Optimal Mixing Evolutionary Algorithms for Large-Scale Real-Valued Optimization Including Real-World Medical Applications

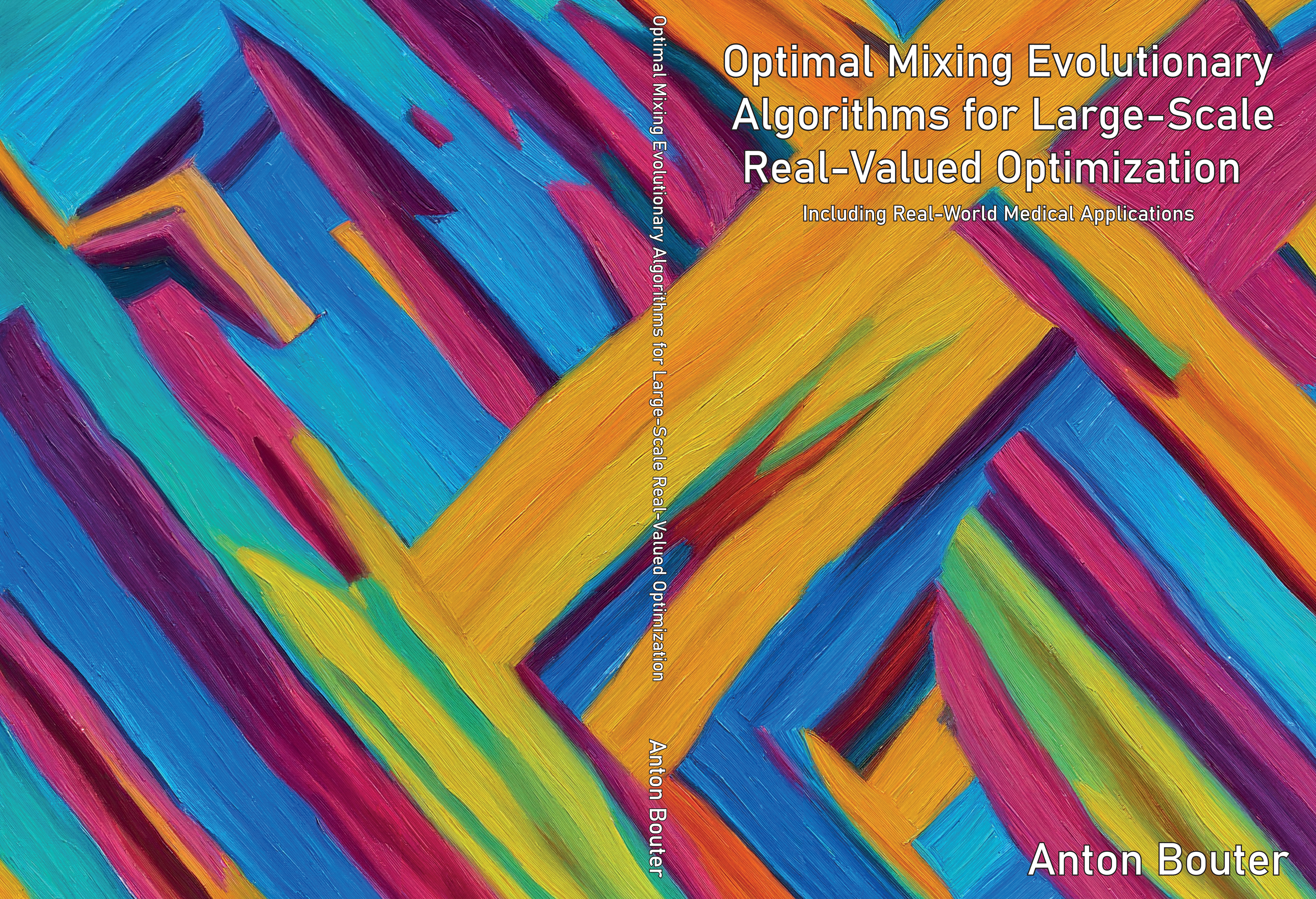
Bouter, Anton

DOI
10.4233/uuid:0e03913c-898e-4392-8de5-072a7ead7fd6

Publication date
2023

Document Version
Final published version

Citation (APA)
Bouter, A. (2023). *Optimal Mixing Evolutionary Algorithms for Large-Scale Real-Valued Optimization: Including Real-World Medical Applications*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:0e03913c-898e-4392-8de5-072a7ead7fd6



Optimal Mixing Evolutionary Algorithms for Large-Scale Real-Valued Optimization

Including Real-World Medical Applications

Anton Bouter

Optimal Mixing Evolutionary Algorithms for Large-Scale Real-Valued Optimization Anton Bouter

Optimal Mixing Evolutionary Algorithms for Large-Scale Real-Valued Optimization

including real-world medical applications

OPTIMAL MIXING EVOLUTIONARY ALGORITHMS FOR LARGE-SCALE REAL-VALUED OPTIMIZATION

INCLUDING REAL-WORLD MEDICAL APPLICATIONS

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 13 februari 2023 om 12:30 uur

door

Pieter Anthonie BOUTER

Master of Science in Computer Science,
Technische Universiteit Delft, Nederland,
geboren te Capelle aan den IJssel, Nederland.

Dit proefschrift is goedgekeurd door de

promotor: Prof. dr. P.A.N. Bosman
copromotor: Dr. T. Alderliesten

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. P.A.N. Bosman,	Centrum Wiskunde & Informatica, Technische Universiteit Delft, promotor
Dr. T. Alderliesten,	Leids Universitair Medisch Centrum, Technische Universiteit Delft, copromotor
Onafhankelijke leden:	
Prof. dr. ir. K. I. Aardal	Technische Universiteit Delft
Prof. dr. L. J. A. Stalpers	Amsterdam Universitair Medische Centra, Universiteit van Amsterdam
Prof. dr. A. D. Todor	Virginia Commonwealth University, Verenigde Staten
Dr. N. Hansen,	Inria, Frankrijk
Prof. dr. M. M. de Weerdt	Technische Universiteit Delft, reservelid

This thesis was part of the research programme IPPSI-TA (project number 628.006.003), financed by the Netherlands Organisation for Scientific Research (NWO) and Elekta, and was carried out in collaboration with the Amsterdam University Medical Centers (UMC), location Academic Medical Center (AMC).

SIKS Dissertation Series No. 2023-05
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Keywords:	Evolutionary algorithms, gene-pool optimal mixing, gray-box optimization, large-scale optimization, real-valued optimization, multi-objective optimization, graphics processing unit, CUDA, brachytherapy, treatment planning, deformable image registration
Printed by:	Ipskamp Printing
Front & Back:	Generated using Stable Diffusion 2 by StabilityAI (`stability.ai`)

ISBN 978-94-6366-648-0

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

CONTENTS

SUMMARY

In recent years, the use of Artificial Intelligence (AI) has become prevalent in a large number of societally relevant, real-world problems, e.g., in the domains of engineering and health care. The field of Evolutionary Computation (EC) can be considered to be a sub-field of AI, concerning optimization using Evolutionary Algorithms (EAs), which are population-based (meta-)heuristics that employ the Darwinian principles of evolution, i.e., variation and selection. Such EAs are historically mainly considered for the optimization of difficult, non-linear problems in a Black-Box Optimization (BBO) setting, because EAs can effectively optimize such problems even when very little is known about the optimization problem and its structure. This is in contrast to optimization methods that are specifically designed for certain problems of which the definition and structure are known, i.e., a White-Box Optimization (WBO) setting.

However, classic EAs are highly unlikely to scale well to optimization problems with large numbers of variables, because their variation operators do not take into account the structure of the optimization problem. As a consequence, there is an increasing probability of violating important dependencies as the number of variables grows, leading to inefficiency. Instead, in Model-Based Evolutionary Algorithms (MBEAs), the structure of the optimization problem is attempted to be captured in a model, which can be exploited to improve the effectiveness of the optimization process. Such a model is generally learned during optimization, but it may also be learned before optimization based on a limited amount of available knowledge about the structure of the optimization problem, if such information is available. If this is the case, the optimization setting is considered to be a Gray-Box Optimization (GBO) setting.

Of particular interest is the Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA), which is an MBEA originally introduced for discrete optimization that captures the dependency structure of the optimization problem in what is called a linkage model. The linkage model is then used to apply variation to (small) subsets of variables, only accepting variation steps that lead to an improvement in fitness. This led to excellent performance and scalability for optimization problems with substructures of limited complexity. Besides exploiting domain knowledge to learn a linkage model, one of the main benefits of GOMEA in a GBO setting is that it can exploit partial evaluations, which are relatively computationally efficient function evaluations that are used to update the fitness of a solution after its partial modification. Such partial evaluations can only be applied to optimization problems for which the structure allows for partial updates of the fitness. Moreover, this requires a GBO setting; otherwise it cannot be assumed that any knowledge about the structure of the optimization problem is known.

The main research goal of this thesis is to design real-valued MBEAs capable of efficiently and effectively scaling to large-scale optimization problems with thousands or even millions of variables. In particular, we consider two real-world medical applications: the optimization of treatment plans for High-Dose-Rate (HDR) brachytherapy, i.e., a type of internal radiation treatment, for prostate cancer, and Deformable Image Registration (DIR), i.e., an application that requires finding an anatomically plausible deformation from a given source image to a given target image with the purpose of determining how tissue has deformed during the period between the acquisition of the source and the target image. Both these applications require the optimization of multiple, conflicting objectives. For such multi-objective optimization problems, EAs are known to be among the state of the art. Furthermore, partial evaluations are possible within each of these applications, enabling their optimization in a GBO setting. However, prior to the start of this thesis, only limited work had been done on exploiting a GBO setting in the domain of real-valued optimization, despite the fact the aforementioned applications, and many other real-world applications, are real-valued.

For this purpose, in Chapter 2, the Real-Valued GOMEA (RV-GOMEA), a variant of GOMEA for the domain of real-valued optimization, is introduced, because GOMEA was previously known to achieve excellent performance in a discrete GBO setting. This real-valued adaptation of GOMEA, like the discrete GOMEA, uses a linkage model to exploit the problem structure for more effective variation. Essentially, variation is applied to multiple subsets of problem variables for each solution. To bring the main strengths of GOMEA to the domain of real-valued optimization, it is combined with the well-known real-valued optimization algorithms AMaLGaM and CMA-ES, leading to two versions of RV-GOMEA. Both versions are shown to achieve similar, excellent performance and scalability in a GBO setting, which can be orders of magnitude better than that of EAs unable to efficiently exploit the GBO setting.

Though RV-GOMEA can achieve excellent performance in a GBO setting, it has difficulties optimizing problems with strong, low-order dependencies between overlapping subsets of variables. Despite the fact that such problems are difficult to optimize, they still allow for partial evaluations, enabling a potentially large increase in performance in a GBO setting. Therefore, in Chapter 3, a new type of linkage model that is better capable of capturing overlapping dependencies, i.e., the conditional linkage model, is introduced to be used with RV-GOMEA. These linkage models are compared to traditional, non-conditional linkage models, and to VkD-CMA, a state-of-the-art variant of CMA-ES for large-scale optimization that uses a decomposition to improve its efficiency. We find that RV-GOMEA with a conditional linkage model achieves the best performance on most optimization problems, and that conditional linkage models perform similar or better than non-conditional linkage models. Therefore, the introduction of conditional linkage models enlarges the class of problems which RV-GOMEA can efficiently and effectively find good solutions for in a GBO setting.

For many real-world applications, it is important that the optimization process is performed as fast as possible. Besides employing algorithmic improvements for this purpose, the use of large-scale computing power can also lead to a substantial increase in efficiency. Because the frequency of single processor cores are nearing their physical limits, parallelization is a very commonly used technique for the use of large-scale computing power. Moreover, EAs can trivially be parallelized due to their population-based search. However, in a GBO setting that allows for partial evaluations, it may be possible to exploit an even larger scale of parallelism, because it is known that the optimization problem consists of a large number of subfunctions that may be computed in parallel. In Chapter 4, the potentially achievable speed-ups of the optimization of real-world problems are analyzed for different large-scale computing hardware. It was found that the best performing parallelization technique depends on the computational complexity, and parallelization potential of the subfunctions. In most scenarios that would occur in real-world problems, i.e., with at least a moderate number of subfunctions, it was found that the use of a Graphics Processing Unit (GPU) is the most efficient. This analysis was then specifically repeated for the application of prostate HDR brachytherapy, for which it was also found that the GPU is the best approach to improve performance through parallelization.

In Chapter 5, GPU parallelization, along with a small number of algorithmic enhancements, is applied to the real-world problem of treatment planning for HDR prostate brachytherapy. In particular, a novel bi-objective formulation of this problem is considered with the purpose of reducing the time required to find high-quality solutions under this formulation using EAs from one hour to a time that is clinically acceptable. The required time before convergence of the optimization, and the quality of the optimized treatment plans, is analyzed for a set of 18 patients for two different settings of the number of dose calculation points, i.e., a setting that determines the accuracy of the radiation dose calculation. It was found that the setting for the number of dose calculation points that previously required one hour of computation time, now requires as little as 30 seconds due to the aforementioned efficiency enhancements when leveraging a GPU. A more accurate setting corresponding to a higher number of dose calculation points resulted in higher quality treatment plans within 3 minutes of computation, which is still considered to be clinically acceptable. Furthermore, additional experiments with even higher numbers of dose calculation points showed no substantial improvement in the quality of the treatment plans being optimized.

For various applications in a GBO setting, there is further potential for parallelization when using RV-GOMEA, because RV-GOMEA applies variation to small subsets of variables. When such subsets of variables are conditionally independent, variation steps may be applied to each of these subsets in parallel. For high-dimensional problems, the fact that this extra potential for parallelizability can be combined with the parallelization of the population, leads to a substantial increase in overall parallelizability. In Chapter 6, a parallelization of the GOM variation operator is implemented on a GPU to exploit this large parallelization porential, and applied to the DIR problem. It was found that similar high-quality results can be achieved up to two orders of magnitude faster due to the parallel application of GOM.

When an optional specific third objective of DIR is used to guide the optimization, the resulting set of solutions is a 3-dimensional approximation front, i.e., a 2-dimensional manifold in 3-dimensional space, from which a clinical expert has to select the deformation that is the most suitable for the respective patient. Intuitively selecting the preferred deformation can be difficult due to the fact that it is difficult to navigate through a 3-dimensional approximation front of solutions, and the fact that this front may have an irregular shape. For this reason, in Chapter 7, a visualization technique is introduced that maps a 3-dimensional front to a set of points within a

2-dimensional simplex. This mapping requires the optimization of a bi-objective problem, with a trade-off between optimally scattering the points and maintaining the relative positions of points on the original front. We tackled this problem with RV-GOMEA in a GBO setting. It was found that the mapping of the 3-dimensional front to a 2-dimensional simplex results in a representation where each of the points is easily distinguishable and located in an intuitive location relative to the nearby points in the original front.

Given the results presented in this thesis, it can be concluded that RV-GOMEA has great potential for large-scale optimization, in particular in case of a GBO setting, and even more so when (large-scale) parallel computing power is available. Its introduction in this thesis therefore opens the door to many potential future real-world applications. This has particularly been acknowledged with the clinical introduction of BRIGHT, i.e., the bi-objective optimization of treatment plans for prostate HDR brachytherapy using GOMEA, in March 2020, at the Amsterdam University Medical Centers (UMC), location Academic Medical Center (AMC).

SAMENVATTING

In de afgelopen jaren is het gebruik van Kunstmatige Intelligentie (KI) gangbaar geworden in een groot aantal maatschappelijk relevante technieken, bijvoorbeeld op het gebied van machinebouw en gezondheidszorg. Het gebied van Evolutionaire Computatie (EC) kan worden beschouwd als een subgebied van KI, met betrekking tot optimalisatie met behulp van Evolutionaire Algoritmen (EA's). Dit zijn op-populatie-gebaseerde (meta-)heuristieken die de Darwiniaanse principes van evolutie gebruiken, d.w.z. variatie en selectie. Dergelijke EA's worden van oudsher voornamelijk overwogen voor de optimalisatie van moeilijke, niet-lineaire problemen in een *Black-Box* Optimalisatie (BBO)-omgeving, omdat EA's dergelijke problemen effectief kunnen optimaliseren, zelfs als er heel weinig bekend is over het optimalisatieprobleem en de structuur ervan. Dit in tegenstelling tot optimalisatiemethoden die specifiek zijn ontworpen voor bepaalde problemen waarvan de definitie en structuur bekend zijn. In dit geval spreken we van een *White-Box* Optimalisatie (WBO)-omgeving.

Het is echter zeer onwaarschijnlijk dat klassieke EA's goed schalen naar optimalisatieproblemen met grote aantallen variabelen, omdat hun variatie-operatoren geen rekening houden met de structuur van het optimalisatieprobleem. Als gevolg hiervan is er een toenemende kans op het schenden van belangrijke afhankelijkheden naarmate het aantal variabelen groeit, wat leidt tot inefficiëntie. In plaats daarvan wordt in ModelgeBaseerde Evolutionaire Algoritmen (MBEA's) geprobeerd de structuur van het optimalisatieprobleem vast te leggen in een model dat kan worden benut om de effectiviteit van het optimalisatieproces te verbeteren. Een dergelijk model wordt over het algemeen geleerd tijdens de optimalisatie, maar kan ook worden geleerd vóór de optimalisatie op basis van een beperkte hoeveelheid beschikbare kennis over de structuur van het optimalisatieprobleem, mits dergelijke informatie beschikbaar is. In dit geval wordt de optimalisatie-omgeving beschouwd als een *Gray-Box* Optimalisatie (GBO)-omgeving.

Van bijzonder belang is het *Gene-pool Optimal Mixing* Evolutionair Algoritme (GOMEA), een MBEA dat oorspronkelijk is geïntroduceerd voor discrete optimalisatie en dat expliciet de afhankelijkheidsstructuur van het optimalisatieprobleem vastlegt in een zogenaamd koppelingsmodel. Het koppelingsmodel wordt vervolgens gebruikt om variatie toe te passen op (kleine) deelverzamelingen van variabelen, waarbij alleen variatiestappen geaccepteerd worden die leiden tot een verbetering van de fitheid, wat leidt tot uitstekende prestaties en schaalbaarheid op optimalisatieproblemen met substructuren van beperkte complexiteit. Naast het benutten van domeinkennis om een koppelingsmodel te leren, is een van de belangrijkste voordelen van GOMEA in een GBO-omgeving dat het gedeeltelijke evaluaties kan benutten. Rekenkundig gezien zijn dit relatief efficiënte functie-evaluaties die worden gebruikt om de fitheid van een oplossing bij te werken na de gedeeltelijke wijziging ervan. Deze gedeeltelijke evaluaties kunnen alleen worden toegepast op optimalisatieproblemen waarvan de structuur gedeeltelijke updates van de fitheid toelaat. Bovendien vereist deze techniek een GBO-omgeving. Als dit niet het geval is, kan niet worden aangenomen dat enige kennis over de structuur van het optimalisatieprobleem bekend is.

Het belangrijkste onderzoeksdoel van dit proefschrift is het ontwerpen van continue MBEA's die efficiënt en effectief kunnen worden geschaald naar grootschalige optimalisatieproblemen met duizenden of zelfs miljoenen variabelen. In het bijzonder beschouwen we twee medische toepassingen: de optimalisatie van behandelplannen voor *High-Dose-Rate* (HDR), ofwel hoog dosistempo, brachytherapie, d.w.z. een type interne bestralingsbehandeling, voor prostaatkanker, en *Deformable Image Registration* (DIR), ofwel vervormbare beeldregistratie, d.w.z. een toepassing waarbij een anatomisch plausibele vervorming moet worden gevonden van een bepaald bronbeeld naar een bepaald doelbeeld met als doel te bepalen hoe weefsel zich heeft verplaatst en/of vervormd tijdens de periode tussen het maken van het bron- en het doelbeeld. Beide toepassingen vereisen de optimalisatie van meerdere, tegenstrijdige doelstellingen, waarvan bekend is dat EA's tot de huidige meest vooraanstaande techniek behoren. Bovendien zijn binnen elk van deze toepassingen gedeeltelijke evaluaties mogelijk, waardoor ze geoptimaliseerd kunnen worden in een GBO-omgeving. Echter, voorafgaand aan de start van dit proefschrift was er slechts beperkt werk verricht op het gebied van het exploiteren van een GBO-omgeving in het domein van continue optimalisatie, ondanks het feit dat de bovengenoemde toepassingen, en vele andere echte toepassingen, continue variabelen hebben.

Met dit onderzoeksdoel voor ogen wordt in Hoofdstuk 2 de *Real-Valued GOMEA* (RV-GOMEA), ofwel Continue GOMEA, een variant van GOMEA voor het domein van continue optimalisatie, geïntroduceerd, omdat van GOMEA bekend was dat het uitstekende prestaties behaalde in een discrete GBO-omgeving. Deze continue aanpassing van GOMEA, evenals de discrete GOMEA, gebruikt een koppelingsmodel om de probleemstructuur te benutten voor effectievere variatie. In wezen wordt hierbij, voor elke oplossing, variatie toegepast op meerdere deelverzamelingen van de probleemvariabelen. Om de belangrijkste sterke punten van GOMEA naar het domein van continue optimalisatie te brengen, wordt het gecombineerd met de bekende continue optimalisatie-algoritmen AMaLGaM en CMA-ES, wat leidt tot twee versies van RV-GOMEA. Van beide versies wordt aangetoond dat ze vergelijkbare, uitstekende prestaties en schaalbaarheid behalen in een GBO-omgeving, die meerdere orden van grootte beter kunnen zijn dan die van EA's die de GBO-omgeving niet efficiënt kunnen benutten.

Hoewel RV-GOMEA uitstekende prestaties kan leveren in een GBO-omgeving, heeft het problemen met het optimaliseren van problemen met sterke lage-orde afhankelijkheden tussen overlappende deelverzamelingen van variabelen. Ondanks het feit dat dergelijke problemen moeilijk te optimaliseren zijn, laten ze toch gedeeltelijke evaluaties toe, wat een potentieel grote prestatieverhoging in een GBO-omgeving mogelijk maakt. Daarom wordt in Hoofdstuk 3 een nieuw type koppelingsmodel, namelijk het voorwaardelijke koppelingsmodel, geïntroduceerd voor gebruik in RV-GOMEA. Dit type koppelingsmodellen, dat beter in staat is om overlappende afhankelijkheden vast te leggen, wordt vergeleken met traditionele, niet-voorwaardelijke koppelingsmodellen en met VkD-CMA, een moderne variant van CMA-ES voor grootschalige optimalisatie, dat een decompositie gebruikt om de efficiëntie van de optimalisatie te verbeteren. We observeren dat RV-GOMEA in combinatie met een voorwaardelijk koppelingsmodel de beste prestaties levert op de meeste optimalisatieproblemen, en dat voorwaardelijke koppelingsmodellen vergelijkbaar of beter presteren dan niet-voorwaardelijke koppelingsmodellen. Daarom vergroot de introductie van voorwaardelijke koppelingsmodellen de klasse van problemen waarop RV-GOMEA efficiënt kan worden toegepast in een GBO-omgeving.

Voor veel toepassingen in reële situaties is het belangrijk dat het optimalisatieproces zo snel mogelijk wordt uitgevoerd. Naast het toepassen van algoritmische verbeteringen, kan ook het gebruik van grootschalige rekenkracht leiden tot een substantiële verhoging van de efficiëntie. Omdat de frequentie van processorkernen haar fysieke limiet nadert, is parallellisatie een veelgebruikte techniek voor het gebruik van grootschalige rekenkracht. Bovendien kunnen EA's triviaal worden geparallelliseerd vanwege het gebruik van een populatie. In een GBO-omgeving die gedeeltelijke evaluaties mogelijk maakt, kan het echter mogelijk zijn om een nog grotere schaal van parallellisme te benutten, omdat bekend is dat het optimalisatieprobleem bestaat uit een groot aantal deelfuncties die parallel kunnen worden berekend. In Hoofdstuk 4 worden de potentieel haalbare versnellingen van reële problemen geanalyseerd voor meerdere soorten computerhardware voor grootschalige berekeningen. Het bleek dat de best presterende parallellisatietechniek afhangt van de vereiste rekentijd en het parallellisatiepotentieel van de deelfuncties. In de meeste scenario's die zich zouden voordoen bij reële toepassingen, d.w.z. met ten minste een matig aantal deelfuncties, bleek het gebruik van een *Graphics Processing Unit* (GPU), ofwel een grafische processor, het meest efficiënt te zijn. Deze analyse werd vervolgens specifiek herhaald voor de toepassing van HDR-brachytherapie voor de prostaat, waarbij ook werd vastgesteld dat het gebruikt van een GPU de beste aanpak is ter verbetering van de efficiïentie d.m.v. parallelizatie.

In Hoofdstuk 5 wordt GPU-parallellisatie, samen met een klein aantal algoritmische verbeteringen, toegepast op het proces van behandelingsplanning voor HDR-prostaatbrachytherapie. In het bijzonder wordt een nieuwe bi-objectieve, d.w.z. met twee tegenstrijdige doelstellingen, formulering van deze toepassing overwogen met als doel de tijd die nodig is om met behulp van EA's hoogwaardige oplossingen te vinden te verminderen van één uur tot een tijd die klinisch acceptabel is. De benodigde tijd voor convergentie van de optimalisatie, en de kwaliteit van de geoptimaliseerde behandelplannen, wordt geanalyseerd voor een verzameling van achttien patiënten voor twee verschillende instellingen van het aantal dosisberekeningspunten, d.w.z. een instelling die de nauwkeurigheid van de dosisberekening bepaalt. Het bleek dat, door de bovengenoemde efficiëntieverbeteringen, de rekentijd voor de instelling van het aantal dosisberekeningspunten die voorheen een uur rekentijd vergde, is gereduceerd tot dertig seconden. Een nauwkeurigere instelling die overeenkomt met een hoger aantal dosisberekeningspunten resulteerde in behandelplannen van hogere kwaliteit binnen drie minuten rekentijd, wat nog steeds als klinisch acceptabel wordt beschouwd. Bovendien lieten aanvullende experimenten met nog hogere aantallen dosisberekeningspunten geen substantiële verbetering zien in de kwaliteit van de geoptimaliseerde behandelplannen.

Voor verschillende toepassingen in een GBO-omgeving is er extra potentieel voor parallellisatie bij gebruik van RV-GOMEA, omdat RV-GOMEA variatie toepast op kleine deelverzamelingen van variabelen. Wanneer dergelijke deelverzamelingen van variabelen voorwaardelijk onafhankelijk zijn, kunnen variatiestappen parallel worden toegepast op elk van deze deelverzamelingen. Gezien het feit dat dit extra potentieel voor parallelliseerbaarheid wordt gecombineerd met de parallellisatie van de populatie, leidt dit voor hoogdimensionale problemen tot een aanzienlijke toename van de algehele parallelliseerbaarheid. In Hoofdstuk 6 wordt een parallellisatie van de GOM-variatie-operator geïmplementeerd op een GPU, ter exploitatie van de grootschalige parallelizatiecapaciteit daarvan, en toegepast op de DIR-toepassing. Het bleek dat vergelijkbare resultaten van hoge kwaliteit tot twee ordes van grootte sneller kunnen worden bereikt door de parallelle toepassing van GOM.

Wanneer een optionele specifieke derde doelstelling van DIR wordt gebruikt om de optimalisatie te begeleiden, is de resulterende reeks oplossingen een driedimensionaal benaderingsfront, d.w.z. een tweedimensionale variëteit in een driedimensionale ruimte, waaruit een klinisch expert de vervorming die het meest geschikt is voor de betreffende patiënt dient te selecteren. Het intuïtief selecteren van de gewenste vervorming kan moeilijk zijn vanwege het feit dat het moeilijk is om door een driedimensionaal benaderingsfront van oplossingen te navigeren en het feit dat dit front een onregelmatige vorm kan hebben. Om deze reden wordt in Hoofdstuk 7 een visualisatietechniek geïntroduceerd die een driedimensionaal front afbeeldt op een reeks punten binnen een tweedimensionale simplex. Deze afbeelding vereist de optimalisatie van een bi-objectief probleem met een afweging tussen het optimaal verstrooien van de punten en het behouden van de relatieve posities van punten op het oorspronkelijke front. Dit probleem is geoptimaliseerd met RV-GOMEA in een GBO-omgeving. Hieruit bleek dat het in kaart brengen van een driedimensionaal front op een tweedimensionale simplex resulteert in een representatie waarin elk van de punten gemakkelijk te onderscheiden is en zich op een intuïtieve locatie bevindt ten opzichte van nabijgelegen punten in het oorspronkelijke front.

Gezien de resultaten die in dit proefschrift worden gepresenteerd, kan worden geconcludeerd dat RV-GOMEA een groot potentieel heeft voor grootschalige optimalisatie, voornamelijk in een GBO-omgeving, maar vooral als er beschikking is over (grootschalige) parallelle rekenkracht. De introductie ervan in dit proefschrift opent daarom de deur naar veel potentiële toekomstige reële toepassingen. Een erkenning hiervan is met name de klinische introductie van BRIGHT, d.w.z. de bi-objectieve optimalisatie van behandelplannen voor HDR-brachytherapie voor prostaatkanker met behulp van GOMEA, in maart 2020, in de Amsterdam Universitair Medische Centra (UMC), locatie Academisch Medisch Centrum (AMC).

1

Introduction

The explosive growth of research on Artificial Intelligence (AI) in recent years is increasingly leading to many societally relevant applications. Even in the mainstream media AI has gained an increasing amount of attention. Today, applications of AI range from self-driving cars [17] to advanced medical diagnostic tools [43], or stock market trading [45].

It is oftentimes mainly the field of deep learning, a sub-field of AI, that is associated with AI, but many more fields of research exist within the field of AI. Many of such forms of AI, in particular machine learning methods, are essentially built on statistics and optimization methods. Similarly, the field of Evolutionary Computation (EC) is a sub-field of optimization, as well as AI, concerning population-based search (meta)heuristics inspired by the Darwinian cornerstones of evolution: variation and selection. Though this field started with algorithms using relatively simple concepts [18, 59], many more sophisticated algorithms have been introduced, often applicable to a wide range of societally relevant applications, e.g., in engineering [36].

These Evolutionary Algorithms (EAs) [13, 37, 59], have mainly been advocated and used for Black-Box Optimization (BBO) scenarios in which very little (e.g., only the types of variables) is assumed to be known about a problem to be solved. Often, these optimization problems are assumed to have properties that make them difficult to optimize for conventional, more efficient, White-Box Optimization (WBO) techniques like gradient descent. Such properties include non-differentiable objectives, non-convex problem landscapes, or multiple conflicting objectives. Even (real-world) optimization problems for which the exact definition is known, can have such properties, making them very difficult to optimize, and thus still warranting the use of advanced search (meta)heuristics like EAs. Classic EAs can effectively optimize such problems, though they do not exploit the fact that the structure of the optimization problem may very well be known, and they generally do not scale to problems with a large number of problem variables. The variation operators used by classic EAs are often said to be *blind*, meaning that they do not take into account the structure of the optimization problem. Knowing the structure of your problem and having the ability to exploit this knowledge can make a very big impact on the efficiency of an EA [113]. For this reason, in Model-Based EAs (MBEAs) [49, 62, 115] the structure of the optimization problem is attempted to be captured in a model that can be learned during optimization, allowing the application of variation in a way that better fits the optimization problem. Of particular interest is the relatively recent type of MBEA known as the Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA) [115] that learns an explicit dependency model and uses this model to exploit the dependency structure of all problem variables. This is interesting, because large optimization problems are generally not completely jointly dependent, but rather have substructures of a bounded complexity that are jointly dependent. Therefore, exploiting such substructures has high potential for improved scalability or even parallelizability. Prior to the start of this thesis, there was only limited work on this for real-valued variables, despite the fact that many real-world problems are indeed real-valued.

1.1. GUIDE TO THE THESIS

The main research goal of this thesis was to design real-valued MBEAs capable of efficiently scaling to large-scale optimization problems with thousands or even millions of variables. In particular, the goal was to understand to what extent optimal mixing could be used for real-valued variables and how modern means of parallel computing, such as Graphics Processing Units (GPUs), could be exploited. At the same time, this thesis was part of a project focused on a real-world medical application: the treatment of prostate cancer using High-Dose-Rate (HDR) prostate brachytherapy [32], a form of internal radiation treatment. A second application of interest is that of Deformable Image Registration (DIR) [109], which concerns finding an anatomically plausible non-rigid deformation between a pair of medical images.

The research described in this thesis was part of the research program named *ICT-based Innovations in the Battle against Cancer - Next-Generation Patient-Tailored Brachytherapy Cancer Treatment Planning*, a collaboration between researchers from Centrum Wiskunde & Informatica (CWI) and the Amsterdam University Medical Centers (Amsterdam UMC), location Academic Medical Center (AMC), financed by the Netherlands Organisation for Scientific Research (NWO) and Elekta. As a result of this project, treatment planning with BRIGHT (BRachytherapy via artificially Intelligent GOMEA-Heuristic based Treatment planning) for prostate HDR brachytherapy was clinically introduced in the Amsterdam UMC, location AMC, on March 17, 2020. This thesis describes various works of research that were essential for the development and clinical introduction of BRIGHT.

This thesis is structured in such a way that fundamental research on MBEAs is discussed first. In Chapter 2 the real-valued variant of GOMEA, Real-Valued GOMEA (RV-GOMEA), is introduced, which is a real-valued MBEA that explicitly models problem dependencies, and uses Gaussian probability distributions estimated with maximum likelihood to explore the search space. RV-GOMEA is then enhanced with a different type of dependency model, one that can better deal with optimization problems with strong overlapping dependencies, in Chapter 3. In Chapter 4, the groundwork is laid for the use of state-of-the-art parallel computing methods within RV-GOMEA, by comparing different methods in different scenarios ranging from low to high potential for parallelization. In Chapter 5, these findings are then applied to speed up the application of RV-GOMEA to treatment planning for prostate HDR brachytherapy. Chapter 6 shows how an even larger scale of parallelization may be achieved, in particular on the DIR application. In Chapter 7, RV-GOMEA is used to improve the intuitiveness of exploring a 3-dimensional solution set that is the result of the optimization of the DIR application, by finding an intuitive mapping of these solutions to a 2-dimensional space. In the remainder of this introduction, the research questions of these aforementioned chapters are introduced, and related work is discussed. Finally, in Chapter 8, it is discussed to what degree these research questions have been answered, which research questions remain, and if the work described in this thesis has led to new research questions.

1.2. OPTIMIZATION

A (real-valued, single-objective) optimization problem is primarily defined by a function $f : \mathbb{R}^\ell \rightarrow \mathbb{R}$ mapping from an ℓ-dimensional problem space to a one-dimensional objective space. The problem space consists of the problem variables $\boldsymbol{X} = [X_1, X_2, \ldots, X_\ell]$, which are indexed through $\boldsymbol{I} = [1, 2, \ldots, \ell]$. This problem space may be constrained by a constraint function $g : \mathbb{R}^\ell \rightarrow \mathbb{R}$ that maps a potential solution to a constraint value that indicates the (in)feasibility of the solution. A potential solution to such an optimization problem is an ℓ-dimensional vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_\ell]$ that represents the realized values of all problem variables. The optimum of f is denoted $f^* = \min_{\boldsymbol{x} \in \mathbb{R}^\ell} f(\boldsymbol{x})$.

Exact algorithms, e.g., (integer) linear programming [63, 119], branch and bound [26, 74], or dynamic programming [14], offer a provable guarantee of finding an optimal solution, but are either restricted to a very narrow class of problems, or have very high computational complexity, meaning that their execution times increase dramatically with the problem size, making them virtually unusable for many problems of even moderate size.

Furthermore, real-world problems, especially when accounting for realistic details in the problem, may have all kinds of properties that make them difficult to optimize, e.g., a non-smooth, non-continuous problem landscape. In order to make the optimization process easier, the real-world problem may be formulated in a way that results in a problem landscape with desirable properties, for example by smoothing the problem landscape. This will, however, result in a problem landscape that no longer matches the true problem landscape of the real-world problem, which could result in misalignment of the global optimum of the reformulated problem and that of the true, real-world optimum. The use of an exact algorithm therefore no longer guarantees that the global optimum of the true problem landscape will be found.

Non-exact algorithms, i.e., approximation algorithms and heuristics, do not guarantee to find the optimum of the optimization problem, but they are more efficient or more generally applicable than exact algorithms. Approximation algorithms guarantee to approximate the optimum with a given precision, whereas heuristics may not give such a guarantee. Approximation algorithms and heuristics are generally designed for a particular type of problem, making them White-Box Optimization (WBO) methods. Metaheuristics, e.g., evolutionary algorithms [13, 37, 59], simulated annealing [118], or tabu search [48], are a different type of algorithms that are instead often designed for problems of which little or no domain knowledge is available.

A Gray-Box Optimization (GBO) setting is one where a limited amount of domain knowledge is used to aid the optimization process. Various kinds of GBO settings have been used in the field of EAs as a way of enhancing the performance of these metaheuristics on specific problems, e.g., by introducing problem-specific variation operators [41], applying function transforms [122], or using a more efficient way of calculating the objective value [11].

1.3. MULTI-OBJECTIVE OPTIMIZATION

Appropriately modeling a real-world problem can be a difficult task. An important factor contributing to this may be that the real-world problem does not only have a single objective (of interest), but it has multiple conflicting ones. We denoted this as a vector of objectives $f(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})]$. In this case, the optimization problem would be considered a multi-objective, or multi-criterion, problem [40, 53, 93]. Given that the objectives are conflicting, no single solution exists that achieves the optimal value for each of the objectives. Because this hypothetical optimal solution is impossible to achieve, a trade-off must be made between each of the objectives, and each solution achieving an optimal trade-off is said to be Pareto optimal. For any solution that is Pareto optimal, no solution exists that Pareto dominates it, meaning that no solution exists that can be considered to have a strictly higher quality from a multi-objective perspective. A solution is considered to dominate a different solution when it is at least equally good in all objectives, and strictly better in at least one objective. The set of Pareto optimal solutions is called the Pareto set, and the set of points that the Pareto set maps to in objective space is called the Pareto front.

Examples of multi-objective optimization may be found in various real-world problems, such as radiation treatment planning [27, 30, 81], vehicle routing [64, 68], other types of planning problems [4, 52], or a range of problems in the field of machine learning [7, 47, 66].

Multiple approaches exist for the optimization of multi-objective optimization problems. One fairly straightforward approach is using linear scalarization to define a weighted average of all objectives, and using a single-objective optimization method that produces a single solution [65]. A downside of this approach is that it may be difficult and unintuitive to determine appropriate weights for each of the objectives. Therefore, finding the solution with the best trade-off between the multiple objectives may require many repetitions of the single-objective optimization method.

A different approach is one where the goal of the optimization method is not to find just one appropriate solution, but a set of appropriate solutions with different trade-offs. A set of solutions is desired that is as close as possible to the Pareto front, i.e., a Pareto-approximation front, and which uniformly covers the Pareto front. The quality of a Pareto-approximation front can be evaluated using different metrics, such as the Inverted Generational Distance (IGD) [23, 34], or hypervolume [128]. From the Pareto-approximation front, an appropriate solution is selected by a decision maker, generally an expert in the field of the application, in an a-posteriori decision-making process [15, 31]. By allowing an expert to select from the set of optimized solutions, valuable experience is incorporated in the decision-making process, some of which would be difficult to formalize within one of the objectives of the optimization model.

The decision-making process may be guided in various ways, e.g., by incorporating the preferences of the decision maker a priori and guiding the search towards these preferences [101, 112], or by guiding the decision maker during the process [50, 104].

To find a set of solutions for an a-posteriori decision making process, it is possible to repeat single-objective scalarization a large number of times with different weights, and combining the results into a set of solutions that approximates the Pareto set. Such an approach, however, is tied to a number of disadvantages. Firstly, it may require a large amount of computation time, because the optimization procedure is repeated a large number of times. A second disadvantage is the fact that an even distribution of weights may not lead to an evenly distributed set of solutions along the Pareto front [39]. Thirdly, it is impossible to find solutions on concave parts of a Pareto front when using linear scalarization [40]. Instead, population-based search heuristics, such as EAs, have a natural synergy with the desire for a large set of diverse, high-quality solutions that arises in multi-objective optimization. Such solutions are generally maintained during optimization in a set of non-dominated solutions, named the elitist archive [23, 80, 127]. Mainly for this reason, EAs are considered to be among the state of the art for multi-objective optimization [38].

1.4. Evolutionary Algorithms

EAs [13, 37, 59] are metaheuristics inspired by key Darwinian principles of evolution: variation and selection. In an EA, a population of potential solutions is subject to repeated variation and selection, with the purpose of exploring diversity and exploiting quality of above-average solutions, respectively, so as to improve the quality of the population as a whole over time, i.e., across multiple so-called generations. A solution in the population is often referred to as an individual, and the problem variables of a solution map to the alleles of an individual, which together form its genotype. The quality of a solution is often referred to as its fitness value. Often, a one-to-one mapping from the problem variables to the alleles of an individual, and from the objective value of a solution to its fitness value, is used in EAs. Different mappings can, however, be used in order to achieve different search behavior.

During each generation of the EA, the population is subject to variation, resulting in a number of offspring solutions. Selection is then applied to determine which of the solutions in the current population and the offspring will survive, and which will be discarded. The population of the EA in the next generation will then consist of all solutions that are selected.

Variation in an EA generally consists of crossover and/or mutation. During variation, genetic material is exchanged between more than one individual. An example variation operator is uniform crossover, which is applied to two parent solutions and builds two offspring solutions; one for which each allele is selected from either of the parents with equal probability, and its complement. In one-point crossover, a crossover point is randomly selected within the genotype of the parents, and the parents exchange all genes to the right of the crossover point. Examples of the application of these two types of crossover are displayed in Figure 1.1.

The application of most EAs to an optimization problem does not require any domain-specific knowledge. Instead, only the definition of an objective function is

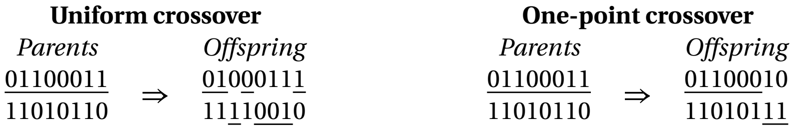


Figure 1.1: Uniform and one-point crossover applied to examples of arbitrary solutions of 8 binary variables.

required, making them straightforwardly usable for BBO without applying relaxations to the optimization problem, which may change the landscape of the problem or even the location of the global optimum. Furthermore, due to the population-based search, EAs are not as prone to get stuck in local optima as single-solution-based optimization methods, are well-suited for the optimization of problems with non-smooth landscapes, and are considered to be among the state-of-the-art for multi-objective optimization [39]. Thanks to these properties, EAs have been successfully applied to difficult real-world problems, such as antenna design [60] and various other engineering applications [36].

Though EAs are powerful algorithms that can deal with very difficult problems, this comes at the cost of being relatively computationally expensive. Furthermore, the performance of EAs greatly depends on the effectiveness of the variation operator for the respective problem. For example, the standard Genetic Algorithm (GA) requires an exponentially large population size for various difficult problems, due to its ineffective mixing operator [113].

Various types of methods have been introduced to improve the scalability of EAs on difficult optimization problems, for example by decomposing the problem [100], or by learning a model that approximates the problem structure in order to improve the efficiency of variation [49, 62, 115]. Such EAs are named Model-Based Evolutionary Algorithms (MBEAs).

Some of the earliest MBEAs use an explicit probability distribution to model the search space and its dependency structure, which is learned based on promising solutions. These MBEAs are classified as Estimation of Distribution Algorithms (EDAs) [58, 79]. In contrast to classic EAs that apply variation through crossover, EDAs explore the search space by sampling new solutions from this probability distribution that is estimated based on promising solutions in the population. The first EDAs, e.g., the Univariate Marginal Distribution Algorithm (UMDA) [90] or compact Genetic Algorithm (cGA) [56], estimated univariate probability distributions, thereby not modeling interactions between problem variables. Bivariate and multivariate interactions were later modeled in, e.g., the Bivariate Marginal Distribution Algorithm (BMDA) [98] and the Extended Compact Genetic Algorithm (ECGA) [57], respectively. The ECGA, however, decomposes the optimization problem into partitions of dependent variables, which is unable to capture large numbers of overlapping interactions between variables. More complex interactions can be modeled by Bayesian networks, which are used by the Bayesian Optimization Algorithm (BOA) [97] and hierarchical BOA (hBOA) [96]. It can, however, be very expensive to learn and store an entire probability distribution. A more recent type of MBEAs, in particular the algorithms in the family of the Gene-pool Optimal Mixing Evolutionary Algorithm

(GOMEA) [115], therefore does not learn an entire probability distribution, but instead learns a model that explicitly models which dependencies are present in the optimization problem.

