088.2021.9493409](https://doi.org/10.1109/MOCASST52088.2021.9493409)

Kostas Touloupas and Paul Peter Sotiriadis. “Analog and RF Circuit Constrained Optimization Using Multi-Objective Evolutionary Algorithms”. In: *2021 IEEE 12th Latin America Symposium on Circuits and System (LASCAS)*. 2021, pp. 1–4. DOI: [10.1109/LASCAS51355.2021.9459145](https://doi.org/10.1109/LASCAS51355.2021.9459145)