1.5. GENE-POOL OPTIMAL MIXING EVOLUTIONARY ALGORITHMS

GOMEA is a type of MBEA that was originally introduced for single-objective [115] and multi-objective [83] discrete optimization. The multi-objective variant of GOMEA is referred to as the Multi-Objective GOMEA (MO-GOMEA). GOMEA was introduced as the successor to the Linkage Tree Genetic Algorithm (LTGA) [114], but it can be considered to be a family of algorithms based around the Gene-pool Optimal Mixing (GOM) variation operator. The GOM variation operator applies variation to subsets of variables defined by a linkage model, which explicitly models dependencies between variables, also referred to as linkage, present in the optimization problem. Such a linkage model can be set a priori based on domain knowledge, or, in a BBO setting, learned at the start of each generation.

The linkage model used by GOMEA is defined by a Family Of Subsets (FOS) $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_q\}$, with $\mathcal{F}_i \subseteq \boldsymbol{I}$ for each linkage set $\mathcal{F}_i \in \mathcal{F}$. The simplest linkage model is the univariate model, i.e., $\mathcal{F} = \{1, 2, \ldots, \ell\}$, modeling complete independence between problem variables, which would fit perfectly in case of a completely separable problem. A marginal product FOS models groups of dependent variables with no overlap, i.e., $\mathcal{F}_i \cap \mathcal{F}_j = \emptyset$ for $\mathcal{F}_i, \mathcal{F}_j \in \mathcal{F}$. This means that each variable is present in exactly one linkage set. A FOS capable of capturing more complex dependencies is the linkage tree model, as displayed in Figure 1.2, which is a hierarchical model that contains linkage sets ranging from very small to large sizes in order to capture both low-order and high-order dependencies. Each multivariate linkage set in the linkage tree is the union of two smaller sets. A linkage tree is often built by first initializing it as the collection of all possible univariate linkage sets. Iteratively, the two linkage sets with the highest joint dependency, often measured in terms of mutual information, are merged and added to the linkage tree, until no more merges are possible. A linkage set that has been merged into a larger linkage set remains in the linkage tree, but can no longer be a candidate to be merged with a different linkage set. For discrete optimization, the linkage set containing all problems variables is omitted.

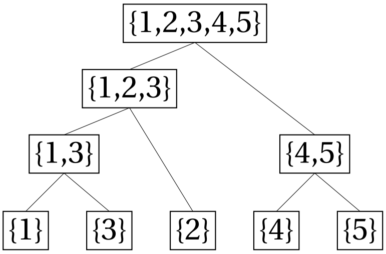


Figure 1.2: An example of a linkage tree for 5 problem variables, where each node indicates that a linkage set exists containing the problem variables with the denoted indices.

During each generation, GOM is applied to each individual in the population. In the discrete case, the application of GOM to an individual, i.e., the parent, consists of the iteration over all linkage sets in a random order. For each linkage set, a random donor is selected from the population, and the linkage set is used as a crossover mask to copy variables from the donor to the parent. In case of single-objective optimization, this modification of the parent is then evaluated, and accepted if it resulted in a solution with an equal or better fitness. Otherwise, the parent is returned to its previous state. In MO-GOMEA, a modification is accepted when it results in a solution that dominates the parent solution, or it is non-dominated by the elitist archive.

If an individual has not been modified during GOM for an entire generation, or it has not improved for a specified number of generations, the forced improvement procedure is applied [24]. During this procedure, GOM is applied to the individual using the elitist solution, i.e., the best known solution, as the donor, but the forced improvement procedure is terminated directly after any such iteration of GOM is accepted. If no such iteration is accepted, the individual to which forced improvements are applied, is replaced by a copy of the elitist solution. In the MO-GOMEA, the forced improvement procedure is applied in a similar way, but the solution in the elitist archive that is closest to the parent solution (Euclidean distance in objective space) is used.

The goal of the GOM variation operator is to prevent the disruption of building blocks, i.e., configurations of small blocks of variables with above-average fitness, while still mixing the solution contents as much as possible by applying variation to dependent subsets of problem variables. Due to the fact that only variation steps with non-decreasing fitness are allowed during GOM, an optimal building block present in one of the parents is guaranteed to be reproduced in the offspring. In contrast, in a standard GA, optimal building blocks are more likely to be disrupted due to random variation being applied to all variables and due to collateral noise coming from other parts of the solution and their contribution to fitness, leading to an exponentially large required population size for various difficult optimization problems [113]. Efficient mixing with GOM, however, relies on knowing a priori or otherwise learning the right linkage structure, for which many techniques have been introduced over the years [24, 25, 33, 115].

GOMEA has strong synergy with a GBO setting that allows for partial evaluations, because GOMEA frequently applies modifications to small subsets of variables, which can be efficiently evaluated in such a setting. This GBO setting requires that a decomposition of the objective function into (possibly overlapping) subfunctions is known. Each subfunction is then still considered to be a black-box function over a known subset of variables. Knowing such a decomposition does, however, not mean that the problem is trivial in any way, as such problems can still be non-smooth, non-separable, and deceptive.

1.6. REAL-VALUED EVOLUTIONARY ALGORITHMS

One of the first EAs for real-valued optimization was the Evolution Strategies (ES) [16, 102], which mainly employs mutation and selection. The mutation strength, or step size in each individual separately, of ES is governed by a Gaussian distribution for which the standard deviation is encoded using one or more strategy parameters, which also undergo a fully predetermined form of mutation. Through selection better strategy parameters are automatically preferred, since they result in better individuals. This process is called self-adaptation [107].

The indirect optimization of strategy parameters through self-adaptation, however, has various shortcomings, which can mainly be overcome by derandomization of the strategy parameters [55, 95]. This derandomization has led to the introduction of the Covariance Matrix Adaptation Evolution Strategies (CMA-ES) [54, 55], which has been the subject of a large amount of research [120, 121], and is considered to be among the state of the art in BBO. CMA-ES uses a (multivariate) Gaussian distribution based on a full covariance matrix to sample new solutions, which is iteratively updated each generation based on the selection and following various update rules that are based, among other things, on previously evaluated solutions and their path through solution space in previous generations.

As CMA-ES maintains and updates a Gaussian distribution to sample new solutions, CMA-ES can be considered to be an EDA. It was however developed along the research line into ES. The EDA research line, that originated for discrete search spaces, had a separate development into real-valued optimization [22]. For example, the Adapted Maximum-Likelihood Gaussian Model Iterated Density Estimation Evolutionary Algorithm (AMaLGaM-ID𝔼A, or AMaLGaM for short) [20, 21] uses Gaussian probability distributions to model the distribution of the problem variables. The Gaussian probability distribution used by AMaLGaM is estimated with maximum likelihood based on the selection of the current generation, whereas the Gaussian probability distribution estimated by CMA-ES also depends on the distributions of previous generations. Alternatively, the real-coded Bayesian Optimization Algorithm (rBOA) [1] uses a Bayesian factorization of a mixture of Gaussian probability distributions.

To have a high search capacity (i.e., to be able to solve a large class of problems), the capacity of the model that determines the most promising areas of the search space to sample new solutions in should be as large as needed. For CMA-ES and AMaLGaM, an ℓ-dimensional covariance matrix can be used to have the largest model capacity (within the confines of the Gaussian parametric model). Larger capacity often comes with larger complexity, however, and the computational complexity of the construction of each of these models is at least $\mathcal{O}(\ell^3)$, which could become a bottleneck for large-scale optimization problems with relatively efficient evaluations. Various variants of CMA-ES now exist that are aimed at improving its scalability for large-scale optimization, e.g., Limited-Memory CMA-ES (LM-CMA-ES) [78] or VkD-CMA [2, 3]. Though these variants have been shown to be very efficient at BBO, they lack the capacity to efficiently and straightforwardly exploit domain knowledge that may be available in a GBO setting, as discussed in Section 6.2. Being able to do so, may allow for large additional speed-ups.

An adaptation of GOMEA for the real-valued domain, name Real-Valued GOMEA (RV-GOMEA) [11, 12], was recently introduced, which is capable of exploiting exactly this type of domain knowledge in a GBO setting. The main benefit of RV-GOMEA is that it can use partial evaluations to efficiently evaluate variation steps of small subsets of variables, without requiring the evaluation of the entire solution. A second benefit is that domain knowledge may be incorporated in the linkage model of RV-GOMEA, which indicates the dependency structure of the optimization problem. In Chapter 2 this work is extended by exploring how the concepts of GOMEA can be used for real-valued optimization in different ways, based on different real-valued optimizers. Therefore, we attempt to answer the following research question in Chapter 2:

Research Question: Chapter 2

How can the key strengths of GOMEA be exploited in real-valued optimization to achieve highly scalable GBO?

In a GBO setting that allows for partial evaluations, a decomposition of the objective function into (possibly overlapping) subfunctions is known. Each subfunction is treated as a black-box function over a given subset of problem variables. From this, a Variable Interaction Graph (VIG) [116] can be created that models each problem variable as a node, and has an edge between each pair of dependent variables, i.e., variables that are input to the same subfunction. An example of this is shown in Figure 1.3. If the VIG of a problem consists of disconnected subgraphs, this means that the problem is separable, and each of the separable problems can be optimized independently. If this results in a large number of small sub-problems, these sub-problems can straightforwardly be optimized in a parallel or interleaved way with a BBO solver. Therefore, partial evaluations will not lead to substantial improvements in efficiency over such an approach. Also, for problems where all variables are jointly dependent, partial evaluations are not expected to result in large improvements in efficiency, because partial evaluations require variation to be applied to a small number of variables, but this type of variation would not be effective for a fully jointly dependent problem.

Instead, the potential increase in performance gained by partial evaluations is largest when the VIG is a sparsely connected graph, because the problem is then non-separable, but partial evaluations are relatively efficient compared to full evaluations. In problems with such a VIG, each problem variable is (strongly) dependent on a small number of neighboring problem variables, but not directly dependent on other variables, resulting in a chain of dependencies. Problem variables that are not directly dependent, but are connected by such a chain of dependencies, are conditionally dependent.

An apparent problem of tackling such problems with GOMEA is that the application of variation to any subset of variables will disregard a number of potential dependencies with variables outside of this subset, because no independent subset of variables exists if the problem is non-separable. For this reason, when a strong chain of

$$f(x) = f_1(x_1, x_2, x_3) + f_2(x_3, x_4) + f_3(x_4, x_5) + f_4(x_5, x_6, x_7, x_8)$$

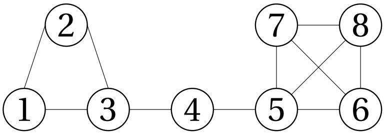


Figure 1.3: An example of an optimization function $f(x)$, defined in terms of a sum of subfunctions, and its corresponding VIG.

dependencies exists, applying GOM to small subsets of variables is unlikely to be efficient. Only applying variation to large subsets of variables would, however, not get the most benefit out of the efficient partial evaluations enabled by the GBO setting. It is therefore unclear how much benefit can be obtained from a GBO setting for problems with overlapping strong dependencies. We therefore aim to answer the following research question in Chapter 3:

Research Question: Chapter 3

To what extent is RV-GOMEA limited in its effectiveness to exploit partial evaluations in a GBO setting in case of optimization problems with overlapping strong dependencies, and how could we improve this?

1.7. Large-Scale Parallelization

Beside algorithmic advancements, using a large amount of parallel computing power can lead to non-negligible efficiency enhancements of many computational processes. In fact, much of the popularity of modern-day deep learning is due to its suitability to leverage the massive parallel computing power of today's affordable GPUs. For EAs, due to their population-based search, there is inherent potential for parallelization, because the evaluation of each individual can be done in parallel. Various parallel EAs have therefore been introduced in the past [5, 111].

Given the fact that the frequency of CPUs is nearing its physical limit [9], due to a cubic relation with their power use [28], GPUs have been gaining an increasing amount of attention for the purpose of high-performance computing [29]. This has been reinforced by the popularity of deep learning, and the essential role of GPUs in their application [88]. Also other real-world problems, such as biochemical simulation [126] or weather modeling [87], rely on the large-scale computing power of GPUs. Furthermore, in the field of EC, GPUs have been used to speed up various EAs [77, 123]. However, none of these works consider the additional parallelization potential that comes from problem decomposition.

In a GBO setting, of which a more detailed description can be found in Section 6.2, the potential of parallelization is increased even further, because the evaluation of different subfunctions may be performed in parallel. This results in a speed-up of the evaluation function, which is not only a benefit for the GOMEA family of algorithms, known to exhibit a synergistic effect with partial evaluations, but also for EAs targeted at BBO. The limiting factor with this increase in parallelizability is the number of subfunctions that are required to be simultaneously evaluated, which depends on the number of modified variables. Altogether there is undeniable large parallelization potential within the GOMEA family, especially for the GBO setting. We therefore aim to answer the following research question in Chapter 4:

Research Question: Chapter 4

Which parallelization techniques are the most suitable for large-scale optimization in a GBO setting, particularly for GOMEA, and in which scenarios?

1.8. High-Dose-Rate Prostate Brachytherapy

Brachytherapy [32] is a type of internal radiation treatment used to treat patients with various types of cancer, e.g., cervix [91], breast [70], or prostate cancer [124]. During brachytherapy, radiation is delivered using a radioactive source in or close to the tumor. This type of treatment can either be used as the primary radiation treatment modality, or as a boost after treatment with external-beam radiotherapy [35, 61]. In particular, the focus of this thesis is on HDR prostate brachytherapy [35], though applications of this research to other tumor sites are closely related and are interesting areas of future research.

In prostate HDR brachytherapy, as performed in the Amsterdam UMC, radiation is delivered to the prostate using a radioactive source that is moved through a number of catheters, generally ranging from 14 to 20, that are temporarily surgically inserted into the prostate. These catheters are inserted through the transperineal area guided by a rectal ultrasound probe. Following the placement of the catheters, an image of the tumor site is captured using Magnetic Resonance Imaging (MRI), which is used to delineate the target volumes, and the Organs At Risk (OARs), and to reconstruct the catheters. During treatment, the source is guided through each catheter one by one, halting it at predefined positions, named dwell positions, for a specific amount of time, named the dwell time. As a result of this, radiation dose is deposited inside the patient and a dose distribution is formed. Figure 1.4 shows a schematic three-dimensional view of the catheters that are inserted into the prostate and the seminal vesicles, and the surrounding organs. An example of a dose distribution is shown in Figure 1.5.

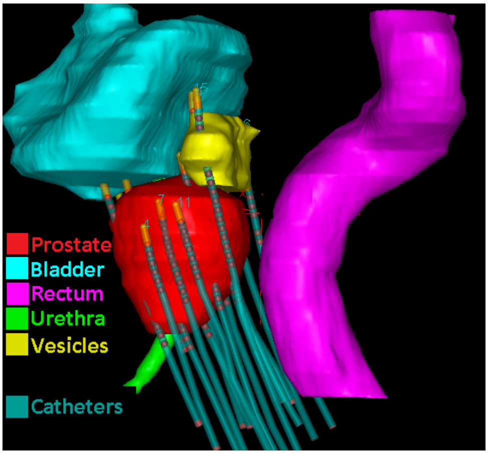


Figure 1.4: A schematic three-dimensional view of the catheters that are inserted into the prostate and the seminal vesicles, and the surrounding organs. Dwell positions are marked by red spheres.

One of the goals of prostate HDR brachytherapy is to deliver a sufficient amount of radiation to the target volumes, e.g., the prostate. In many cases, also the seminal vesicles, a pair of glands responsible for producing many of the constituents of semen that are located between the bladder and the rectum, are among the target volumes of the radiation treatment, because of potential invasion of the tumor into these organs

[86]. However, radiation of the target volumes inherently causes radiation to be delivered to surrounding healthy OARs, such as the bladder, rectum, and urethra. A different goal of prostate HDR brachytherapy is therefore to minimize the amount of dose that is delivered to surrounding OARs. These goals are generally specified by a clinical protocol that includes criteria for a number of dosimetric indices, or Dose/Volume Indices (DVIs), of both the target volumes and the OARs. A volume-index is denoted V_d^o and indicates the volume in organ o (specified in cm^3 or as a percentage of the total organ volume) that receives at least a dose d (specified in Gy or as a percentage of the Planning-aim Dose (PD)). For example, the $V_{100\%}^{prostate}$ is the volume of the prostate that receives at least 100% of the PD. A dose-index is denoted D_v^o and indicates the minimum dose delivered to any part of organ o with a total volume of v that receives the highest amount of dose within this organ. For example, the D_{1cm3}^{rectum} is the minimum dose delivered to the 1 cm^3 of all of the rectum that receives the highest amount of dose. DVIs are estimated by randomly sampling a large number of dose calculation points in the specified organ, each representing a small amount of volume of the organ, and calculating the dose in each of these points using the AAPM TG-43 dose calculation model [92, 103]. The accuracy of the DVI calculation is greatly dependent on the number of dose calculation points, and thousands of points are required to accurately estimate such values [117].

The goal of treatment planning is to set the dwell times in such a way that the best possible treatment plan is found. Ideally, this means that all criteria specified by the clinical protocol are satisfied, but this may not be achievable for all patients due to the placement of the catheters relative to the surrounding anatomy of the patient. Moreover, the protocol specifies desired values, but going beyond these, if possible, is preferred, making it an optimization problem that is specific to each patient. A possible way of optimizing a treatment plan is through Graphical Optimization (GrO) [89], which involves dragging and dropping of isodose lines in a graphical interface to adjust the dose distribution, but this approach can take up to 1-2 hours [72, 89]. Inverse treatment planning methods, such as Inverse Planning by Simulated Annealing (IPSA) [75, 76] and Hybrid Inverse treatment Planning and Optimization (HIPO) [69], instead optimize the dose distribution by minimizing a predefined cost function based on a set of weights. A study into the tuning of these weights has shown that the optimal setting of these weights is patient-specific, and a weighted-sum approach may not be able to find plans that satisfy the clinical protocol [84]. Furthermore, it is known that patient-specific manual tuning of the weights is an unintuitive and time-consuming process [44].

Though plans optimized by IPSA were found to be not significantly different from mathematically optimal plans found with an exact Linear Programming (LP) method [8], these mathematically optimal plans are only optimal with respect to the cost function, the weights used, and the respective set of dose calculation points. Due to uncertainties in DVI calculation [117], and the fact that the cost function used by IPSA and the LP is only weakly correlated with the quality of relevant DVIs [51], such mathematically optimal plans may be far from clinically optimal.

As an alternative to optimizing a surrogate cost function, DVIs can be optimized directly. Various existing methods [42, 108] optimize a weighted linear sum of DVIs, but identical to optimization with IPSA and HIPO, this requires an unintuitive and time-consuming process of finding appropriate weights [44].

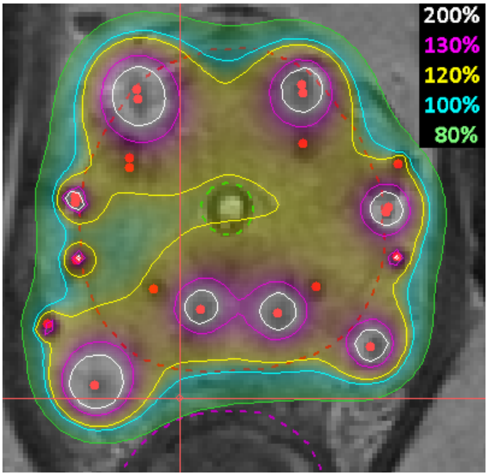


Figure 1.5: An example dose distribution for one slice of the MRI scan of a patient, showing colored isodose lines (as percentage of the planning-aim dose), dwell positions as red points, and organ contours as dashed lines (prostate in red, rectum in purple).

1.8.1. BI-OBJECTIVE TREATMENT PLANNING

A trade-off exists between covering the target volumes with a sufficient amount of dose and sparing the OARs, and the desirable trade-off can depend on patient-specific characteristics, such as age or possible side-effects from prior treatment. Because it may be difficult to define the desired trade-off prior to optimization, multi-objective optimization, through which a set of plans with different trade-offs are optimized, has previously been applied to treatment planning [73, 81]. Such an a-posteriori approach involves the medical expert in a way that allows them to use their expertise in the decision making process in an effective and efficient manner by judging already near-optimal treatment plans and selecting the best one based on their additional medical insights and patient-specific details.

Though multi-objective optimization has previously been applied to treatment planning for brachytherapy [73], direct optimization of DVIs in a multi-objective setting was first done in [81, 82]. In these latter works, all dose maximization criteria, i.e., those with the purpose of covering the tumor with a sufficient amount of dose, are grouped into one objective, and all dose minimization criteria, i.e., those with the purpose of sparing the OARs, are grouped into the other objective. Accordingly, the Least Coverage Index (LCI) and Least Sparing Index (LSI) are defined and maximized as the objectives of a bi-objective optimization model. The LCI and LSI are defined such that positive values indicate that all coverage or sparing criteria are satisfied, respectively. Specifically, the grouped planning criteria follow the worst-case principle, i.e., of all planning criteria in a group, the one that is furthest from its desired value, is the value that is used for the entire group. Therefore, when viewing an approximation front in objective space, it is immediately clear which plans satisfy the clinical protocol, because they lie in the top-right corner, which is appropriately named the golden corner.

Due to the multi-objective nature of this problem, EAs are a logical choice for its optimization. The Multi-Objective RV-GOMEA (MO-RV-GOMEA), introduced in Chapter 2, is an EA that can exploit the fact that domain knowledge of the brachytherapy problem is available. In particular, one of the strengths of MO-RV-GOMEA is that it can use partial evaluations to more efficiently calculate the dose distribution of a solution after the modification of a small number of dwell times. This is the case in the AAPM TG-43 dose calculation model that is used, because the planned dose at each dose calculation point is simply the sum of dose contributions from each dwell position. An updated dose distribution can therefore be calculated by subtracting the previous dose contribution and adding the current dose contribution of modified dwell times.

Furthermore, MO-RV-GOMEA applies variation using an explicitly defined dependency model, which can be based on available domain knowledge. For the brachytherapy application, the dependency model is defined based on the distance between dwell positions, because nearby dwell positions are strongly dependent, whereas far away dwell positions are weakly dependent. The performance of MO-RV-GOMEA was compared to that of other EAs, and it was found that MO-RV-GOMEA was able to find sets of plans that reached further into the golden corner than the other EAs [81].

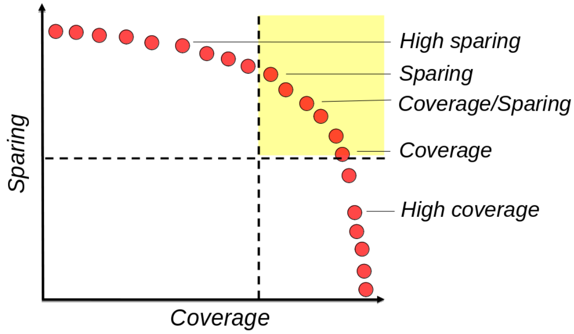


Figure 1.6: An example of a front resulting from bi-objective treatment planning with RV-GOMEA, where each treatment plan is shown as a red circle. The five labeled plans are those shown to clinical experts in a clinical observer study [85].

High quality of solutions in the bi-objective problem formulation is, however, no guarantee that resulting plans are clinically acceptable. The clinical acceptability of plans optimized by MO-RV-GOMEA was studied in a retrospective clinical observer study [85], where clinical experts were blindly presented with a number of plans optimized by MO-RV-GOMEA as well as the clinical plan with which the patient was treated, and were requested to rank these plans in terms of clinical preference. Five plans optimized by MO-RV-GOMEA were shown to each of the clinical experts, with each plan having a different trade-off between coverage and sparing, as shown in Figure

1.6. This study showed that in 98% of the cases, a clinical expert preferred a plan optimized by MO-RV-GOMEA over the clinical plan. Plans satisfying all clinical planning criteria were found for 14 of 18 patients, whereas only 4 of 18 clinical plans satisfied all clinical planning criteria. Finally, the study confirmed the benefit of multi-objective optimization for treatment planning, because various clinical experts were positive about the prospects of being able to inspect a set of optimized treatment plans with different trade-offs, i.e., with either a higher amount of coverage or a higher amount of sparing. However, optimizing a set of treatment plans with MO-RV-GOMEA required up to one hour of computation time [81], which is not acceptable in clinical practice for HDR brachytherapy, due to the tight clinical workflow, and the fact that treatment planning must be done while the patient is awaiting treatment, following the implantation of the catheters. Therefore, in Chapter 5, we pose the following research question:

Research Question: Chapter 5

How can bi-objective treatment planning with MO-RV-GOMEA be accelerated so that it can be performed in a clinically acceptable amount of time?

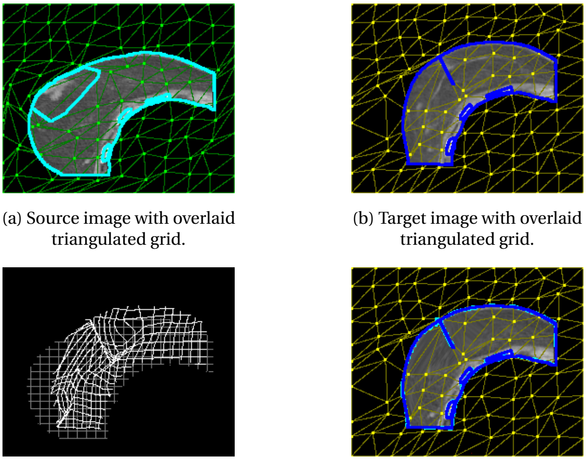
1.9. DEFORMABLE IMAGE REGISTRATION

Deformable Image Registration (DIR) [94, 109, 125] is a real-world problem that holds much potential for medical applications, where the objective is to find an anatomically plausible non-rigid deformation from a given source image to a given target image. For example, DIR may be used to find the deformation from a pre-operative scan of a cancer patient to a post-operative scan, which is used to guide post-operative radiotherapy treatment to the original tumor location [99, 110].

A frequently used deformation model, e.g., in the `elastix` toolbox [71], is one based on a linear combination of B-spline functions on a regular grid of control points [105, 106]. The benefits of such a B-spline model are that it is relatively computationally inexpensive, and that it is very effective at modeling smooth deformations. However, this also makes such models unable to effectively deal with large deformations and (dis)appearing structures, i.e., content mismatch.

An alternative deformation model is the dual-dynamic deformation model [6], which is a symmetric model that is more effective at dealing with large deformations and (dis)appearing structures, i.e., content mismatch, as is for instance the case for the pre- and post-operative application of DIR. Moreover, in its simplex-correspondence form, it employs a finite-element-like transformation model that has the benefit of having a natural augmentation to support bio-mechanical modeling of underlying tissues. This does however result in the model having non-linear and non-smooth objectives, and being relatively computationally expensive. Furthermore, the fact that the dual-dynamic deformation model is capable of modeling large deformations, is also a matter of concern, because such large deformations must not only result in a good similarity with the target image. They must also represent anatomically plausible deformations, because very large deformations may not by definition be anatomically plausible. Because it is difficult to define what constitutes an anatomically plausible deformation, DIR can be tackled as a multi-objective optimization problem, maximizing image similarity and minimizing deformation magnitude. A medical expert is then asked to select the best deformation from a set of possible deformations with different trade-offs between image similarity and deformation magnitude.

The dual-dynamic deformation model consists of two triangulated grids, one overlaying the source image and the other overlaying the target image, as displayed in Figure 1.7, which together define a transformation of the source image. Each triangle in the source grid has an associated triangle in the target grid; in our application, the triangle with the same initial coordinates. Given the two grids, the source image is transformed by shrinking or expanding the contents of each triangle in the source grid to the size and location of the corresponding triangle in the target grid. For the optimization of the dual-dynamic deformation model, the x- and y-coordinates of all points in the source and target triangulation are encoded as real-valued problem variables.



(a) Source image with overlaid triangulated grid.

(b) Target image with overlaid triangulated grid.

(c) Deformation applied to a uniform grid.

(d) Deformation applied to source image.

Figure 1.7: Example of an 11 × 11 dual-dynamic grid applied to a pre/post-operative breast cancer case, including guidance contours (in cyan and blue). Images by courtesy of the Netherlands Cancer Institute - Antoni van Leeuwenhoek Hospital (NKI-AVL).

Because of the complexity of the dual-dynamic transformation model, and the fact that it concerns multiple conflicting objectives, it was initially optimized using the multi-objective version of AMaLGaM, named MAMaLGaM [6, 19]. In order to exploit the fact that DIR can be optimized in a GBO setting, it was later optimized using (MO-)RV-GOMEA, leading to substantial improvements in performance and scalability [10].

Though DIR is a non-separable problem, each problem variable is only directly dependent on the variables that encode the coordinates of neighboring points. Therefore, for high-dimensional grids, it is possible to evaluate local modifications in different areas of the triangulated grids simultaneously. Together with the typically large number of grid points, this leads to a large potential for parallelization of the variation steps performed by RV-GOMEA. This potential for parallelization results in the following research question:

Research Question: Chapter 6

Can RV-GOMEA be accelerated by leveraging the potential for large-scale parallelization caused by the dependency structure and large dimensionality of problems such as DIR?

A third, optional, objective of DIR is the maximization of guidance information similarity between the deformed source image and the target image. Such guidance information, i.e., landmark or contour points, may be used to guide the optimization by providing information on how certain clinical features have been transformed. This information is typically manually obtained by medical experts, and cannot be used as a hard constraint due to inter- and intra-observer variability [46, 67]. Therefore, it is used as an additional objective. As DIR has three conflicting optimization objectives in this scenario, the resulting set of solutions is a 3-dimensional approximation front, which may be difficult to navigate through. These 3-dimensional fronts, furthermore, tend to have an irregular distribution of points due to differences in the relative difficulties of the three objectives. In order to help navigation through such 3-dimensional fronts, the following research question is posed in Chapter 7:

Research Question: Chapter 7

Can irregularly distributed multi-dimensional approximation fronts be intuitively mapped to a lower-dimensional space to help navigation and solution selection?

REFERENCES

[1] Ahn, C. W., Ramakrishna, R. S., and Goldberg, D. E. (2004). Real-coded Bayesian optimization algorithm: Bringing the strength of BOA into the continuous world. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 840–851. Springer.

[2] Akimoto, Y. and Hansen, N. (2016a). Online model selection for restricted covariance matrix adaptation. In *International Conference on Parallel Problem Solving from Nature*, pages 3–13. Springer.

[3] Akimoto, Y. and Hansen, N. (2016b). Projection-based restricted covariance matrix adaptation for high dimension. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 197–204. ACM.

[4] Alarcon-Rodriguez, A., Ault, G., and Galloway, S. (2010). Multi-objective planning of distributed energy resources: A review of the state-of-the-art. *Renewable and Sustainable Energy Reviews*, 14(5):1353–1366.

[5] Alba, E. (2006). *Parallel evolutionary computations*, volume 22. Springer.

[6] Alderliesten, T., Bosman, P. A. N., and Bel, A. (2015). Getting the most out of additional guidance information in deformable image registration by leveraging multi-objective optimization. In Ourselin, S. and Styner, M. A., editors, *Medical Imaging 2015: Image Processing*, volume 9413, pages 469 – 475. International Society for Optics and Photonics, SPIE.

[7] Alexandropoulos, S.-A. N., Aridas, C. K., Kotsiantis, S. B., and Vrahatis, M. N. (2019). Multi-objective evolutionary optimization algorithms for machine learning: A recent survey. In *Approximation and Optimization*, pages 35–55. Springer.

[8] Alterovitz, R., Lessard, E., Pouliot, J., Hsu, I.-C. J., O'Brien, J. F., and Goldberg, K. (2006). Optimization of HDR brachytherapy dose distributions using linear programming with penalty costs. *Medical Physics*, 33(11):4012–4019.

[9] Asanović, K., Bodik, R., Catanzaro, B. C., Gebis, J. J., Husbands, P., Keutzer, K., Patterson, D. A., Plishker, W. L., Shalf, J., Williams, S. W., and Yelick, K. A. (2006). The landscape of parallel computing research: A view from Berkeley. (UCB/EECS-2006-183).

[10] Bouter, A., Alderliesten, T., and Bosman, P. A. N. (2017a). A novel model-based evolutionary algorithm for multi-objective deformable image registration with content mismatch and large deformations: benchmarking efficiency and quality. In Styner, M. A. and Angelini, E. D., editors, *Medical Imaging 2017: Image Processing*, volume 10133, pages 304 – 311. International Society for Optics and Photonics, SPIE.

[11] Bouter, A., Alderliesten, T., Witteveen, C., and Bosman, P. A. N. (2017b). Exploiting linkage information in real-valued optimization with the real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 705–712. ACM.

[12] Bouter, A., Luong, N. H., Witteveen, C., Alderliesten, T., and Bosman, P. A. N. (2017c). The multi-objective real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 537–544. ACM.

[13] Bäck, T., Fogel, D. B., and Michalewicz, Z. (1997). Handbook of evolutionary computation. *Release*, 97(1):B1.

[14] Bellman, R. (1966). Dynamic programming. *Science*, 153(3731):34–37.

[15] Benayoun, R., De Montgolfier, J., Tergny, J., and Laritchev, O. (1971). Linear programming with multiple objective functions: Step method (STEM). *Mathematical Programming*, 1(1):366–375.

[16] Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution strategies – A comprehensive introduction. *Natural Computing*, 1(1):3–52.

[17] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

[18] Booker, L. B., Goldberg, D. E., and Holland, J. H. (1989). Classifier systems and genetic algorithms. *Artificial Intelligence*, 40(1-3):235–282.

[19] Bosman, P. A. N. (2010). The anticipated mean shift and cluster registration in mixture-based EDAs for multi-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 351–358. ACM.

[20] Bosman, P. A. N., Grahl, J., and Thierens, D. (2009). AMaLGaM IDEAs in noiseless black-box optimization benchmarking. In *Proceedings of the Genetic and Evolutionary Computation Conference: Late Breaking Papers*, pages 2247–2254. ACM.

[21] Bosman, P. A. N., Grahl, J., and Thierens, D. (2013). Benchmarking parameter-free AMaLGaM on functions with and without noise. *Evolutionary Computation*, 21(3):445–469.

[22] Bosman, P. A. N. and Thierens, D. (2000). Expanding from discrete to continuous estimation of distribution algorithms: The ID𝔼A. In *International Conference on Parallel Problem Solving from Nature*, pages 767–776. Springer.

[23] Bosman, P. A. N. and Thierens, D. (2003). The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):174–188.

[24] Bosman, P. A. N. and Thierens, D. (2012). Linkage neighbors, optimal mixing and forced improvements in genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 585–592. ACM.

[25] Bosman, P. A. N. and Thierens, D. (2013). More concise and robust linkage learning by filtering and combining linkage hierarchies. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 359–366. ACM.

[26] Boyd, S. and Mattingley, J. (2007). Branch and bound methods. *Notes for EE364b, Stanford University*, pages 2006–07.

[27] Breedveld, S., Craft, D., Van Haveren, R., and Heijmen, B. (2019). Multi-criteria optimization and decision-making in radiotherapy. *European Journal of Operational Research*, 277(1):1–19.

[28] Brodtkorb, A. R., Dyken, C., Hagen, T. R., Hjelmervik, J. M., and Storaasli, O. O. (2010). State-of-the-art in heterogeneous computing. *Scientific Programming*, 18(1):1–33.

[29] Brodtkorb, A. R., Hagen, T. R., and Sætra, M. L. (2013). Graphics processing unit (GPU) programming strategies and trends in GPU computing. *Journal of Parallel and Distributed Computing*, 73(1):4–13.

[30] Cabrera, G., Ehrgott, M., Mason, A., and Philpott, A. (2014). Multi-objective optimisation of positively homogeneous functions and an application in radiation therapy. *Operations Research Letters*, 42(4):268–272.

[31] Chankong, V. and Haimes, Y. Y. (2008). *Multiobjective decision making: Theory and methodology*. Courier Dover Publications.

[32] Chargari, C., Deutsch, E., Blanchard, P., Gouy, S., Martelli, H., Guérin, F., Dumas, I., Bossi, A., Morice, P., Viswanathan, A. N., et al. (2019). Brachytherapy: An overview for clinicians. *CA: A Cancer Journal for Clinicians*, 69(5):386–401.

[33] Chen, Y. P., Yu, T.-L., Sastry, K., and Goldberg, D. E. (2007). A survey of linkage learning techniques in genetic and evolutionary algorithms. *IlliGAL report*, 2007014.

[34] Coello, C. A. C. and Sierra, M. R. (2004). A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In *Mexican International Conference on Artificial Intelligence*, pages 688–697. Springer.

[35] Crook, J., Marbán, M., and Batchelar, D. (2020). HDR prostate brachytherapy. In *Seminars in Radiation Oncology*, volume 30, pages 49–60. Elsevier.

[36] Dasgupta, D. and Michalewicz, Z. (2013). *Evolutionary algorithms in engineering applications*. Springer Science & Business Media.

[37] De Jong, K. A. (2006). *Evolutionary Computation: A Unified Approach*. MIT Press.

[38] Deb, K. (1999). Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3):205–230.

[39] Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc.

[40] Deb, K. (2014). Multi-objective optimization. In *Search Methodologies*, pages 403–449. Springer.

[41] Deb, K. and Myburgh, C. (2016). Breaking the billion-variable barrier in real-world optimization using a customized evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 653–660. ACM.

[42] Deist, T. M. and Gorissen, B. L. (2016). High-dose-rate prostate brachytherapy inverse planning on dose-volume criteria by simulated annealing. *Physics in Medicine & Biology*, 61(3):1155.

[43] Dilsizian, S. E. and Siegel, E. L. (2014). Artificial intelligence in medicine and cardiac imaging: Harnessing big data and advanced computing to provide personalized medical diagnosis and treatment. *Current Cardiology Reports*, 16(1):441.

[44] Dinkla, A. M., van der Laarse, R., Kaljouw, E., Pieters, B. R., Koedooder, K., van Wieringen, N., and Bel, A. (2015). A comparison of inverse optimization algorithms for hdr/pdr prostate brachytherapy treatment planning. *Brachytherapy*, 14(2):279–288.

[45] Ferreira, F. G. D. C., Gandomi, A. H., and Cardoso, R. T. N. (2021). Artificial intelligence applied to stock market trading: A review. *IEEE Access*, 9:30898–30917.

[46] Fiorino, C., Reni, M., Bolognesi, A., Cattaneo, G. M., and Calandrino, R. (1998). Intra-and inter-observer variability in contouring prostate and seminal vesicles: Implications for conformal treatment planning. *Radiotherapy and Oncology*, 47(3):285–292.

[47] Freitas, A. A. (2004). A critical review of multi-objective optimization in data mining: A position paper. *ACM SIGKDD Explorations Newsletter*, 6(2):77–86.

[48] Glover, F. and Laguna, M. (1998). Tabu search. In *Handbook of Combinatorial Optimization*, pages 2093–2229. Springer.

[49] Goldman, B. W. and Punch, W. F. (2014). Parameter-less population pyramid. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 785–792. ACM.

[50] Gong, M., Liu, F., Zhang, W., Jiao, L., and Zhang, Q. (2011). Interactive MOEA/D for multi-objective decision making. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 721–728.

[51] Gorissen, B. L., Den Hertog, D., and Hoffmann, A. L. (2013). Mixed integer programming improves comprehensibility and plan quality in inverse optimization of prostate HDR brachytherapy. *Physics in Medicine & Biology*, 58(4):1041.

[52] Grond, M., Luong, N., Morren, J., and Slootweg, J. G. (2012). Multi-objective optimization techniques and applications in electric power systems. In *2012 47th International Universities Power Engineering Conference (UPEC)*, pages 1–6. IEEE.

[53] Gunantara, N. (2018). A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, 5(1):1502242.

[54] Hansen, N., Müller, S., and Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18.

[55] Hansen, N. and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195.

[56] Harik, G. R., Lobo, F. G., and Goldberg, D. E. (1999). The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 3(4):287–297.

[57] Harik, G. R., Lobo, F. G., and Sastry, K. (2006). Linkage learning via probabilistic modeling in the extended compact genetic algorithm (ECGA). In Pelikan, M., Sastry, K., and Cantú-Paz, E., editors, *Scalable optimization via probabilistic modeling*, volume 33 of *Studies in Computational Intelligence*, pages 39–61. Springer.

[58] Hauschild, M. and Pelikan, M. (2011). An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, 1(3):111–128.

[59] Holland, J. H. (1962). Outline for a logical theory of adaptive systems. *Journal of the ACM (JACM)*, 9(3):297–314.

[60] Hornby, G., Globus, A., Linden, D., and Lohn, J. (2006). Automated antenna design with evolutionary algorithms. In *Space 2006*, page 7242.

[61] Hoskin, P. J., Rojas, A. M., Bownes, P. J., Lowe, G. J., Ostler, P. J., and Bryant, L. (2012). Randomised trial of external beam radiotherapy alone or combined with high-dose-rate brachytherapy boost for localised prostate cancer. *Radiotherapy and Oncology*, 103(2):217–222.

[62] Hsu, S.-H. and Yu, T.-L. (2015). Optimization by pairwise linkage detection, incremental linkage set, and restricted/back mixing: DSMGA-II. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 519–526. ACM.

[63] Ignizio, J. P. and Cavalier, T. M. (1994). *Linear Programming*. Prentice-Hall, Inc.

[64] Jabir, E., Panicker, V. V., and Sridharan, R. (2015). Multi-objective optimization model for a green vehicle routing problem. *Procedia-Social and Behavioral Sciences*, 189:33–39.

[65] Jahn, J. (1985). Scalarization in multi objective optimization. In *Mathematics of Multi Objective Optimization*, pages 45–88. Springer.

[66] Jin, Y. (2006). *Multi-objective Machine Learning*, volume 16. Springer Science & Business Media.

[67] Joskowicz, L., Cohen, D., Caplan, N., and Sosna, J. (2019). Inter-observer variability of manual contour delineation of structures in ct. *European Radiology*, 29(3):1391–1399.

[68] Jozefowiez, N., Semet, F., and Talbi, E.-G. (2008). Multi-objective vehicle routing problems. *European Journal of Operational Research*, 189(2):293–309.

[69] Karabis, A., Giannouli, S., and Baltas, D. (2005). HIPO: A hybrid inverse treatment planning optimization algorithm in HDR brachytherapy. *Radiotherapy and Oncology*, 76(2):S29.

[70] Keisch, M., Vicini, F., Kuske, R. R., Hebert, M., White, J., Quiet, C., Arthur, D., Scroggins, T., and Streeter, O. (2003). Initial clinical experience with the MammoSite breast brachytherapy applicator in women with early-stage breast cancer treated with breast-conserving therapy. *International Journal of Radiation Oncology* Biology* Physics*, 55(2):289–293.

[71] Klein, S., Staring, M., Murphy, K., Viergever, M. A., and Pluim, J. P. (2009). Elastix: A toolbox for intensity-based medical image registration. *IEEE Transactions on Medical Imaging*, 29(1):196–205.

[72] Kolkman-Deurloo, I.-K. K., Deleye, X. G. J., Jansen, P. P., and Koper, P. C. M. (2004). Anatomy based inverse planning in hdr prostate brachytherapy. *Radiotherapy and Oncology*, 73(1):73–77.

[73] Lahanas, M., Baltas, D., and Zamboglou, N. (2003). A hybrid evolutionary algorithm for multi-objective anatomy-based dose optimization in high-dose-rate brachytherapy. *Physics in Medicine & Biology*, 48(3):399.

[74] Lawler, E. L. and Wood, D. E. (1966). Branch-and-bound methods: A survey. *Operations Research*, 14(4):699–719.

[75] Lessard, E. (2004). Development and clinical introduction of an inverse planning dose optimization by simulated annealing (IPSA) for high dose rate brachytherapy: Ph. D. theses abstracts. *Medical Physics*, 31(10):2935–2935.

[76] Lessard, E. and Pouliot, J. (2001). Inverse planning anatomy-based dose optimization for HDR-brachytherapy of the prostate using fast simulated annealing algorithm and dedicated objective function. *Medical Physics*, 28(5):773–779.

[77] Li, S.-C. and Yu, T.-L. (2017). Speeding up DSMGA-II on CUDA platform. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 809–816. ACM.

[78] Loshchilov, I. (2014). A computationally efficient limited memory CMA-ES for large scale optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 397–404. ACM.

[79] Lozano, J. A., Larrañaga, P., Inza, I., and Bengoetxea, E. (2006). *Towards a new evolutionary computation: Advances on estimation of distribution algorithms*, volume 192 of *Studies in Fuzziness and Soft Computing*. Springer.

[80] Luong, H. N. and Bosman, P. A. N. (2012). Elitist archiving for multi-objective evolutionary algorithms: To adapt or not to adapt. In *International Conference on Parallel Problem Solving from Nature*, pages 72–81. Springer.

[81] Luong, N. H., Alderliesten, T., Bel, A., Niatsetski, Y., and Bosman, P. A. N. (2018). Application and benchmarking of multi-objective evolutionary algorithms on high-dose-rate brachytherapy planning for prostate cancer treatment. *Swarm and Evolutionary Computation*, 40:37–52.

[82] Luong, N. H., Bouter, A., van der Meer, M. C., Niatsetski, Y., Witteveen, C., Bel, A., Alderliesten, T., and Bosman, P. A. N. (2017). Efficient, effective, and insightful tackling of the high-dose-rate brachytherapy treatment planning problem for prostate cancer using evolutionary multi-objective optimization algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1372–1379. ACM.

[83] Luong, N. H., La Poutré, H., and Bosman, P. A. N. (2014). Multi-objective gene-pool optimal mixing evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 357–364. ACM.

[84] Maree, S. C., Bosman, P. A. N., van Wieringen, N., Niatsetski, Y., Pieters, B. R., Bel, A., and Alderliesten, T. (2020). Automatic bi-objective parameter tuning for inverse planning of high-dose-rate prostate brachytherapy. *Physics in Medicine & Biology*, 65(7):075009.

[85] Maree, S. C., Luong, N. H., Kooreman, E. S., van Wieringen, N., Bel, A., Hinnen, K. A., Westerveld, H., Pieters, B. R., Bosman, P. A. N., and Alderliesten, T. (2019). Evaluation of bi-objective treatment planning for high-dose-rate prostate brachytherapy – A retrospective observer study. *Brachytherapy*, 18(3):396–403.

[86] Marks, L. B. and Anscher, M. S. (1992). Radiotherapy for prostate cancer: Should the seminal vesicles be considered target? *International Journal of Radiation Oncology* Biology* Physics*, 24(3):435–440.

[87] Michalakes, J. and Vachharajani, M. (2008). GPU acceleration of numerical weather prediction. *Parallel Processing Letters*, 18(04):531–548.

[88] Mittal, S. and Vaishay, S. (2019). A survey of techniques for optimizing deep learning on GPUs. *Journal of Systems Architecture*, 99:101635.

[89] Morton, G. C., Sankreacha, R., Halina, P., and Loblaw, A. (2008). A comparison of anatomy-based inverse planning with simulated annealing and graphical optimization for high-dose-rate prostate brachytherapy. *Brachytherapy*, 7(1):12–16.

[90] Mühlenbein, H. and Paaß, G. (1996). From recombination of genes to the estimation of distributions I. Binary parameters. In *International Conference on Parallel Problem Solving from Nature*, pages 178–187. Springer.

[91] Nag, S., Erickson, B., Thomadsen, B., Orton, C., Demanes, J. D., Petereit, D., and Society, A. B. (2000). The American brachytherapy society recommendations for high-dose-rate brachytherapy for carcinoma of the cervix. *International Journal of Radiation Oncology* Biology* Physics*, 48(1):201–211.

[92] Nath, R., Anderson, L. L., Luxton, G., Weaver, K. A., Williamson, J. F., and Meigooni, A. S. (1995). Dosimetry of interstitial brachytherapy sources: Recommendations of the AAPM radiation therapy committee task group no. 43. *Medical Physics*, 22(2):209–234.

[93] Odu, G. O. and Charles-Owaba, O. E. (2013). Review of multi-criteria optimization methods–Theory and applications. *IOSR Journal of Engineering (IOSRJEN)*, 3(10):1–14.

[94] Oh, S. and Kim, S. (2017). Deformable image registration in radiation therapy. *Radiation Oncology Journal*, 35(2):101.

[95] Ostermeier, A., Gawelczyk, A., and Hansen, N. (1994). A derandomized approach to self-adaptation of evolution strategies. *Evolutionary Computation*, 2(4):369–380.

[96] Pelikan, M. (2005). Hierarchical Bayesian optimization algorithm. In *Hierarchical Bayesian optimization algorithm*, volume 170 of *Studies in Fuzziness and Soft Computing*, pages 105–129. Springer.

[97] Pelikan, M., Goldberg, D. E., and Cantú-Paz, E. (1999). BOA: The Bayesian optimization algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 525–532. Morgan Kaufmann Publishers Inc.

[98] Pelikan, M. and Mühlenbein, H. (1999). The bivariate marginal distribution algorithm. In Roy, R., Furuhashi, T., and Chawdhry, P. K., editors, *Advances in Soft Computing*, pages 521–535. Springer.

[99] Petersen, R. P., Truong, P. T., Kader, H. A., Berthelet, E., Lee, J. C., Hilts, M. L., Kader, A. S., Beckham, W. A., and Olivotto, I. A. (2007). Target volume delineation for partial breast radiotherapy planning: Clinical characteristics associated with low interobserver concordance. *International Journal of Radiation Oncology* Biology* Physics*, 69(1):41–48.

[100] Potter, M. A. and Jong, K. A. D. (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29.

[101] Rachmawati, L. and Srinivasan, D. (2006). Preference incorporation in multi-objective evolutionary algorithms: A survey. In *2006 IEEE International Conference on Evolutionary Computation*, pages 962–968. IEEE.

[102] Rechenberg, I. (1978). Evolutionsstrategien. In *Simulationsmethoden in der Medizin und Biologie*, pages 83–114. Springer.

[103] Rivard, M. J., Coursey, B. M., DeWerd, L. A., Hanson, W. F., Saiful Huq, M., Ibbott, G. S., Mitch, M. G., Nath, R., and Williamson, J. F. (2004). Update of AAPM task group no. 43 report: A revised AAPM protocol for brachytherapy dose calculations. *Medical Physics*, 31(3):633–674.

[104] Roijers, D. M. and Whiteson, S. (2017). Multi-objective decision making. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 11(1):1–129.

[105] Rueckert, D., Sonoda, L. I., Hayes, C., Hill, D. L., Leach, M. O., and Hawkes, D. J. (1999). Nonrigid registration using free-form deformations: Application to breast MR images. *IEEE Transactions on Medical Imaging*, 18(8):712–721.

[106] Schreibmann, E., Chen, G. T. Y., and Xing, L. (2006). Image interpolation in 4D CT using a BSpline deformable registration model. *International Journal of Radiation Oncology* Biology* Physics*, 64(5):1537–1550.

[107] Schwefel, H.-P. P. (1993). *Evolution and optimum seeking: The Sixth Generation.* John Wiley & Sons, Inc.

[108] Siauw, T., Cunha, A., Atamtürk, A., Hsu, I.-C., Pouliot, J., and Goldberg, K. (2011). IPIP: A new approach to inverse planning for HDR brachytherapy by directly optimizing dosimetric indices. *Medical Physics*, 38(7):4045–4051.

[109] Sotiras, A., Davatzikos, C., and Paragios, N. (2013). Deformable medical image registration: A survey. *IEEE Transactions on Medical Imaging*, 32(7):1153–1190.

[110] Struikmans, H., Wárlám-Rodenhuis, C., Stam, T., Stapper, G., Tersteeg, R. J., Bol, G. H., and Raaijmakers, C. P. (2005). Interobserver variability of clinical target volume delineation of glandular breast tissue and of boost volume in tangential breast irradiation. *Radiotherapy and Oncology*, 76(3):293–299.

[111] Sudholt, D. (2015). Parallel evolutionary algorithms. In *Springer Handbook of Computational Intelligence*, pages 929–959. Springer.

[112] Thiele, L., Miettinen, K., Korhonen, P. J., and Molina, J. (2009). A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary Computation*, 17(3):411–436.

[113] Thierens, D. (1999). Scalability problems of simple genetic algorithms. *Evolutionary Computation*, 7(4):331–352.

[114] Thierens, D. (2010). The linkage tree genetic algorithm. In *International Conference on Parallel Problem Solving from Nature*, pages 264–273. Springer.

[115] Thierens, D. and Bosman, P. A. N. (2011). Optimal mixing evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 617–624. ACM.

[116] Tintos, R., Whitley, D., and Chicano, F. (2015). Partition crossover for pseudo-boolean optimization. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*, pages 137–149. ACM.

[117] van der Meer, M. C., Bosman, P. A. N., Pieters, B. R., Niatsetski, Y., van Wieringen, N., Alderliesten, T., and Bel, A. (2019). Sensitivity of dose-volume indices to computation settings in high-dose-rate prostate brachytherapy treatment plan evaluation. *Journal of Applied Clinical Medical Physics*, 20(4):66–74.

[118] Van Laarhoven, P. J. M. and Aarts, E. H. L. (1987). Simulated annealing. In *Simulated Annealing: Theory and Applications*, pages 7–15. Springer.

[119] Vanderbei, R. J. et al. (2015). *Linear Programming*, volume 3. Springer.

[120] Varelas, K., Auger, A., Brockhoff, D., Hansen, N., ElHara, O. A., Semet, Y., Kassab, R., and Barbaresco, F. (2018). A comparative study of large-scale variants of CMA-ES. In *International Conference on Parallel Problem Solving from Nature*, pages 3–15. Springer.

[121] Vermetten, D., van Rijn, S., Bäck, T., and Doerr, C. (2019). Online selection of CMA-ES variants. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 951–959.

[122] Whitley, D., Aguirre, H., and Sutton, A. (2020). Understanding transforms of pseudo-boolean functions. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 760–768.

[123] Wong, M.-L., Wong, T.-T., and Fok, K.-L. (2005). Parallel evolutionary algorithms on graphics processing unit. In *IEEE Congress on Evolutionary Computation*, volume 3, pages 2286–2293. IEEE.

[124] Yamada, Y., Rogers, L., Demanes, D. J., Morton, G., Prestidge, B. R., Pouliot, J., Gil'ad, N. C., Zaider, M., Ghilezan, M., and Hsu, I.-C. (2012). American brachytherapy society consensus guidelines for high-dose-rate prostate brachytherapy. *Brachytherapy*, 11(1):20–32.

[125] Yuen, J., Barber, J., Ralston, A., Gray, A., Walker, A., Hardcastle, N., Schmidt, L., Harrison, K., Poder, J., Sykes, J. R., et al. (2020). An international survey on the clinical use of rigid and deformable image registration in radiotherapy. *Journal of Applied Clinical Medical Physics*, 21(10):10–24.

[126] Zhou, Y., Liepe, J., Sheng, X., Stumpf, M. P. H., and Barnes, C. (2011). GPU accelerated biochemical network simulation. *Bioinformatics*, 27(6):874–876.

[127] Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195.

[128] Zitzler, E. and Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms — A comparative case study. In *International Conference on Parallel Problem Solving from Nature*, pages 292–301. Springer.

2

Real-Valued Gene-Pool Optimal Mixing Evolutionary Algorithms

This chapter is based on the following publication: Bouter, A., Alderliesten, T., and Bosman, P. A. N. (2021). Achieving highly scalable evolutionary real-valued optimization by exploiting partial evaluations. *Evolutionary Computation*, 29(1):129–155

Abstract

It is known that to achieve efficient scalability of an Evolutionary Algorithm (EA), dependencies (also known as linkage) must be properly taken into account during variation. In a Gray-Box Optimization (GBO) setting, exploiting prior knowledge regarding these dependencies can greatly benefit optimization. We specifically consider the setting where partial evaluations are possible, meaning that the partial modification of a solution can be efficiently evaluated. Such problems are potentially very difficult, e.g., non-separable, multi-modal, and multi-objective. The Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA) can effectively exploit partial evaluations, leading to a substantial improvement in performance and scalability. GOMEA was recently shown to be extendable to real-valued optimization through a combination with the real-valued estimation of distribution algorithm AMaLGaM. In this chapter, we definitively introduce the Real-Valued GOMEA (RV-GOMEA), and introduce a new variant, constructed by combining GOMEA with what is arguably the best-known real-valued EA, the Covariance Matrix Adaptation Evolution Strategies (CMA-ES). Both variants of GOMEA are compared to L-BFGS and the Limited Memory CMA-ES (LM-CMA-ES). We show that both variants of RV-GOMEA achieve excellent performance and scalability in a GBO setting, which can be orders of magnitude better than that of EAs unable to efficiently exploit the GBO setting.

2.1. INTRODUCTION

Evolutionary Algorithms (EAs) are frequently used to solve optimization problems that are considered too difficult for more efficient algorithms such as local search in case of discrete problems or gradient descent in case of real-valued problems. This is the case when, for example, the problem is multi-modal or when plateaus in the problem landscape prevent gradient information from directing the search towards the global optimum. Furthermore, EAs are known to be among the state of the art for the optimization of Multi-Objective (MO) optimization problems [15]. In most cases, the optimization problem is considered to be a Black-Box Optimization (BBO) problem, where only function evaluations can reveal information about the structure of the optimization problem at hand.

Considering problems in a BBO setting is however not always a necessity, and the exploitation of problem-specific information, when possible, could substantially increase the performance of an EA. Some exploitable form of problem information can be available even when the optimization problem is considered very difficult. For example, an EA with problem-specific recombination and mutation operators has been able to find near-optimal solutions for a binary optimization problem with a billion variables [16]. Near-linear scalability in the number of problem variables was achieved, despite the problem being non-separable due to a combination of equality and inequality constraints. In comparison, conventional integer linear programming solvers were unable to solve a 2000-variable version of this problem.

Partition crossover [54] is a more general instance of a custom binary crossover operator that exploits problem information. This operator has access to a variable interaction graph, describing whether or not interactions exist between pairs of problem variables. Partition crossover was previously applied to the Traveling Salesman Problem (TSP) [54], and NK-landscapes [51]. It substantially outperformed state-of-the-art local-search solvers for the Maximal Satisfiability (MAXSAT) problem [12].

Partial evaluations are a way to exploit problem-specific information in the domain of either discrete or real-valued optimization, allowing to efficiently evaluate the objective value(s) of a solution for which only a small number of problem variables have been modified. When such partial evaluations are possible, we speak of a Gray-Box Optimization (GBO) setting.

In real-valued optimization, which our work is focused on, proper use of partial evaluations has been shown to lead to a substantial increase in performance and scalability on a wide range of benchmark problems [4], as well as the real-world problems of medical deformable image registration [2] and the optimization of brachytherapy treatment plans for prostate cancer [32], when combined with a real-valued version of GOMEA (RV-GOMEA) [4, 5]. The two aforementioned real-world problems are both instances of non-separable, multi-modal, multi-objective problems, making them very suitable for optimization with an EA. Despite the complexity of these problems, their definitions allow for the application of partial evaluations, leading to substantial improvements in performance and scalability. To the best of our knowledge, RV-GOMEA is the first real-valued EA to exploit partial evaluations.

In this chapter, we highlight the benefits of a GBO setting in which partial evaluations are possible, and the requirements for such a setting. Furthermore, we definitively introduce RV-GOMEA, extending previously published work [4, 5] by comparisons with state-of-the-art large-scale optimization algorithms including the well-known gradient-based optimization method L-BFGS [28], and the Limited-Memory CMA-ES (LM-CMA-ES) [29], a large-scale variant of CMA-ES, on various types of single-objective and multi-objective problems. Furthermore, we introduce a novel variation of RV-GOMEA, by making a new combination with the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [21], arguably the most well-known and state-of-the-art EA for real-valued optimization.

The remainder of this chapter is organized as follows. In Section 2.2 we discuss work on Model-Based EAs (MBEAs) related to GOMEA. Section 2.3 then gives a general introduction to GOMEA, and Section 2.4 describes how linkage structure is modeled in GOMEA. Section 6.2 describes the GBO setting that we consider, which allows for the use of partial evaluations, and the benefits and restrictions of such a setting are discussed. The extension of GOMEA to the real-valued domain, i.e., RV-GOMEA, is introduced in Section 2.6, after which two instantiations (one novel instance and one previously introduced) of RV-GOMEA are discussed in Section 2.7. Multi-objective variants of these algorithms are subsequently designed in Section 2.8. In Section 2.9, the performance of all discussed algorithms, combined with a variety of different linkage models, is then analyzed through scalability experiments on a set of well-known benchmark problems. Results are then discussed in Section 2.10, followed by stating a number of conclusions drawn from said experiments in Section 2.11.

2.2. Related Work

In the standard Genetic Algorithm (GA) [23], variation is performed completely at random. In comparison, the goal of MBEAs is to perform variation in a more informed manner, so as to substantially improve performance. MBEAs exploit the problem structure or the optimization landscape throughout the optimization process guided by an explicit model that describes key characteristics of the problem at hand. This model may be learned online, while optimization is being performed, or it may be (partially) instantiated with problem-specific knowledge.

In the domain of real-valued optimization, which this work focuses on, a Gaussian distribution is frequently used to describe how to generate the values for the problem variables of new solutions. This is done in, e.g., the Covariance Matrix Adaptation Evolution Strategies (CMA-ES) [21], the Adapted Maximum-Likelihood Gaussian Model Iterated Density Estimation Algorithm (AMaLGaM-ID𝔼A or AMaLGaM for short) [8], and Natural Evolution Strategies (NES) [55].

A key aspect of problem structure is the linkage structure, which describes the structure of an optimization problem in terms of the dependencies between problem variables. Linkage information, which is generally derived from the population or known beforehand due to problem-specific knowledge, is used in many MBEAs to estimate a statistical model that captures the underlying linkage structure of the optimization problem. In Estimation of Distribution Algorithms (EDAs) [30, 43], a specific type of MBEA, a probability distribution is used to model the linkage structure.

In discrete optimization, the Bivariate Marginal Distribution Algorithm (BMDA) [42], an extension of the Univariate Marginal Distribution Algorithm (UMDA) [36], incorporates bivariate dependencies between problem variables in order to model dependencies. Models of larger capacity were later introduced, with the Extended Compact Genetic Algorithm (ECGA) [22] using a marginal product model, and the Estimation of Bayesian Network Algorithm (EBNA) [17], Bayesian Optimization Algorithm (BOA) [40], and the hierarchical Bayesian Optimization Algorithm (hBOA) [41] using Bayesian networks to model linkage structure. In the aforementioned Gaussian-based MBEAs, CMA-ES [21], AMaLGaM [8], and NES [55], a covariance matrix describes the linkage structure.

Recently, there has been an increasing focus on MBEAs explicitly modeling and exploiting linkage information to guide mixing operators, rather than using the linkage information to guide a statistical model that samples new problem variables. This trend was first observed for discrete optimization [18, 24, 49, 50], and more recently for real-valued optimization [4]. Of these new types of MBEAs, we focus on GOMEA [50] here, because GOMEA is so far the only one among these MBEAs to be extended beyond discrete optimization.

2.3. GENE-POOL OPTIMAL MIXING

GOMEA has its roots in the Linkage Tree Genetic Algorithm (LTGA) [49] for binary variables, which was later generalized and renamed [50]. Subsequent efficiency enhancements, restart mechanisms, and multi-objective extensions, adapted GOMEA to its current state [11, 34], which is considered to be among the state-of-the-art for discrete optimization.

A central concept in GOMEA that is the key to its performance, is to exploit linkage information by applying variation not to all problem variables simultaneously, but instead to apply variation to small subsets of dependent variables, and only accepting a variation operation if it leads to an improved offspring solution. This latter concept is called optimal mixing. A so-called linkage model is used to describe (small) subsets of variables that are considered to be dependent. Moreover, when the entire population, is used as potential donor information during optimal mixing, the procedure is generally called Gene-pool Optimal Mixing (GOM).

After its introduction in GOMEA, optimal mixing was included also in other state-of-the-art EAs, such as DSMGA-II [24] and P3 [18]. Moreover, even though GOMEA was originally introduced in the domain of discrete optimization, the concept of linkage information exploitation of GOMEA is more widely applicable, as was recently shown through RV-GOMEA [4]. RV-GOMEA combines key aspects of GOMEA with those of the real-valued EDA known as AMaLGaM [8]. The dependency model of GOMEA is used in order to exploit linkage information, and the distribution-based sampling methods used by AMaLGaM are used in order to match the continuous nature of the search space.

The use of AMaLGaM as the method of sampling new problem-variable values is however not an absolute necessity. Instead, other real-valued EAs could be combined with GOMEA in order to benefit from its model-building capabilities. This not only applies to real-valued optimization, but could apply to EAs in other domains of optimization. Indeed, novel variants of GOMEA have recently appeared, e.g., for permutation spaces [9] and genetic programming [53].

2.4. Modeling Linkage Structure

In GOMEA, the linkage structure of a problem is modeled as a Family Of Subsets (FOS), denoted $\mathcal{F}$, which is a subset of the power set of $\mathcal{I}$, which contains the indices of all ℓ problem variables, i.e., $\mathcal{F} \subseteq \mathcal{P}(\mathcal{I})$, where $\mathcal{I} = \{0, 1, \ldots, \ell - 1\}$. Each element $\mathcal{F}_i \in \mathcal{F}$ contains a number of indices of problem variables, all of which are considered to be mutually dependent by this linkage model. Such an element of a FOS is named a linkage set, as it describes a set of problem variables between which linkage information is modeled. Virtually any FOS can be used to model linkage structure, though generally speaking, each index $i \in \mathcal{I}$ appears in at least one linkage set $\mathcal{F}_j$.

The linkage model can either be determined prior to optimization, and be fixed throughout optimization, or it can be learned based on the population at the start of every generation. We refer to linkage models that are fixed as static linkage models, and to linkage models that change throughout optimization as dynamic linkage models.

In the remainder of this section, we discuss four potentially useful FOS models. Depending on the optimization problem, and how much domain-specific knowledge is available, one can select the linkage model that is the most appropriate.

2.4.1. Marginal Product FOS

The marginal product FOS is a FOS where each problem variable index exists in exactly one linkage set, i.e., $\mathcal{F}_i \cap \mathcal{F}_j = \emptyset$ for each $\mathcal{F}_i, \mathcal{F}_j \in \mathcal{F}$ with $i \neq j$, and $\bigcup_{\mathcal{F}_i \in \mathcal{F}} \mathcal{F}_i = \mathcal{I}$. A simple instance of a marginal product FOS is the univariate FOS, which contains each problem variable index in a separate linkage set, i.e., $\mathcal{F} = \{\{0\}, \{1\}, \ldots, \{\ell - 1\}\}$.

We also define the k-block FOS as a specific marginal product FOS where each linkage set consists of blocks of k subsequent variables, i.e., $\mathcal{F} = \{\{0, 1, \ldots, k-1\}, \{k, k+1, \ldots, 2k-1\}, \ldots\}$. This FOS structure is defined to match the problem structure of the Sum of Rotated Ellipsoid Blocks (SoREB) benchmark problem discussed in Section 2.9.1.

2.4.2. Linkage Tree

The linkage tree FOS, first introduced in [49], is a hierarchical linkage model. It contains linkage sets of varying sizes, including all linkage sets of one problem variable, and the linkage set that includes all problem variables. Moreover, each linkage set with a size larger than one consists of the elements of two smaller linkage sets combined, i.e., for any $\mathcal{F}_i \in \mathcal{F}$ with $|\mathcal{F}_i| > 1$ there exist $\mathcal{F}_j, \mathcal{F}_k \in \mathcal{F}$ such that $i \neq j \neq k$, $\mathcal{F}_i = \mathcal{F}_j \cup \mathcal{F}_k$, and $\mathcal{F}_j \cap \mathcal{F}_k = \emptyset$.

The linkage tree FOS can be learned through agglomerative bottom-up hierarchical clustering. For this, a similarity notion is required between problem variables. In the dynamic case, an often used similarity notion is mutual information [26], which is estimated from the population. In real-valued optimization, when estimating Gaussian distributions, the mutual information between a pair of problem variables can for instance be estimated through the Pearson product-moment correlation coefficient.

The Unweighted Pair Grouping Method with Arithmetic-mean (UPGMA) clustering approach, whereby the similarity between two sets is the average similarity of all pairwise combinations, is then used for the construction of the linkage tree. This approach has a complexity of $\mathcal{O}(\ell^2)$ [19].

2.4.3. Bounded Fixed Linkage Tree

For certain problems, and arguably many large-scale real-valued problems, using a linkage tree that models dependencies up to the set of all problem variables is a waste of resources, because only relatively low-order dependencies are present in the problem. In this case, the Bounded Fixed Linkage Tree (BFLT) model, which was introduced by [4], can be used. The BFLT model is constructed in the same way as any linkage tree, but any merge of two linkage sets that would lead to a linkage set of a size larger than k is avoided, and the construction of the BFLT is terminated when no two available linkage sets can be merged into a new linkage set of size up to k.

A BFLT can be determined using a problem-specific notion of similarity between problem variables, e.g., Euclidean distance in various real-world problems [2, 32]. For this reason, a BFLT is not learned at the start of each generation, but prior to optimization.

Even though bounding the linkage tree can have a large impact on the efficiency of optimization, the construction of a BFLT with any $k > 1$ has the same complexity as the construction of a full linkage tree, i.e., $\mathcal{O}(\ell^2)$, as the calculation of all pairwise similarities is still required.

2.5. Partial Evaluations in Gray-Box Optimization

We speak of a GBO setting when the optimization problem allows for the application of partial evaluations, because in contrast to the BBO setting, this requires some knowledge of the optimization problem. A partial evaluation efficiently calculates the objective function(s) of a solution after the modification of a small number of problem variables, by subtracting the contribution of these variables to the objective function before modification, and adding their contribution after modification.

2.5.1. Definition

An optimization problem must be suitable for the application of partial evaluations. This is the case when it consists of k subfunctions, and it is known which variables each subfunction depends on. A given $\mathbf{I} = \{\mathcal{I}_0, \mathcal{I}_1, \ldots, \mathcal{I}_{k-1}\}$ defines that subfunction f_j^M has a dependency on all variables for which the index is included in $\mathcal{I}_j$. Other than this, each subfunction is considered to be a black box. A function to which partial evaluations can be applied, can be formulated in the following way [1]:

$$f(\boldsymbol{x}) = f^P\left(f_0^M(\boldsymbol{x}|_{\mathcal{I}_0}) \oplus f_1^M(\boldsymbol{x}|_{\mathcal{I}_1}) \oplus \cdots \oplus f_{k-1}^M(\boldsymbol{x}|_{\mathcal{I}_{k-1}})\right) = f^P\left(\bigoplus_{j=0}^{k-1} f_j^M(\boldsymbol{x}|_{\mathcal{I}_j})\right), \qquad (2.1)$$

where for each $j \in [0, \ldots, k-1]$, $f_j^M : \mathbb{R}^{|\mathcal{I}_j|} \to \mathbb{R}$ is a function of $\boldsymbol{x}|_{\mathcal{I}_j}$, a restricted number of elements of $\boldsymbol{x}$. The indices of $\boldsymbol{x}$ that $\boldsymbol{x}|_{\mathcal{I}_j}$ is restricted to are defined by a given $\mathcal{I}_j \subseteq$

$\mathcal{I} = \{0, 1, \ldots, \ell - 1\}$. The operator $\oplus$ can be any commutative binary operator for which we know an inverse operator $\ominus$, for example the summation or multiplication operators, though using the multiplication operator requires extra care to avoid division by zero [1].

In the case that $f^P : \mathbb{R} \to \mathbb{R}$ is the identity function, the partial evaluation of a solution $\boldsymbol{x}'$ that is the result of a solution $\boldsymbol{x}$ following the modification of $\boldsymbol{x}_i$, requires the calculation of each subfunction that has a dependency with $\boldsymbol{x}_i$, i.e.,:

$$f^{\mathtt{part}}(\boldsymbol{x}, f_{\boldsymbol{x}}, \boldsymbol{x}', i) = f_{\boldsymbol{x}} \ominus \bigoplus_{\mathcal{I}_j \ni i} f_j^M(\boldsymbol{x}|_{\mathcal{I}_j}) \oplus \bigoplus_{\mathcal{I}_j \ni i} f_j^M(\boldsymbol{x}'|_{\mathcal{I}_j}), \tag{2.2}$$

with $f_{\boldsymbol{x}}$ the given objective function of $\boldsymbol{x}$, and $\mathcal{I}_j \ni i$ shorthand for $\{\mathcal{I}_j \in \mathbf{I} | i \in \mathcal{I}_j\}$.

If f^P is any non-invertible function, the application of a partial evaluation requires maintaining the sum of all subfunctions Σ in memory. This Σ is updated following each partial evaluation. Given a solution $\boldsymbol{x}$ and

$$\Sigma = \bigoplus_{j=0}^{k-1} f_j^M(\boldsymbol{x}|_{\mathcal{I}_j}), \tag{2.3}$$

a partial evaluation following the modification of $\boldsymbol{x}_i$ can be performed as follows

$$f^{\mathtt{part}}(\boldsymbol{x}, \Sigma, \boldsymbol{x}', i) = f^P\left(\Sigma \ominus \bigoplus_{\mathcal{I}_j \ni i} f_j^M(\boldsymbol{x}|_{\mathcal{I}_j}) \oplus \bigoplus_{\mathcal{I}_j \ni i} f_j^M(\boldsymbol{x}'|_{\mathcal{I}_j})\right). \tag{2.4}$$

Note that the definition of a function can reveal information about its separability, because a variable interaction graph can be built based on $\mathbf{I}$.

2.5.2. Application

Because variation is applied to small subsets of variables by GOMEA, followed by an evaluation to determine whether the variation step should be accepted or rejected, the possibility of applying partial evaluations substantially improves the performance of GOMEA. Partial evaluations are used to efficiently update the objective value of a parent solution that is modified during GOM. If the parent solution is now in a state that achieves a better objective value than before its modification, this state of the parent solution is maintained in the population. Otherwise, the parent solution is returned to the state before its modification.

Knowing the separability of a problem, and how partial evaluations can be applied, does however not necessarily mean that the optimal linkage structure is immediately clear. For example, some non-separable problems can best be solved with a marginal product FOS due to relatively weak interactions between variables.

In contrast to GOMEA, state-of-the-art EAs like CMA-ES [21] and some of its variants [29, 47] cannot take full advantage of the possibility of partial evaluations, because solutions are only ever evaluated after all of their variables have been sampled anew. CMA-ES can benefit from a GBO scenario by using a covariance matrix that is restricted based on the decomposability of the problem, as this can be derived from the GBO problem definition. However, this benefit is marginal compared to the full potential of partial evaluations, because sep-CMA-ES [47] performed multiple orders of magnitude worse than RV-GOMEA on various decomposable benchmark problems in a GBO setting, in terms of time and number of function evaluations [4].

In Section 2.9.1 we discuss how partial evaluations can be applied to a series of benchmark problems. This includes problems that are non-separable, multi-modal, or multi-objective. Furthermore, partial evaluations have previously been applied to the real-world problems of medical deformable image registration [2] and the optimization of brachytherapy treatment plans for prostate cancer [32]. Both of these problems are non-separable, multi-objective and multi-modal, making them very well suited for optimization with an EA. The possibility of applying partial evaluations to these problems then gives GOMEA a substantial advantage compared to different algorithms that cannot benefit as much from partial evaluations.

2.6. REAL-VALUED GOMEA

RV-GOMEA [4] was introduced as a combination of GOMEA [50] and AMaLGaM [8]. The use of AMaLGaM as the method of sampling new problem-variable values is however not an absolute necessity. Instead, other real-valued EAs could be combined with GOMEA in order to benefit from its model-building capabilities. In this section, we present a general outline of RV-GOMEA, and we show how a different EA can be combined with GOMEA to form variations of RV-GOMEA. In Section 2.7.2 we then apply this to combine GOMEA with CMA-ES.

In RV-GOMEA, a population of solutions is maintained, and a linkage model is used to describe the linkage structure of the optimization problem (see Section 2.4). Until any one of the termination criteria are satisfied, possibly a budget in terms of time or number of evaluations, generations are performed, each consisting of variation and selection. The general outline of a generation of RV-GOMEA consists of applying GOM with each linkage set to each solution in the population $\mathcal{P}$ of size n, apart from the one elitist solution from the previous generation that is copied to the current generation. At the end of each generation, the parameters of the sampling model, e.g., the covariance matrix, are updated in a way defined by the sampling model itself. The application of GOM with some linkage set $\mathcal{F}_j$ to some parent solution $\boldsymbol{x}$ then consists of sampling a new partial solution $\boldsymbol{o}$, which contains values for all variables described by linkage set $\mathcal{F}_j$. All values of the partial solution $\boldsymbol{o}$ are then inserted into the parent $\boldsymbol{x}$, and the parent is evaluated by applying partial evaluations.

Partial evaluations can lead to numerical errors in the objective values of solutions, of which the magnitude depends on the optimization problem and the dimensionality of the instance being solved. Therefore, whenever a solution seems to have reached the Value-To-Reach (VTR), it is completely reevaluated to remove any numerical errors caused by partial evaluations. Moreover, errors in the objective values of solutions can have an impact on the overall optimization process. For this reason, the entire population is reevaluated every 50 generations by default.

Pseudo-code of the general outline of RV-GOMEA is shown in Algorithm 6.8, and pseudo-code of the GOM procedure is given in Algorithm 6.9. How GOM is performed, and how a sampling model is initialized and updated, strictly depends on how this sampling model is defined. The Anticipated Mean Shift (AMS) (see Section 2.7.3) and

Algorithm 2.1 RV-GOMEA

```
procedure RV-GOMEA($n$)
    $\mathcal{P} \leftarrow$ InitializeAndEvaluatePopulation($n$)
    $\mathcal{F} \leftarrow$ InitializeLinkageModel()
    for $j \in \{0, \ldots, |\mathcal{F}|-1\}$ do
        $\mathcal{M}_j \leftarrow$ InitializeSamplingModel($\mathcal{F}_j$)
    while $\neg$TerminationCriterionSatisfied() do
        $\mathcal{P}_0 \leftarrow \boldsymbol{x}^{\texttt{elitist}}$
        for $j \in \{0, \ldots, |\mathcal{F}|-1\}$ do                    ▷ Random order
            for $\boldsymbol{x} \in \mathcal{P}_{[1 \ldots n-1]}$ do
                GOM($\boldsymbol{x}, \mathcal{F}_j, \mathcal{M}_j$)
            IntermediateUpdate($\mathcal{M}_j$)
        for $\boldsymbol{x} \in \mathcal{P}_{[1 \ldots n_{\text{AMS}}]}$ do AnticipatedMeanShift($\boldsymbol{x}$)
        for $\boldsymbol{x} \in \mathcal{P}_{[1 \ldots n-1]}$ do
            if $\text{NIS}(\boldsymbol{x}) > \text{NIS}^{\text{MAX}}$ then
                ForcedImprovement($\boldsymbol{x}$)
        for $j \in \{0, \ldots, |\mathcal{F}|-1\}$ do
            UpdateSamplingModel($\mathcal{M}_j$)
```

forced-improvement procedures (see Section 2.7.4) are used for the real-valued instantiation of GOMEA, as these procedures are specifically targeted at real-valued optimization, and proved to be successful [4]. The forced-improvement procedure is only applied to solutions that have not been improved for a number of generations larger than the maximum No-Improvement Stretch (NIS) [8].

Algorithm 2.2 Gene-pool Optimal Mixing

```
procedure GOM($\boldsymbol{x}, \mathcal{F}_j, \mathcal{M}_j$)
    for $u \in \{0, \ldots, |\mathcal{F}_j|-1\}$ do
        $\boldsymbol{b}[u] \leftarrow \boldsymbol{x}[\mathcal{F}_j[u]]$
    $\boldsymbol{o} \leftarrow$ SamplePartialSolution($\mathcal{M}_j$)
    for $u \in \{0, \ldots, |\mathcal{F}_j|-1\}$ do
        $\boldsymbol{x}[\mathcal{F}_j[u]] \leftarrow \boldsymbol{o}[u]$
    $\boldsymbol{f_o} \leftarrow$ PartialEvaluation($\boldsymbol{x}, \boldsymbol{f_x}, \mathcal{F}_j$)
    if $\boldsymbol{f_o} < \boldsymbol{f_x}$ or $\mathcal{U}(0,1) < p^{\texttt{accept}}$ then
        $\boldsymbol{f_x} = \boldsymbol{f_o}$
    else
        for $u \in \{0, \ldots, |\mathcal{F}_j|-1\}$ do
            $\boldsymbol{x}[\mathcal{F}_j[u]] \leftarrow \boldsymbol{b}[u]$
```

2.7. Single-Objective Optimization

In this section we describe the application of two different sampling models to RV-GOMEA. The first application uses a sampling model based on AMaLGaM, as was introduced in [4], and the second application uses a sampling model based on CMA-ES [20, 21]. We refer to the former as RV-GOMEA$^{\text{A}}$, and to the latter as RV-GOMEA$^{\text{C}}$. The combination of GOMEA and CMA-ES was selected, because CMA-ES is a well-known algorithm that is arguably the state of the art of EAs for real-valued BBO. Both in AMaLGaM and in CMA-ES a multivariate normal probability distribution is used to sample new variables, but a different approach to determining the parameters of this probability distribution is taken in these algorithms. The AMS and forced improvement procedures are applied to both variants of RV-GOMEA, as displayed in Algorithm 6.8.

2.7.1. AMALGAM SAMPLING MODEL

We now define the structure and parameters of the sampling model used by this instance of RV-GOMEA, based on AMaLGaM [8]. For the sampling model of each FOS element, a multivariate normal probability distribution is maintained, with each of the dimensions of this probability distribution corresponding to one of the variables included in the linkage set. This probability distribution $\mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{C}_j)$ is defined in terms of the mean vector $\boldsymbol{\mu}_j$ and the covariance matrix $\boldsymbol{C}_j$, describing the means of all variables in $\mathcal{F}_j$, and the covariances of all pairs of variables in $\mathcal{F}_j$, respectively. Each such a probability distribution is estimated with maximum-likelihood based on the selection, and then multiplied by the distribution multiplier $c_j^{\texttt{Mul}}$, which is described in Section 2.7.1. The selection $\mathcal{S}$ consists of the τn best solutions in the population, with $\tau = 0.35$, as in [8]. Selection and the estimation of the probability distribution are performed during the initialization of each sampling model, and during the update procedure of each sampling model. Pseudo-code of this instance of RV-GOMEA is displayed in Algorithm 2.3.

GENE-POOL OPTIMAL MIXING

The normal probability distribution that is maintained, is used to sample new partial solutions, i.e., values for subsets of variables, during the GOM phase. In this phase, for each linkage set, GOM is applied to all but $n^{\texttt{elitist}} = 1$ solutions in the population. During the GOM phase of a linkage set $\mathcal{F}_j$, partial variation is applied to all solutions in the population by sampling new partial solutions describing new values for all variables in $\mathcal{F}_j$. In order to sample from the distribution $\mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{C}_j)$, a Cholesky decomposition is applied to $\boldsymbol{C}_j$ to find $\boldsymbol{C}_j = \boldsymbol{L}_j \boldsymbol{L}_j^*$. A sample $\boldsymbol{o} \sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{C}_j)$ is then obtained through

$$\boldsymbol{o} = \boldsymbol{\mu}_j + \boldsymbol{L}_j \cdot \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}). \tag{2.5}$$

The resulting offspring solution is then evaluated, and the modification of the parent solution is accepted if the offspring has a better objective value than the parent. If no better objective value is achieved by the offspring, the modification is accepted with a probability of $p^{\texttt{accept}} = 0.05$, based on preliminary experiments, because this encourages exploration and increases the likelihood of escaping local optima.

ANTICIPATED MEAN SHIFT

A fraction 0.5τ, corresponding to [8], of newly sampled partial solutions is subject to the AMS, moving them in the direction of generational improvement. This is done by

Algorithm 2.3 RV-GOMEA$^{\text{A}}$

```
procedure InitializeSamplingModel($\mathcal{F}_j$)
    $\boldsymbol{L} \leftarrow \boldsymbol{I}_{|\mathcal{F}_j|}$                                   ▷ $|\mathcal{F}_j| \times |\mathcal{F}_j|$
    $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}^{\texttt{AMS}} \leftarrow \boldsymbol{0}$
    $c^{\texttt{Mul}} \leftarrow 1$
    $\mathcal{M}_j \leftarrow \{\mathcal{F}_j, \boldsymbol{L}, \boldsymbol{\mu}, \boldsymbol{\mu}^{\texttt{AMS}}, c^{\texttt{Mul}}\}$
    UpdateSamplingModel($\mathcal{M}_j$)
    return $\mathcal{M}_j$
procedure UpdateSamplingModel($\{\mathcal{F}_j, \boldsymbol{L}, \boldsymbol{\mu}, \boldsymbol{\mu}^{\texttt{AMS}}, c^{\texttt{Mul}}\}$)
    $\mathcal{S} \leftarrow \lfloor \tau n \rfloor$ best solutions
    for $u \in \{0, \ldots, |\mathcal{F}_j| - 1\}$ do
        $\boldsymbol{\mu}^{\texttt{new}}[u] \leftarrow \frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{s} \in \mathcal{S}} \boldsymbol{s}[u]$
    if NumberOfGenerations $> 0$ then
        $\boldsymbol{\mu}^{\texttt{AMS}} \leftarrow \boldsymbol{\mu}^{\texttt{new}} - \boldsymbol{\mu}$
    $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}^{\texttt{new}}$
    for $u, v \in \{0, \ldots, |\mathcal{F}_j| - 1\}$ do
        $\boldsymbol{C}[u, v] \leftarrow \frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{s} \in \mathcal{S}} (\boldsymbol{s}[u] - \boldsymbol{\mu}[u])(\boldsymbol{s}[v] - \boldsymbol{\mu}[v])$
        $\boldsymbol{C}[u, v] \leftarrow c^{\texttt{Mul}} \boldsymbol{C}[u, v]$
    $\boldsymbol{L}\boldsymbol{L}^* \leftarrow$ CholeskyDecomposition($\boldsymbol{C}$)
    $f^{\texttt{elitist}} \leftarrow \min_{\boldsymbol{x} \in \mathcal{P}}\{f(\boldsymbol{x})\}$
procedure IntermediateUpdate($\{\mathcal{F}_j, \boldsymbol{L}, \boldsymbol{\mu}, \boldsymbol{\mu}^{\texttt{AMS}}, c^{\texttt{Mul}}\}$)
    $\boldsymbol{X}^{\texttt{Improved}} \leftarrow \{\boldsymbol{x} : f(\boldsymbol{x}) < f^{\texttt{elitist}}\}$
    if $\boldsymbol{X}^{\texttt{Improved}} = \emptyset$ then
        if $c^{\texttt{Mul}} > 1$ then $c^{\texttt{Mul}} = \eta^{\texttt{DEC}} \cdot c^{\texttt{Mul}}$
        if $c^{\texttt{Mul}} < 1$ then $c^{\texttt{Mul}} = 1$
    else
        if $c^{\texttt{Mul}} < 1$ then $c^{\texttt{Mul}} = 1$
        for $u \in \mathcal{F}_j$ do
            $\boldsymbol{x}^{\texttt{Avg-imp}}[u] = \frac{1}{|\boldsymbol{X}^{\texttt{Improved}}|} \sum_{\boldsymbol{x} \in \boldsymbol{X}^{\texttt{Improved}}} \boldsymbol{x}[u]$
        $\boldsymbol{z}^{\texttt{Avg-imp}} = \boldsymbol{L}^{-1}\left(\boldsymbol{x}^{\texttt{Avg-imp}} - \boldsymbol{\mu}\right)$
        SDR $\leftarrow \max_{u \in \mathcal{F}_j}\{|\boldsymbol{z}^{\texttt{Avg-imp}}[u]|\}$
        if SDR $> \theta^{\text{SDR}}$ then $c^{\texttt{Mul}} = \eta^{\texttt{INC}} \cdot c^{\texttt{Mul}}$
    $f^{\texttt{elitist}} \leftarrow \min_{\boldsymbol{x} \in \mathcal{P}}\{f(\boldsymbol{x})\}$
procedure SamplePartialSolution($\{\mathcal{F}_j, \boldsymbol{L}, \boldsymbol{\mu}, \boldsymbol{\mu}^{\texttt{AMS}}, c^{\texttt{Mul}}\}$)
    $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$                           ▷ $|\mathcal{F}_j|$-dimensional vector
    $\boldsymbol{o} \leftarrow \boldsymbol{\mu} + \boldsymbol{z}\boldsymbol{L}$                                 ▷ Eq. 2.5
    if ApplyAMS then                                  ▷ applied to $\tau n/2$ solutions
        $\boldsymbol{o} \leftarrow \boldsymbol{o} + \delta^{\texttt{AMS}} c^{\texttt{Mul}} \boldsymbol{\mu}^{\texttt{AMS}}$                   ▷ Eq. 2.6
    return $\boldsymbol{o}$
```

adding the difference between the means of consecutive generations, times a multiplicative factor, to the partial solution $\boldsymbol{o}$, i.e.,

$$\boldsymbol{o}^{\texttt{AMS}} = \boldsymbol{o} + \delta^{\texttt{AMS}} c_j^{\texttt{Mul}} \left(\boldsymbol{\mu}_j^{(g)} - \boldsymbol{\mu}_j^{(g-1)} \right), \tag{2.6}$$

where $\delta^{\texttt{AMS}} = 2$, $\boldsymbol{\mu}_j^{(g)}$ is the mean vector of all variables in $\mathcal{F}_j$ in generation g, and $c_j^{\texttt{Mul}}$ is the distribution multiplier that is described in Section 2.7.1.

Adaptive Variance Scaling

Each covariance matrix is scaled by a distribution multiplier that is used to adaptively control the kernel size of the probability distribution. The procedure of adapting the distribution multiplier is called Adaptive Variance Scaling (AVS) [8]. Parameters of the AVS are set corresponding to [8]. The value of the distribution multiplier is initialized to 1 and scaled by either the factor $\eta^{\texttt{DEC}} = 0.9$ or the factor $\eta^{\texttt{INC}} = 1/\eta^{\texttt{DEC}}$ at the end of each generation, depending on the success of the optimization procedure. More specifically, to update the distribution multiplier $c_j^{\texttt{Mul}}$ of a sampling model $\mathcal{M}_j$, we first find the set of solutions $\boldsymbol{X}_j^{\texttt{Improved}}$. This set contains the offspring solutions that were produced by the GOM phase of $\mathcal{M}_j$ and obtained an objective value better than that of the elitist solution at the start of the GOM phase of $\mathcal{M}_j$. If no such improvements were found, i.e., $\boldsymbol{X}_j^{\texttt{Improved}} = \emptyset$, $c_j^{\texttt{Mul}}$ is multiplied by the factor $\eta^{\texttt{DEC}}$ to narrow down the size of the search space.

If $\boldsymbol{X}_j^{\texttt{Improved}}$ is non-empty, $c^{\texttt{Mul}}$ is first reset to 1 if it was smaller than 1. Subsequently, $c^{\texttt{Mul}}$ is adapted if improvements were found far away from the sample mean, determined by the Standard Deviation Ratio (SDR) approach. For this purpose, the vector $\boldsymbol{x}_j^{\texttt{Avg-imp}}$ defines, for all variables in $\mathcal{F}_j$, the average values of all solutions in $\boldsymbol{X}_j^{\texttt{Improved}}$. A transformation is applied to $\boldsymbol{x}_j^{\texttt{Avg-imp}}$ in order to find the vector of standard deviation ratios $\boldsymbol{z}_j^{\texttt{Avg-imp}} = \boldsymbol{L}_j^{-1} \left(\boldsymbol{x}_j^{\texttt{Avg-imp}} - \boldsymbol{\mu}_j \right)$, which describes the average direction of improvement in terms of the standard normal probability distribution $\mathcal{N}(0,1)$ for each variable. If any dimension of $\boldsymbol{z}_j^{\texttt{Avg-imp}}$ is larger than $\theta^{\texttt{SDR}} = 1$ in absolute terms, the average improvement is considered to be far away from the sample mean, resulting in the multiplication of $c_j^{\texttt{Mul}}$ by the factor $\eta^{\texttt{INC}}$, because the kernel of the probability distribution should be enlarged to find solutions far away from the mean.

2.7.2. CMA-ES Sampling Model

In this section we describe our implementation of a CMA-ES-based [20, 21] sampling model in RV-GOMEA. Pseudo-code of this instance of RV-GOMEA is displayed in Algorithm 2.4. For readability we omit the subscript j that indicates that these parameters belong to a sampling model $\mathcal{M}_j$, because all these parameters are adapted in the same way, with the only difference being the number of parameters that are modeled.

Each sampling model $\mathcal{M}_j$ consists of all parameters of CMA-ES as in [21], modeling the probability distribution of the set of variables in $\mathcal{F}_j$. Each of these sampling models based on CMA-ES maintains the parameters that define the $\mathcal{F}_j$-dimensional normal

Algorithm 2.4 RV-GOMEA$^{\mathrm{C}}$

```
procedure InitializeSamplingModel($\mathcal{F}_j$)
    $\boldsymbol{C} \leftarrow \boldsymbol{B} \leftarrow \boldsymbol{D} \leftarrow \boldsymbol{I}_{|\mathcal{F}_j|}$    ▷ $|\mathcal{F}_j| \times |\mathcal{F}_j|$
    $\boldsymbol{p}_c \leftarrow \boldsymbol{p}_\sigma \leftarrow \boldsymbol{0}$
    $\sigma \leftarrow \frac{1}{2|\mathcal{F}_j|} \sum_{u \in \mathcal{F}_j} [\texttt{MaxInit}(u) - \texttt{MinInit}(u)]$
    return $\{\boldsymbol{C}, \boldsymbol{B}, \boldsymbol{D}, \boldsymbol{p}_c, \boldsymbol{p}_\sigma, \sigma\}$
procedure UpdateSamplingModel($\{\boldsymbol{C}, \boldsymbol{B}, \boldsymbol{D}, \boldsymbol{p}_c, \boldsymbol{p}_\sigma, \sigma\}$)
    $\mathcal{S} \leftarrow \lfloor \tau n \rfloor$ best solutions
    $\boldsymbol{p}_c \leftarrow (1 - c_c) \cdot \boldsymbol{p}_c + c_c^u \cdot \frac{c_w}{\sigma} \left( \langle \boldsymbol{x} \rangle_w^{(g+1)} - \langle \boldsymbol{x} \rangle_w^{(g)} \right)$    ▷ Eq. 2.7
    $\boldsymbol{C} \leftarrow (1 - c_{\mathrm{cov}}) \cdot \boldsymbol{C} + c_{\mathrm{cov}} \cdot \boldsymbol{p}_c \left( \boldsymbol{p}_c \right)^{\mathrm{T}}$    ▷ Eq. 2.8
    $\boldsymbol{p}_\sigma \leftarrow (1 - c_\sigma) \cdot \boldsymbol{p}_\sigma + c_\sigma^u \cdot c_w \boldsymbol{B} \langle \boldsymbol{z} \rangle_w$    ▷ Eq. 2.9
    $\sigma \leftarrow \sigma \cdot \exp\left( \frac{1}{d_\sigma} \cdot \frac{\|\boldsymbol{p}_\sigma\| - \widehat{\chi}_j}{\widehat{\chi}_j} \right)$    ▷ Eq. 2.10
    $(\boldsymbol{B}, \boldsymbol{D}) \leftarrow \texttt{EigenDecomposition}(\boldsymbol{C})$
procedure IntermediateUpdate($\{\boldsymbol{C}, \boldsymbol{B}, \boldsymbol{D}, \boldsymbol{p}_c, \boldsymbol{p}_\sigma, \sigma\}$)
    // No intermediate update is required.
procedure SamplePartialSolution($\{\boldsymbol{C}, \boldsymbol{B}, \boldsymbol{D}, \boldsymbol{p}_c, \boldsymbol{p}_\sigma, \sigma\}$)
    $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{|\mathcal{F}_j|})$    ▷ $|\mathcal{F}_j|$-dimensional vector
    $\boldsymbol{o} \leftarrow \langle \boldsymbol{x} \rangle_w + \sigma \boldsymbol{B} \boldsymbol{D} \boldsymbol{z}$    ▷ Eq. 2.11
    return $\boldsymbol{o}$
```

probability distribution $\mathcal{N}(\boldsymbol{\mu}_j, \sigma_j^2 \boldsymbol{C}_j)$ from which new solutions are sampled, where $\boldsymbol{\mu}_j$ is the mean vector, $\boldsymbol{C}_j$ is the covariance matrix, and σ_j is the step size parameter. All these parameters are updated at the start of each generation, separately for each sampling model.

Parameter Adaptation

In order to update μ and $\boldsymbol{C}$, truncation selection is performed to find the selection $\mathcal{S}$ consisting of the best $\lfloor \tau n \rfloor$ individuals in $\mathcal{P}$. The vectors $\langle \boldsymbol{x} \rangle_w$ and $\langle \boldsymbol{z} \rangle_w$ are the weighted average of the solutions in $\mathcal{S}$ and the weighted average of the $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$-distributed samples $\boldsymbol{z}$ of these individuals, respectively, which are computed based on the selection. The weight of a solution with rank $0 \le r \le n-1$ is $\log(|\mathcal{S}|+1) - \log(r+1)$, and all weights are normalized such that their sum is equal to 1.

The evolution path $\boldsymbol{p}_c$ is then adapted, which determines how the covariance matrix $\boldsymbol{C}$ is adapted. This is done according to [21], i.e.,

$$\boldsymbol{p}_c^{(g+1)} = (1 - c_c) \cdot \boldsymbol{p}_c^{(g)} + c_c^u \cdot \frac{c_w}{\sigma^{(g)}} \left(\langle \boldsymbol{x} \rangle_w^{(g+1)} - \langle \boldsymbol{x} \rangle_w^{(g)} \right), \tag{2.7}$$

$$\boldsymbol{C}^{(g+1)} = (1 - c_{\mathrm{cov}}) \cdot \boldsymbol{C}^{(g)} + c_{\mathrm{cov}} \cdot \boldsymbol{p}_c^{(g+1)} \left(\boldsymbol{p}_c^{(g+1)} \right)^{\mathrm{T}}, \tag{2.8}$$

with c_c the cumulation constant, and c_c^u and c_w normalization factors. These parameters are set corresponding to an $|\mathcal{F}_j|$-dimensional instance of CMA-ES [21].

The step size parameter σ is adapted based on a separate evolution path $\boldsymbol{p}_\sigma$ in the following way, according to [21]:

$$\boldsymbol{p}_\sigma^{(g+1)} = (1 - c_\sigma) \cdot \boldsymbol{p}_\sigma^{(g)} + c_\sigma^u \cdot c_w \boldsymbol{B}^{(g)} \langle \boldsymbol{z} \rangle_w^{(g+1)}, \tag{2.9}$$

$$\sigma^{(g+1)} = \sigma^{(g)} \cdot \exp\left(\frac{1}{d_\sigma} \cdot \frac{\|\boldsymbol{p}_\sigma^{(g+1)}\| - \widehat{\chi}_j}{\widehat{\chi}_j} \right), \tag{2.10}$$

with c_σ the cumulation constant, c_σ^u a normalization factor, d_σ the damping parameter, and $\widehat{\chi}_j$ (an approximation of) the expected length of a vector drawn from the $|\mathcal{F}_j|$-dimensional normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. These parameters are set corresponding to an $|\mathcal{F}_j|$-dimensional instance of CMA-ES [21].

The update rule for the evolution path of σ is dependent on $\boldsymbol{B}\langle \boldsymbol{z} \rangle_w$, i.e., the rotation matrix $\boldsymbol{B}$ and the weighted mean of the sample vector $\boldsymbol{z}$, because this indicates the direction of improvement. However, $\boldsymbol{B}$ is updated every generation, while $\boldsymbol{z}$ could have been sampled more than one generation ago, because later samples were rejected. Additionally, the weight of each solution that contributes to $\langle \boldsymbol{z} \rangle_w$ changes every generation, because it depends on the relative quality of the solution. This means that, if $\boldsymbol{B}$ and $\boldsymbol{z}$ originate from a different generation, the product $\boldsymbol{zB}$ results in a vector unrelated to the direction of improvement. Instead $\boldsymbol{z}$ and $\boldsymbol{B}$, both originating from the same generation, would have to be stored until all values of $\boldsymbol{z}$ have been replaced by newer samples. This would increase the overall memory complexity to $\mathcal{O}(n\ell^2)$, which should be avoided. Therefore we set the value of $\boldsymbol{z}$ to 0 if the sample in the current generation is rejected, because this would lead to a mismatch between $\boldsymbol{z}$ and $\boldsymbol{B}$.

GENE-POOL OPTIMAL MIXING

To sample from the distribution with covariance matrix $\boldsymbol{C}$, an eigendecomposition is applied to find $\boldsymbol{C} = \boldsymbol{BD}^2\boldsymbol{B}^T$, where $\boldsymbol{B}$ is an orthogonal matrix that determines the coordinate system, and $\boldsymbol{D}$ is a diagonal matrix that scales the dimensions of this coordinate system accordingly. For a sampling model $\mathcal{M}_j$, a $|\mathcal{F}_j|$-dimensional partial solution $\boldsymbol{o}$ is sampled from $\mathcal{N}(\boldsymbol{\mu}, \sigma^2\boldsymbol{C})$ as follows:

$$\boldsymbol{o}^{(g+1)} = \langle \boldsymbol{x} \rangle_w^{(g)} + \sigma^{(g)} \boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)} \boldsymbol{z}^{(g+1)}, \tag{2.11}$$

with $\boldsymbol{z}^{(g+1)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

As discussed in Section 2.6, the population has to be periodically reevaluated to avoid large numerical errors. However, reevaluating the population once every 50 generations, as is done in RV-GOMEA$^{\text{A}}$, appeared insufficient in preliminary experiments to efficiently solve high-dimensional problems with RV-GOMEA$^{\text{C}}$, because the numerical errors have too much of an influence on the optimization process. This can be seen in the convergence plots in Figure 2.1. Figure 2.1 shows runs of RV-GOMEA$^{\text{C}}$ with three different settings for the number of generations after which the complete population is reevaluated. Note that all runs took fewer than 50 generations, so the population was never reevaluated for the setting with reevaluation every 50 generations. If the population is reevaluated every 50 generations, RV-GOMEA$^{\text{C}}$ gets ‘stuck’ near the VTR, because each solution that appears to reach the VTR is reevaluated, but then turns out to be worse than the VTR due to numerical errors. This is the cause of a large inefficiency, most notably in highly-dimensional problems, and can even cause premature convergence. For RV-GOMEA$^{\mathrm{C}}$ we therefore choose to reevaluate the population every 10 generations, as this is a reasonable setting given the results in Figure 2.1.

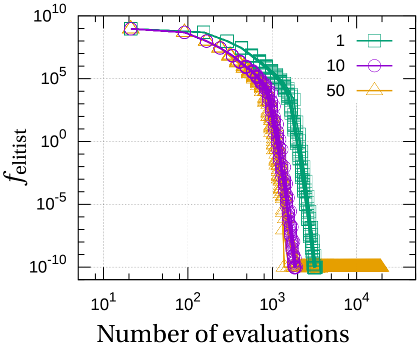


Figure 2.1: Convergence plots of typical runs of RV-GOMEA$^{\mathrm{C}}$ on the 80960-dimensional sphere problem, with different settings for the number of generations after which the complete population is reevaluated.

2.7.3. Anticipated Mean Shift

Similar to the AMS applied to partial solutions in Section 2.7.1, AMS is applied to all variables of a fraction 0.5τ of the population directly following GOM, with the purpose of moving solutions in the direction of generational improvement for all problem variables simultaneously. This application of AMS is only accepted when it leads to an improvement, or with a probability of p^{accept}. AMS is applied as follows:

$$\boldsymbol{x}^{\mathtt{AMS}} = \boldsymbol{x} + \delta^{\mathtt{AMS}}\left(\boldsymbol{\mu}^{(g)} - \boldsymbol{\mu}^{(g-1)}\right), \tag{2.12}$$

where $\delta^{\mathtt{AMS}} = 2$, and $\boldsymbol{\mu}_j^{(g)}$ is the mean vector of all variables in generation g.

2.7.4. Forced Improvements

A phase of forced improvements was introduced in the discrete GOMEA to force solutions out of local optima if they have not improved for a certain number of generations. For each solution $\boldsymbol{x}$, the number of generations of no improvement NIS($\boldsymbol{x}$) is kept track of. This counter is reset whenever GOM or AMS resulted in an improvement of $\boldsymbol{x}$. Note that NIS($\boldsymbol{x}$) is not reset when a step of GOM is accepted due to $p^{\mathtt{accept}}$. Whenever NIS($\boldsymbol{x}$) reaches NIS$^{\mathtt{MAX}} = 100$, the forced improvement phase is applied to $\boldsymbol{x}$. This phase consists of multiple rounds, during each of which GOM is applied to $\boldsymbol{x}$ for each linkage set. However, all new partial solutions that are inserted into $\boldsymbol{x}$ are a weighted average of the variables of $\boldsymbol{x}$ and the elitist solution $\boldsymbol{x}^{\mathtt{elitist}}$, where the weight of $\boldsymbol{x}$ is α and the weight of $\boldsymbol{x}^{\mathtt{elitist}}$ is $1-\alpha$. If the modification of $\boldsymbol{x}$ leads to an improvement of its objective value, this modification is accepted and the forced improvement phase terminates. After one round of GOM operations has been applied to $\boldsymbol{x}$, but no improvements have been found, the value of α is multiplied by 0.5, and the next round of GOM operations starts. This process continues until α has reached a value below 0.01, at which point a copy of $\boldsymbol{x}^{\texttt{elitist}}$ replaces $\boldsymbol{x}$ in the population.

2.8. MULTI-OBJECTIVE OPTIMIZATION

Optimization problems often involve multiple conflicting objective functions that require optimization. Again, without loss of generality, we assume that all objective functions need to be minimized. A weighted average of multiple objective functions could be optimized by a single-objective algorithm, but such an approach might not always result in the desired outcome, because such weights can be difficult to tune. Instead, the goal of multi-objective optimization is finding a set of solutions, each of which has a different optimal trade-off between each of the objective functions.

In the setting of multi-objective optimization, a solution $\boldsymbol{x}$ is considered to dominate a solution $\boldsymbol{y}$, denoted $\boldsymbol{x} \succ \boldsymbol{y}$, if it is better than $\boldsymbol{y}$ in at least one objective function, and at least as good as $\boldsymbol{y}$ in all other objective functions. If neither $\boldsymbol{x} \succ \boldsymbol{y}$ nor $\boldsymbol{y} \succ \boldsymbol{x}$ holds, $\boldsymbol{x}$ and $\boldsymbol{y}$ are said to be non-dominating. A solution is Pareto optimal if no solution exists that dominates it. All Pareto-optimal solutions comprise the so-called Pareto set. In objective space, the solutions in a Pareto set form a so-called Pareto front. EAs are known to be among the state of the art in multi-objective optimization [15]. Multi-objective EAs produce a so-called approximation set, a set of solutions that is as close as possible to the Pareto front, and is spread across all regions of the Pareto front.

2.8.1. MULTI-OBJECTIVE RV-GOMEA

Extending the RV-GOMEA introduced in Section 2.3 from single-objective optimization to multi-objective optimization is relatively simple with the multi-objective framework introduced by [7]. This framework employs selection based on non-dominated sorting and a clustering procedure in order to spread the search effort across different regions of the objective space, because the goal of multi-objective optimization is to find many solutions, presumably consisting of widely varying problem variable values, near each region of the Pareto front. Clustering was previously shown to be beneficial for finding a good spread of solutions across the regions of the Pareto front [44]. [7] applied the aforementioned multi-objective framework to AMaLGaM, resulting in the Multi-Objective AMaLGaM (MAMaLGaM) where basically each cluster is provided its own AMaLGaM estimation and sampling procedure. Cluster registration, i.e., the one-to-one matching of clusters in generation g and those in generation $g+1$, is performed, in order to apply various mechanisms that rely on parameter settings from multiple generations, such as AMS and AVS.

Similar to how the multi-objective framework is applied to AMaLGaM, we can apply it to RV-GOMEA by maintaining a sampling model for each cluster. Each cluster should also have the possibility of having a different linkage model, because different dependencies could be of importance in different regions of the Pareto front. Moreover, even if the same linkage model is used by each cluster, the optimal distribution variable values of a specific linkage set are presumably different at different regions of the Pareto front. Therefore, a model $\mathcal{M}_{ij}$ is maintained for each linkage set $\mathcal{F}_j$ in each cluster $\mathcal{C}_i$.

Clustering

To distinguish the different regions of the Pareto front that are to be approached by different clusters, the selection $\mathcal{S}$ is clustered into q possibly overlapping clusters, each consisting of $c = 2\tau\frac{n}{q}$ solutions. All distances in the clustering procedure are Euclidean distances in objective space, where each dimension is normalized by the range of the current selection. A so-called single-objective cluster is first created for each objective, each consisting of the best c solutions in that objective. To establish the remaining $q - m$ clusters, $q - m$ far apart cluster leaders are heuristically chosen from the selection. The first cluster leader is a solution that is the maximum in a randomly chosen dimension. All $q - m - 1$ remaining cluster leaders are then iteratively selected by choosing the solution that is furthest away from all previously selected cluster leaders. The process of selecting a subset of far apart solutions is further on referred to as scattered subset selection. Cluster members are then determined by assigning the closest c solutions in the selection to each cluster leader, meaning that some solutions in the selection can be assigned to more than one cluster, or to no cluster.

In contrast to MAMaLGaM, where only the selection $\mathcal{S}$ is clustered, the application of the multi-objective framework to RV-GOMEA will require the clustering of the entire population, because each solution in the population must receive new samples from a sampling model that is directly associated with a cluster. After clusters consisting of solutions in $\mathcal{S}$ have been established, all solutions in the population (including the selection) are assigned to exactly one cluster. To ensure a minimum size of each cluster, we first assign $c = 2\tau\frac{n}{q}$ solutions to each cluster in a round robin fashion. In each round, for each cluster, the closest (with respect to the cluster mean) non-assigned solution is assigned to the cluster, starting with the single-objective clusters. This process is repeated until c solutions are assigned to each cluster. All non-assigned solutions are then assigned to their closest (again with respect to the cluster mean) cluster.

Elitist Archive

An adaptive elitist archive, introduced in [31], is used to keep track of a set of non-dominated solutions with a desired target size. This elitist archive is set to a maximum capacity of 125% of its target size, and the elitist archive is adapted when the maximum capacity is exceeded. This adaptation condition is checked at the end of each generation. While the elitist archive has not been adapted yet, this condition is also checked after each application of GOM, to prevent the elitist archive from growing substantially beyond its maximum capacity.

The objective space of the elitist archive is then discretized into a regular grid, and no more than one solution is allowed in each grid cell. Different grid resolutions are attempted in a binary search manner to find a grid resolution such that the size of the elitist archive is close to 75% of its target size. Whenever a solution would be added to the elitist archive, and it would be located in an already occupied grid cell, either the new solution or the solution preexisting in this cell is selected to be stored in the archive. If either of the solutions is dominating the other, the dominating solution is selected. Otherwise, the preexisting solution remains in the archive.

MIXING

Accepting or rejecting the modification of a solution during GOM is based on whether the modification resulted in an improvement of the solution in question. Contrary to single-objective optimization, however, an improvement is not clearly defined in multi-objective optimization. A modified solution that dominates the previous state of the solution clearly counts as an improvement, but little can be said when the two solutions are non-dominating. A modification leading to a non-dominating state could very well lead to a solution that is further from the Pareto front. We therefore adopt the acceptation criteria that were previously used in the (discrete) Multi-Objective GOMEA (MO-GOMEA) [33]. This means that a modification is only accepted if it leads to a solution that dominates the parent solution, or if the resulting solution is not dominated by any solution in the elitist archive. In the multi-objective framework, we set $p^{\texttt{accept}} = 0$, as the acceptation of sideways (non-dominating) steps serves the same purpose.

2.8.2. AMaLGaM SAMPLING MODEL

The adaptation of GOMEA for multi-objective real-valued optimization was previously introduced by [5] as the Multi-Objective RV-GOMEA (MO-RV-GOMEA) and uses the AMaLGaM sampling model.

2.8.3. CMA-ES SAMPLING MODEL

Extending RV-GOMEA$^{\text{C}}$ to its multi-objective variant, MO-RV-GOMEA$^{\text{C}}$, is done through the straightforward application of the multi-objective framework discussed in Section 2.8.1 and thus in the same way that RV-GOMEA$^{\text{A}}$ is extended to MO-RV-GOMEA$^{\text{A}}$. A separate linkage model is used for each cluster, and each linkage model in each cluster maintains a sampling model for each of its linkage sets. All methods discussed in Section 2.8.1 are then applied.

2.9. EXPERIMENTS

2.9.1. BENCHMARK PROBLEMS

This section introduces the set of single- and multi-objective benchmark problems used in this chapter.

SINGLE-OBJECTIVE

The set of single-objective optimization problems consists of the sphere [13], Rosenbrock [48], Rastrigin [46], Michalewicz [35], SoREB [4] functions, and a step function, all subject to minimization.

From this set of functions, the sphere function is clearly the easiest to optimize, because it has no (non-global) local optima, and it is completely dimension-wise decomposable. The sphere function is defined as:

$$f_{\text{sphere}}(\boldsymbol{x}) = \sum_{i=0}^{\ell-1} \boldsymbol{x}_i^2.$$

Applying partial evaluations to the sphere function is relatively straightforward, and is done in the following way: given a solution $\boldsymbol{x}$ and its objective value $f_{\boldsymbol{x}}$. If some entry $\boldsymbol{x}_i$ is then modified to $\boldsymbol{x}'_i$, the objective value of the entire modified solution $\boldsymbol{x}'$ is calculated through

$$f_{\text{sphere}}(\boldsymbol{x}') = f_{\boldsymbol{x}} - \boldsymbol{x}_i^2 + \boldsymbol{x}'^2_i,$$

which is done in $\mathcal{O}(1)$ time.

The second benchmark problem that we consider is the Rosenbrock function, because this is a relatively difficult function to which partial evaluations can be applied despite the fact that it is not dimension-wise decomposable. The Rosenbrock function is defined as:

$$f_{\text{Rosenbrock}}(\boldsymbol{x}) = \sum_{i=0}^{\ell-2} \left[100\left(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i^2\right)^2 + (1-\boldsymbol{x}_i)^2\right].$$

For the application of partial evaluations to the Rosenbrock function, we define

$$f_{\text{Rb}}(\boldsymbol{x}, i) = 100\left(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i^2\right)^2 + (1-\boldsymbol{x}_i)^2,$$

which is equal to the $i^{\text{th}}(0 \le i \le \ell - 2)$ term of the Rosenbrock function. We then define the function that isolates all terms of the Rosenbrock function that are dependent on some variable with index i $(0 \le i \le \ell - 1)$, as

$$f^p_{\text{Rb}}(\boldsymbol{x}, i) = \begin{cases} f_{\text{Rb}}(\boldsymbol{x}, i) & \text{if } i = 0, \\ f_{\text{Rb}}(\boldsymbol{x}, i) + f_{\text{Rb}}(\boldsymbol{x}, i-1) & \text{if } 1 \le i \le \ell - 2, \\ f_{\text{Rb}}(\boldsymbol{x}, i-1) & \text{if } i = \ell - 1. \end{cases}$$

With $f^p_{\text{Rb}}(\boldsymbol{x}, i)$ now defining the contribution of the variable with index i to the objective value, the partial evaluation of a solution $\boldsymbol{x}$ is calculated in a way similar to that of the sphere function. Given a solution $\boldsymbol{x}$ with objective value $f_{\boldsymbol{x}}$, of which the variable with index i is modified, resulting in a solution $\boldsymbol{x}'$, the objective value of $\boldsymbol{x}'$ is calculated as

$$f_{\text{Rosenbrock}}(\boldsymbol{x}') = f_{\boldsymbol{x}} - f^p_{\text{Rb}}(\boldsymbol{x}, i) + f^p_{\text{Rb}}(\boldsymbol{x}', i).$$

The Rastrigin function is very similar to the sphere function, apart from the fact that the Rastrigin function has many local minima due to the addition of a cosine to the objective function. As the Rastrigin function is dimension-wise decomposable, partial evaluations are performed in a way similar to the sphere function. The Rastrigin function is defined as

$$f_{\text{Rastrigin}}(\boldsymbol{x}) = 10\ell + \sum_{i=0}^{\ell-1} \left[\boldsymbol{x}_i^2 - 10\cos(2\pi\boldsymbol{x}_i)\right].$$

The Michalewicz function also has many local optima. Additionally, when considering the dimensions of the Michalewicz function separately, dimensions with a higher index have more local optima. As the Michalewicz function is dimension-wise decomposable, partial evaluations are performed in a way similar to the sphere function. The Michalewicz function is defined as

$$f_{\text{Michalewicz}}(\boldsymbol{x}) = \sum_{i=0}^{\ell-1}\left[-\sin(\boldsymbol{x}_i)\cdot\sin\left((i+1)\boldsymbol{x}_i^2/\pi\right)^{20}\right].$$

The SoREB function, previously used by [4], is also used, as this is a decomposable problem with relatively difficult sub-problems. We define the SoREB function in terms of the ellipsoid function

$$f_{\text{Ellipsoid}}(\boldsymbol{x}) = \sum_{i=0}^{\ell-1}\left[10^{\frac{6i}{\ell-1}}\boldsymbol{x}_i^2\right],$$

and the rotation function $R_\theta(\boldsymbol{x})$. This function applies a counterclockwise rotation of an angle θ around the origin to a vector $\boldsymbol{x}$. When a nontrivial rotation is applied to the ellipsoid function, dependencies are introduced between each pair of variables, which makes the function difficult to optimize. The SoREB function is defined as

$$f_{\text{SoREB}}(\boldsymbol{x},\beta) = \sum_{i=0}^{\ell/\beta-1}\left[f_{\text{Ellipsoid}}\left(R_\theta\left(\left[\boldsymbol{x}_{\beta i},\ldots,\boldsymbol{x}_{\beta(i+1)-1}\right]\right)\right)\right],$$

where β is the block size. Partial evaluations of the SoREB function require the evaluation of a block of β variables if the value of at least one variable in the block is modified. We use the SoREB function as a benchmark problem, because the variables in each block are highly dependent, but variables in different blocks are not dependent. Specifically, we use block size $\beta = 5$ and rotation angle $\theta = 45°$.

Finally, we use a step-function variant of the sphere function, because the landscape of this function has many plateaus, a landscape characteristic that may well occur in real-world problems that should still be possible to overcome with population-based optimizers, but pose difficulty for gradient-based solvers. The step-function is defined as

$$f_{\text{step}}(\boldsymbol{x}) = \sum_{i=0}^{\ell-1}\lfloor\boldsymbol{x}_i\rfloor^2,$$

as the landscape of a step function can make it relatively difficult to optimize. Partial evaluations for this problem are performed in similar to the sphere function.

MULTI-OBJECTIVE

The set of multi-objective benchmark problems includes the convex generalized Multiple Euclidean Distances (genMED) problem [6], the well-known ZDT1 and ZDT3 problems [14], and a multi-objective variant of the SoREB function, named MOSoREB.

The genMED problem is a relatively simple problem with a convex Pareto front. The first objective of the genMED problem is the normalized Euclidean distance from $\boldsymbol{x}$ to the point $(1,0,0,\ldots)$, and the second objective is the normalized distance to the point $(0,1,0,\ldots)$. Both these objectives are completely dimension-wise decomposable, and straightforward to partially evaluate. The Pareto set consists of all $\boldsymbol{x}$ subject to $\boldsymbol{x}_0 + \boldsymbol{x}_1 = 1$, and $\boldsymbol{x}_i = 0$ for $i > 1$, leading to a convex Pareto front. The genMED function is defined as

$$\begin{aligned} f^0_{\text{genMED}}(\boldsymbol{x}) &= \frac{1}{2}\left((\boldsymbol{x}_0 - 1)^2 + \sum_{i=1}^{\ell-1} \boldsymbol{x}_i^2\right), \\ f^1_{\text{genMED}}(\boldsymbol{x}) &= \frac{1}{2}\left(\boldsymbol{x}_0^2 + (\boldsymbol{x}_1 - 1)^2 + \sum_{i=2}^{\ell-1} \boldsymbol{x}_i^2\right). \end{aligned}$$

We specifically selected the ZDT1 and ZDT3 problems for their distinct properties, as the Pareto fronts for these problems are convex and discontinuous, respectively. Both the ZDT1 and ZDT3 problems are non-separable [27]. The ZDT1 and ZDT3 problems are defined as follows:

$$\begin{aligned} g(\boldsymbol{x}) &= 1 + \frac{9}{\ell - 1}\sum_{i=1}^{\ell-1} \boldsymbol{x}_i, \\ f^0_{\text{ZDT1}}(\boldsymbol{x}) &= \boldsymbol{x}_0, \\ f^1_{\text{ZDT1}}(\boldsymbol{x}) &= g(\boldsymbol{x}) \cdot \left(1 - \sqrt{\frac{\boldsymbol{x}_0}{g(\boldsymbol{x})}}\right), \\ f^0_{\text{ZDT3}}(\boldsymbol{x}) &= \boldsymbol{x}_0, \\ f^1_{\text{ZDT3}}(\boldsymbol{x}) &= g(\boldsymbol{x}) \cdot \left(1 - \sqrt{\frac{\boldsymbol{x}_0}{g(\boldsymbol{x})}} - \frac{\boldsymbol{x}_0}{g(\boldsymbol{x})}\sin(10\pi\boldsymbol{x}_0)\right). \end{aligned}$$

All variables of these problems are constrained to the range $[0,1]$. Partial evaluations can be applied when, for each solution in the population, the sum of all its variables is kept track of at all times. This requires only $\mathcal{O}(n)$ memory, whereas the population itself requires $\mathcal{O}(\ell n)$ memory. Given $\sum_{i=0}^{\ell-1} \boldsymbol{x}_i$, any objective function of any ZDT problem can be partially evaluated in $\mathcal{O}(k)$ time after the modification of k variables.

Finally, the MOSoREB problem consists of the objective function f_0 for which each value is optimal, and an objective function f_1 that is similar to the SoREB function, but conflicts with f_0. The Pareto front of the MOSoREB problem is a straight line between $(0,1)$ and $(1,0)$. The MOSoREB problem is defined as follows:

$$\begin{aligned} f^0_{\text{MOSoREB}}(\boldsymbol{x}) &= \boldsymbol{x}_0, \\ f^1_{\text{MOSoREB}}(\boldsymbol{x}, \beta) &= 1 - \boldsymbol{x}_0 + f_{\text{SoREB}}([\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{\ell-1}], \beta). \end{aligned}$$

The variable $\boldsymbol{x}_0$ is constrained to the range $[0,1]$. All other variables are unconstrained.

2.9.2. EXPERIMENTAL SETUP

For all variants of GOMEA, we use the Interleaved Multi-start Scheme (IMS) [4] to avoid tuning the population size and the number of clusters. With this scheme, one run of an EA consists of several independent instances of the EA with population sizes exponentially growing in size. The generations of these independent instances are interleaved, such that the instance with population size $2n$ performs one generation for each c^{IMS} generations of the instance with population size n. For the multi-objective EAs, the number of clusters is increased by 1 for each new population that is started. The IMS starts with one instance of an EA with a relatively small population size n^{base} and q^{base} clusters (in MO optimization). Each consecutive instance that is started has a population size that is a factor 2 larger, and has one more cluster (in MO optimization). For all experiments $c^{\text{IMS}} = 8$, and $q^{\text{base}} = 5$ (for MO problems) are used. For the SoREB problem we use $n^{\text{base}} = 50$, and for the MOSoREB problem we use $n^{\text{base}} = 50q^{\text{base}}$, conform to the AMaLGaM population-size guideline for a 5-dimensional problem [8]. For all other Single-Objective (SO) experiments $n^{\text{base}} = 10$ is used, and for all other MO experiments $n^{\text{base}} = 10q^{\text{base}}$ is used.

Random restarts are used with L-BFGS, i.e., a run is restarted when convergence is observed while the computational budget has not yet been expended.

An initialization range of $[-115, -100]$ is used for all SO benchmark problems, as this range does not bracket the optimum. An initialization range of $[0, 1]$ is used for all MO benchmark problems, as some of these problems are constrained to this range. Boundary repair is used as a constraint handling mechanism [39].

For the sphere, Rosenbrock, Rastrigin, and SoREB problems we use a VTR of 10^{-10}. For the Michalewicz problem the VTR is set to 95% of the global optimum. In MO optimization, we use the $D_{\mathcal{P}_f \to S}$ metric [10], also known as the inverted generational distance, to evaluate the quality of an approximation set. The $D_{\mathcal{P}_f \to S}$ is the average distance in objective space computed over each point in an approximation set of 5000 equally spread out points on the Pareto front $\mathcal{P}_f$ with its associated nearest point in a given approximation set S. Using this metric requires that the Pareto front is known, but this is the case for all used MO benchmark problems. For all MO benchmark problems, we use a VTR of $D_{\mathcal{P}_f \to S} < 5 \cdot 10^{-3}$. An adaptive elitist archive is used with a target size of 1000.

For all problems except the SoREB and MOSoREB problems, the partial evaluation of k variables is counted as a fraction k/ℓ of an evaluation, because the computational effort for such a partial evaluation takes $\mathcal{O}(k/\ell)$. For the SoREB and MOSoREB problems, the computational effort of a partial evaluation depends on the number of blocks that have to be evaluated. Therefore, the partial evaluation of any number of variables in k different blocks of size β is counted as $k\beta/\ell \leq 1$ evaluations. Note that the partial evaluation of a set of variables consists of the calculation of their contribution to the objective value before modification, and the calculation of their contribution to the objective value after modification. However, if the objective function is computationally expensive, one can choose to save partial contributions to the objective function in memory, which has no impact on the overall memory complexity. For this reason, a partial evaluation is counted as a fraction k/ℓ of an evaluation instead of $2k/\ell$.

Experimental results for RV-GOMEA$^{\text{A}}$ were obtained by running experiments on an Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz. Experimental results for MO-RV-GOMEA$^{\text{A}}$, (MO-)RV-GOMEA$^{\text{C}}$, L-BFGS, and LM-CMA-ES were obtained by running experiments on an Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz. Reported times of RV-GOMEA$^{\text{A}}$ are scaled by a constant to compensate for the difference in CPU clock speed.

2.9.3. Comparison to Gradient-Based Optimization

If each subfunction has a dependency with up to k problem variables, the gradient can be estimated in up to k evaluations in a GBO setting. This makes gradient-based optimization methods, such as L-BFGS [28], very efficient in a GBO setting. However, though a GBO setting provides information on the separability of the optimization problem, each subfunction is considered as a black box. As any such subfunction can have a landscape that is arbitrarily difficult to optimize, gradient-based methods can still have difficulties optimizing such problems.

Figure 2.2 shows convergence plots of L-BFGS for the 40-dimensional Rastrigin problem, given different step sizes for the estimation of the gradient. The open source python library SciPy [25] was used for the implementation of L-BFGS. Figure 2.2 shows that L-BFGS is unable to reach the global optimum of the Rastrigin function, caused by the multimodality of this problem. In contrast, EAs are generally well-suited for such multi-modal problems. This is also the case for the variants of GOMEA, as shown in Figure 2.4.

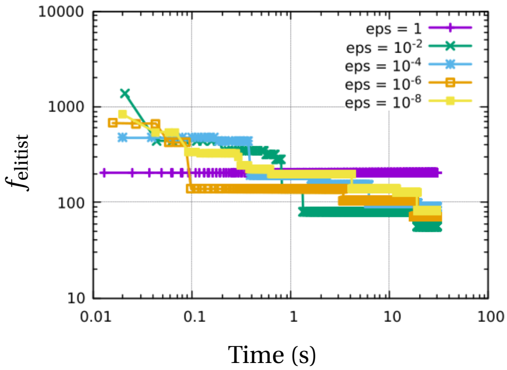


Figure 2.2: Convergence plot for the 40-dimensional Rastrigin problem using L-BFGS with different step size (eps) values for the estimation of the gradient. The vertical axis shows the best objective value found following random restarts of L-BFGS. A total time limit of 30 seconds was used.

L-BFGS can also be used for MO through repeated optimization of linear combinations of the objectives. This however makes it difficult to achieve good coverage of the Pareto front for problems such as ZDT3, as shown in Figure 2.3. In this case, weights of the initial two runs were set to 0.99 and 0.01, and weights of subsequent runs were set to the middle of the largest range of weights not yet covered, as to perform the optimization with a wide range of different weights. Additionally, also in MO, L-BFGS has trouble optimizing difficult problem landscapes, e.g., multi-modal problem landscapes.

2.9.4. Scalability Analysis

As computing power increases and the demand for the optimization of large-scale problems increases, it becomes increasingly important to analyze how well algorithms scale to high-dimensional optimization problems. For this reason, we perform a

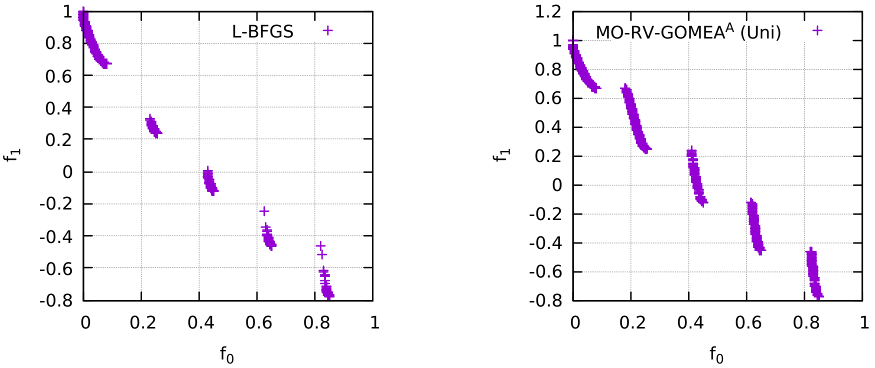


Figure 2.3: Resulting approximation fronts for L-BFGS and MO-RV-GOMEA^A for the 10-dimensional ZDT3 problem. A time limit of 60 seconds was used. The front found by L-BFGS was constructed through repeated optimization of linear combinations of the two objectives.

scalability analysis of all discussed algorithms in a GBO setting using different linkage models on a set of well-known single- and multi-objective benchmark problems.

We analyze the scalability of RV-GOMEA^A, RV-GOMEA^C, MO-RV-GOMEA^A, and MO-RV-GOMEA^C in a GBO setting by applying these algorithms to all benchmark problems introduced in Section 2.9.1, with exponentially increasing dimensionality. We compare these variants of GOMEA to the Limited Memory CMA-ES (LM-CMA-ES) [29], as it was shown that this EA scales the best to high dimensions compared to various large-scale variants of CMA-ES [52]. All default parameter settings are used, and the EA is restarted with double the population size when the convergence criterion has been reached. The comparison to LM-CMA-ES should give an indication of the benefits of a GBO setting compared to a BBO setting in terms of scalability, as LM-CMA-ES is also designed to perform well on large-scale optimization problems. We also compare to L-BFGS [28], as implemented in the SciPy [25] library. This implementation was modified to allow for the more efficient estimation of the gradient through partial evaluations. For the application of L-BFGS to MO benchmark problems, we repeatedly optimize weighted linear combinations of the objective functions.

All variants of GOMEA are tested with the Univariate factorization (Uni) and a BFLT linkage model (see Section 2.4.1) with an upper bound of 100 on the size of each linkage set. The LT model is not used, because including the linkage set of size ℓ should be avoided to achieve good scalability. A 5-Block FOS (5B) linkage model is also tested on the SoREB and MOSoREB problems. For each dimensionality, 30 independent runs of each EA are performed with a time limit of one hour. A run is considered successful when the EA finds the VTR within this time limit. As long as an EA is successful in all 30 runs for a problem of a certain dimensionality, we keep increasing the problem dimensionality by a factor 2 until the algorithm is no longer successful in all 30 runs, up to a dimensionality of 10^7 for the SO benchmark problems and 10^5 for the MO benchmark problems.

Results

The results of all SO scalability experiments are displayed in Figure 2.4, and the results for all MO scalability experiments are displayed in Figure 2.5.

RV-GOMEA$^{\mathrm{A}}$ and RV-GOMEA$^{\mathrm{C}}$ achieve very similar performance on most benchmark problems. Some differences in scalability can be observed for the Michalewicz and SoREB problems.

On the Michalewicz problem, the univariate RV-GOMEA$^{\mathrm{A}}$ scales relatively poorly up to approximately 10^3-dimensional problems. On higher-dimensional instances, RV-GOMEA$^{\mathrm{A}}$ however achieves scalability similar to the sphere problem. This is caused by the fact that a VTR equal to a fraction (95%) of the global optimum is used, which becomes easier to reach as the dimension of the Michalewicz problems increases, counteracting the fact that the optimization problem becomes more difficult as its dimensionality increases. A changing trend in the scalability of RV-GOMEA$^{\mathrm{C}}$ is not observed, because RV-GOMEA$^{\mathrm{C}}$ is able to consistently solve the Michalewicz problem with a relatively small population size, whereas RV-GOMEA$^{\mathrm{A}}$ requires a larger population size to efficiently solve this problem for medium-sized dimensionalities. This can also be observed through generally smaller variance in the results of RV-GOMEA$^{\mathrm{C}}$ compared to RV-GOMEA$^{\mathrm{A}}$.

On the SoREB problem, the scalability of RV-GOMEA$^{\mathrm{A}}$ and RV-GOMEA$^{\mathrm{C}}$ is largely the same when a 5B linkage model is used. RV-GOMEA$^{\mathrm{C}}$ however scales better than RV-GOMEA$^{\mathrm{A}}$ when a BFLT model is used, due to the fact that CMA-ES inherently scales better than AMaLGaM in case a full covariance matrix is used and the BFLT has linkage sets up to 100 variables.

LM-CMA-ES was unsuccessful at consistently solving the Rastrigin and Michalewicz problems for any dimensionality. Both of these problems are highly multi-modal. On all other problems, the scalability of LM-CMA-ES was substantially worse than that of all GOMEA variants, showing the benefits of a GBO setting for GOMEA.

L-BFGS unsurprisingly had the best performance on the sphere function, but failed to solve the Rastrigin, Michalewicz and step functions for any number of dimensions. On the Rosenbrock problem, L-BFGS was very efficient, but suffered from errors in numerical precision for high-dimensional problems. The performance of L-BFGS on the SoREB problem was very similar to RV-GOMEA variants with a 5B FOS in terms in time. Note that a Python implementation of L-BFGS is used, whereas the competing algorithms are implemented in C, affecting computation time results in an absolute sense, but not in terms of scalability. A more efficient implementation of L-BFGS may therefore be expected to achieve better computation times than RV-GOMEA on the SoREB, genMED, and possibly ZDT1 problems.

On the set of MO benchmark problems, MO-RV-GOMEA$^{\mathrm{C}}$ achieves the best performance on the genMED, ZDT1 and ZDT3 problems. MO-RV-GOMEA$^{\mathrm{C}}$ and MO-RV-GOMEA$^{\mathrm{A}}$ generally achieve the same scalability when the same linkage model is used. MO-RV-GOMEA$^{\mathrm{A}}$ achieves slightly better performance than MO-RV-GOMEA$^{\mathrm{C}}$ on the MOSoREB problem, but both achieve the same scalability. Though the ZDT1 and ZDT3 problems are non-separable [27], the performance of different linkage models is very similar for these problems.

	Sphere	Rosenbrock	Rastrigin	Michalewicz	SoREB	Step
Uni	A^{***}		C^{***}	A^{***}	C^{***}	C
BFLT	A^{***}		C^{***}	C^{***}	C^{***}	C^{***}
B5					A^{***}	

Table 2.1: Statistical comparison of RV-GOMEA$^{\mathrm{A}}$ and RV-GOMEA$^{\mathrm{C}}$ for all single-objective benchmark problems, for the highest dimensionality achieved by both algorithms in Figure 2.4. A cell marked by a letter (A or C) indicates that this variant of RV-GOMEA performed statistically significantly better (in terms of function evaluations) than the other variant, for the given linkage model and benchmark problem. The number of asterisks indicates the level of significance, with 0 to 3 asterisks indicating p-values smaller than 10^{-2}, 10^{-3}, 10^{-4}, and 10^{-5}, respectively.

	genMED	ZDT1	ZDT3	MOSoREB
Uni	A^{***}	C^{***}	C^{*}	C^{***}
BFLT	A^{***}	C^{***}		C^{***}
B5				C^{***}

Table 2.2: Statistical comparison of RV-GOMEA$^{\mathrm{A}}$ and RV-GOMEA$^{\mathrm{C}}$ for all multi-objective benchmark problems, for the highest dimensionality achieved by both algorithms in Figure 2.5. A cell marked by a letter (A or C) indicates that this variant of RV-GOMEA performed statistically significantly better (in terms of function evaluations) than the other variant, for the given linkage model and benchmark problem. The number of asterisks indicates the level of significance, with 0 to 3 asterisks indicating p-values smaller than 10^{-2}, 10^{-3}, 10^{-4}, and 10^{-5}, respectively.

L-BFGS performs very well on the easy genMED problem, but is outperformed by GOMEA on the ZDT1 and ZDT3 problems. L-BFGS was unable to reach the VTR on the ZDT3 and MOSoREB problems for any tested problem dimensionality. These results show that EAs are generally better at optimizing MO problems than repeated SO approaches.

In Tables 2.1 and 2.2 we show a statistical comparison of results shown in Figures 2.4 and 2.5. In particular, we compare the RV-GOMEA$^{\mathrm{A}}$ and RV-GOMEA$^{\mathrm{C}}$ variants of RV-GOMEA given identical linkage models. We restrict the analysis to these comparisons, because comparisons to L-BFGS and LM-CMA-ES are generally directly clear from Figures 2.4 and 2.5. For each linkage model and each benchmark problem, a t-test is performed to compare the number of evaluations of the two variants of RV-GOMEA for the highest dimensionality reached by both algorithms. Though the absolute differences in the number of evaluations are generally small between the two variants, as shown in Figures 2.4 and 2.5, RV-GOMEA$^{\mathrm{C}}$ is statistically significantly better ($p < 0.01$) than RV-GOMEA$^{\mathrm{A}}$ in 13 of 22 cases. In 6 of 22 cases RV-GOMEA$^{\mathrm{A}}$ is statistically significantly better ($p < 0.01$) than RV-GOMEA$^{\mathrm{C}}$.

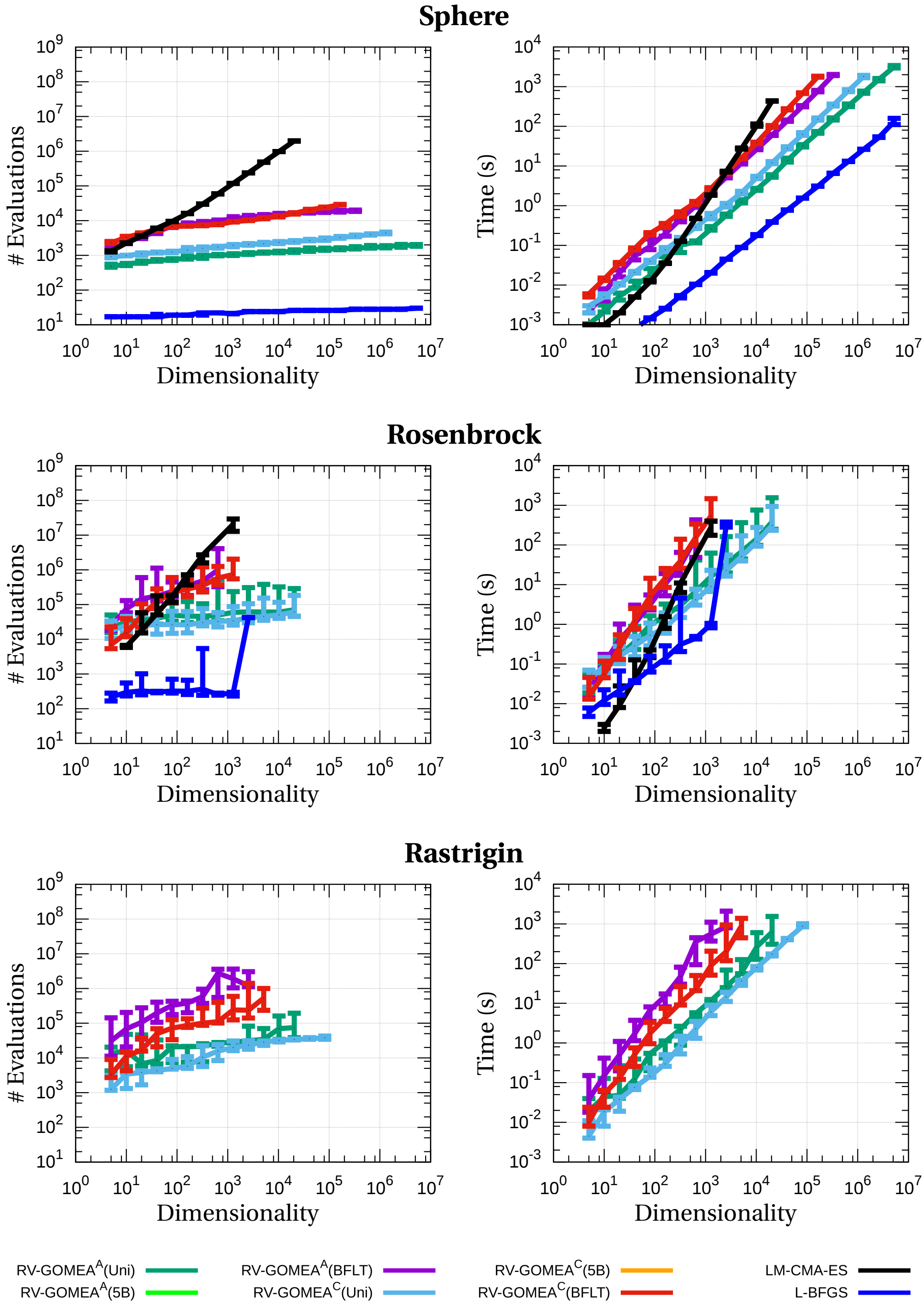


Figure 2.4: Medians and interdecile ranges of single-objective GBO scalability experiments with each data point the median of 30 successful runs, and error bars showing the interdecile range.

2.10. Discussion

First and foremost, we have shown that the exploitation of partial evaluations in a GBO setting can substantially improve the performance and scalability of EAs compared to a BBO setting. For this reason, exploiting what a GBO setting has to offer should be strongly considered if possible when optimizing a real-world problem, as substantial gains in performance can be achieved if the problem supports partial evaluations. Such partial evaluations are certainly not restricted to trivial problems, as they have previously been applied to real-world problems that are multi-objective, non-separable, and multi-modal [2, 32]. For such difficult problems, gradient-based optimizers are likely to converge to a local optimum. In contrast, EAs are considered state-of-the-art for such optimization problems.

In particular, we have introduced two variants of the real-valued version of GOMEA, based on previous work [4, 5], which is designed to exploit partial evaluations in a GBO setting. To the best of our knowledge, RV-GOMEA is the only real-valued EA capable of exploiting partial evaluations in a GBO setting. Different real-valued EAs are not directly suitable to get the most out of a GBO setting. For example, variants of CMA-ES can use a restricted covariance matrix to account for available problem information. Using the optimal restriction of the covariance matrix however still leads to worse performance than RV-GOMEA in a GBO setting [4].

Based on the results of comparisons between (MO-)RV-GOMEA$^{\text{A}}$ and (MO-)RV-GOMEA$^{\text{C}}$, we conclude that both algorithms are valid and attractive alternatives to employ for optimization in a GBO setting. Our results show that the core principles of GOMEA have been successfully applied to CMA-ES, as we were able to achieve results similar, and in some cases superior, to the results of (MO-)RV-GOMEA$^{\text{A}}$ that were previously shown to be excellent compared to the state-of-the-art [4, 5].

Future work includes research into automatically determining the best linkage structure for GBO problems. Though this may be trivial for problems such as SoREB, it may be difficult for others. For instance, we found that a univariate linkage structure achieved the best performance on the non-separable Rosenbrock problem. It is further not immediately clear what the best linkage structure would be, e.g., when all pairs of variables have some degree of dependence, but some relations are clearly stronger than others. Various techniques could be considered to learn problem structure, such as techniques used in cooperative coevolution [45]. Such techniques analyze interactions between problem variables to decompose the optimization problem, for example using adaptive weighting [56], delta grouping [38], or differential grouping [37].

Further future work consists of trying to improve the performance of RV-GOMEA in a BBO setting. This could require fundamental changes to GOM, because GOM requires evaluations to be done after only a small number of variables have been modified. For this reason, GOM is very efficient in a GBO setting, but not as efficient in a BBO setting. Potentially, combinations with surrogate models that are efficient to evaluate could have an impact here.

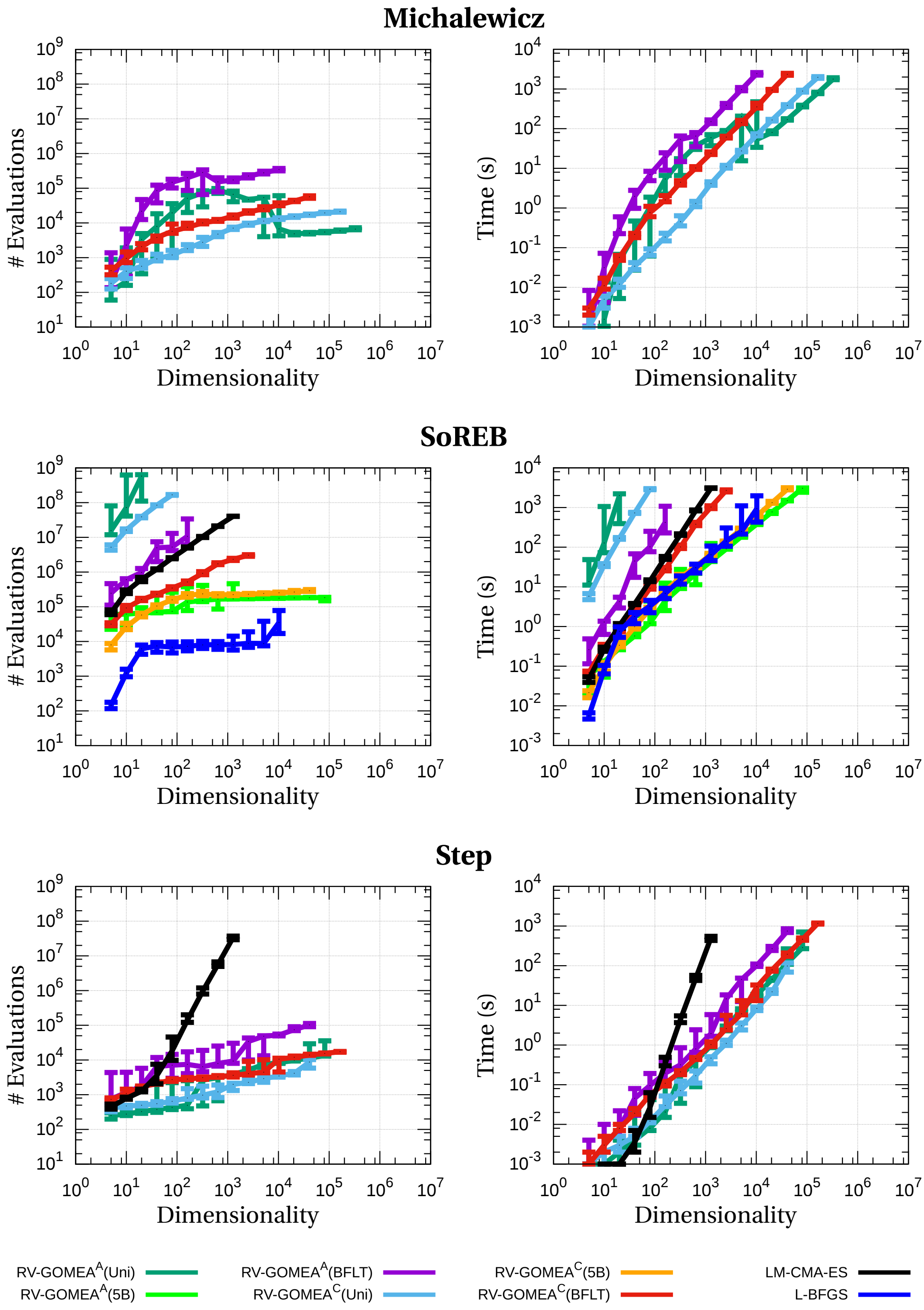


Figure 2.4: Medians and interdecile ranges of single-objective GBO scalability experiments with each data point the median of 30 successful runs, and error bars showing the interdecile range.

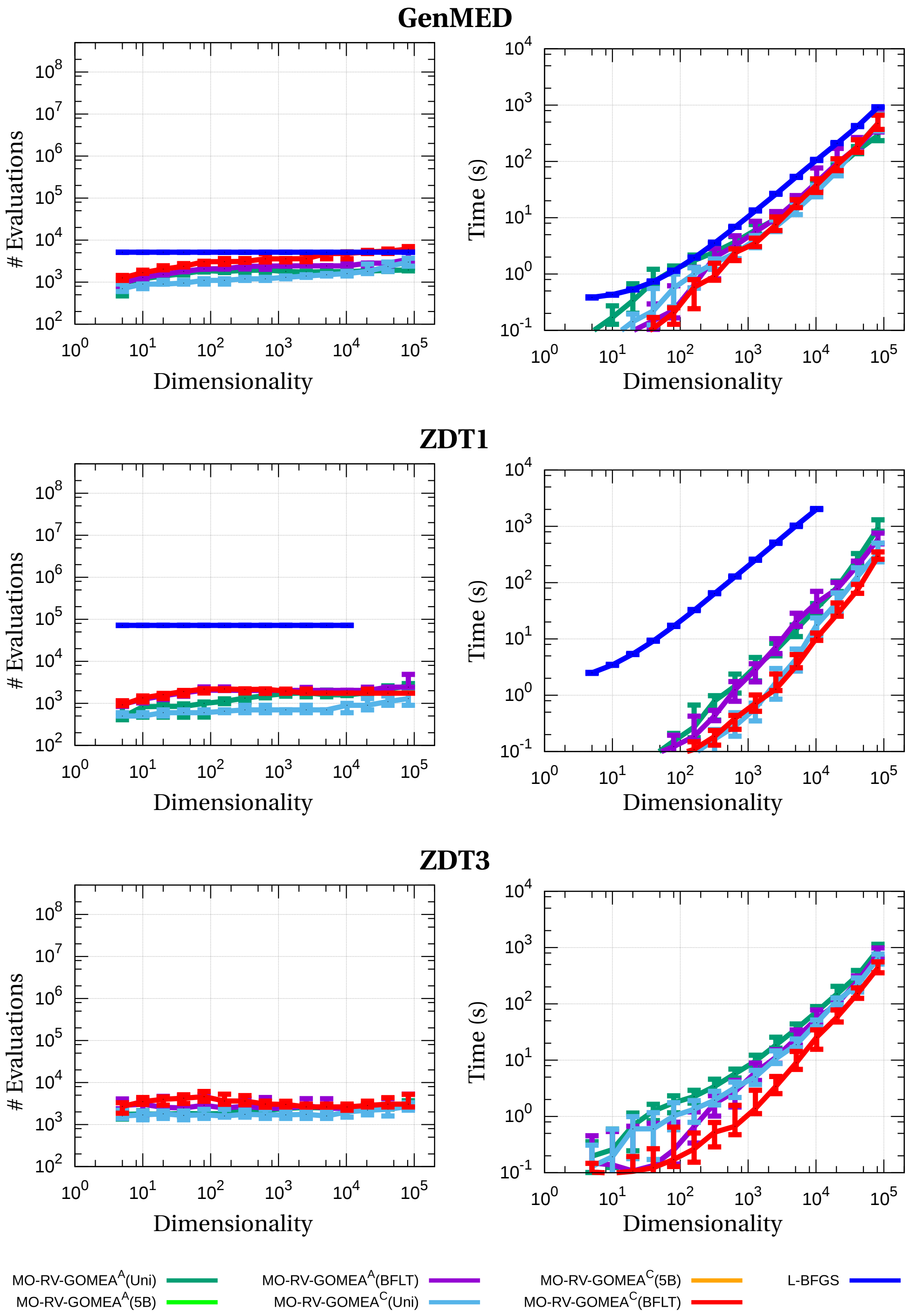


Figure 2.5: Medians and interdecile ranges of multi-objective GBO scalability experiments with each data point the median of 30 successful runs, and error bars showing the interdecile range.

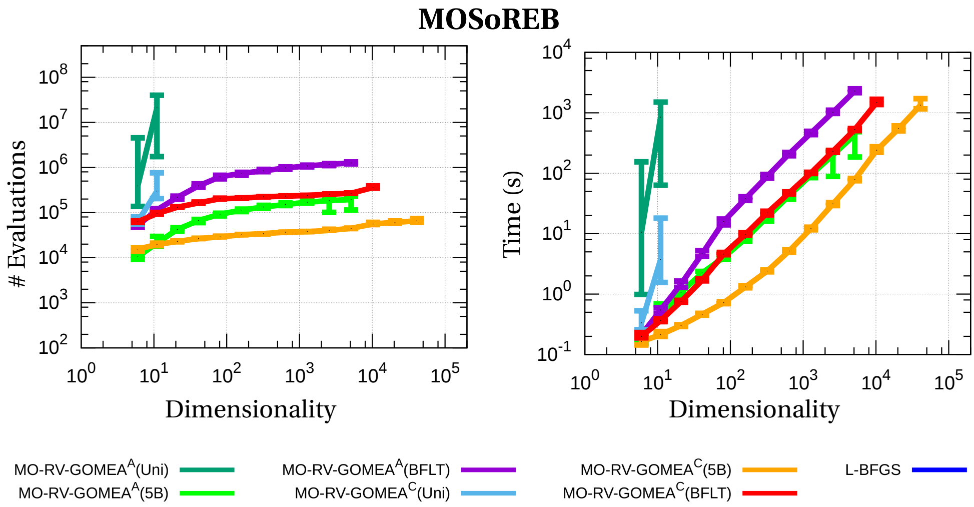


Figure 2.5: Medians and interdecile ranges of multi-objective GBO scalability experiments with each data point the median of 30 successful runs, and error bars showing the interdecile range.

2.11. CONCLUSIONS

Effectively exploiting linkage information can lead to substantial improvements in the performance of EAs, including for problems with real-valued variables. In this chapter, we showed that exploiting such information in a GBO setting, specifically one that allows for partial evaluations, can lead to substantial improvements in performance and scalability.

We described combinations of the state-of-the-art real-valued EAs AMaLGaM and CMA-ES [21], with GOMEA, a state-of-the-art EA framework designed around effectively mixing partial solutions. The new GOMEA variants, referred to as RV-GOMEA, were also extended to be suitable for multi-objective optimization. Scalability analyses were performed on a set of well-known benchmark problems.

The performance of RV-GOMEA was compared to that of LM-CMA-ES, a state-of-the-art EA for large-scale BBO, and L-BFGS, a well-known gradient-based optimization method. This showed that using RV-GOMEA in a GBO setting leads to substantially better performance than the state-of-the-art EAs in BBO and the ability to scalably solve a much richer class of optimization problems than L-BFGS, while achieving similar, or for MO, even better scalability in a GBO setting. Therefore, RV-GOMEA can be considered a valuable addition to the field of EAs, in particular for the purpose of solving complex (large-scale) optimization problems when facing a GBO scenario that permits efficient partial evaluations.

REFERENCES

[1] Bouter, A., Alderliesten, T., Bel, A., Witteveen, C., and Bosman, P. A. N. (2018). Large-scale parallelization of partial evaluations in evolutionary algorithms for real-world

problems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1199–1206. ACM.

[2] Bouter, A., Alderliesten, T., and Bosman, P. A. N. (2017a). A novel model-based evolutionary algorithm for multi-objective deformable image registration with content mismatch and large deformations: benchmarking efficiency and quality. In Styner, M. A. and Angelini, E. D., editors, *Medical Imaging 2017: Image Processing*, volume 10133, pages 304 – 311. International Society for Optics and Photonics, SPIE.

[3] Bouter, A., Alderliesten, T., and Bosman, P. A. N. (2021). Achieving highly scalable evolutionary real-valued optimization by exploiting partial evaluations. *Evolutionary Computation*, 29(1):129–155.

[4] Bouter, A., Alderliesten, T., Witteveen, C., and Bosman, P. A. N. (2017b). Exploiting linkage information in real-valued optimization with the real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 705–712. ACM.

[5] Bouter, A., Luong, N. H., Witteveen, C., Alderliesten, T., and Bosman, P. A. N. (2017c). The multi-objective real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 537–544. ACM.

[6] Bosman, P. A. N. (2012). On gradients and hybrid evolutionary algorithms for real-valued multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, 16(1):51–69.

[7] Bosman, P. A. N. and Alderliesten, T. (2012). Incremental Gaussian model-building in multi-objective EDAs with an application to deformable image registration. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 241–248. ACM.

[8] Bosman, P. A. N., Grahl, J., and Thierens, D. (2013). Benchmarking parameter-free AMaLGaM on functions with and without noise. *Evolutionary Computation*, 21(3):445–469.

[9] Bosman, P. A. N., Luong, N. H., and Thierens, D. (2016). Expanding from discrete Cartesian to permutation gene-pool optimal mixing evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 637–644. ACM.

[10] Bosman, P. A. N. and Thierens, D. (2003). The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):174–188.

[11] Bosman, P. A. N. and Thierens, D. (2013). More concise and robust linkage learning by filtering and combining linkage hierarchies. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 359–366. ACM.

[12] Chen, W., Whitley, D., Tinós, R., and Chicano, F. (2018). Tunneling between plateaus: Improving on a state-of-the-art MAXSAT solver using partition crossover. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 921–928. ACM.

[13] De Jong, K. A. (1975). An analysis of the behavior of a class of genetic adaptive systems [Ph.D. thesis]. *University of Michigan.*

[14] Deb, K. (1999). Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3):205–230.

[15] Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley & Sons, Inc.

[16] Deb, K. and Myburgh, C. (2016). Breaking the billion-variable barrier in real-world optimization using a customized evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 653–660. ACM.

[17] Etxeberria, R. and Larrañaga, P. (1999). Global optimization using Bayesian networks. In *Second Symposium on Artificial Intelligence (CIMAF-99)*, pages 332–339.

[18] Goldman, B. W. and Punch, W. F. (2014). Parameter-less population pyramid. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 785–792. ACM.

[19] Gronau, I. and Moran, S. (2007). Optimal implementations of UPGMA and other common clustering algorithms. *Information Processing Letters*, 104(6):205–210.

[20] Hansen, N. and Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 312–317. IEEE.

[21] Hansen, N. and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195.

[22] Harik, G. R., Lobo, F. G., and Sastry, K. (2006). Linkage learning via probabilistic modeling in the extended compact genetic algorithm (ECGA). In Pelikan, M., Sastry, K., and Cantú-Paz, E., editors, *Scalable optimization via probabilistic modeling*, volume 33 of *Studies in Computational Intelligence*, pages 39–61. Springer.

[23] Holland, J. (1975). Adaptation in natural and artificial systems. *Ann Arbor: University of Michigan Press.*

[24] Hsu, S.-H. and Yu, T.-L. (2015). Optimization by pairwise linkage detection, incremental linkage set, and restricted/back mixing: DSMGA-II. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 519–526. ACM.

[25] Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python. [Online].

[26] Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. *Physical Review E*, 69(6):066138.

[27] Li, K., Omidvar, M. N., Deb, K., and Yao, X. (2016). Variable interaction in multi-objective optimization problems. In *International Conference on Parallel Problem Solving from Nature*, pages 399–409. Springer.

[28] Liu, D. C. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528.

[29] Loshchilov, I. (2014). A computationally efficient limited memory CMA-ES for large scale optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 397–404. ACM.

[30] Lozano, J. A., Larrañaga, P., Inza, I., and Bengoetxea, E. (2006). *Towards a new evolutionary computation: Advances on estimation of distribution algorithms*, volume 192 of *Studies in Fuzziness and Soft Computing*. Springer.

[31] Luong, H. N. and Bosman, P. A. N. (2012). Elitist archiving for multi-objective evolutionary algorithms: To adapt or not to adapt. In *International Conference on Parallel Problem Solving from Nature*, pages 72–81. Springer.

[32] Luong, N. H., Alderliesten, T., Bel, A., Niatsetski, Y., and Bosman, P. A. N. (2018a). Application and benchmarking of multi-objective evolutionary algorithms on high-dose-rate brachytherapy planning for prostate cancer treatment. *Swarm and Evolutionary Computation*, 40:37–52.

[33] Luong, N. H., La Poutré, H., and Bosman, P. A. N. (2014). Multi-objective gene-pool optimal mixing evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 357–364. ACM.

[34] Luong, N. H., La Poutré, H., and Bosman, P. A. N. (2018b). Multi-objective gene-pool optimal mixing evolutionary algorithm with the interleaved multi-start scheme. *Swarm and Evolutionary Computation*, 40:238–254.

[35] Michalewicz, Z. and Janikow, C. Z. (1991). Genetic algorithms for numerical optimization. *Statistics and Computing*, 1(2):75–91.

[36] Mühlenbein, H. (1997). The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5(3):303–346.

[37] Omidvar, M. N., Li, X., Mei, Y., and Yao, X. (2014). Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Transactions on Evolutionary Computation*, 18(3):378–393.

[38] Omidvar, M. N., Li, X., and Yao, X. (2010). Cooperative co-evolution with delta grouping for large scale non-separable function optimization. In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE.

[39] Orvosh, D. and Davis, L. (1994). Using a genetic algorithm to optimize problems with feasibility constraints. In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, pages 548–553. IEEE.

[40] Pelikan, M. (2005a). Bayesian optimization algorithm. In *Hierarchical Bayesian optimization algorithm*, volume 170 of *Studies in Fuzziness and Soft Computing*, pages 31–48. Springer.

[41] Pelikan, M. (2005b). Hierarchical Bayesian optimization algorithm. In *Hierarchical Bayesian optimization algorithm*, volume 170 of *Studies in Fuzziness and Soft Computing*, pages 105–129. Springer.

[42] Pelikan, M. and Mühlenbein, H. (1999). The bivariate marginal distribution algorithm. In Roy, R., Furuhashi, T., and Chawdhry, P. K., editors, *Advances in Soft Computing*, pages 521–535. Springer.

[43] Pelikan, M., Sastry, K., and Cantú-Paz, E. (2007). *Scalable optimization via probabilistic modeling: From algorithms to applications*, volume 33 of *Studies in Computational Intelligence*. Springer.

[44] Pelikan, M., Sastry, K., and Goldberg, D. E. (2005). Multiobjective hBOA, clustering, and scalability. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 663–670. ACM.

[45] Potter, M. A. and Jong, K. A. D. (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29.

[46] Rastrigin, L. A. (1974). Systems of extremal control. *Nauka*.

[47] Ros, R. and Hansen, N. (2008). A simple modification in CMA-ES achieving linear time and space complexity. In *International Conference on Parallel Problem Solving from Nature*, pages 296–305. Springer.

[48] Rosenbrock, H. (1960). An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184.

[49] Thierens, D. (2010). The linkage tree genetic algorithm. In *International Conference on Parallel Problem Solving from Nature*, pages 264–273. Springer.

[50] Thierens, D. and Bosman, P. A. N. (2011). Optimal mixing evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 617–624. ACM.

[51] Tinós, R., Whitley, D., and Chicano, F. (2015). Partition crossover for pseudo-boolean optimization. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*, pages 137–149. ACM.

[52] Varelas, K., Auger, A., Brockhoff, D., Hansen, N., ElHara, O. A., Semet, Y., Kassab, R., and Barbaresco, F. (2018). A comparative study of large-scale variants of CMA-ES. In *International Conference on Parallel Problem Solving from Nature*, pages 3–15. Springer.

[53] Virgolin, M., Alderliesten, T., Witteveen, C., and Bosman, P. A. N. (2017). Scalable genetic programming by gene-pool optimal mixing and input-space entropy-based building-block learning. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1041–1048. ACM.

[54] Whitley, D., Hains, D., and Howe, A. (2009). Tunneling between optima: Partition crossover for the traveling salesman problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 915–922. ACM.

[55] Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., Peters, J., and Schmidhuber, J. (2014). Natural evolution strategies. *Journal of Machine Learning Research*, 15(1):949–980.

[56] Yang, Z., Tang, K., and Yao, X. (2008). Large scale evolutionary optimization using cooperative coevolution. *Information Sciences*, 178(15):2985–2999.

3

Leveraging Conditional Linkage Models

This chapter is based on the following publication: Bouter, A., Maree, S. C., Alderliesten, T., and Bosman, P. A. N. (2020). Leveraging conditional linkage models in gray-box optimization with the real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 603–611

ABSTRACT

Often, real-world problems are of the gray-box type. It has been shown that the Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm (RV-GOMEA) is in principle capable of exploiting such a Gray-Box Optimization (GBO) setting using linkage models that capture dependencies between problem variables, resulting in excellent performance and scalability on both benchmark and real-world problems that allow for partial evaluations. However, linkage models proposed for RV-GOMEA so far cannot explicitly capture overlapping dependencies. Consequently, performance degrades if such dependencies exist. In this chapter, we therefore introduce various ways of using conditional linkage models in RV-GOMEA. Their use is compared to that of non-conditional models, and to VkD-CMA, whose performance is among the state of the art, on various benchmark problems with overlapping dependencies. We find that RV-GOMEA with conditional linkage models achieves the best scalability on most problems, with conditional models leading to similar or better performance than non-conditional models. We conclude that the introduction of conditional linkage models to RV-GOMEA is an important contribution, as it expands the set of problems for which optimization in a GBO setting results in substantially improved performance and scalability. In future work, conditional linkage models may prove to benefit the optimization of real-world problems.

3.1. INTRODUCTION

Traditionally, research in the field of Evolutionary Algorithms (EAs) has mainly been focused on the optimization of Black-Box Optimization (BBO) problems. In a BBO setting, no prior information of the objective function is assumed to be available, and only function evaluations can reveal the structure of the problem to be optimized. However, in various real-world problems [15, 25] the underlying structure, or even an exact definition, of the optimization problem is known. Despite this knowledge, such problems may still be difficult to optimize. As such, problem-specific knowledge has recently been used in EAs to greatly improve performance, e.g., through the use of problem-specific variation operators [15], partition crossover [14, 38], or partial evaluations [6].

Partial evaluations are used to efficiently calculate the objective value of a solution following the modification of a subset of its variables. The possibility to use partial evaluations is by no means an indicator of the difficulty of the problem, as partial evaluations may be possible for problems that are, e.g., non-separable, non-smooth, multi-modal, and/or multi-objective. It has been shown that partial evaluations can be efficiently leveraged in the Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm (RV-GOMEA) [6]. RV-GOMEA achieves excellent performance and scalability on single-objective and multi-objective benchmark problems [6, 7], but also on real-world problems such as medical deformable image registration [5], and treatment planning for prostate cancer [25]. However, the problems considered thus far are either separable or have relatively weak dependencies.

It is as of yet unclear if it is possible to efficiently leverage partial evaluations in case the optimization problem has strong overlapping dependencies, because partial evaluations are only beneficial when some form of selection is applied after the modification of a subset of variables of a solution. Applying variation not to all variables at once, but only to subsets of variables, may however be detrimental to the optimization of problems with strong overlapping dependencies. Such problems cannot be decomposed into disjoint sets of independent variables, and are therefore generally difficult to solve for optimization methods relying on decompositions if the dependencies themselves are strong, i.e., a rotated ellipsoid with high condition number [26, 36]. They can however be efficiently optimized by other decomposition methods like VkD-CMA [2, 3], a variant of the well-known Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [20], using a covariance matrix that is parameterized by a smaller number of parameters than the problem dimensionality. VkD-CMA can however not directly benefit from the possibility of partial evaluations, as variation is always applied to all problem variables at once.

In this chapter, we aim to improve the performance of RV-GOMEA on problems with strong overlapping dependencies in a GBO setting. For this purpose, we explore the use of different novel linkage models, which are used by RV-GOMEA to model dependencies between problem variables. In particular, we explore linkage models that can capture conditional dependencies, as these models may benefit the optimization of problems with overlapping dependencies.

We first introduce the GBO setting in more detail in Section 3.2, followed by background on RV-GOMEA in Section 3.3. In Section 3.4 we discuss a number of conditional linkage models, and how they can be employed in RV-GOMEA. In Section 3.5 the performance and scalability of these conditional linkage models is compared to non-conditional models, and to VkD-CMA. The results are discussed in Section 3.6, followed by our conclusions in Section 3.7.

3.2. GRAY-BOX OPTIMIZATION

We consider the optimization of an objective function $f(\boldsymbol{x}) : \mathbb{R}^\ell \rightarrow \mathbb{R}$, subject to minimization. We refer to problem variables as $\boldsymbol{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_\ell\}$, which is indexed through the set $\mathcal{I} = [1, \ldots, \ell]$. A realization of the problem variables, i.e., a solution, is denoted $\boldsymbol{x} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\ell\}$.

In particular, we focus on a GBO setting (as previously defined [4]) that allows for partial evaluations, meaning that the objective value of a solution can be efficiently computed after the modification of a subset of its variables. Let $Y \subseteq \mathcal{I}$ refer to a subset of problem variables. We denote by $\boldsymbol{x}_Y$ the variables of $\boldsymbol{x}$ corresponding to the indices specified by Y. In this setting, the objective function is composed of q subfunctions $\boldsymbol{F} = \{f_1, f_2, \ldots, f_q\}$. A subfunction $f_j(\boldsymbol{x}_{\mathcal{I}_j}) \in \boldsymbol{F}$ takes problem variables $\boldsymbol{x}_i$ with $i \in \mathcal{I}_j \subseteq \mathcal{I}$ as input, where $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_q\}$ is given by the problem definition. Each subfunction itself is treated as a black box. Moreover, a subfunction f_i is assumed not to be separable itself.

Problem variables $\boldsymbol{x}_u$ and $\boldsymbol{x}_v$ with $u, v \in \mathcal{I}$ are considered to be directly dependent, i.e., $\boldsymbol{x}_u \leftrightarrow \boldsymbol{x}_v$, when a subfunction $f_j(\boldsymbol{x}_{\mathcal{I}_j}) \in F$ exists with $\{u, v\} \subseteq \mathcal{I}_j$. Problem variables $\boldsymbol{x}_u$ and $\boldsymbol{x}_v$ are indirectly dependent when a set $\{u, \ldots, v\} \subseteq \mathcal{I}$ exists such that $\boldsymbol{x}_u \leftrightarrow \cdots \leftrightarrow \boldsymbol{x}_v$ but not $\boldsymbol{x}_u \leftrightarrow \boldsymbol{x}_v$.

An objective function in a GBO setting can be written as:

$$f(\boldsymbol{x}) = g\left(f_1(\boldsymbol{x}_{\mathcal{I}_1}) \oplus f_2(\boldsymbol{x}_{\mathcal{I}_2}) \oplus \cdots \oplus f_q(\boldsymbol{x}_{\mathcal{I}_q})\right), \tag{3.1}$$

with $\oplus$ a binary operator that has a known inverse $\ominus$ (e.g., addition or multiplication[4]), and $g : \mathbb{R} \rightarrow \mathbb{R}$ any (potentially non-linear) function.

A change in some variable $\boldsymbol{x}_u$ requires the evaluation of each subfunction $f_j(\boldsymbol{x}_{\mathcal{I}_j})$ for which $u \in \mathcal{I}_j$. If g is not the identity function, the intermediate result of $f_1(\boldsymbol{x}_{\mathcal{I}_1}) \oplus \cdots \oplus f_q(\boldsymbol{x}_{\mathcal{I}_q})$ must be stored in memory for each solution in the population.

We assume that the computational complexity of each subfunction is approximately equal. Therefore, given a total of q subfunctions, a partial evaluation of n subfunctions is counted as a fraction n/q of an evaluation in a GBO setting.

Given the GBO definition of an optimization problem, as in Equation (6.1), a Variable Interaction Graph (VIG) [38] can be created. A VIG is an undirected graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ where each vertex $v \in V$ corresponds to a problem variable $\boldsymbol{x}_v$, and each edge $(u, v) \in E$ denotes that problem variables $\boldsymbol{x}_u$ and $\boldsymbol{x}_v$ are directly dependent. Problem variables $\boldsymbol{x}_u$ and $\boldsymbol{x}_v$ are indirectly dependent if a path between vertices u and v exists in the VIG, but not an edge (u, v).

3.3. Real-Valued Gene-Pool Optimal Mixing Evolutionary Algorithm

In this section we present an outline of RV-GOMEA. A more detailed description is provided in [6].

3.3.1. Linkage Model

In RV-GOMEA, dependencies between problem variables are explicitly modeled by a linkage model. Such linkage models are described by a Family of Subsets (FOS) $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_m\}$ that is a subset of the powerset of $\mathcal{I}$, i.e., $\mathcal{F}_i \subseteq \mathcal{I}$. Each FOS element $\mathcal{F}_i \in \mathcal{F}$ models a group of variables that is considered to be jointly dependent.

The univariate FOS $\mathcal{F} = \{\{1\}, \{2\}, \dots, \{\ell\}\}$, which models all problem variables to be independent, is considered to be the simplest FOS model. A marginal product FOS models disjoint sets of dependent variables, i.e., $\mathcal{F}_i \cap \mathcal{F}_j = \emptyset$ for $\mathcal{F}_i, \mathcal{F}_j \in \mathcal{F}$. The full FOS models all variables as jointly dependent, i.e., $\mathcal{F} = \{\{1, \dots, \ell\}\}$.

A linkage tree is a linkage model, which was first introduced for the discrete GOMEA [37], that is capable of modeling hierarchical dependencies. A linkage tree FOS firstly includes all singleton elements. All other FOS elements are the union of two smaller FOS elements. Formally stated, for each $\mathcal{F}_i \in \mathcal{F}$ there exist $\mathcal{F}_j, \mathcal{F}_k \in \mathcal{F}$ such that $\mathcal{F}_j \cap \mathcal{F}_k = \emptyset$ and $\mathcal{F}_j \cup \mathcal{F}_k = \mathcal{F}_i$. Building a linkage tree is often done using the Unweighted Pair Grouping Method with Arithmetic-mean (UPGMA) [19]. Therefore, pairs of FOS elements with the strongest dependencies are the first to be merged. The dependency strength of variables can either be determined based on problem-specific knowledge or based on mutual information.

The elements of a linkage tree may be restricted to a maximum size k. In this case, the linkage tree is generally learned offline based on problem-specific information, in which case we refer to it as the Bounded Fixed Linkage Tree (BFLT) [6].

3.3.2. Gene-Pool Optimal Mixing

RV-GOMEA uses Gene-pool Optimal Mixing (GOM) as its variation operator. During GOM, variation is applied based on the dependencies defined by the linkage model $\mathcal{F}$. During each generation, one iteration of GOM is applied to each solution in the population for each FOS element $\mathcal{F}_i$. One iteration of GOM, applied to a parent solution in the population, consists of the sampling of new values for the problem variables contained in the FOS element in question. These values are inserted into the parent and this solution is evaluated. Because few variables change, as specified by $\mathcal{F}_i$, a partial evaluation can be performed. The modification of the parent is only accepted if it results in an improvement. Otherwise, the modification is accepted with probability p^{accept}. If the modification is rejected, the parent is returned to its previous state.

A sampling model is used to generate new problem variables. This can be taken from any EA that has an identifiable sampling model, such as CMA-ES [20]. Here, we build on the original version of RV-GOMEA that uses the sampling model of AMaLGaM [10].

Sampled values for problem variables are generated from a Multivariate Gaussian (MVG) that is estimated with maximum likelihood based on the selection $\mathcal{S}$, consisting of the $\lfloor \tau n \rfloor$ best solutions in the population $\mathcal{P}$, with n the population size. For each FOS element $\mathcal{F}_j$, an MVG $\mathcal{N}(\hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{C}}_j)$ is estimated for all problem variables in $\mathcal{F}_j$. Each such distribution is scaled by a distribution multiplier that is updated each generation based on the location of improvements found. This process is called Adaptive Variance Scaling (AVS).

Furthermore, RV-GOMEA includes a generational shift that is applied to n^{AMS} solutions in the population, named the Anticipated Mean Shift (AMS), and a procedure aimed at avoiding stagnation caused by spreading the population across multiple local minima, named the Forced Improvement (FI) procedure. The FI procedure is triggered when a solution has not been improved for a number of generations larger than the maximum No-Improvement Stretch (NIS). These procedures are described in more detail in [6]. High-level pseudo-code of RV-GOMEA is shown in Algorithm 3.5. The GOM procedure is shown in Algorithm 3.6.

Algorithm 3.5 RV-GOMEA

```
procedure RV-GOMEA($n$,$\tau$)
    $\mathcal{P} \leftarrow$ InitializeAndEvaluatePopulation($n$)
    $\mathcal{F} \leftarrow$ InitializeLinkageModel()
    while $\neg$TerminationCriterionSatisfied() do
        $\mathcal{P}_1 \leftarrow \boldsymbol{x}^{\texttt{elitist}}$
        for $\mathcal{F}_j \in \mathcal{F}$ do                                    ▷ Random order
            $\mathcal{S} \leftarrow \lfloor \tau n \rfloor$ best in $\mathcal{P}$
            $P(\{\boldsymbol{X}_u : u \in \mathcal{F}_j\}) \leftarrow$ MaxLikelihoodEstimate($\mathcal{S}$)
            for $\boldsymbol{x} \in \mathcal{P}_{2\ldots n}$ do
                GenepoolOptimalMixing($\boldsymbol{x}$,$\mathcal{F}_j$)
            AdaptiveVarianceScaling($\mathcal{F}_j$)
        for $\boldsymbol{x} \in \mathcal{P}_{2\ldots n_{\mathrm{AMS}}+1}$ do
            AnticipatedMeanShift($\boldsymbol{x}$)
        for $\boldsymbol{x} \in \mathcal{P}_{2\ldots n}$ do
            if NIS($\boldsymbol{x}$) > NIS$^{\mathrm{MAX}}$ then
                ForcedImprovement($\boldsymbol{x}$)
```

Algorithm 3.6 Gene-pool Optimal Mixing

```
procedure GenepoolOptimalMixing($\boldsymbol{x}$,$\mathcal{F}_j$)
    $\boldsymbol{b} \leftarrow \boldsymbol{x}_{\mathcal{F}_j}$
    $\boldsymbol{x}_{\mathcal{F}_j} \leftarrow P(\{\boldsymbol{X}_u : u \in \mathcal{F}_j\})$          ▷ Sampling
    $\boldsymbol{f_o} \leftarrow$ PartialEvaluation($\boldsymbol{x}, \boldsymbol{f_x}, \mathcal{F}_j$)
    if $\boldsymbol{f_o} < \boldsymbol{f_x}$ or $\mathcal{U}(0,1) < p^{\texttt{accept}}$ then $\boldsymbol{f_x} \leftarrow \boldsymbol{f_o}$
    else $\boldsymbol{x}_{\mathcal{F}_j} \leftarrow \boldsymbol{b}$
```

Here, we deviate from the original RV-GOMEA definition in that selection is now performed at the start of each iteration of GOM, whereas previously, selection was only performed at the start of each generation [6]. This modification is important when the linkage model is not a marginal product. In such a case, variation is applied to a problem variable more than once per generation. The estimated distribution may then become outdated throughout a generation, making GOM less effective and efficient. Supporting experiments are included in Appendix 3.A.

3.4. Linkage Modeling

Different methods for the modeling of dependencies between variables have previously been used for the design of Estimation of Distribution Algorithms (EDAs) [23]. Some of the models used by early EDAs include univariate [22, 29], tree-based [13], or multivariate [21] models. Marginal product models were however shown to be too restrictive to solve difficult problems [11]. Alternatively, Bayesian networks were later used in EDAs for discrete [27, 28] and real-valued optimization [1, 9, 12, 17]. Also Markov networks, or Markov random fields [35], have been proposed for use in (discrete) EDAs [34], for example in the Markovianity-based Optimization Algorithm (MOA) [33], the Distribution Estimation using Markov networks (DEUM) [31] algorithm, or the Markov Network Estimation of Distribution Algorithm (MN-EDA) [30]. Factorized Gaussian Markov networks were previously used in real-valued optimization in the Gaussian Markov Random Field EDA (GMRF-EDA) [24], though the latter method is focused on problems with disjoint sets of dependent variables. Here, we focus on the introduction of conditional linkage models, in particular based on Markov and Bayesian networks, because such models appear most suitable for problems with overlapping dependencies and integration with RV-GOMEA.

3.4.1. Non-Conditional Linkage Models

A marginal product linkage model is an appropriate model for problems with disjoint groups of dependent variables. However, this is not efficient in case long chains of (pairwise) dependent variables exist, as such dependencies must be modeled by a single joint group. This requires the estimation of an ℓ-variate MVG, which becomes computationally expensive to sample from for large ℓ and requires a relatively large population to accurately estimate.

We illustrate the difficulty of optimizing problems with strong overlapping dependencies through the $f^{REBChain}(\boldsymbol{x}, c, \theta)$ function, defined in Equation (3.2), where $R^{\theta}(\boldsymbol{x})$ applies a rotation to $\boldsymbol{x}$ of θ degrees around the origin in a clockwise direction for each pair of variables. This function is the sum of a 2-dimensional rotated ellipsoid function with condition number c (see Equation (3.3)) for each pair of consecutive variables. In Figure 3.1 we show the number of evaluations required to solve the 20-dimensional $f^{REBChain}$ function for varying rotation angles and condition numbers using RV-GOMEA with different marginal linkage models. In all cases, the Interleaved Multistart Scheme (IMS) [6] was used, which scales the population size over time by interleaving runs with gradually increasing population sizes, using an initial population size of 20.

Figure 3.1a clearly shows that a univariate linkage model is not adequate in case of a large condition number and large rotation angles, i.e., when dependencies are strong. Results for the BFLT, shown in Figure 3.1b, shows that a hierarchical model that does not include the joint distribution, leads to only minor improvements. The best performance for strongly dependent problems is obtained with the full linkage model, shown in Figure 3.1c. It is however known that the full linkage model does not scale well to very high-dimensional problems, and cannot benefit from a GBO setting. Furthermore, a full linkage model is often overkill, as most variables are not directly and/or strongly dependent in real-world problems.

$$f^{REBChain}(\boldsymbol{x}, c, \theta) = \sum_{i=1}^{\ell-1} f^{2DEllipsoid}(R^{\theta}(\{\boldsymbol{x}_i, \boldsymbol{x}_{i+1}\}), c), \tag{3.2}$$

$$f^{2DEllipsoid}(\boldsymbol{x}, c) = \boldsymbol{x}_1^2 + 10^c \boldsymbol{x}_2^2. \tag{3.3}$$

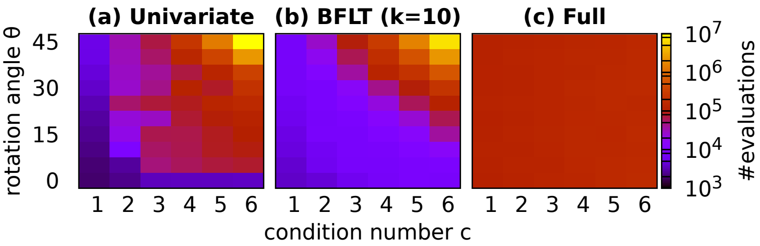


Figure 3.1: Median (30 runs) number of evaluations (color-coded) required to find the optimum with a precision of 10^{-10} of the 20-dimensional $f^{REBChain}(\boldsymbol{x}, c, \theta)$, or the budget of 10^7 function evaluations was used, for RV-GOMEA using different marginal linkage models.

3.4.2. Bayesian Factorization

Updating variables $\mathcal{I}_j$ pertaining to a single subfunction f_j while taking variables pertaining to overlapping subfunctions into account, can be done by conditional sampling. To be precise, instead of sampling new values for problem variables $\boldsymbol{x}_{\mathcal{I}_j}$ using the multivariate marginal $P(\boldsymbol{X}_{\mathcal{I}_j})$, one could sample conditionally on the other variables of $\boldsymbol{x}$, i.e., $\boldsymbol{x}_{\neg\mathcal{I}_j}$ with $\neg\mathcal{I}_j = \mathcal{I} \setminus \mathcal{I}_j$, giving,

$$\boldsymbol{x}_{\mathcal{I}_j} \leftarrow P(\boldsymbol{X}_{\mathcal{I}_j} | \boldsymbol{X}_{\neg\mathcal{I}_j} = \boldsymbol{x}_{\neg\mathcal{I}_j}). \tag{3.4}$$

In this way, partial evaluations can still be performed.

As we use an MVG distribution to sample from, its conditional distribution is also an MVG, which is easy to sample from [16]. However, directly using this conditional distribution to sample from is also computationally expensive, as conditioning on all other variables has a computational complexity of $\mathcal{O}(\ell^3)$. To reduce computational cost, we can exploit the VIG. The VIG $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ represents a Markov network (or field), and specifically a Gaussian Markov Field (GMF), between problem variables. In a GMF, the probability density function of a variable is completely defined in terms of a conditional probability function given its neighbors. Therefore, samples can be drawn from a GMF without modeling the joint probability distribution of all variables [31, 32]. Let $\mathrm{N}(\mathcal{I}_j)$ be the set of all neighbors of $\mathcal{I}_j$ in the VIG. Then, the conditional probability in Equation (3.4) simplifies to,

$$P\left(\boldsymbol{X}_{\mathcal{I}_j} | \boldsymbol{X}_{\neg\mathcal{I}_j} = \boldsymbol{x}_{\neg\mathcal{I}_j}\right) = P\left(\boldsymbol{X}_{\mathcal{I}_j} | \boldsymbol{X}_{N(\mathcal{I}_j)} = \boldsymbol{x}_{N(\mathcal{I}_j)}\right). \tag{3.5}$$

This has two advantages. First, it reduces computational cost of sampling from the conditional distribution, as sampling from $P(\boldsymbol{X})$ has a complexity of $\mathcal{O}(\ell d^3)$ when the maximum degree of the VIG is bounded by a constant d. Second, the sample distribution explicitly takes conditional independencies into account. Estimating the sample distribution of a GMF, instead of the full MVG, requires fewer parameters to be estimated, which can therefore be done with a smaller population size, furthermore improving efficiency.

Partitioning

More in general, large groups of variables may be sampled from a joint MVG. Given a partitioning of variables $\mathcal{C}$, the variables in each partition $\mathcal{C}_i$ can be jointly sampled conditioned on $\boldsymbol{X}_{N(\mathcal{C}_i)}$. For a non-overlapping partitioning of variables $\mathcal{C} \subset \mathcal{P}(\mathcal{I})$, i.e., with $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ for $\mathcal{C}_i, \mathcal{C}_j \in \mathcal{C}$, this results in,

$$P(\boldsymbol{X}) = \prod_{\mathcal{C}_j \in \mathcal{C}} P\left(\boldsymbol{X}_{\mathcal{C}_j} | \boldsymbol{X}_{N(\mathcal{C}_j)}\right). \tag{3.6}$$

The larger the clusters $\mathcal{C}_j$, the less the independence structure imposed by the VIG is leveraged. This effect is smaller if the variables in $\mathcal{C}_j$ form a clique in the VIG, and guaranteed to be minimal if the clusters are set to be the singletons. If a univariate partitioning, i.e., $\mathcal{C} = \{\{1\}, \ldots, \{\ell\}\}$, is used, Equation (3.6) represents a univariate conditional factorization. We refer to this factorization as UCond.

When each factor $\mathcal{C}_i$ may consist of multiple variables, Equation (3.6) represents a multivariate conditional factorization. We aim to have these multivariate factors correspond to the subfunctions, as all variables required for a subfunction may be jointly sampled. As such, these factors are constructed by traversing $\boldsymbol{G}$ in a breadth-first order, and finding maximal cliques. Given the current vertex u and its previously visited neighbors, a maximal clique including both u and these neighbors is constructed, if possible. A factor $\mathcal{C}_i$ is then defined as this clique, excluding the previously visited neighbors, and is added to $\mathcal{C}$. This process continues until each variable (or node) is contained by a factor $\mathcal{C}_i$. We refer to this factorization as MCond.

Note that the univariate and multivariate conditional factorizations may define the same probability distribution. Their distinction becomes important when a selection step is performed after sampling variables for some factor $\mathcal{C}_i$, as discussed in Section 3.4.3.

Sampling

We use forward sampling [34] to sample from the distribution described in Equation (3.6). This requires a topological ordering of the GMF. As each partition $\mathcal{C}_i$ is jointly sampled, it is sufficient to direct the edges between vertices in different partitions. This is done by defining an order $\boldsymbol{O}$ in which the partitions of $\mathcal{C}$ are sampled. The order $\boldsymbol{O}$ is defined as a function mapping the index of a partition to its precedence. As such, $\mathcal{C}_u$ precedes $\mathcal{C}_v$ if $\boldsymbol{O}(u) < \boldsymbol{O}(v)$, indicating that each edge $(q, w) \in \boldsymbol{E}$ with $q \in \mathcal{C}_u$ and $w \in \mathcal{C}_v$ is directed as $w \leftarrow q$. Variables are then sampled conditioned on all variables from which an incoming edge exists.

Directing the edges of the GMF will generally result in a different joint probability distribution than represented by the GMF, and may break dependencies between variables due to d-separation [18]. This is undesirable, because we aim to model a conditional dependency between each pair of variables between which a path exists in the VIG. To minimize the number of breaks, we traverse the network in a breadth-first manner, starting from the partition of a randomly selected vertex. This ensures that only the first partition is sampled independently from all others, and all remaining variables are sampled conditionally dependent. For each partition $\mathcal{C}_j$, values are sampled for problem variables $\boldsymbol{X}_u$ with $u \in \mathcal{C}_j$. These values are sampled from a normal probability distribution that is conditioned on the problem variables corresponding to the neighboring vertices of $u \in \mathcal{C}_j$ that have already been sampled. The values of the starting partition are sampled independently of all other variables. Doing so simplifies the probability distribution in Equation (3.6) to,

$$P\left(\boldsymbol{X}_{\mathcal{C}_j} | \boldsymbol{X}_{N(\mathcal{C}_j)} = \boldsymbol{x}_{N(\mathcal{C}_j)}\right) \approx P\left(\boldsymbol{X}_{\mathcal{C}_j} | \boldsymbol{X}_{\pi(\mathcal{C}_j)} = \boldsymbol{x}_{\pi(\mathcal{C}_j)}\right), \tag{3.7}$$

with $\pi(\mathcal{C}_j) = \{u \in N(\mathcal{C}_j) : \boldsymbol{O}(u) < \boldsymbol{O}(v) \text{ for all } v \in \mathcal{C}_j\}$ the neighbors of $\mathcal{C}_j$ that preceded $\mathcal{C}_j$ in the sampling order $\boldsymbol{O}$.

Figure 3.2 shows a VIG of the 13-dimensional REB5SmallOverlap (Equation (3.15)) problem, given a UCond factorization. Figure 3.3 then shows the same VIG color coded according to the MCond factorization. An edge $u \rightarrow v$ indicates that $\boldsymbol{X}_v$ is sampled conditioned on $\boldsymbol{X}_u$. Edges are directed corresponding to an arbitrary (breadth-first) ordering starting from vertex 7.

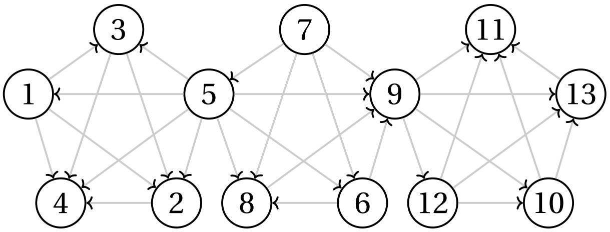


Figure 3.2: A VIG for the 13-dimensional REB5SmallOverlap problem, showing a potential breadth-first ordering starting from randomly selected vertex 7.

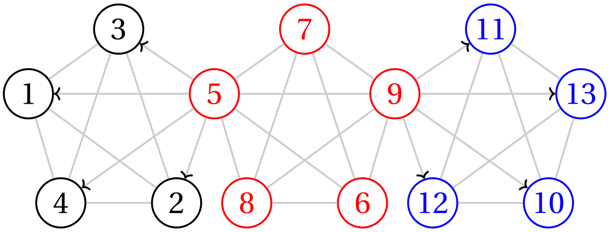


Figure 3.3: A VIG for the 13-dimensional REB5SmallOverlap problem, showing a color-coded partitioning starting from randomly selected vertex 7.

3.4.3. GOM with Conditional Linkage Model

Because the application of GOM is the main strength of RV-GOMEA, and the way in which GOM is applied substantially influences performance and scalability, we explore different options of using GOM when variables are sampled from a conditional distribution.

FOS of Conditional Linkage Model

How GOM is applied using a conditional linkage model is still defined by a FOS, now indicated $\mathcal{F}^{\mathcal{C}}$. However, each FOS element $\mathcal{F}^{\mathcal{C}}_i$ now indicates which factors of $\mathcal{C}$ are resampled during one iteration of GOM. Applying GOM for a FOS element $\mathcal{F}^{\mathcal{C}}_i$ to a solution $\boldsymbol{x}$ then amounts to sampling from the Bayesian factorization for each factor $\mathcal{C}_j$ with $j \in \mathcal{F}^{\mathcal{C}}_i$, i.e.,

$$P\left(\boldsymbol{X}_{\mathcal{F}^{\mathcal{C}}_i}\right) = \prod_{j \in \mathcal{F}^{\mathcal{C}}_i} P\left(\boldsymbol{X}_{\mathcal{C}_j} | \boldsymbol{X}_{\pi(\mathcal{C}_j)}\right), \tag{3.8}$$

followed by the selection step of GOM. We consider three different options for the FOS $\mathcal{F}^{\mathcal{C}}$ used in conditional linkage models.

First we consider the option where all variables are sampled during each iteration of GOM, i.e., $\mathcal{F}^{\mathcal{C}} = \{\{1,\ldots,|\mathcal{C}|\}\}$. The selection step of GOM is therefore only applied after all variables have been sampled. In this case RV-GOMEA reverts to a classic EDA. Because of the estimation and sampling used, it is similar to the Bayesian factorized version of AMaLGaM [10], but without learning the factorization structure each generation. As all variables are sampled during each iteration of GOM, partial evaluations do not benefit the optimization in this case. The benefit of a GBO setting is the fact that a Markov network can be constructed from the VIG. This option is referred to as Generational GOM (GG), as each solution in the population is subject to one iteration of GOM per generation.

In the second option we consider, we aim to benefit from partial evaluations by sampling only the variables for one factor $\mathcal{C}_i$ during each iteration of GOM, i.e., $\mathcal{F}^{\mathcal{C}} = \{\{1\},\ldots,\{|\mathcal{C}|\}\}$. This option is referred to as Factorized GOM (FG). As a selection step is applied after sampling a subset of variables, optimization for strongly dependent problems may be hindered, because the selection step between these univariate variation operations prohibit moving a solution in a strongly correlated multi-dimensional cone of improvement.

Thirdly, we combine GG and FG to potentially combine their strengths. As such, GOM is first applied to each factor separately, followed by all factors jointly, i.e., $\mathcal{F}^{\mathcal{C}} = \{\{1\},\ldots,\{|\mathcal{C}|\},\{1,\ldots,|\mathcal{C}|\}\}$. We refer to this option as Hybrid GOM (HG).

Application of GOM

A conditional linkage model is a combination of an underlying Bayesian factorization and a FOS $\mathcal{F}^{\mathcal{C}}$ that describes how GOM is performed.

Algorithm 3.7 describes how GOM is applied to a solution $\boldsymbol{x}$ for a FOS element $\mathcal{F}^{\mathcal{C}}_j$ of a conditional linkage model.

In Figure 3.4 we illustrate results similar to those shown in Figure 3.1 for a number of introduced conditional linkage models. This shows that UCond-FG and UCond-GG

Algorithm 3.7 Conditional Gene-pool Optimal Mixing

```
procedure GenepoolOptimalMixing($\boldsymbol{x}, \mathcal{F}^{\mathcal{C}}_j, (\boldsymbol{V}, \boldsymbol{E}), \boldsymbol{O}$)
    $\boldsymbol{b} \leftarrow \boldsymbol{x}_{\mathcal{F}^{\mathcal{C}}_j}$
    for $u \in \mathcal{F}^{\mathcal{C}}_j$ do                                    ▷ In order $\boldsymbol{O}$
        $\boldsymbol{Y} \leftarrow \{w \in \mathbb{I} : w \in \pi(\mathcal{C}_u)\}$
        $\boldsymbol{x}_{\mathcal{C}_u} \leftarrow P(\boldsymbol{X}_{\mathcal{C}_u} \mid \boldsymbol{X}_{\boldsymbol{Y}} = \boldsymbol{x}_{\boldsymbol{Y}})$          ▷ Eq. (3.7)
    $\boldsymbol{f_o} \leftarrow$ PartialEvaluation($\boldsymbol{x}, \boldsymbol{f_x}, \mathcal{F}^{\mathcal{C}}_j$)
    if $\boldsymbol{f_o} < \boldsymbol{f_x}$ or $\mathcal{U}(0,1) < p^{\text{accept}}$ then $\boldsymbol{f_x} \leftarrow \boldsymbol{f_o}$
    else $\boldsymbol{x}_{\mathcal{F}^{\mathcal{C}}_j} \leftarrow \boldsymbol{b}$
```

perform better than their non-conditional counterparts, i.e., univariate and full, respectively. Furthermore, the hybrid model UCond-HG achieves the best performance on strongly dependent problems, but also benefits from the univariate steps on weakly dependent problems. The MCond models are not shown here, as they are almost identical to the UCond models for this problem.

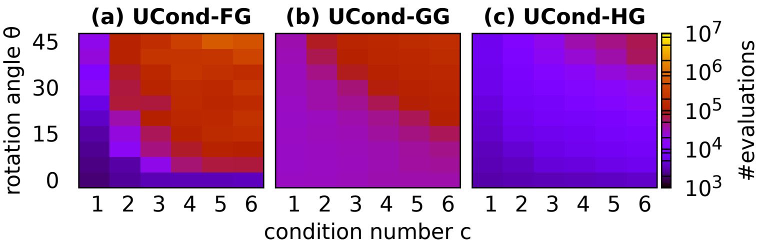


Figure 3.4: Median (30 runs) number of evaluations (color-coded) required to find the optimum with a precision of 10^{-10} of the 20-dimensional $f^{REBChain}(\boldsymbol{x}, c, \theta)$ for RV-GOMEA using different conditional linkage models.

3.5. EXPERIMENTS

In this section, we analyze the performance of RV-GOMEA with conditional linkage models, and compare it to non-conditional linkage models. We only consider benchmark problems that are non-separable, because disjoint sets of variables in the VIG are independent, and can therefore be reduced to multiple non-separable problems that can be solved independently if g (see Equation (6.1)) is the identity function. All experiments are performed in a GBO setting that allows for partial evaluations.

We compare the univariate (Uni), full, UCond-GG, UCond-FG, UCond-HG, and MCond-HG linkage models. As the MCond-GG model is essentially identical to the UCond-GG model for our benchmark problems, and the MCond-FG was never found to be more effective than the UCond-FG, no results are shown for MCond.

We furthermore compare the performance of RV-GOMEA to VkD-CMA [2, 3]. VkD-CMA is among the current state of the art in continuous optimization, and is also capable of exploiting the fact that the number of dependencies is less than all possible dependencies, albeit through a means that is incompatible with exploiting partial evaluations.

3.5.1. BENCHMARK PROBLEMS

We firstly consider the Rosenbrock function, defined as:

$$f^{Rosenbrock}(\boldsymbol{x}) = \sum_{i=1}^{\ell-1} \left[100(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i^2)^2 + (1-\boldsymbol{x}_i)^2\right]. \tag{3.9}$$

Each remaining benchmark problem we consider is constructed using rotated ellipsoid subfunctions, as defined in Equation (3.10). For a large condition number and rotation angle, the ellipsoid function is strongly dependent and cannot be solved efficiently using a linkage model that does not model the variables of the ellipsoid as jointly dependent. Furthermore, using rotated ellipsoid subfunctions allows us to analyze benchmark problems with different overlapping dependency structures.

Secondly, we introduce the *Rotated Ellipsoid Blocks* (REB) problem in Equation (3.11), defined as the sum of a number of rotated ellipsoid functions (Equation (3.10)) of size k with condition number 10^c, rotation angle θ, and stride $1 \le s \le k$. The REB function is a generalization of the Sum of Rotated Ellipsoid Blocks (SoREB) function [6]. The function $R^\theta(\boldsymbol{x})$ applies a counter-clockwise rotation to $\boldsymbol{x}$ of θ degrees around the origin for each pair of variables. The stride s determines the number of overlapping variables, as the starting indices of consecutive ellipsoid blocks are s apart. For $s = k$, each ellipsoid block is disjoint and therefore independent.

$$f^{Ellipsoid}(\boldsymbol{x}, c) = \sum_{i=1}^{|\boldsymbol{x}|} 10^{c(i-1)/(|\boldsymbol{x}|-1)} \boldsymbol{x}_i^2. \tag{3.10}$$

$$f^{REB}(\boldsymbol{x}, c, \theta, k, s) = \sum_{i=0}^{\frac{|\boldsymbol{x}|-k}{s}} f^{Ellipsoid}(R^\theta(\boldsymbol{x}[1+is : k+is]), c). \tag{3.11}$$

To have more flexibility in testing problems with overlapping dependencies, we define the *EllipsoidGraph* problem in Equation (3.12), of which the dependency structure is defined by the graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$. For each vertex $v \in \boldsymbol{V}$, a rotated ellipsoid of v and its neighboring vertices $N(v)$ is added, i.e.,

$$f^{REBGraph}(\boldsymbol{x}, c, \theta, (\boldsymbol{V}, \boldsymbol{E})) = \sum_{v \in \boldsymbol{V}} f^{Ellipsoid}(R^\theta(\{v\} \cup N(v)), c). \tag{3.12}$$

For convenience, we define the following functions:

$$f^{REBChainWeak}(\boldsymbol{x}) = f^{REB}(\boldsymbol{x}, c=1, \theta=5, k=5, s=4), \tag{3.13}$$

$$f^{REBChainStrong}(\boldsymbol{x}) = f^{REB}(\boldsymbol{x}, c=6, \theta=45, k=5, s=4), \tag{3.14}$$

$$f^{REB5SmallOverlap}(\boldsymbol{x}) = f^{REB}(\boldsymbol{x}, c=6, \theta=45, k=5, s=4), \tag{3.15}$$

$$f^{REB5LargeOverlap}(\boldsymbol{x}) = f^{REB}(\boldsymbol{x}, c=6, \theta=45, k=5, s=1), \tag{3.16}$$

$$f^{REBGrid}(\boldsymbol{x}) = f^{REBGraph}(\boldsymbol{x}, c=6, \theta=45, \boldsymbol{G}^{\texttt{Grid}}), \tag{3.17}$$

$$f^{REBTorus}(\boldsymbol{x}) = f^{REBGraph}(\boldsymbol{x}, c=6, \theta=45, \boldsymbol{G}^{\texttt{Torus}}), \tag{3.18}$$

$$f^{REBCube}(\boldsymbol{x}) = f^{REBGraph}(\boldsymbol{x}, c=6, \theta=45, \boldsymbol{G}^{\texttt{Cube}}), \tag{3.19}$$

with $\boldsymbol{G}^{\texttt{Grid}}$ a graph where the vertices are arranged into a $\sqrt{\ell} \times \sqrt{\ell}$ square grid, and an edge is added between each horizontally or vertically neighboring pair of vertices, with no wrap-around. The graph $\boldsymbol{G}^{\texttt{Grid}}$ is therefore a planar graph with $2(\ell - \sqrt{\ell})$ edges, and the degree of each vertex is at least 2 and at most 4.

The graph $\boldsymbol{G}^{\texttt{Torus}}$ is identical to $\boldsymbol{G}^{\texttt{Grid}}$, but has wrap-around at the edges of the square grid. Therefore, the $\boldsymbol{G}^{\texttt{Torus}}$ has 2ℓ edges and the degree of each edge is 4. We only consider the REBGrid and REBTorus problems for a square number of variables.

For the graph $\boldsymbol{G}^{\texttt{Cube}}$, the vertices are arranged into a $\sqrt[3]{\ell} \times \sqrt[3]{\ell} \times \sqrt[3]{\ell}$ simple cubic lattice. An edge is added between pairs of vertices neighboring in the x, y, or z-direction, with no wrap-around. The total number of edges is therefore $3(\ell - \ell^{2/3})$, and each vertex has a degree of at least 3 and at most 6. We only consider the REBCube problem for a cubic number of variables.

3.5.2. EXPERIMENTAL SET-UP

In each run, a budget of 10^7 function evaluations was used, and a time limit of 3 hours. In case of premature convergence before exceeding the budget, a run is restarted using the same population size. An implementation of RV-GOMEA [6] in C++, and an implementation of VkD-CMA [2, 3] in Python were used. Unless stated otherwise, default parameters were used for these algorithms. Source code for these algorithms is available on the web pages of the respective authors. The additions to RV-GOMEA introduced in this chapter were also implemented in C++, for which source code is publicly available[1].

3.5.3. OPTIMAL POPULATION SIZE

We first determine how the optimal population size scales with the increase of problem dimensionality for these linkage models, as this can be seen as a clear indicator of how well the linkage model matches the problem structure. For this purpose, we report the median population size of a number of bisections (1 for VkD-CMA, 5 for RV-GOMEA, due to substantial differences in required computation time) in Figure 3.5, with each data point in the bisection the median of 30 optimization runs. Estimated optimal population size for larger dimensions (dashed lines) were linearly extrapolated (on a

[1] `https://homepages.cwi.nl/~bosman/`

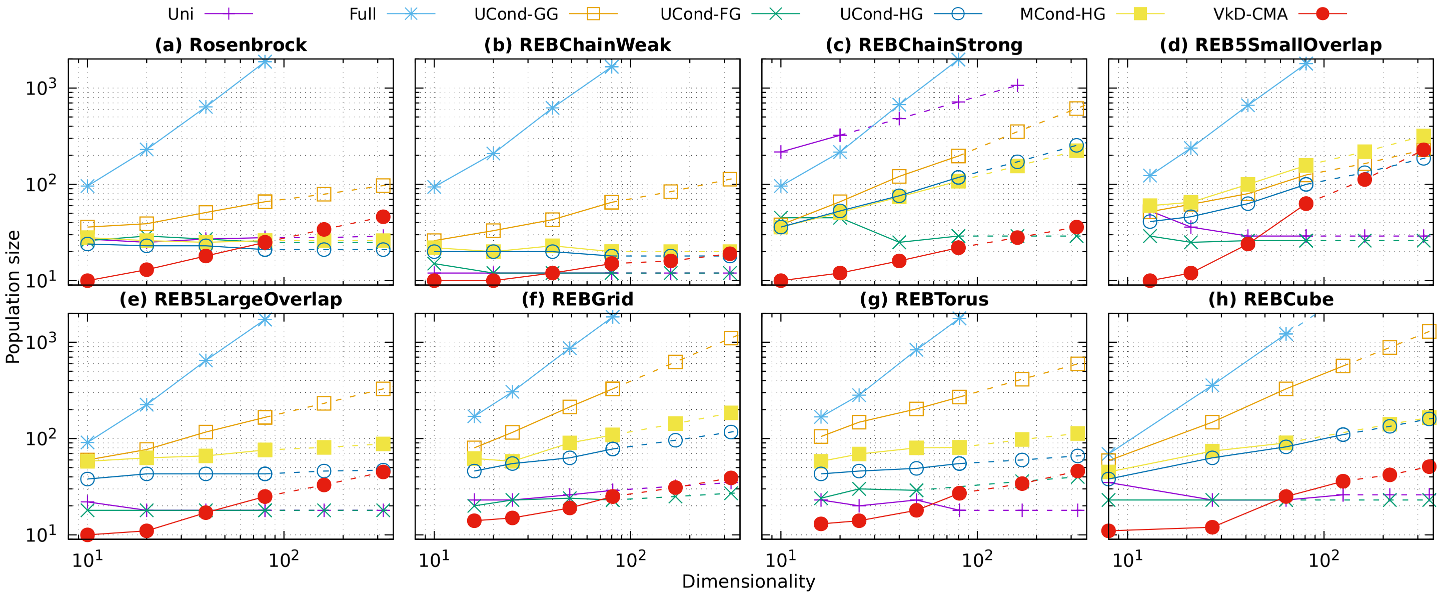


Figure 3.5: Estimated optimal population size for various linkage models for various benchmark problems, determined as median population size of a number (5 for RV-GOMEA, 1 for VkD-CMA) bisections, with each data point of a bisection the median number of evaluations of 30 runs.

log-log scale) with non-decreasing slope, as these populations were used to obtain results shown in Figure 3.6. Running bisections for all problem dimensionalities would require an exceptionally large amount of computation time.

3.5.4. SCALABILITY

In Figure 3.6 we show the scalability of the number of function evaluations needed by RV-GOMEA with conditional linkage models, RV-GOMEA with non-conditional linkage models, and VkD-CMA. The population size is set to the value found through bisections or extrapolation, displayed in Figure 3.5.

In Figure 3.6 we generally observe the same trends as in Figure 3.5. The UCond-HG model is found to achieve the best performance on all problems with strong dependencies, except on the REB5SmallOverlap problem, where it performs slightly worse than the UCond-GG model, but seems to have better scalability. Furthermore, on the problems with weak overlapping dependencies, the UCond-HG was only a constant factor worse than the univariate models, which obtained the best performance.

Interestingly, the Uni and UCond-FG linkage models performed fairly well on the REB5LargeOverlap, REBGrid, REBTorus and REBCube problems, despite these problems consisting of 3-dimensional to 6-dimensional rotated ellipsoids with high condition number and rotation angle, indicating that the optimization problem becomes easier when many dependencies overlap. This is caused by the fact that the problem landscape of a subset of variables Y becomes less ill-conditioned when overlapping additive rotated ellipsoid subfunctions exist.

Furthermore, though the Uni and UCond-FG linkage models required over 10^6 function evaluations for even small dimensionalities of the REB5SmallOverlap function, this remained roughly constant for larger problem dimensionalities. This is caused by the fact that the subfunctions in this problem do not overlap much and are each relatively difficult to solve for univariate linkage models. Problem difficulty does however not increase with dimensionality if more such subfunctions are added, due to GOM.

3.6. DISCUSSION

Based on the results discussed in Sections 3.5.3 and 3.5.4, we argue that conditional linkage models have a clear benefit over non-conditional linkage models in RV-GOMEA. In particular, we find that using a conditional linkage model with Hybrid GOM (HG) achieves the best or close to the best performance on all problems, while also achieving the best scalability on all problems. The UCond-HG variant seems to be the best of the two conditional variants, as it was only slightly worse on the REB5SmallOverlap problem, and better on all other problems.

We find that VkD-CMA generally requires a smaller population size than RV-GOMEA, and performed better than the variants of RV-GOMEA that do not benefit from partial evaluations (full and UCond-GG). This is likely a result of the fact that variants of CMA-ES are known to obtain excellent performance on problems with quadratic surfaces, such as the benchmark problems considered here. Though RV-GOMEA here used a sampling model based on AMaLGaM [10], as it was originally

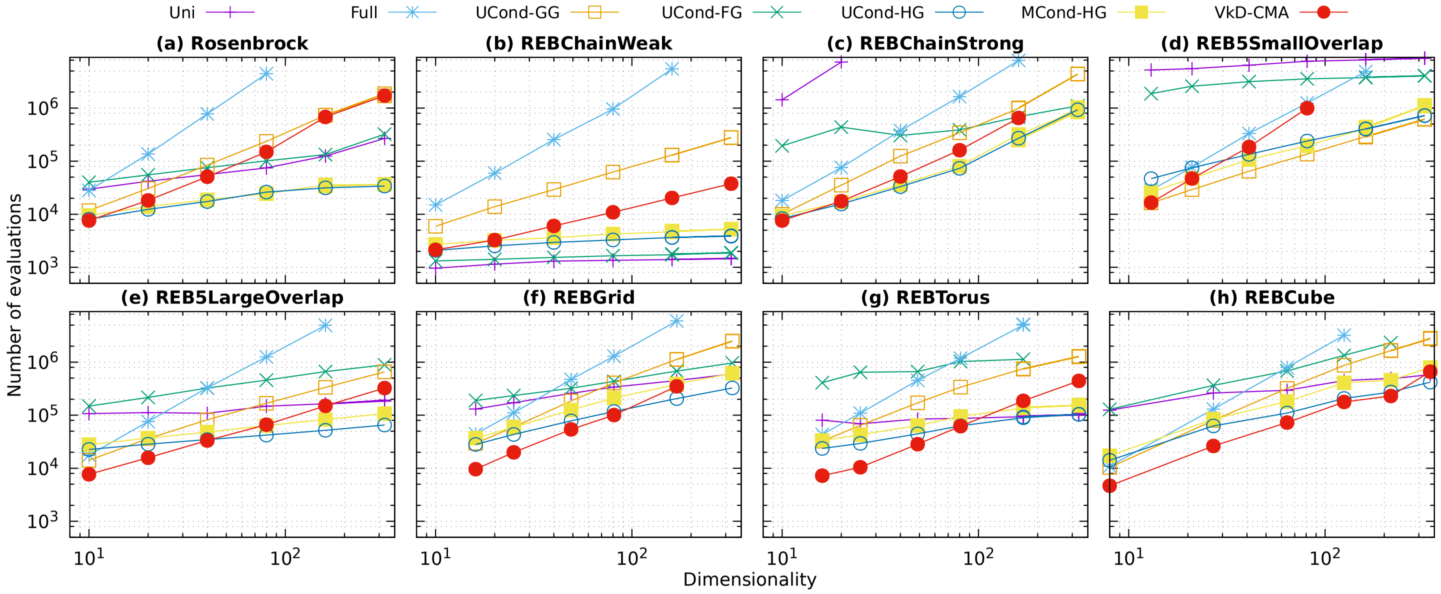


Figure 3.6: Median (30 runs) number of evaluations for various linkage models for various benchmark problems in a GBO setting using the population size shown in Figure 3.5.

introduced [6], it is possible to incorporate a sampling model based on CMA-ES into RV-GOMEA, potentially decreasing the required population size, and improving performance. This is left to future work.

In this work, we have only considered artificial benchmark problems, in particular benchmark problems constructed using rotated ellipsoids. In these problems, the dependency strength is equally strong for each rotated ellipsoid, though real-world problems may have varying degrees of dependency for different groups of variables, even if strictly speaking the problem is completely jointly dependent. For example, this is the case for HDR brachytherapy treatment planning for prostate cancer [25], where each variable is associated with a physical location, and the dependency strength of two variables correlates to the distance between their respective physical locations. To deal with such problems, it may be necessary to limit the linkage model to only the strongest dependencies, for example by removing edges corresponding to weak dependencies in the VIG. This does however require a notion of dependency strength with associated cut-off value, which we currently lack.

In this work, we limited our study to a GBO setting, because this is where RV-GOMEA mainly excels. The scalability of the UCond-HG linkage model will become worse in a BBO setting, because the univariate operations of GOM cannot benefit from partial evaluations. The UCond-GG linkage model does however not benefit from partial evaluations, and only benefits from the fact that the VIG is known a-priori, which would need to be learned online in a BBO setting. In contrast, the performance of VkD-CMA is identical in the GBO and BBO settings.

3.7. Conclusions

In this chapter, we aimed to improve the performance of RV-GOMEA on problems with strong overlapping dependencies in a GBO setting that allows for partial evaluations. For this purpose, various new sorts of conditional linkage models capable of capturing conditional dependencies were introduced to be combined with RV-GOMEA.

The efficiency of RV-GOMEA with these new conditional models was compared to when non-conditional linkage models are used, and to the efficiency of VkD-CMA, which is considered to be among the state of the art for continuous optimization with EAs.

To compare performance, a number of benchmark problems with overlapping dependencies were were constructed to have different dependency structures and different dependency strength.

We observed that the UCond-HG linkage model with RV-GOMEA obtained the best overall performance, and scaled better than or equally good as VkD-CMA on all considered benchmark problems. Note that all benchmark problems were considered in a GBO setting that allows for partial evaluations, from which RV-GOMEA can benefit, unlike VkD-CMA. This GBO setting has previously appeared in various real-world problems [5, 25], which also have an overlapping dependency structure.

Furthermore, the UCond-HG linkage model was able to match the scalability of univariate linkage models on problems with weak overlapping dependencies, and that of full linkage models on problems with strong overlapping dependencies. For two benchmark problems, the UCond-HG linkage model scaled better than all other considered linkage models.

We conclude that the introduction of conditional linkage models can greatly benefit the performance of RV-GOMEA on problems with overlapping dependencies in a GBO setting. These results are promising and may benefit the optimization of real-world problems with similar dependency structures [5, 25] in future work.

REFERENCES

[1] Ahn, C. W., Ramakrishna, R. S., and Goldberg, D. E. (2004). Real-coded Bayesian optimization algorithm: Bringing the strength of BOA into the continuous world. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 840–851. Springer.

[2] Akimoto, Y. and Hansen, N. (2016a). Online model selection for restricted covariance matrix adaptation. In *International Conference on Parallel Problem Solving from Nature*, pages 3–13. Springer.

[3] Akimoto, Y. and Hansen, N. (2016b). Projection-based restricted covariance matrix adaptation for high dimension. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 197–204. ACM.

[4] Bouter, A., Alderliesten, T., Bel, A., Witteveen, C., and Bosman, P. A. N. (2018). Large-scale parallelization of partial evaluations in evolutionary algorithms for real-world problems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1199–1206. ACM.

[5] Bouter, A., Alderliesten, T., and Bosman, P. A. N. (2017a). A novel model-based evolutionary algorithm for multi-objective deformable image registration with content mismatch and large deformations: benchmarking efficiency and quality. In Styner, M. A. and Angelini, E. D., editors, *Medical Imaging 2017: Image Processing*, volume 10133, pages 304 – 311. International Society for Optics and Photonics, SPIE.

[6] Bouter, A., Alderliesten, T., Witteveen, C., and Bosman, P. A. N. (2017b). Exploiting linkage information in real-valued optimization with the real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 705–712. ACM.

[7] Bouter, A., Luong, N. H., Witteveen, C., Alderliesten, T., and Bosman, P. A. N. (2017c). The multi-objective real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 537–544. ACM.

[8] Bouter, A., Maree, S. C., Alderliesten, T., and Bosman, P. A. N. (2020). Leveraging conditional linkage models in gray-box optimization with the real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 603–611.

[9] Bosman, P. A. N. (2009). On empirical memory design, faster selection of Bayesian factorizations and parameter-free Gaussian EDAs. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 389–396. ACM.

[10] Bosman, P. A. N., Grahl, J., and Thierens, D. (2013). Benchmarking parameter-free AMaLGaM on functions with and without noise. *Evolutionary Computation*, 21(3):445–469.

[11] Bosman, P. A. N. and Thierens, D. (1999). Linkage information processing in distribution estimation algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 60–67. Morgan Kaufmann Publishers Inc.

[12] Bosman, P. A. N. and Thierens, D. (2000). Expanding from discrete to continuous estimation of distribution algorithms: The ID𝔼A. In *International Conference on Parallel Problem Solving from Nature*, pages 767–776. Springer.

[13] De Bonet, J. S., Isbell Jr, C. L., and Viola, P. A. (1997). MIMIC: Finding optima by estimating probability densities. In *Advances in Neural Information Processing Systems*, pages 424–430.

[14] de Carvalho, O. Q., Tinós, R., Whitley, D., and Sanches, D. S. (2019). A new method for identification of recombining components in the generalized partition crossover. In *Brazilian Conference on Intelligent Systems (BRACIS)*, pages 36–41. IEEE.

[15] Deb, K. and Myburgh, C. (2016). Breaking the billion-variable barrier in real-world optimization using a customized evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 653–660. ACM.

[16] Eaton, M. (1983). Multivariate statistics: A vector space approach. *John Wiley & Sons, Inc.*

[17] Echegoyen, C., Lozano, J. A., Santana, R., and Larrañaga, P. (2007). Exact Bayesian network learning in estimation of distribution algorithms. In *IEEE Congress on Evolutionary Computation*, pages 1051–1058. IEEE.

[18] Geiger, D., Verma, T., and Pearl, J. (1990). Identifying independence in Bayesian networks. *Networks*, 20(5):507–534.

[19] Gronau, I. and Moran, S. (2007). Optimal implementations of UPGMA and other common clustering algorithms. *Information Processing Letters*, 104(6):205–210.

[20] Hansen, N., Müller, S., and Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18.

[21] Harik, G. (1999). Linkage learning via probabilistic modeling in the ECGA. *IlliGAL report*, 99010.

[22] Harik, G. R., Lobo, F. G., and Goldberg, D. E. (1999). The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 3(4):287–297.

[23] Hauschild, M. and Pelikan, M. (2011). An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, 1(3):111–128.

[24] Karshenas, H., Santana, R., Bielza, C., and Larrañaga, P. (2012). Continuous estimation of distribution algorithms based on factorized Gaussian Markov networks. In *Markov Networks in Evolutionary Computation*, pages 157–173. Springer.

[25] Luong, N. H., Alderliesten, T., Bel, A., Niatsetski, Y., and Bosman, P. A. N. (2018). Application and benchmarking of multi-objective evolutionary algorithms on high-dose-rate brachytherapy planning for prostate cancer treatment. *Swarm and Evolutionary Computation*, 40:37–52.

[26] Omidvar, M. N., Li, X., Mei, Y., and Yao, X. (2013). Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Transactions on Evolutionary Computation*, 18(3):378–393.

[27] Pelikan, M. (2005). Hierarchical Bayesian optimization algorithm. In *Hierarchical Bayesian optimization algorithm*, volume 170 of *Studies in Fuzziness and Soft Computing*, pages 105–129. Springer.

[28] Pelikan, M., Goldberg, D. E., and Cantú-Paz, E. (1999). BOA: The Bayesian optimization algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 525–532. Morgan Kaufmann Publishers Inc.

[29] Pelikan, M. and Mühlenbein, H. (1998). Marginal distributions in evolutionary algorithms. In *Proceedings of the International Conference on Genetic Algorithms Mendel*, volume 98, pages 90–95.

[30] Santana, R. (2005). Estimation of distribution algorithms with kikuchi approximations. *Evolutionary Computation*, 13(1):67–97.

[31] Shakya, S., Brownlee, A., McCall, J., Fournier, F., and Owusu, G. (2010). DEUM–a fully multivariate EDA based on Markov networks. In *Exploitation of Linkage Learning in Evolutionary Algorithms*, pages 71–93. Springer.

[32] Shakya, S. and McCall, J. (2007). Optimization by estimation of distribution with DEUM framework based on Markov random fields. *International Journal of Automation and Computing*, 4(3):262–272.

[33] Shakya, S. and Santana, R. (2008). An EDA based on local Markov property and Gibbs sampling. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 475–476. ACM.

[34] Shakya, S. and Santana, R. (2012). *Markov Networks in Evolutionary Computation*. Springer.

[35] Spitzer, F. (1971). Markov random fields and Gibbs ensembles. *The American Mathematical Monthly*, 78(2):142–154.

[36] Sun, Y., Kirley, M., and Halgamuge, S. K. (2017). A recursive decomposition method for large scale continuous optimization. *IEEE Transactions on Evolutionary Computation*, 22(5):647–661.

[37] Thierens, D. and Bosman, P. A. N. (2011). Optimal mixing evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 617–624. ACM.

[38] Tintos, R., Whitley, D., and Chicano, F. (2015). Partition crossover for pseudo-boolean optimization. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*, pages 137–149. ACM.

APPENDICES

3.A. FREQUENCY OF SELECTION IN RV-GOMEA

In this section we show how applying selection (and an update of the sampling relevant sampling distributions) at the start of each iteration of GOM affects the performance of RV-GOMEA [6], compared to previous experiments. We compare the originally used generational selection (Gen.Sel.) to selection applied at the start of each iteration of GOM (GOM.Sel.).

The sphere, Rosenbrock and SoREB benchmark problems are considered, as in [6]. For each benchmark problem, two reasonable linkage models are considered. Results for these experiments are shown in Figure 3.A.1. Medians of 30 runs are shown, along with error bars displaying 10th and 90th percentiles. Population sizes were determined by running a bisection.

As expected, we observe no difference between the two selection methods for the marginal product linkage models (Univariate and MPM5). Minor differences in performance are observed for the Bounded Fixed Linkage Tree (BFLT) model, most notable on the Rosenbrock problem, for which selection at the start of each iteration of GOM was favored. Note that large error bars on the Rosenbrock problem were the effect of runs converged to the local minimum, as no restart mechanism was used.

Based on the results shown in Figure 3.A.1, we argue that selection at the start of each iteration of GOM has no negative effect on the performance of RV-GOMEA on previously considered benchmark problems. As argued in the main chapter, we therefore employ this form of selection, as it benefits the application of conditional linkage models to RV-GOMEA.

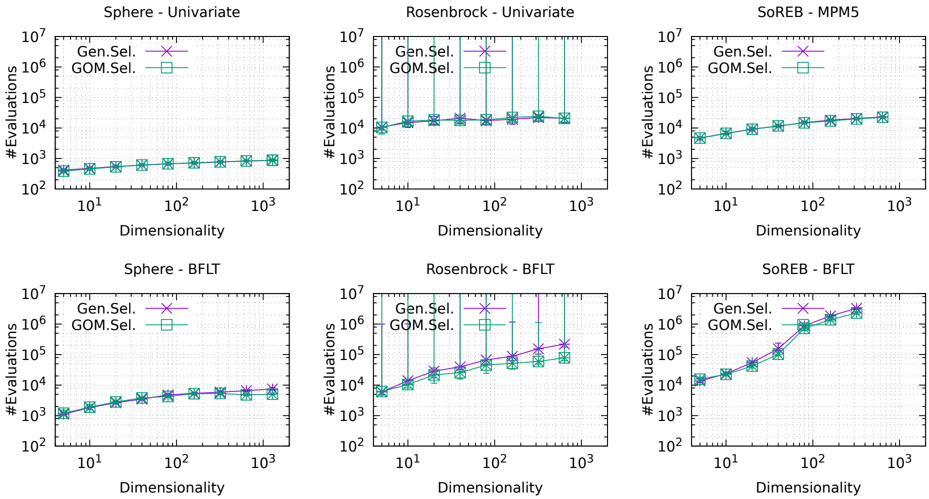


Figure 3.A.1: Median, 10th percentile and 90th percentile (30 runs) number of evaluations for various linkage models for various benchmark problems in a GBO setting. Selection was either used at the start of each generation (Gen.Sel.), or at the start of each iteration of GOM (GOM.Sel.).

4

Large-Scale Parallelization of Partial Evaluations for Real-World Problems

This chapter is based on the following publication: Bouter, A., Alderliesten, T., Bel, A., Witteveen, C., and Bosman, P. A. N. (2018). Large-scale parallelization of partial evaluations in evolutionary algorithms for real-world problems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1199–1206. ACM

ABSTRACT

The importance and potential of Gray-Box Optimization (GBO) with evolutionary algorithms is becoming increasingly clear lately, both for benchmark and real-world problems. We consider the GBO setting where partial evaluations are possible, meaning that sub-functions of the evaluation function are known and can be exploited to improve optimization efficiency. In this chapter, we show that the efficiency of GBO can be greatly improved through large-scale parallelism, exploiting the fact that each evaluation function requires the calculation of a number of independent subfunctions. This is especially interesting for real-world problems where often the majority of the computational effort is spent on the evaluation function. Moreover, we show how the best parallelization technique largely depends on factors including the number of subfunctions and their required computation time, revealing that for different parts of the optimization the best parallelization technique should be selected based on these factors. As an illustration, we show how large-scale parallelization can be applied to optimization of high-dose-rate brachytherapy treatment plans for prostate cancer. We find that use of a modern Graphics Processing Unit (GPU) was the most efficient parallelization technique in all realistic scenarios, leading to substantial speed-ups up to a factor of 73.

4.1. INTRODUCTION

Recent advances in evolutionary computation have shown that problems with millions of variables can be efficiently solved in a Gray-Box Optimization (GBO) setting, in the domains of both discrete [8] and real-valued optimization [3]. Optimization with Evolutionary Algorithms (EAs) in a GBO setting has also led to large improvements in optimization speed for various real-world problems, such as the traveling salesman problem [23], multi-objective deformable image registration [2], and the optimization of High-Dose-Rate (HDR) brachytherapy (BT) treatment plans for prostate cancer [17]. These problems have well-understood objective functions, but are nevertheless non-trivial to optimize. In the GBO setting that we consider, the objective functions are defined in a way that allows partial evaluations to be performed, meaning that the objective values of a solution can be efficiently updated after the modification of only a subset of variables.

The efficiency of an EA is frequently said to be easily improved through parallelization. The evaluation of all solutions in the population can be trivially parallelized, leading to large speed-ups for problems with a computationally expensive evaluation function. However, as the clock speed of Central Processing Units (CPUs) is starting to reach its physical limits [7], high-performance computing trends have recently focused on large-scale parallelism, for example using Graphics Processing Units (GPUs). Modern GPUs are equipped with several thousands of cores, but are restricted by the fact that they require a fine-grained parallelization model to be effective. In the field of evolutionary computation, GPUs are becoming widely used to speed-up various algorithms through the parallelization of their most time-consuming operations [9, 16, 20, 24].

In various real-world problems, evaluations require the majority of the computation time. Large-scale parallelization of the internal mechanisms of the EA will in that case only lead to marginal improvements when in a black-box setting. Instead, the parallelization of function evaluations could then lead to much larger improvements in performance. The degree of parallelization of evaluations is however limited by the population size, which rarely comes close to the thousands of cores that current-day GPUs are equipped with.

In a GBO setting, a much higher degree of parallelism can be achieved compared to a Black-Box Optimization (BBO) setting, because the objective function consists of known subfunctions that can be calculated independently. For each partial evaluation, a subset of these subfunctions has to be recalculated, which can be done in parallel. Because the population size, the number of functions to which partial evaluations can be applied, and the number of subfunctions in each decomposable evaluation function, all multiplicatively increase the possible degree of parallelism, large-scale parallelization quickly becomes feasible in GBO. We therefore apply large-scale parallelization to evaluation functions in a GBO setting, more specifically to partial evaluations, in order to greatly reduce the required computation time of partial evaluations.

In this chapter, we introduce a general approach for the large-scale parallelization of partial evaluations in an EA in a GBO setting. We compare parallelization techniques using a GPU, an Intel Xeon Phi, and multiple Central Processing Unit (CPU) cores, and we analyze which kind of parallelization approach is preferred in a range of different scenarios, depending on factors including the computational effort of a partial evaluation and the number of subfunctions that are calculated in parallel. We then extend and validate this analysis with the real-world optimization problem of HDR BT treatment planning for prostate cancer.

4.2. Gray-Box Optimization

In a GBO setting, some information on the optimization function is available. For example, a setting in which (part of) the definition of the optimization function is known can be considered a GBO setting if this information can be exploited to improve optimization.

One way of exploiting the GBO setting in (discrete) pseudo-Boolean optimization is by using partition crossover [22]. This approach does not easily generalize to real-valued optimization, as it works by finding the best among a finite set of possible offspring.

A more general way of exploiting GBO is by using partial evaluations. These are performed to efficiently update the objective value after the modification of a (small) subset of variables by recomputing the contribution of the modified variables to the objective value. Such partial evaluations were first discussed in [6] to be used with the discrete Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA) [21], and were recently demonstrated with the Real-Valued GOMEA (RV-GOMEA) [3] to have the potential to substantially (by orders of magnitude) improve performance.

As a simple example of a problem to which partial evaluations can be applied, consider the well-known multi-objective benchmark problem ZDT1 [10], defined as

$$\begin{aligned} f_1^{\text{ZDT1}}(\boldsymbol{x}) &= \boldsymbol{x}_1, \\ f_2^{\text{ZDT1}}(\boldsymbol{x}) &= g(\boldsymbol{x}) \cdot h(f_1^{\text{ZDT1}}(\boldsymbol{x}), g(\boldsymbol{x})), \\ g(\boldsymbol{x}) &= 1 + \frac{9}{\ell - 1} \sum_{i=2}^{\ell} \boldsymbol{x}_i, \\ h(f(\boldsymbol{x}), g(\boldsymbol{x})) &= 1 - \sqrt{\frac{f(\boldsymbol{x})}{g(\boldsymbol{x})}}, \end{aligned}$$

with ℓ the number of variables, which are real-valued in ZDT1.

Consider a solution $\boldsymbol{x}^g$ at time g, and the solution $\boldsymbol{x}^{g+1}$, which is the state of solution $\boldsymbol{x}$ at time $g+1$ after the modification of q variables. The $\mathcal{O}(\ell)$ time complexity of the ZDT1 function is caused by the calculation of $\sum_{i=2}^{\ell} \boldsymbol{x}_i^{g+1}$ required for the function $g(\boldsymbol{x}^{g+1})$. However, given the result of $\sum_{i=2}^{\ell} \boldsymbol{x}_i^{g}$, the result of $\sum_{i=2}^{\ell} \boldsymbol{x}_i^{g+1}$ can be computed in $\mathcal{O}(q)$ time, because new values of the q modified variables can be added to the sum, and previous values can be subtracted. For this reason, we maintain the result of $\sum_{i=2}^{\ell} \boldsymbol{x}_i^{g}$ in memory, and update it after any modification of $\boldsymbol{x}$. The value of $g(\boldsymbol{x}^{g+1})$ can then be calculated in constant time, resulting in a total complexity of $\mathcal{O}(q)$ if q variables are modified. In a similar way, partial evaluations can be applied to the other ZDT functions [10].

Applying partial evaluations to the ZDT functions requires only $\mathcal{O}(n)$ memory for a population $\boldsymbol{P}$ of size n, but introduces the possibility of applying partial evaluations with a time complexity of $\mathcal{O}(q)$, compared to full evaluations with a time complexity of $\mathcal{O}(\ell)$.

4.2.1. GENERAL REPRESENTATION

Given k subfunctions of $f(\boldsymbol{x})$, a function to which partial evaluations can be applied, let $\boldsymbol{I} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k\}$ where $\mathcal{I}_j$ is the set of indices of problem variables on which subfunction f_j^M is dependent. Given a solution $\boldsymbol{x}$, a general representation of $f(\boldsymbol{x})$ is:

$$\begin{aligned} f(\boldsymbol{x}) &= f^P\left(f_1^M(\boldsymbol{x}|_{\mathcal{I}_1}) \oplus f_2^M(\boldsymbol{x}|_{\mathcal{I}_2}) \oplus \cdots \oplus f_k^M(\boldsymbol{x}|_{\mathcal{I}_k})\right) \\ &= f^P\left(\bigoplus_{j=1}^{k} f_j^M(\boldsymbol{x}|_{\mathcal{I}_j})\right), \end{aligned}$$

where each $f_j^M(\boldsymbol{x}|_{\mathcal{I}_j})$ is a subfunction that depends on $\boldsymbol{x}|_{\mathcal{I}_j}$, which is a subset of $\boldsymbol{x}$ restricted to the indices in $\mathcal{I}_j \subseteq \{1, \ldots, \ell\}$.

To apply partial evaluations to $f(\boldsymbol{x})$, consider the state $\boldsymbol{x}^g$ of solution $\boldsymbol{x}$ at time g, and the state $\boldsymbol{x}^{g+1}$ at time $g+1$. In this example we assume that only the variable $\boldsymbol{x}_i$ was modified, i.e., $\boldsymbol{x}_j^g = \boldsymbol{x}_j^{g+1}$ for $j \neq i$. If the operator $\oplus$ has an inverse $\ominus$, then:

$$f(\boldsymbol{x}^{g+1}) = f^P\left(\bigoplus_{j=1}^{k} f_j^M(\boldsymbol{x}^g|_{\mathcal{I}_j}) \oplus \bigoplus_{\mathcal{I}_j \ni i} f_j^M(\boldsymbol{x}^{g+1}|_{\mathcal{I}_j}) \ominus \bigoplus_{\mathcal{I}_j \ni i} f_j^M(\boldsymbol{x}^g|_{\mathcal{I}_j})\right),$$

with $\mathcal{I}_j \ni i$ shorthand for $\{\mathcal{I}_j \in \boldsymbol{I} | i \in \mathcal{I}_j\}$. For all dependent subfunctions, the contributions of $\boldsymbol{x}^{g+1}$ are added with the $\oplus$ operator, while the contributions of $\boldsymbol{x}^g$ are subtracted with the $\ominus$ operator. The calculation of $f(\boldsymbol{x}^{g+1})$ can be made more efficient by maintaining the sum of subfunctions, denoted s^g, in memory. The value of s^g is recursively updated as follows:

$$\begin{aligned} s^1 &= \bigoplus_{j=1}^{k} f_j^M(\boldsymbol{x}^1|_{\mathcal{I}_j}), \\ s^{g+1} &= s^g \oplus \left[\bigoplus_{\mathcal{I}_j \ni i} f_j^M(\boldsymbol{x}^{g+1}|_{\mathcal{I}_j})\right] \ominus \left[\bigoplus_{\mathcal{I}_j \ni i} f_j^M(\boldsymbol{x}^g|_{\mathcal{I}_j})\right]. \end{aligned}$$

For the modification of more than one variable, i.e., variables $\boldsymbol{x}_i$ with indices $i \in \mathcal{F} \subseteq \{1, \ldots, \ell\}$, s^g is updated as follows:

$$s^{g+1} = s^g \oplus \left[\bigoplus_{(\mathcal{F} \cap \mathcal{I}_j) \ni i} f_j^M(\boldsymbol{x}^{g+1}|_{\mathcal{I}_j})\right] \ominus \left[\bigoplus_{(\mathcal{F} \cap \mathcal{I}_j) \ni i} f_j^M(\boldsymbol{x}^g|_{\mathcal{I}_j})\right].$$

with $(\mathcal{F} \cap \mathcal{I}_j) \ni i$ shorthand for $\left\{\mathcal{I}_j \in \boldsymbol{I} | i \in (\mathcal{I}_j \cap \mathcal{F})\right\}$. After the recursive update of s^{g+1}, the partial evaluation of $f(\boldsymbol{x}^{g+1})$ is then calculated as:

$$f(\boldsymbol{x}^{g+1}) = f^P(s^{g+1}).$$

If the $\oplus$ operator is the sum operator, then it trivially follows that $\ominus$ is the minus operator. However, if $\oplus$ is the multiplication operator, then using division for the $\ominus$ operator leads to problems if the result of a subfunction is equal to zero. To overcome this problem, two values are maintained in memory instead of just $s^{\texttt{partial}}$: the product of all non-zero subfunctions $p^{\texttt{non-zero}}$, and the number of subfunctions equal to zero $n^{\texttt{zero}}$. The result of a subfunction (a, b) is then added to these values using the operators $\oplus$ and $\ominus$ in the following way:

$$\begin{aligned}(p^{\texttt{non-zero}}, n^{\texttt{zero}}) \oplus (a, b) &= \left(a \cdot p^{\texttt{non-zero}}, n^{\texttt{zero}} + b\right), \\ (p^{\texttt{non-zero}}, n^{\texttt{zero}}) \ominus (a, b) &= \left(\frac{p^{\texttt{non-zero}}}{a}, n^{\texttt{zero}} - b\right),\end{aligned}$$

with each subfunction returning a tuple: $(1, 1)$ if its result is equal to zero, and $\left(f_j^M(\boldsymbol{x}^g|_{\mathcal{I}_j}), 0\right)$ otherwise. The final result of $f(\boldsymbol{x}^{g+1})$ is then equal to:

$$f(\boldsymbol{x}^{g+1}) = \begin{cases} f^P(0) & \text{if } n^{\texttt{zero}} > 0 \\ f^P\left(p^{\texttt{non-zero}}\right) & \text{if } n^{\texttt{zero}} = 0. \end{cases}$$

Partial evaluations can also be applied to more complex functions when the objective function can be written as a set of nested functions of which a subfunction decomposition is known. Moreover, the function f^P could require the calculation of multiple functions to which partial evaluations can be applied, as is the case in the optimization of HDR BT treatment plans, described in Section 4.6. These functions can also be calculated in parallel.

4.2.2. Gray-Box Optimization Algorithms

Partial evaluations have recently been applied to RV-GOMEA [3] and the Multi-Objective RV-GOMEA (MO-RV-GOMEA) [4], achieving excellent scalability on various benchmark problems [3] and real-world problems [2, 17] in a GBO setting. Partial evaluations are well-suited for use with algorithms of the GOMEA family, because the Gene-pool Optimal Mixing (GOM) variation operator applies variation to (small) subsets of variables, and reverts this change if it did not lead to an improved objective value. This selection step means that the change of a small subset of variables must be evaluated, which is done efficiently using partial evaluations.

Current state-of-the-art EAs for real-valued BBO [11, 13] apply variation to all variables of a solution, meaning that partial evaluations provide no benefit by themselves. However, when such algorithms are used in a GBO setting that allows partial evaluations, subfunctions of the evaluation function can still be calculated in parallel. This means that EAs aimed at BBO can still benefit from large-scale parallelization, as discussed in Section 4.4.

4.3. Parallel Co-Processor Architectures

In this section we discuss two frequently used parallel co-processors and their benefits, as these architectures could be used for large-scale parallelization in GBO. We discuss the Intel Xeon Phi and the NVIDIA Pascal architectures, with the former aimed at coarse-grained parallelism, and the latter at fine-grained parallelism.

4.3.1. Intel Xeon Phi

An Intel Xeon Phi is a co-processor consisting of a large number of older, but general purpose, CPUs [14]. Some benefits of an Intel Xeon Phi are that it takes minimal effort to program applications for it, and that a coarse parallelization approach can be used. We specifically use an Intel Xeon Phi of the 5100 series with the Knight's Corner architecture. This co-processor has 60 cores at a base clock speed of 1.05 GHz each.

The Intel Xeon Phi co-processor can be used either in native, or in offload mode. In native mode, any program is executed on the co-processor similar to how it is executed on any multi-core computer. One of the cores of the co-processor executes the serial code, and the remaining cores of the co-processor are used to execute parallel regions of the code. A benefit of using the native mode is that there is no overhead caused by the copying of memory. Furthermore, an application designed for parallel execution on a multi-core computer can be directly executed on the co-processor in native mode.

In offload mode, the serial code of the program is executed on a core of the host machine, and parallel regions in the code can be offloaded to be executed on the co-processor. A benefit of the offload mode is that it can be combined with different parallel co-processors, such as a GPU, because the serial code is executed on the host machine. This comes at the cost of overhead caused by memory transfer between the host and the co-processor device.

4.3.2. NVIDIA Pascal

One of the most recent architectures used by NVIDIA GPUs, succeeding the Maxwell architecture, is the Pascal architecture. The Pascal architecture is used by the NVIDIA Titan X GPU, which we use in this chapter to run experiments.

The NVIDIA Titan X has 12 GB of main device memory, which is completely separated from the main host memory. Any data that has to be accessed on the GPU needs to be copied from the host to the device memory.

The Pascal architecture consists of a large number of Compute Unified Device Architecture (CUDA) cores that perform computations, which are distributed among a number of Streaming Multiprocessors (SMs). Each SM contains a number of CUDA cores, schedulers, cache memory, and registers. Access of the shared memory of an SM is much faster than access of the global GPU memory, so frequently reused data should be stored in shared memory.

Different types of CUDA cores are used for single-precision or double-precision operations. Each SM of the NVIDIA Titan X is equipped with 128 CUDA cores for single-precision, and 4 CUDA cores for double-precision operations. Due to this 32:1 ratio, double-precision operations should be avoided on similar GPUs. The NVIDIA Titan X has 28 SMs, for a total of 3584 single-precision CUDA cores, each at a base clock speed of 1.4 GHz.

GPUs use the Single Instruction Multiple Data (SIMD) computation model. This means that each instruction is applied to a large number of data points in parallel. The CUDA [19] programming language, which we use to program on an NVIDIA GPU, applies the SIMD computation model by using *kernels*, which are functions that are executed by a grid of threads in parallel. Such a grid consists of a large number of blocks, and each block itself is a grid of threads. Each thread executes the same operations applied to different data, depending on the coordinates of the thread and the block it is located in. A block should consist of a multiple of 32 threads, because each SM applies operations to sets of 32 threads, called *warps*, in parallel.

4.4. Parallel Gray-Box Optimization

We distinguish three phases in the application of partial evaluations, because this allows us to identify sections to which a fine-grained parallelization approach can be applied. Given the modification of variable $\boldsymbol{x}_i^g$ of solution $\boldsymbol{x}^g$, resulting in a new solution $\boldsymbol{x}^{g+1}$, these phases are described as follows:

1. *Map*
   - Compute $f_j^M(\boldsymbol{x}^g|_{\mathcal{I}_j})$ and $f_j^M(\boldsymbol{x}^{g+1}|_{\mathcal{I}_j})$ for all $\mathcal{I}_j \ni i$ in parallel.
2. *Reduce*
   - $\Delta s = \bigoplus_{\mathcal{I}_j \ni i} f_j^M(\boldsymbol{x}^{g+1}|_{\mathcal{I}_j}) \ominus \bigoplus_{\mathcal{I}_j \ni i} f_j^M(\boldsymbol{x}^g|_{\mathcal{I}_j})$.
   - $s^{g+1} = s^g \oplus \Delta s$.
3. *Process*
   - $f(\boldsymbol{x}^{g+1}) = f^P\left(s^{g+1}\right)$.

4.4.1. CPU

When a number of CPU cores much smaller than the population size is used, no more can be achieved than the straightforward parallel evaluation of a number of solutions in the population. Therefore, each thread simply performs each phase sequentially for a single solution in the population at a time.

When using the Intel Xeon Phi co-processor, partial evaluations are applied to different solutions in the population in parallel, and any available threads are used for the parallel calculation of subfunctions during the *map* phase.

4.4.2. GPU

On a GPU, each phase is applied in parallel to the entire population $\boldsymbol{P}^g$, and synchronization is required between consecutive phases. The results of the *map* phase are saved in a $(k \times 2n)$ matrix $\boldsymbol{B}$, with k the number of calculated subfunctions. For each modified solution $\boldsymbol{P}_j^{g+1}$ the row $\boldsymbol{B}_{*,2j-1}$ is used to store the results of all subfunctions, and the results for all subfunctions of solution $\boldsymbol{P}_j^g$ are saved in the row $\boldsymbol{B}_{*,2j}$, as displayed in Figure 4.1. The CUDA Thrust library [5], which includes many CUDA utility functions, is then used to calculate the inclusive prefix sum of $\boldsymbol{B}$, denoted $\boldsymbol{B}^{\texttt{pre}}$. Each element $\boldsymbol{B}_{i,j}^{\texttt{pre}}$ contains the sum of elements of $\boldsymbol{B}$ up to and including $\boldsymbol{B}_{i,j}$, given the fact that $\boldsymbol{B}$ is stored as the concatenation of its rows in GPU memory. This means that the sum of a row is equal to the last element of this row minus the last element of the previous row. As a result of this, the value of s_j^{g+1} of $\boldsymbol{P}_j$ can be calculated in constant

$f_1^M\left(\boldsymbol{P}_1^{g+1}\big\rvert_{\mathcal{J}_1}\right)$	$f_2^M\left(\boldsymbol{P}_1^{g+1}\big\rvert_{\mathcal{J}_2}\right)$	$\cdots$	$f_k^M\left(\boldsymbol{P}_1^{g+1}\big\rvert_{\mathcal{J}_k}\right)$
$f_1^M\left(\boldsymbol{P}_1^{g}\rvert_{\mathcal{J}_1}\right)$	$f_2^M\left(\boldsymbol{P}_1^{g}\rvert_{\mathcal{J}_2}\right)$	$\cdots$	$f_k^M\left(\boldsymbol{P}_1^{g}\rvert_{\mathcal{J}_k}\right)$
$f_1^M\left(\boldsymbol{P}_2^{g+1}\big\rvert_{\mathcal{J}_1}\right)$	$f_2^M\left(\boldsymbol{P}_2^{g+1}\big\rvert_{\mathcal{J}_2}\right)$	$\cdots$	$f_k^M\left(\boldsymbol{P}_2^{g+1}\big\rvert_{\mathcal{J}_k}\right)$
$f_1^M\left(\boldsymbol{P}_2^{g}\rvert_{\mathcal{J}_1}\right)$	$f_2^M\left(\boldsymbol{P}_2^{g}\rvert_{\mathcal{J}_2}\right)$	$\cdots$	$f_k^M\left(\boldsymbol{P}_2^{g}\rvert_{\mathcal{J}_k}\right)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$f_1^M\left(\boldsymbol{P}_n^{g+1}\big\rvert_{\mathcal{J}_1}\right)$	$f_2^M\left(\boldsymbol{P}_n^{g+1}\big\rvert_{\mathcal{J}_2}\right)$	$\cdots$	$f_k^M\left(\boldsymbol{P}_n^{g+1}\big\rvert_{\mathcal{J}_k}\right)$
$f_1^M\left(\boldsymbol{P}_n^{g}\rvert_{\mathcal{J}_1}\right)$	$f_2^M\left(\boldsymbol{P}_n^{g}\rvert_{\mathcal{J}_2}\right)$	$\cdots$	$f_k^M\left(\boldsymbol{P}_n^{g}\rvert_{\mathcal{J}_k}\right)$

Figure 4.1: Matrix $\boldsymbol{B}$, containing the results of the *map* phase for the new population $\boldsymbol{P}^{g+1}$ and its previous state $\boldsymbol{P}^{g}$.

time through:

$$s_j^{g+1} = s_j^g \oplus \left(\boldsymbol{B}_{k,2j-1}^{\mathtt{pre}} \ominus \boldsymbol{B}_{k,2j-2}^{\mathtt{pre}}\right) \ominus \left(\boldsymbol{B}_{k,2j}^{\mathtt{pre}} \ominus \boldsymbol{B}_{k,2j-1}^{\mathtt{pre}}\right).$$

4.4.3. HYBRIDIZATION

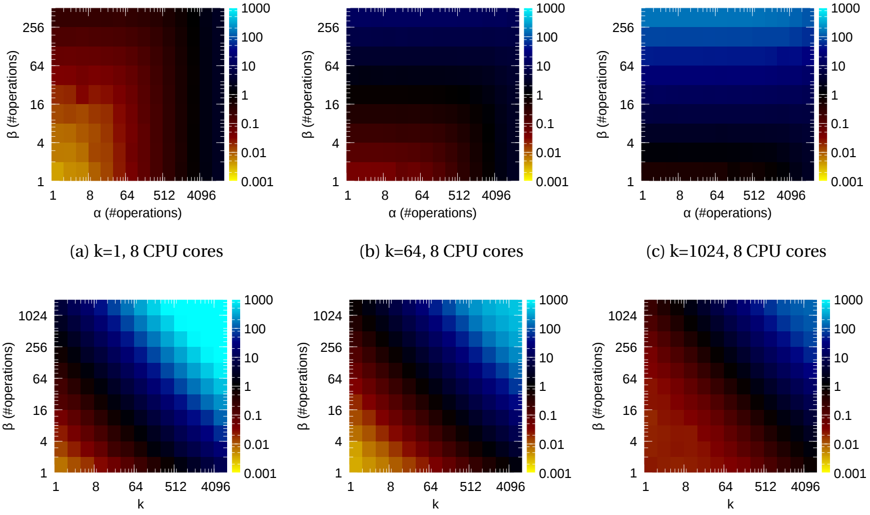
It is important to realize that the best performing parallelization technique can be different for partial evaluations of different complexities. Because of this, we write the time required for a partial evaluation of k subfunctions as $T = \alpha + k\beta$, where α indicates the complexity of the *process* phase, and β indicates the complexity of the *map* and *reduce* phases, assuming that the complexity of each subfunction is roughly identical.

Based on k, α and β, the most appropriate parallelization technique should be selected at different points throughout optimization. For example, because a linkage tree is used in the optimization of HDR BT with MO-RV-GOMEA [17], variation of any parent solution can consist of the modification of any number between 1 and ℓ variables. In this optimization problem, the number of modified variables determines the complexity of each subfunction β.

Criteria that determine the optimal parallelization technique can be determined prior to optimization based on the required computation time of a partial evaluation for various values of k. Based on these criteria, a hybrid parallelization approach can be used during optimization, meaning that a different parallelization technique is selected for partial evaluations of different complexities.

4.5. EXPERIMENTS

We compare different parallelization techniques for applying partial evaluations to a population of solutions, given different settings for the number of parallel subfunctions of the evaluation function, and the required computation time of each subfunction. We compare the use of 8 CPU cores to an Intel Xeon Phi, and an NVIDIA Titan X GPU. The native mode was used for all experiments with the Intel Xeon Phi, because preliminary experiments showed that the Intel Xeon Phi offload mode was never superior to the



(a) k=1, 8 CPU cores

(b) k=64, 8 CPU cores

(c) k=1024, 8 CPU cores

(d) 1 CPU core, $\alpha = 0$

(e) 8 CPU cores, $\alpha = 0$

(f) Intel Xeon Phi, $\alpha = 0$

Figure 4.2: Speed-ups achieved by the GPU compared to other parallelization techniques, for different values of the number of subfunctions k and the subfunction complexity indicators α and β.

GPU. This is caused by the fact that both techniques require memory transfer to the device, while the GPU has more computational power.

4.5.1. Experimental Set-Up

Each experiment consists of the partial evaluation of a population of n solutions consisting of ℓ real-valued variables represented by single-precision (32 bit) floating point numbers. Each partial evaluation requires the calculation of k independent subfunctions, and each subfunction has a complexity β. subfunction complexity β is presented as a number of operations c, meaning that the arbitrary function $i^{3.14}$ for $i \in \{1,2,3,\ldots,c\}$ was performed to artificially increase the subfunction complexity.

All experiments using an Intel Xeon Phi co-processor are performed on a computer with a 5100 series Intel Xeon Phi using the Knight's Corner architecture, having 60 cores at 1.05 GHz. All remaining experiments are performed on a computer with 10 Intel Xeon CPU E5-2630 at 2.2 GHz, and an NVIDIA Titan X GPU. Parallelization on multiple CPU cores is implemented in OpenMP. Each reported result is the average of 1000 independent runs.

4.5.2. Experimental Results

In Figures 4.2a to 4.2c we show speed-ups achieved by the GPU compared to 8 CPU cores, given different settings for k, α, and β. The color of each block indicates the speed-up given the parameters of the block's bottom-left corner. A population size of $n = 128$ is used, as preliminary experiments showed that achievable speed-ups were similar among different settings for the population size.

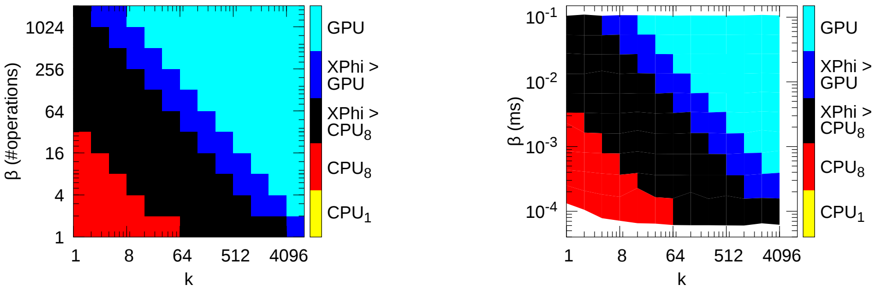


Figure 4.3: Best parallelization technique for different values of the number of subfunctions k and their complexity β.

Figures 4.2a to 4.2c show that the effect of increasing α is only apparent for small values of k. Furthermore, for $k = 1$ and $k = 64$, an increase in α still leads to improved relative performance of the GPU, due to its larger number of cores. Despite this, an effective slow-down is observed due to overhead caused by memory transfer, and the GPU being unable to fully utilize its processing power. Finally, Figure 4.2c shows that a sufficiently large β is required to achieve any speed-up, caused by the overhead of copying memory to the GPU and back.

Figures 4.2d to 4.2f show the speed-ups achieved by the GPU compared to 1 CPU core, 8 CPU cores, and the Intel Xeon Phi, for a population size of $n = 128$, and $\alpha = 0$. We clearly see that the GPU performs very well at computing a large number of subfunctions with a high complexity. For 4096 subfunctions with $\beta = 1024$, the GPU achieved a speed-up of a factor 2803 compared to a single CPU core, a speed-up of a factor 446 compared to 8 CPU cores, and a speed-up of a factor 121 compared to the Intel Xeon Phi.

Figure 4.3 shows the best performing parallelization technique for different settings of the number of subfunctions and their complexity β, with CPU$_m$ denoting parallel execution on m CPU cores, and XPhi denoting the Intel Xeon Phi. The left figure displays the subfunction complexity in terms of the number of operations, and the right figure displays the subfunction complexity in terms of the average computation time of a subfunction, which is calculated by dividing the average required computation time of 1 CPU core by kn. In scenarios where the Intel Xeon Phi resulted in the best performance, the second best performing parallelization technique is also displayed, as this gives more insight that can be used for a hybrid approach using only a GPU and a number of CPU cores.

4.5.3. DISCUSSION

The optimal technique for the parallelization of a partial evaluation clearly depends on factors including the number of subfunctions and their computational complexities. Figure 4.3 shows that the GPU performs very well at problems with a large number of subfunctions. In the largest-scale setting of this figure, with $k = 4096$ and $\beta = 0.1$ ms, the serial evaluation of one solution still only takes 0.4 seconds for small values of α. Performing a reasonable number of 10^5 evaluations of this complexity then takes roughly 11 hours, which could be reduced to a time of less than a minute by using

Table 4.1: Clinical protocol for HDR BT at the AMC, with dose thresholds defined as a percentage of the planning-aim dose of 13 Gy.

Prostate	Bladder	Rectum	Urethra	Vesicles
$V_{100} > 95\%$	$D_{1cm3} < 86\%$	$D_{1cm3} < 78\%$	$D_{0.1cm3} < 110\%$	$V_{80} > 95\%$
$V_{150} < 50\%$	$D_{2cm3} < 74\%$	$D_{2cm3} < 74\%$		
$V_{200} < 20\%$				

large-scale parallelization on a GPU, assuming little global memory access is required. We therefore see a lot of potential in the use of a GPU for the large-scale parallelization of real-world problems in GBO settings.

4.6. BRACHYTHERAPY

We now consider the optimization of BT treatment plans as described in [17]. Recently, experimental results showed that MO-RV-GOMEA [4] using partial evaluations achieved the best performance among a set of state-of-the-art MOEAs when using a single CPU. Furthermore, the formulation of this real-world problem seems well-suited for large-scale parallelization.

BT is a type of cancer treatment where cancerous tissue is irradiated from inside the body of the patient. We specifically address the HDR BT treatment for prostate cancer. This type of treatment consists of the insertion of 14-20 catheters into the body of a patient through the transperineal skin. These catheters are placed under general anaesthetics into the prostate and the seminal vesicles, which are the target volumes of this treatment. The Organs At Risk (OARs) surrounding the prostate, i.e., the bladder, rectum, and urethra, should receive as little radiation as possible. Radiation is delivered to the target volumes by passing a radioactive source through the catheters, and stopping this radioactive source at certain *dwell positions* for a certain amount of time, i.e., the *dwell time*. The longer a certain dwell time, the higher the delivered radiation dose to the tissue surrounding the dwell position.

In order to create a treatment plan, Magnetic Resonance Imaging (MRI) scans are taken of the patient after catheter placement. A clinician then contours the target volumes and the OARs, and specifies the locations of the catheters on the MRI scans. As a treatment plan is uniquely defined by its dwell times, these dwell times are often optimized using inverse planning algorithms [12]. At our collaborative medical center, the Academic Medical Center (AMC) in Amsterdam, this is followed by manual optimization by a clinician.

The quality of a treatment plan is assessed by visual inspection of the dose distribution, and in terms of a clinical protocol. A clinical protocol includes a set of planning criteria that describe clinically acceptable upper/lower bounds for the radiation dose delivered to the target volumes and the OARs. The clinical protocol at the AMC consists of a set of planning criteria, as listed in Table 4.1, defined in terms of a dose-volume-index (DVI) and a threshold.

The volume-index V_d^o is the volume of the organ o that receives at least d% of the planning-aim dose. For example, the planning criterion $V_{100}^{\text{prostate}} > 95\%$ indicates that at least 95% of the prostate volume should receive at least 100% of the planning-aim dose. A dose-index D^o_v specifies the minimum dose delivered to the most irradiated subvolume v of organ o. For example, the planning criterion $D^{\text{bladder}}_{1cm3} < 86\%$ indicates that the minimum dose in the most irradiated 1 cm^3 of the bladder should be less than 86% of the planning-aim dose.

DVIs of a treatment plan $\boldsymbol{t}$ are estimated with Monte Carlo sampling, using a vector $\boldsymbol{d}$ describing the dose received by the set of *dose calculation points*. Increasing the number of dose calculation points leads to more accurate estimation of DVIs, but increases computation time. The total dose received at a certain dose calculation point is the sum of the dose received from each dwell position. A pre-processing step is performed to compute a *dose-rate matrix* $\boldsymbol{R}$ [18], where $\boldsymbol{R}_{i,j}$ defines the dose contribution (in Gy/s) of the dwell position corresponding to $\boldsymbol{t}_i$, to the dose calculation point $\boldsymbol{d}_j$. As such, $\boldsymbol{d}$ is calculated as $\boldsymbol{d} = \boldsymbol{R}\boldsymbol{t}$.

A dose-index D^o_v is estimated by first sorting the points in $\boldsymbol{d}$ that are inside organ o in descending order, denoted as $\boldsymbol{d}^{s,o}$. The dose-index D^o_v is then equal to the dose point with index $\lfloor v/v^o \rfloor$ in $\boldsymbol{d}^{s,o}$, i.e., $D^o_v = \boldsymbol{d}^{s,o}_{\lfloor v/v^o \rfloor}$, where v^o is the volume corresponding to each point in organ o. A volume-index V^o_d is estimated by calculating the fraction of points in o that receive a dose of at least d. Given $\boldsymbol{d}$, the calculation of a dose-index therefore has time complexity $\mathcal{O}(|d|\log|d|)$, and the calculation of a volume-index has time complexity $\mathcal{O}(|d|)$.

4.6.1. TREATMENT PLAN OPTIMIZATION

Because the OARs are in close proximity of the target volumes, with the urethra even passing right through the prostate, there is a clear trade-off between covering the target volumes and sparing the OARs. The aim of HDR BT treatment planning is then to find dwell times such that an acceptable trade-off between target coverage and sparing is achieved. Multi-objective EAs are among the state-of-the-art for multi-objective optimization [10], and they result in a large set of solutions having high-quality trade-offs between the objectives of interest. From this set of solutions, the most appropriate treatment plan can then be selected *a posteriori* by a clinician. Among other reasons, this is why a multi-objective optimization approach was used in [17]. However, such an approach is relatively time-consuming, while limited time is available in clinical practice.

Because a set of dwell times uniquely defines a treatment plan, each solution in the population is described by a vector of dwell times $\boldsymbol{t}$ of length ℓ. The planning criteria $V^{\text{prostate}}_{150} < 50\%$, and $V^{\text{prostate}}_{200} < 20\%$ are used as hard constraints for the optimization, i.e., treatment plans not satisfying these criteria are considered infeasible. The remaining set of planning criteria is condensed into two objectives: the Least Coverage Index (LCI) and the Least Sparing Index (LSI), indicating how close the DVIs of a treatment plan are to the thresholds defined by the clinical protocol. LCI and LSI are defined such that positive values indicate that all planning criteria are satisfied with regard to coverage and sparing, respectively.

The LCI of a treatment plan $\boldsymbol{t}$ is defined as

$$\begin{aligned}\mathrm{LCI}(\boldsymbol{t}) &= \min\left\{\delta_v(V_{100}^{\text{prostate}}), \delta_v(V_{80}^{\text{vesicles}})\right\},\\ \delta_v(V_d^o) &= V_d^o - V_d^{o,\min},\end{aligned}$$

with $V_d^{\text{o,min}}$ the threshold defined by the clinical protocol.

The LSI of a treatment plan $\boldsymbol{t}$ is defined as

$$\begin{aligned}\mathrm{LSI}(\boldsymbol{t}) = \min\Big\{&\delta_d(D_{1cm3}^{\text{bladder}}), \delta_d(D_{2cm3}^{\text{bladder}}),\\ &\delta_d(D_{1cm3}^{\text{rectum}}), \delta_d(D_{2cm3}^{\text{rectum}}), \delta_d(D_{0.1cm3}^{\text{urethra}})\Big\},\\ \delta_d(D_v^o) = D_v^{o,\max} - D_v^o,&\end{aligned}$$

with $D_v^{\text{o,max}}$ the threshold defined by the clinical protocol.

Partial evaluations can be used to efficiently update $\boldsymbol{d}$ if only a subset of elements of $\boldsymbol{t}$ is modified. Given a dose distribution and a change in dwell times $\Delta\boldsymbol{t}$, the effect of this change in dwell times, resulting in $\boldsymbol{d}'$, can be efficiently computed as

$$\begin{aligned}\boldsymbol{d}' &= \boldsymbol{d} + \Delta\boldsymbol{d}\\ &= \boldsymbol{d} + \boldsymbol{R}\Delta\boldsymbol{t}.\end{aligned}$$

For each element $\Delta\boldsymbol{t}_i = 0$, the column $\boldsymbol{R}_{i,*}$ can be skipped in the calculation of $\Delta\boldsymbol{d}$. This means that $\Delta\boldsymbol{d}$ can be calculated in $\mathcal{O}(k|d|)$ time given a vector $\Delta\boldsymbol{t}$ with k non-zero elements, as opposed to $\mathcal{O}(\ell|d|)$ for the calculation of $\boldsymbol{d}$. For each solution in the population, the dose distribution $\boldsymbol{d}$ is maintained in memory, allowing it to be updated efficiently. This requires $\mathcal{O}(n|d|)$ memory for a population of size n.

4.6.2. GBO Parallelization

We now apply large-scale GBO parallelization, as introduced in Section 4.4, to HDR BT treatment plan optimization. In this application, the calculation of the two objectives, LCI and LSI, relies on the intermediate step of the calculation of $\boldsymbol{d}$. We focus on the parallelization of the calculation of $\boldsymbol{d}$, because this is the part of the evaluation where partial evaluations can be applied, and where subfunctions can be identified.

As discussed in Subsection 4.6.1, partial evaluations can be applied by calculating $\Delta\boldsymbol{d}$ and adding it to $\boldsymbol{d}$. The calculation of each element $\boldsymbol{d}_j$ can be considered a function to which partial evaluations can be applied, described as a sum of subfunctions, as follows:

$$\boldsymbol{d}_j = \sum_{i=0}^{\ell-1} \boldsymbol{R}_{i,j}\boldsymbol{t}_i.$$

As such, the number of partial evaluations that can be computed in parallel is equal to $|d|$, where the complexity of one partial evaluation of k variables is equal to $\mathcal{O}(k)$. Given the fact that all solutions in the population can be evaluated in parallel, this leads to a total of $n|d|$ operations that can be performed in parallel.

In the following way, the phases described in Section 4.4 are applied to HDR BT treatment plan optimization:

1. *Map*
   - In parallel for each $\Delta \boldsymbol{t}_i \neq 0$, calculate $\boldsymbol{R}_{i,j} \Delta \boldsymbol{t}_i$.
2. *Reduce*
   - In parallel for each $\boldsymbol{d}_j$, calculate $\sum_{\Delta \boldsymbol{t}_i \neq 0} \boldsymbol{R}_{i,j} \Delta \boldsymbol{t}_i$.
3. *Process*
   - $\boldsymbol{d}' = \boldsymbol{d} + \Delta \boldsymbol{d}$
   - Calculation of DVIs; time complexity $\mathcal{O}(|d| \log |d|)$.

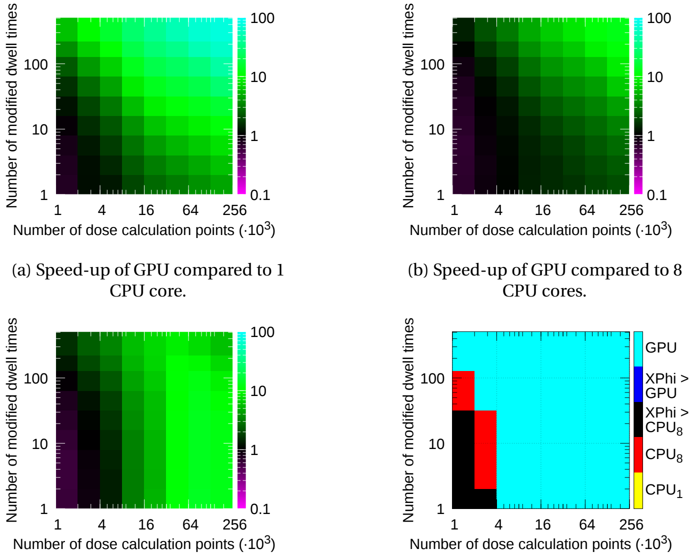
The above phases are only synchronized between all solutions in the population when GPU parallelization is used. Note that the combined *map* and *reduce* phases come down to the matrix-vector multiplication $\boldsymbol{R}\Delta\boldsymbol{t}$. Due to the synchronization on the GPU, this can be done in parallel for the entire population given the matrix $\boldsymbol{T}$, with $\boldsymbol{T}_{*,k}$ the vector $\Delta\boldsymbol{t}$ for solution $\boldsymbol{P}_k$, resulting in the single matrix-matrix multiplication $\boldsymbol{RT}$ being sufficient for the *map* and *reduce* phases of the entire population.

4.6.3. EXPERIMENTS

We now compare different parallelization techniques applied to partial evaluations with different complexities, depending on the number of modified dwell times and the number of dose calculation points, for HDR BT treatment plan optimization. Typically, 1000 dose calculation points are used for inverse planning approaches [15], and 20000 points are used for multi-objective optimization [17]. Because the *process* phase is relatively time consuming, GPU parallelization is always applied to this phase when possible. This is however not possible when the Intel Xeon Phi is used in native mode, because all computation is done on the Intel Xeon Phi, and the host machine that can also host a GPU, is not directly used.

In Figures 4.4a to 4.4c we show the speed-ups achieved by the GPU compared to the three other parallelization techniques for a population size of 112. This population size was selected, because a population size of approximately 100 was found to be sufficient for the optimization, a population size divisible by 8 is preferred for the parallelization on 8 cores, and a population size divisible by 16 is preferred for GPU parallelization, due to the dimensions of the grid discussed in Section 4.3.

We observe that the GPU performs slightly worse than both 1 CPU core and 8 CPU cores for the setting of 1000 dose calculation points and 1 modified dwell time, by factors 0.74 and 0.68 respectively. For 128000 dose calculation points and 220 modified dwell times, the GPU achieves a speed-up of a factor 73 compared to a single CPU core, and a speed-up of a factor 12 compared to 8 CPU cores. A decrease in speed-up compared to the Intel Xeon Phi appears for a large number of dose calculation points and modified number of dwell times. This is caused by the fact that the *process* phase is not performed on a GPU when the Intel Xeon Phi is used, leading to a relatively slow performance of the Intel Xeon Phi when a large number of dose calculation points is used, but a small number of dwell times is modified.



(a) Speed-up of GPU compared to 1 CPU core.

(b) Speed-up of GPU compared to 8 CPU cores.

(c) Speed-up of GPU compared to the Intel Xeon Phi.

(d) Best performing parallelization technique.

Figure 4.4: Results on the parallelization of HDR BT partial evaluations, showing speed-ups achieved by the GPU and the best parallelization technique given different parameter settings. A different color scheme is intentionally used, because speed-ups are shown on a different scale than before.

The best performing parallelization techniques for different settings are displayed in Figure 4.4d, with CPU$_n$ denoting parallel execution on n CPU cores, and XPhi denoting the Intel Xeon Phi. We observe that the Intel Xeon Phi performs best for a small number of dose calculation points and modified dwell times, but is quickly overtaken by the GPU when over 4000 points are used.

4.6.4. Discussion

We observed that the GPU is only outperformed by different parallelization techniques when up to 2000 dose calculation points are used. However, it was previously shown that high-quality treatment plans were unable to be found when only 4000 dose calculation points were used [17]. This shows that GPU parallelization is very well suited for HDR BT treatment plan optimization.

4.7. Conclusions

We have introduced an approach for the large-scale parallelization of partial evaluations in GBO, for example using a GPU or a many-core architecture such as the Intel Xeon Phi, which evaluates a large number of subfunctions of the evaluation function in parallel. This approach is aimed at real-world problems with time-consuming evaluations to which GBO can be applied. We analyzed the performance of the various parallel co-processor architectures in a large number of scenarios depending on, for example, the number of subfunctions of a partial evaluation that can be computed in parallel, and the complexity of such subfunctions. We note that, if partial evaluations are possible, EAs can likely benefit from hybrid parallelization approaches, where the most appropriate parallelization approach is selected for various parts of the optimization process. Furthermore, we conclude that large-scale parallelization is likely to be applicable to time-consuming real-world problems that can be solved in a GBO setting. Applying GPU parallelization resulted in speed-ups of factors over 2000 in the most favorable scenarios of our experiments.

We have applied the large-scale parallelization model to HDR BT treatment plan optimization, and we have found that parallelization on a GPU outperforms other parallelization techniques in all scenarios with realistic parameter settings, resulting in speed-ups of the partial evaluations by up to a factor of 73. These are very promising results that bring us closer to the application of multi-objective optimization of HDR BT treatment plans in a time-constrained clinical setting.

REFERENCES

[1] Bouter, A., Alderliesten, T., Bel, A., Witteveen, C., and Bosman, P. A. N. (2018). Large-scale parallelization of partial evaluations in evolutionary algorithms for real-world problems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1199–1206. ACM.

[2] Bouter, A., Alderliesten, T., and Bosman, P. A. N. (2017a). A novel model-based evolutionary algorithm for multi-objective deformable image registration with content mismatch and large deformations: benchmarking efficiency and quality. In Styner, M. A. and Angelini, E. D., editors, *Medical Imaging 2017: Image Processing*, volume 10133, pages 304 – 311. International Society for Optics and Photonics, SPIE.

[3] Bouter, A., Alderliesten, T., Witteveen, C., and Bosman, P. A. N. (2017b). Exploiting linkage information in real-valued optimization with the real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 705–712. ACM.

[4] Bouter, A., Luong, N. H., Witteveen, C., Alderliesten, T., and Bosman, P. A. N. (2017c). The multi-objective real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 537–544. ACM.

[5] Bell, N. and Hoberock, J. (2011). Thrust: A productivity-oriented library for CUDA. *GPU Computing Gems Jade Edition*, 2:359–371.

[6] Bosman, P. A. N. and Thierens, D. (2011). The roles of local search, model building and optimal mixing in evolutionary algorithms from a BBO perspective. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 663–670. ACM.

[7] Brodtkorb, A. R., Hagen, T. R., and Sætra, M. L. (2013). Graphics processing unit (GPU) programming strategies and trends in GPU computing. *Journal of Parallel and Distributed Computing*, 73(1):4–13.

[8] Chicano, F., Whitley, D., Ochoa, G., and Tinós, R. (2017). Optimizing one million variable nk landscapes by hybridizing deterministic recombination and local search. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 753–760. ACM.

[9] D'Agostino, D., Pasquale, G., and Merelli, I. (2014). A fine-grained CUDA implementation of the multi-objective evolutionary approach NSGA-II: Potential impact for computational and systems biology applications. In *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, pages 273–284. Springer.

[10] Deb, K. (1999). Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3):205–230.

[11] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

[12] Dinkla, A. M., van der Laarse, R., Kaljouw, E., Pieters, B. R., Koedooder, K., van Wieringen, N., and Bel, A. (2015). A comparison of inverse optimization algorithms for hdr/pdr prostate brachytherapy treatment planning. *Brachytherapy*, 14(2):279–288.

[13] Hansen, N., Müller, S., and Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18.

[14] Jeffers, J., Reinders, J., and Sodani, A. (2016). *Intel Xeon Phi Processor High Performance Programming: Knights Landing Edition*. Morgan Kaufmann.

[15] Lessard, E. and Pouliot, J. (2001). Inverse planning anatomy-based dose optimization for HDR-brachytherapy of the prostate using fast simulated annealing algorithm and dedicated objective function. *Medical Physics*, 28(5):773–779.

[16] Li, S.-C. and Yu, T.-L. (2017). Speeding up DSMGA-II on CUDA platform. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 809–816. ACM.

[17] Luong, N. H., Alderliesten, T., Bel, A., Niatsetski, Y., and Bosman, P. A. N. (2018). Application and benchmarking of multi-objective evolutionary algorithms on high-dose-rate brachytherapy planning for prostate cancer treatment. *Swarm and Evolutionary Computation*, 40:37–52.

[18] Nath, R., Anderson, L. L., Luxton, G., Weaver, K. A., Williamson, J. F., and Meigooni, A. S. (1995). Dosimetry of interstitial brachytherapy sources: Recommendations of the AAPM radiation therapy committee task group no. 43. *Medical Physics*, 22(2):209–234.

[19] NVIDIA Corporation (2018). CUDA C programming guide v9.1.85.

[20] Ortega, G., Filatovas, E., Garzón, E. M., and Casado, L. G. (2017). Non-dominated sorting procedure for Pareto dominance ranking on multicore CPU and/or GPU. *Journal of Global Optimization*, 69(3):607–627.

[21] Thierens, D. and Bosman, P. A. N. (2011). Optimal mixing evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 617–624. ACM.

[22] Tintos, R., Whitley, D., and Chicano, F. (2015). Partition crossover for pseudo-boolean optimization. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*, pages 137–149. ACM.

[23] Whitley, D. (2017). Next generation genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 922–941. ACM.

[24] Wong, M. L. and Wong, T. T. (2009). Implementation of parallel genetic algorithms on graphics processing units. *Intelligent and Evolutionary Systems*, 187:197–216.

5

GPU-Accelerated Bi-Objective Treatment Planning for Prostate High-Dose-Rate Brachytherapy

This chapter is based on the following publication: Bouter, A., Alderliesten, T., Pieters, B. R., Bel, A., Niatsetski, Y., and Bosman, P. A. N. (2019). GPU-accelerated bi-objective treatment planning for prostate high-dose-rate brachytherapy. *Medical Physics*, 46(9):3776–3787

Abstract

The purpose of this study is to improve upon a recently introduced bi-objective treatment planning method for prostate HDR brachytherapy, both in terms of resulting plan quality and run time requirements, to the extent that its execution time is clinically acceptable. Bi-objective treatment planning is done using a state-of-the-art multi-objective evolutionary algorithm, which produces a large number of potential treatment plans with different trade-offs between coverage of the target volumes and sparing organs at risk. A Graphics Processing Unit (GPU) is used for large-scale parallelization of dose calculations and the calculation of the Dose-Volume (DV) indices of potential treatment plans. Moreover, the objectives of the previously used bi-objective optimization model are modified to produce better results. We applied the GPU-accelerated bi-objective treatment planning method to a set of 18 patients, resulting in a set containing a few hundred potential treatment plans with different trade-offs for each of these patients. Due to accelerations introduced in this chapter, results previously achieved after 1 hour are now achieved within 30 seconds of optimization. We found plans satisfying the clinical protocol for 15 out of 18 patients, whereas this was the case for only 4 out of 18 clinical plans. Higher quality treatment plans are obtained when the accuracy of DV index calculation is increased using more dose calculation points, requiring still no more than 3 minutes of optimization for 100,000 points. We conclude that large sets of high-quality treatment plans that trade off coverage and sparing are now achievable within 30 seconds, due to the GPU-acceleration of a previously introduced bi-objective treatment planning method for prostate HDR brachytherapy. Higher quality plans can be achieved when optimizing for 3 minutes, which we still consider to be clinically acceptable. This allows for more insightful treatment plan selection in a clinical setting.

5.1. INTRODUCTION

Treatment planning for prostate High-Dose-Rate (HDR) brachytherapy (BT) [21] consists of finding a specific dwell time for each dwell position, such that a desirable dose distribution is achieved. The main difficulty of treatment planning comes from the fact that the target volume(s) should receive sufficient dose, while the surrounding Organs At Risk (OARs) should receive as little dose as possible. For this reason, some form of automated optimization is often used to find a set of dwell times such that all clinical criteria are satisfied, or are as close as possible to being satisfied. Such clinical criteria are generally defined in terms of Dose-Volume (DV) planning criteria (see Section 5.2.3).

Despite the fact that a treatment plan is often evaluated based on its DV indices, inverse planning methods frequently used in clinical practice do not directly optimize the DV indices [14, 16]. Instead, for the purpose of efficiency, a penalty-based approach is used, which aims to minimize the sum of penalty values of all Dose Calculation (DC) points, with a penalty value assigned to a DC point when the dose at that point is outside the range of acceptable dose values. This reformulated problem is then optimized using, e.g., simulated annealing [16], linear programming [1], or the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [14]. It was previously shown that such a penalty-based problem formulation weakly correlates with the quality of DV indices [11]. The optimal solution to a penalty-based problem formulation may not satisfy all clinical criteria, even if such a treatment plan does exist for the respective patient. Furthermore, finding optimal patient-specific weights for a penalty-based problem formulation is an arguably unintuitive and labor-intensive process.

Several studies have looked into directly optimizing DV indices using linear programming [28], mixed-integer linear programming [11], simulated annealing [10], or evolutionary algorithms [18]. These approaches generally solve a relaxed version of the problem, or use a small number of dose calculation points, because solving plans to optimality directly using DV indices is known to take far too long for clinical practice [10], necessitating state-of-the-art modern heuristics.

A core aspect of the difficulty of brachytherapy planning is caused by the inherent trade-off between tumor coverage and organ sparing, i.e., any increase in dose to the target volumes also leads to an increase in dose to the OARs.

In order to gain more insight into the trade-off between coverage and sparing, a bi-objective treatment planning model and optimization method was previously introduced [18]. Multi-Objective Evolutionary Algorithms (MOEAs) are known to be among the state of the art when dealing with problems involving 2 or 3 conflicting objectives [9]. MOEAs are capable of finding large sets of potential solutions, with each solution having a different trade-off between the objectives of interest, i.e., tumor coverage and organ sparing in this scenario. The optimization of a set of treatment plans allows for more insightful treatment plan selection, as it is directly clear how much sparing needs to be sacrificed in order to improve the tumor coverage by a certain amount. This can be especially useful in cases where it is not possible to satisfy all criteria defined in the clinical protocol, e.g., due to suboptimal catheter placement, making it more difficult to find a desirable trade-off between coverage and sparing.

The aforementioned bi-objective treatment planning method uses a novel MOEA, named MO-RV-GOMEA [4], as it performed better than other state-of-the-art MOEAs [18]. Moreover, in a recent observer study for 18 prostate cancer patients with 3 physicians as observers, plans resulting from this approach were found to be preferable compared to plans resulting from the current clinical workflow at our clinic [20]. However, this treatment planning method requires approximately one hour of computation time on a single CPU core, exceeding the acceptable amount of computation time in clinical practice.

In this chapter, we improve the quality of plans found by the previously introduced bi-objective treatment planning method [18] by improving the optimization model and method, and we accelerate the method to the extent that its execution time is clinically acceptable. Specifically, we apply large-scale parallelization on a Graphics Processing Unit (GPU). By doing so, the method is sped up so much that it can even be used to intrinsically improve the approach by increasing the number of so-called dose calculation points used for calculating the DV indices, decreasing the effect of overfitting and increasing the quality of treatment plans.

5.2. Materials and Methods

5.2.1. Data

A dataset of 18 patients with intermediate- and high-risk prostate cancer, previously treated at the Amsterdam UMC in Amsterdam, The Netherlands, is used for all experiments in this chapter. These patients were treated between February 2015 and April 2017 with external beam radiotherapy on the prostate and base of the seminal vesicles to a dose of 44 Gy in daily fractions of 2.2 Gy followed by a single dose of 13 Gy HDR brachytherapy on the prostate. A median of 16 (range: 14-20) catheters were implanted, resulting in a median of 413 (range: 250-668) dwell positions. Catheter reconstruction and contouring of Regions Of Interest (ROIs) was done on three orthogonal pelvic T2-weighted turbo spin echo MRI (Ingenia 3T Philips Healthcare, Best, The Netherlands) scans with a resolution of 0.52 × 0.52 mm, and a slice thickness of 3.0 mm with a 0.3 mm gap. Three interpolated contours were added between each contoured slice of each ROI. The clinical plans that these patients were treated with were obtained through optimization with IPSA [16] or HIPO [14] using a standard set of objectives, followed by graphical optimization.

5.2.2. Clinical Protocol

In clinical practice, the quality of a treatment plan is often assessed in terms of the DV histogram or a set of DV indices, and a visual inspection of the dose distribution. At the Amsterdam UMC in Amsterdam, The Netherlands, treatment plans were evaluated based on a clinical protocol consisting of a set of aspiration values for various DV indices. A dose index D_v^o defines the dose received by the most irradiated subvolume v of ROI o. A volume index V_d^o defines the subvolume of ROI o receiving a dose of at least d. Any amount of dose is expressed as a percentage of the Planning-aim Dose (PD). The clinical protocol for HDR prostate BT at the AMC is given in Table 5.1.

Table 5.1: Clinical protocol for HDR prostate BT at the AMC. Aspiration values are defined as a percentage of the PD of 13 Gy.

Prostate	Bladder	Rectum	Urethra	Seminal vesicles
$V_{100} > 95\%$	$D_{1cm3} < 86\%$	$D_{1cm3} < 78\%$	$D_{0.1cm3} < 110\%$	$V_{80} > 95\%$
$V_{150} < 50\%$	$D_{2cm3} < 74\%$	$D_{2cm3} < 74\%$		
$V_{200} < 20\%$				
$D_{90} > 100\%$				

5.2.3. DV INDEX ESTIMATION

DV indices are generally estimated using a randomly generated (Monte Carlo-sampling) or a regularly distributed set of DC points [15]. When using Monte Carlo-sampling, ROIs are represented by a large set of DC points $\boldsymbol{d}$, with $\boldsymbol{d}_i$ describing the total dose received by the DC point with index i. The subset of DC points in an ROI o is defined as $\boldsymbol{d}^o$. In this work, an equal number of DC points is sampled in each ROI.

A volume index V_d^o is estimated by calculating the fraction of DC points in ROI o that receives at least a dose d, i.e.,

$$V_d^o = \frac{|\{\boldsymbol{d}_i \in \boldsymbol{d}^o | \boldsymbol{d}_i \geq d\}|}{|\boldsymbol{d}|}.$$

A dose index D_v^o is estimated by first selecting the subset of DC points in o that receives the highest dose, such that the equivalent volume of this set of points is equal to v. The value of D_v^o is then equal to the minimum dose received by any DC point in this set. Given the sequence $\boldsymbol{d}^{s,o}$, describing the sequence $\boldsymbol{d}^o$ in descending order, D_v^o is defined as

$$D_v^o = \frac{\boldsymbol{d}^{s,o}_{\lfloor v/v^o \rfloor}}{PD}.$$

For an accurate estimation of DV indices, using a sufficiently large set of DC points is necessary [15]. This is especially important when DV indices are directly optimized using automated treatment planning, as this approach is prone to overfitting. This means that DV indices optimized using a certain set of DC points will always become worse when recalculated using a larger, more representative, set of DC points. The larger the set of DC points used for optimization, the more accurate the DV index calculation, and the smaller the effect of overfitting. Using a larger set of DC points does however increase the computational effort to estimate DV indices, making the optimization procedure more time consuming. Since the dose in all DC points can be calculated independently, the efficiency of DV index estimation can be greatly increased by the use of a GPU.

In this chapter, the reported DV indices of any optimized plan are recalculated after optimization on a set of 500,000 independently sampled DC points, identical to the standard setting in Oncentra Brachy (version 4.5, Elekta AB, Stockholm, Sweden).

In Section 5.3.1, we show how the number of DC points affects treatment plan quality of the automated treatment planning method used in this chapter.

5.2.4. Bi-Objective Treatment Planning Model

Previously, we proposed a bi-objective optimization approach [18, 19] using two objectives that are directly defined in terms of the DV indices of a treatment plan. Dwell times of all activated dwell positions, those within a target volume plus 5 mm margin and outside of the urethra plus 1 mm margin, are optimized. This optimization approach was previously validated in a retrospective observer study [20], showing that plans produced by solving this bi-objective model with the multi-objective evolutionary algorithm GOMEA [4] were preferred over the clinically used plan in all cases.

The two objectives of the bi-objective optimization model, the Least Coverage Index (LCI) and Least Sparing Index (LSI), quantify how much the DV indices of a treatment plan deviate from the aspiration values specified in the clinical protocol, with regard to tumor coverage and organ sparing, respectively. Positive values for either LCI or LSI indicate that the clinical protocol has been satisfied with regard to coverage or sparing, respectively.

Given the protocol in Table 5.1, the LCI and LSI of a treatment plan $\boldsymbol{t}$ are defined as

$$\begin{aligned}
\mathrm{LCI}(\boldsymbol{t}) &= \min\left\{\delta_c(V_{100}^{\text{prostate}}), \delta_c(V_{80}^{\text{vesicles}})\right\},\\
\mathrm{LSI}(\boldsymbol{t}) &= \min\left\{\delta_s(D_{1cm3}^{\text{bladder}}), \delta_s(D_{2cm3}^{\text{bladder}}), \delta_s(D_{1cm3}^{\text{rectum}}), delta_s(D_{2cm3}^{\text{rectum}}), \delta_s(D_{0.1cm3}^{\text{urethra}})\right\},
\end{aligned}$$

$$\delta_c(V_d^o) = V_d^o - V_d^{o,\min},$$

$$\delta_s(D_v^o) = D_v^{o,\max} - D_v^o,$$

where $V_d^{o,\min}$, and $D_v^{o,\max}$ are aspiration values defined by the clinical protocol. We further refer to values of δ_c and δ_s as DV delta values.

The aspiration values for $V_{150}^{\text{prostate}}$ and $V_{200}^{\text{prostate}}$ are excluded from the LSI, as is done in previous work [18], because these volume indices have a different unit than the remaining dose indices in the LSI, making them difficult to be compared to each other. Instead, the aspiration values for these volume indices are included as hard constraints, meaning that any plan produced by the optimization procedure must satisfy these aspiration values. Furthermore, the clinical criterion $D_{90}^{\text{prostate}} > 100\%$ is not included in the LCI, because this criterion is always satisfied when $V_{100}^{\text{prostate}} > 95\%$ is satisfied.

A benefit of optimizing the LCI and LSI is that their outcome strictly depends on the most violated DV index, which is the DV index with the lowest corresponding DV delta. All optimization effort is therefore spent on trying to optimize the most violated DV index. However, this means that any small improvement of the most violated DV index will be preferred over a much larger improvement of a different index, as long as this does not change which DV index is the most violated. Furthermore, optimization of remaining DV indices is no longer possible if the most violated DV index has reached its optimal value. Intuitively, it is desirable to spend some optimization effort on DV indices that are not the most violated, especially in the aforementioned cases. For this reason, we define adaptively weighted variations of LCI and LSI, denoted LCI_w and LSI_w.

To compute the LCI_w, the required DV deltas are first sorted in descending (DESC) order to find the order of DV indices from least to most violated. For the LCI_w, the sorted list of DV deltas is defined as

$$\boldsymbol{\Delta}_c^s = \mathrm{sort}^{\mathrm{DESC}}\left(\left[\delta_c(V_{100}^{\text{prostate}}), \delta_c(V_{80}^{\text{vesicles}})\right]\right).$$

Similarly, for the LSI$_w$, the sorted list of DV deltas is defined as

$$\mathbf{\Delta}_s^s = \mathrm{sort}^{\mathrm{DESC}}\left(\left[\delta_s(D_{1cm3}^{\mathrm{bladder}}), \delta_s(D_{2cm3}^{\mathrm{bladder}}), \delta_s(D_{1cm3}^{\mathrm{rectum}}), \delta_s(D_{2cm3}^{\mathrm{rectum}}), \delta_s(D_{0.1cm3}^{\mathrm{urethra}})\right]\right).$$

Given the sorted list of DV deltas, the LCI$_w$ and LSI$_w$ are then defined as a weighted sum of DV deltas, where the DV delta with the highest value, i.e., corresponding to the least violated DV index, receives a weight of 1 and consecutive weights increase exponentially by a factor of $\lambda = 10$. All weights are then normalized to the range between 0 and 1. Note that the weights associated with individual DV deltas are dynamic throughout optimization, because the sorted order may change, and are not set by a user.

LCI$_w$ and LSI$_w$ are defined as

$$\begin{aligned}
\mathrm{LCI}_w(\boldsymbol{t}) &= w_c(\delta_c(V_{100}^{\mathrm{prostate}})) + w_c(\delta_c(V_{80}^{\mathrm{vesicles}})),\\
\mathrm{LSI}_w(\boldsymbol{t}) &= w_s(\delta_s(D_{1cm3}^{\mathrm{bladder}})) + w_s(\delta_s(D_{2cm3}^{\mathrm{bladder}})) + w_s(\delta_s(D_{1cm3}^{\mathrm{rectum}}))\\
&\quad + w_s(\delta_s(D_{2cm3}^{\mathrm{rectum}})) + w_s(\delta_s(D_{0.1cm3}^{\mathrm{urethra}})),\\
w_c(\delta) &= \frac{\lambda^{r-1}}{\sum_{i=1}^{|\mathbf{\Delta}_c^s|}\lambda^{i-1}}\delta \quad \text{with } r \text{ s.t.} \quad \mathbf{\Delta}_c^s[r] = \delta,\\
w_s(\delta) &= \frac{\lambda^{r-1}}{\sum_{i=1}^{|\mathbf{\Delta}_s^s|}\lambda^{i-1}}\delta \quad \text{with } r \text{ s.t.} \quad \mathbf{\Delta}_s^s[r] = \delta,
\end{aligned}$$

where rank r is a one-based index that indicates the position of the respective DV index in the sorted list of indices.

For $\lambda = 1$, LCI$_w$ and LSI$_w$ define equally-weighted linear combinations of their respective DV deltas. This would result in a loss of focus on the most violated DV index, making it more difficult to find solutions satisfying all clinical criteria. As λ increases, the importance of the most violated DV index is increased. For λ approaching infinity, LCI$_w$ and LSI$_w$ are identical to the LCI and LSI. The value of $\lambda = 10$ was empirically determined to produce equally good solutions in terms of LCI and LSI as the previously used bi-objective model [18], while increasing the quality of DV indices that are not the most violated.

In this chapter, we use MO-RV-GOMEA with the LCI$_w$ and LSI$_w$ as optimization objectives, unless stated otherwise. Note that the LCI$_w$ and LSI$_w$ no longer have the property that a positive value indicates that all clinical criteria have been satisfied. For this reason, we generally report LCI and LSI in Section 5.3, even when LCI$_w$ and LSI$_w$ were used as optimization objectives.

In Section 5.3.2 we analyze how the results of the optimization are affected by using the LCI$_w$ and LSI$_w$ as optimization objectives, compared to the LCI and LSI.

5.2.5. Multi-Objective Evolutionary Algorithm

Evolutionary Algorithms (EAs) [13] are computational heuristics that can be distinguished by the fact that a set, called the *population*, of potential solutions is maintained, rather than a single solution. In our case, a population of 96 solutions is used, where each solution represents a treatment plan. The *fitness* of a solution, in this case defined in terms of LCI and LSI, defines its relative quality. During each *generation* of an EA, *variation* is applied in order to generate a set of new potential solutions, called the *offspring*. By applying *selection*, the solutions with the best fitness from the combined population and offspring form the population of the following generation, and non-selected solutions are discarded.

EAs are known to be among the state of the art in the field of multi-objective optimization [9]. The result of a multi-objective EA is a set of solutions that each have a different trade-off with regard to the objectives. In this chapter, we specifically use the Multi-Objective Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm (MO-RV-GOMEA) [4], as it was shown to achieve the best performance among a set of state-of-the-art multi-objective EAs on bi-objective treatment planning for prostate HDR BT [18]. MO-RV-GOMEA uses an archive with a maximum capacity of 1250 to store all plans for which no better plan was previously found. In this context, we consider a plan to be better when it Pareto-dominates the other plan in terms of LCI and LSI, i.e., it has a better LCI value and an LSI value that is at least as good, or vice versa. This archive is constantly updated whenever new plans are found. At the end of an optimization run, all plans in the archive are presented as the results of the optimization.

We refer the interested reader to Appendix 5.A for detailed parameter settings of MO-RV-GOMEA.

5.2.6. GPU-ACCELERATED BI-OBJECTIVE TREATMENT PLANNING

GPUs are widely used for the purpose of high-performance computing, both within the field of medical physics [25], for example for dose calculation [8, 12, 27, 29], and in the field of evolutionary computation [7, 17, 24, 30]. Because the design of GPUs is particularly suited for specific forms of large-scale parallel computation, they are very suitable for high-performance dose calculation, as the dose in all DC points can be calculated in parallel.

By far the most time-consuming procedures of the bi-objective treatment planning method are the dose calculation and the calculation of the DV indices of the treatment plans in the population. Due to the use of an EA, these procedures are always performed on the entire population, further increasing the potential for parallelization. We therefore use a GPU to improve the performance of the population-wide dose and DV index calculation. Specifically, the GPU that is used in this work is an NVIDIA Titan Xp, and all programming for the GPU is done in CUDA [23]. Using an NVIDIA Titan Xp GPU, our treatment planning method currently supports the optimization of sets up to 500,000 DC points.

We use a parallel dose calculation method that we previously introduced [2]. In this method, the dwell times of all treatment plans in the population are stored in a matrix $\boldsymbol{T}$, where each row describes the set of dwell times of one treatment plan. The dose in all DC points is then calculated through the matrix-matrix multiplication $\boldsymbol{D} = \boldsymbol{T}\boldsymbol{R}^T$, where $\boldsymbol{R}$ is a dose-rate matrix according to the TG-43 model [26]. Element $\boldsymbol{R}[i, j]$ describes the dose contribution in Gy/s of the source at dwell position with index i to the DC point with index j. Each row of $\boldsymbol{D}$ describes the dose in each DC point of one potential treatment plan in the population. The number of rows of $\boldsymbol{D}$ is therefore equal to the population size, and the number of columns is equal to the number of DC points.

The used optimization algorithm, MO-RV-GOMEA [4], often finds new potential treatment plans by modifying a small number of dwell times of a potential treatment plan currently existing in the population. When this is the case, the dose distribution $\boldsymbol{D}'$ of the modified treatment plan is calculated more efficiently through $\boldsymbol{D}' = \boldsymbol{D} + \Delta\boldsymbol{T}\boldsymbol{R}^T$, with $\Delta\boldsymbol{T}$ the matrix of differences in dwell times [2]. Matrix $\Delta\boldsymbol{T}$ has few columns with non-zero elements, equal to the number of dwell times that have been modified. Only these columns with non-zero elements are considered for the calculation of $\boldsymbol{D}'$.

The estimation of dose indices requires sorting the dose in the DC points of each ROI for which a dose index must be calculated, for each potential treatment plan. For this purpose, each DC point d_i is assigned an identifier ID_i, such that DC points in the same ROI of the same treatment plan are assigned an identical ID. A list L of dose/ID pairs is then created, i.e., $L = [(d_0, \text{ID}_0), (d_1, \text{ID}_1), \ldots]$. The Thrust library [5], which contains a large number of CUDA utility functions, is then used to sort L by dose value in ascending order. This is followed by the Thrust stable sort procedure, sorting L by ID in ascending order. The stable sort procedure sorts L by ID, but maintains the relative order of items with an identical ID. As the dose values of L were previously sorted, this means that L will be sorted by ID, but dose values with the same ID remain sorted. The above two sorting procedures therefore result in a list of DC points that are firstly sorted by key, and secondly sorted by dose. A dose index is then calculated according to Section 5.2.3.

The estimation of volume indices requires counting the number of DC points that receive at least a certain dose. For this purpose, all required DC points are first compared to their respective aspiration value in parallel, resulting in a large array of boolean values. Thrust is then used to compute a cumulative sum of this array, after which the number of points achieving their respective aspiration value can be trivially computed for any subsequence of this array. A volume index is then calculated according to Section 5.2.3.

The sampling of DC points and the creation of the dose-rate matrix are not included in the optimization times presented in this chapter. Dose points are sampled using 4 CPU cores in parallel, and the dose-rate matrix is calculated on a single CPU core and then copied to the GPU memory. For a set of 100,000 DC points, this takes approximately 2 seconds on average.

In Section 5.3.3 we compare treatment plans optimized with the GPU-accelerated bi-objective treatment planning to the set of clinical plans.

5.3. RESULTS

In Section 5.3.1, we show how the outcome of optimization in terms of LCI and LSI depends on the number of DC points. Section 5.3.2 then shows the effect of the introduction of the LCI_w and LSI_w on the optimization. Finally, the quality of the results achieved by our bi-objective treatment planning method is shown in Section 5.3.3.

5.3.1. DV INDEX ESTIMATION

In Figure 5.1 we illustrate the impact of the number of DC points on the optimization results. This figure shows the treatment plans found after completing 30 runs of the

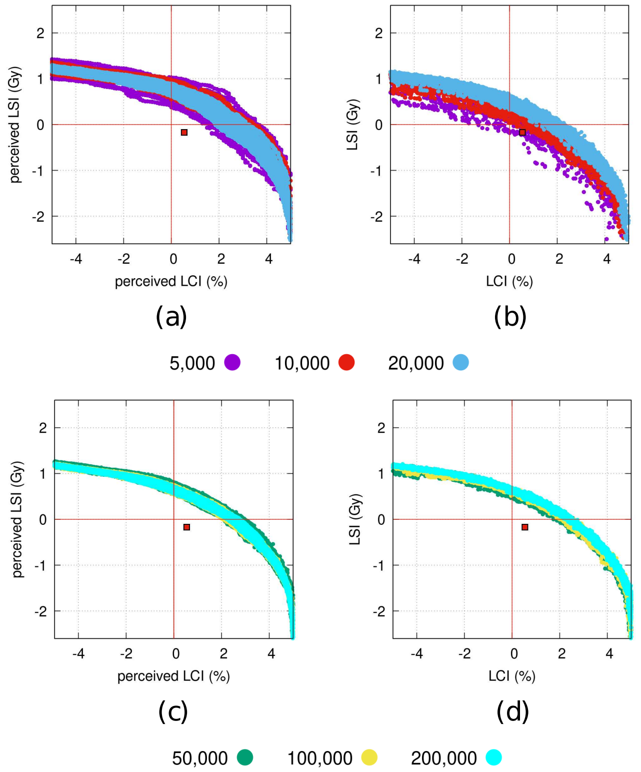


Figure 5.1: Treatment plans found after 10 minutes for patient 1, shown in different colors pertaining to the number of DC points used for optimization. All figures show the combined results of 30 independent runs (to show the degree of variance in outcomes) for each setting of the number of DC points. Perceived LCI and LSI ((a) and (c)) indicate the LCI and LSI values as calculated by the optimization algorithm on the given set of DC points. More accurate LCI and LSI values ((b) and (d)) are calculated using a set of 500,000 independently sampled DC points. The clinical plan is shown as a red square.

optimization with a time limit of 10 minutes on a set of 5,000, 10,000, 20,000, 50,000, 100,000, and 200,000 DC points, respectively. A time limit of 10 minutes was used for all settings, as this time limit was found to be sufficient to obtain the maximum achievable performance for each of the tested number of DC points.

Figure 5.1a shows the perceived LCI and LSI, calculated using sets of 5,000, 10,000, and 20,000 DC points used for optimization. The perceived LCI and LSI values, i.e., as computed by the optimization algorithm, suggest that better treatment plans can be found when smaller numbers of DC points are used. However, Figure 5.1b shows a substantially more accurate calculation of the LCI and LSI values of the identical set of plans, using a set of 500,000 independently sampled DC points. Because Figure 5.1b shows a reduction in LCI and LSI compared to Figure 5.1a, this indicates that the quality of the resulting plans is substantially worse than perceived during optimization. This is caused by the fact that too small sets of DC points are not a good representation of their respective ROI. Due to this effect, no plan satisfying all clinical criteria was found for 8 out of 30 runs when 5,000 DC points were used for optimization. This was still the case for 2 out of 30 runs when 10,000 DC points were used for optimization.

Figures 5.1c and 5.1d show that the difference between perceived and accurate LCI and LSI are much smaller when sets of at least 50,000 DC points are used for optimization. Furthermore, comparing Figures 5.1b and 5.1d shows that the actual LCI and LSI obtained when solving for up to 20,000 DC points are clearly worse than the optimization algorithm perceives, whereas the difference between perceived and accurate LCI and LSI is much smaller for the set of 100,000 DC points.

As Figure 5.1d shows a negligible difference between optimization outcomes for 100,000 and 200,000 DC points, a set of 100,000 DC points seems sufficiently large for a reliable calculation of the LCI and LSI during optimization.

5.3.2. BI-OBJECTIVE TREATMENT PLANNING MODEL

Figure 5.2 shows how the use of LCI$_w$ and LSI$_w$ as optimization objectives, shown in the bottom row, affect optimization results, compared to the use of LCI and LSI, shown in the top row. In the middle plots, all DV indices of each treatment plan are displayed in a separate color. Each treatment plan is therefore represented by five different data points in different colors, by definition aligned vertically, as each plan is defined by a single value for its LCI. In the rightmost plots, the coverage of the prostate and seminal vesicles are shown for each treatment plan. Each treatment plan is represented by two data points in different colors, aligned horizontally.

The leftmost two figures show no perceivable difference of the LCI and LSI between weighted and non-weighted LCI and LSI. This means that both approaches find very similar values for the most violated DV indices in terms of coverage and sparing. However, the remaining subfigures show that optimizing the LCI$_w$ and LSI$_w$ results in better values for the DV indices that are not the most violated.

The middle subfigure of Figure 5.2 shows slight improvements for $D^{\text{bladder}}_{1cm3}$, $D^{\text{bladder}}_{2cm3}$, D^{rectum}_{1cm3}, and D^{rectum}_{2cm3} in case of using LCI$_w$ and LSI$_w$ instead of LCI and LSI. In addition, much better values are found for the V^{vesicles}_{80}, showing that, for this patient, satisfying the clinical criterion for this index is possible regardless of the the achieved LSI. Note that this is not the case for each patient, as shown in Figures 5.B.1 to 5.B.4, because this depends on the patient anatomy and catheter placement. As shown in Figure 5.4, it is still difficult to achieve good vesicle coverage for patient 1, and for various other patients for which results are shown in Figures 5.B.1 to 5.B.4. Overall, however, the results indicate that optimizing the LCI$_w$ and LSI$_w$ leads to better results in terms of individual DV indices than optimizing the LCI and LSI.

5.3.3. GPU-ACCELERATED BI-OBJECTIVE TREATMENT PLANNING

We now show more detailed results obtained by optimizing for sets of 20,000 and 100,000 DC points, as the former setting was used when our bi-objective approach was introduced and validated [18, 20], and the latter setting is one we decided upon in Section 5.3.1. Results obtained by optimizing for sets of 5,000, 10,000, 50,000, and 200,000 DC points are included in Table 5.B.1. Furthermore, Table 5.B.2 contains results for 20,000 and 100,000 DC points using time limits of 30, 60, 180, and 300 seconds. In

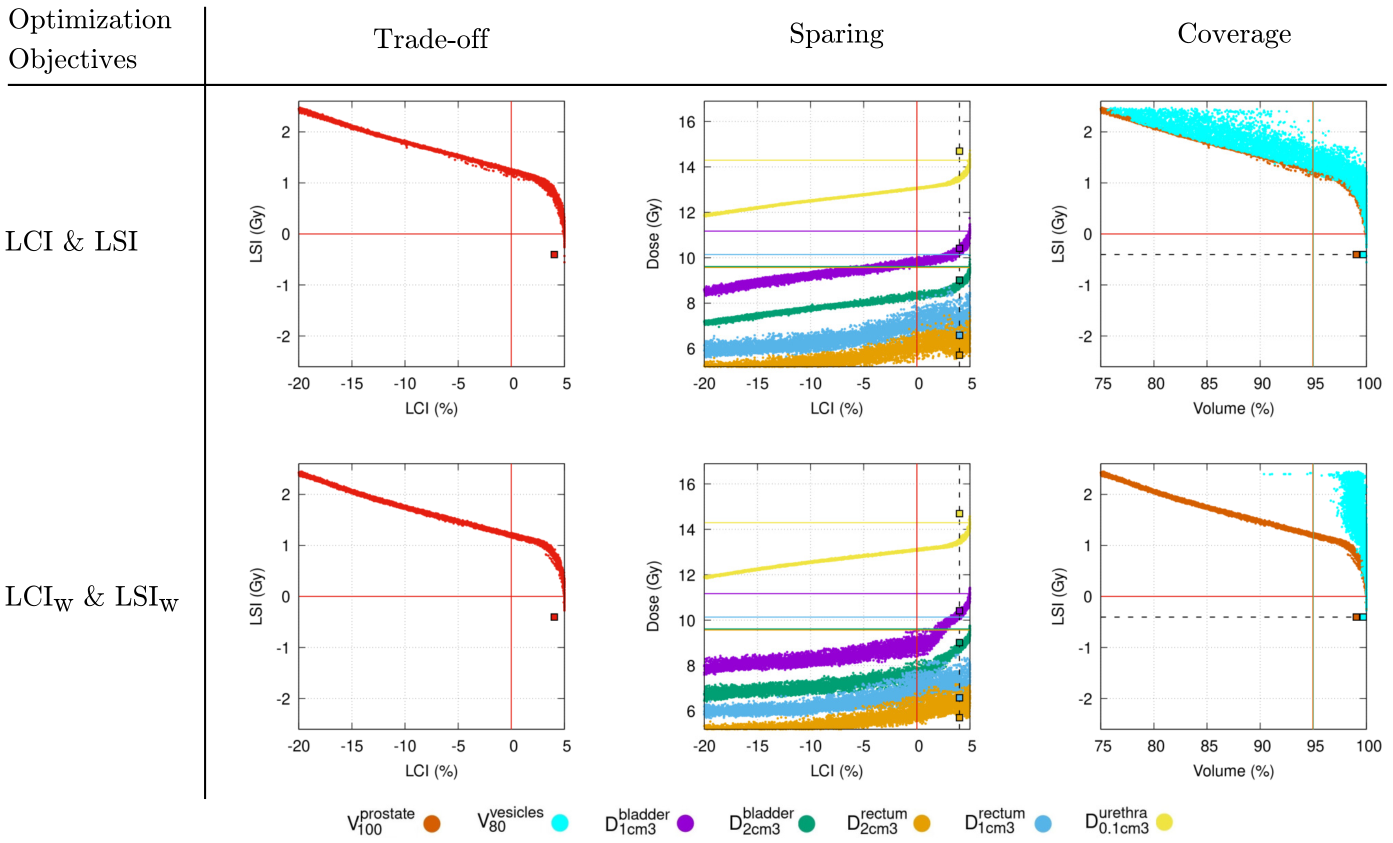


Figure 5.2: Results for patient 2. The top row shows results found when the LCI and LSI are used as optimization objectives. The bottom row shows results found when the LCI$_w$ and LSI$_w$ are used as optimization objectives. Subfigures show all plans found by 30 independent optimization runs, to show variation, using a set of 100,000 DC points, and a time limit of 3 minutes. DV indices of the clinical plan are displayed as squares with identical colors to the optimized treatment plans. In all subfigures, the (non-weighted) LCI and LSI are displayed, calculated using a set of 500,000 DC points.

Figure 5.3 we show the results of 30 optimization runs, each with a newly sampled set of DC points, for patients 1, 2, and 3, optimized on 20,000 and 100,000 DC points.

The optimization with 20,000 DC points resulted in a set of 461 treatment plans on average. After reevaluation using 500,000 DC points, 236 plans remained on average, as any plan was removed from the set of results when it was Pareto-dominated by a different plan. The optimization with 100,000 DC points resulted in a set of 658 treatment plans on average. After reevaluation using 500,000 DC points, 428 plans remained on average.

When 20,000 DC points are used for the optimization, results obtained after 30 seconds are similar to those obtained after 10 minutes. Furthermore, variation in LCI and LSI can be fairly large for this number of DC points. When 100,000 DC points are used, a clear increase in LCI and LSI is observed for time limits up to 180 seconds. Most importantly, there is little variation in the 30 different runs using 100,000 DC points, meaning that this number of DC points more reliably results in a set of high-quality treatment plans. Though there is still a minor difference between optimizing for 180 or 600 seconds, we argue that this difference can be considered irrelevant. Optimizing for 300 or 600 seconds could be considered depending on the strictness of clinical time constraints.

Table 5.2 shows, for each patient used in this study, the clinical plan along with two plans selected from the result of the optimization on 20,000 DC points and 100,000 DC points. A time limit of 30 seconds was used for the optimization on 20,000 DC points, and 3 minutes for the optimization on 100,000 DC points. For both settings of the number of DC points, the optimization was performed 30 times per patient. The selected plan is the plan with the maximum achieved LCI that satisfied all sparing criteria, i.e., achieved an LSI larger than 0. This plan is selected to show how much coverage can be achieved while all sparing criteria are satisfied. Though we show only one plan from each optimization run in Table 5.2, each optimization run produced many plans with different trade-offs between coverage and sparing, as shown in Figure 5.4.

Only 4 out of 18 clinical plans satisfied all clinical criteria, whereas the optimization consistently (30 out of 30 runs) found plans satisfying all clinical criteria for 15 out of 18 patients. No plans satisfying the clinical protocol were found for the remaining 3 patients due to sub-optimal catheter placement. The value of optimizing on a large number of DC points is mostly shown for patient 7, as all runs optimized on 100,000 DC points found a plan satisfying all clinical criteria, whereas this was not the case when optimizing 20,000 DC points.

A common characteristic of the clinical plans is that the urethra received a relatively large amount of dose, as the clinical criterion for the $D^{\text{urethra}}_{0.1cm3}$ was satisfied for only 5 out of 18 clinical plans. Compared to the clinical plans, some sparing of the rectum was sacrificed in many cases of the bi-objectively optimized plans. On average, the D^{rectum}_{1cm3} increased by 0.46 Gy, and the D^{rectum}_{2cm3} increased by 0.43 Gy. However, the $D^{urethra}_{0.1cm3}$ was reduced by 0.68 Gy on average.

Figure 5.4 shows the combined results of 30 optimization runs for patient 1, compared to the clinical plan. All 30 runs are displayed in this figure to show the variance of the resulting treatment plans, caused by the stochastic nature of the optimization, and of random sampling of DC points. The variance of a DV index also depends on the difficulty of satisfying the corresponding criterion defined in the

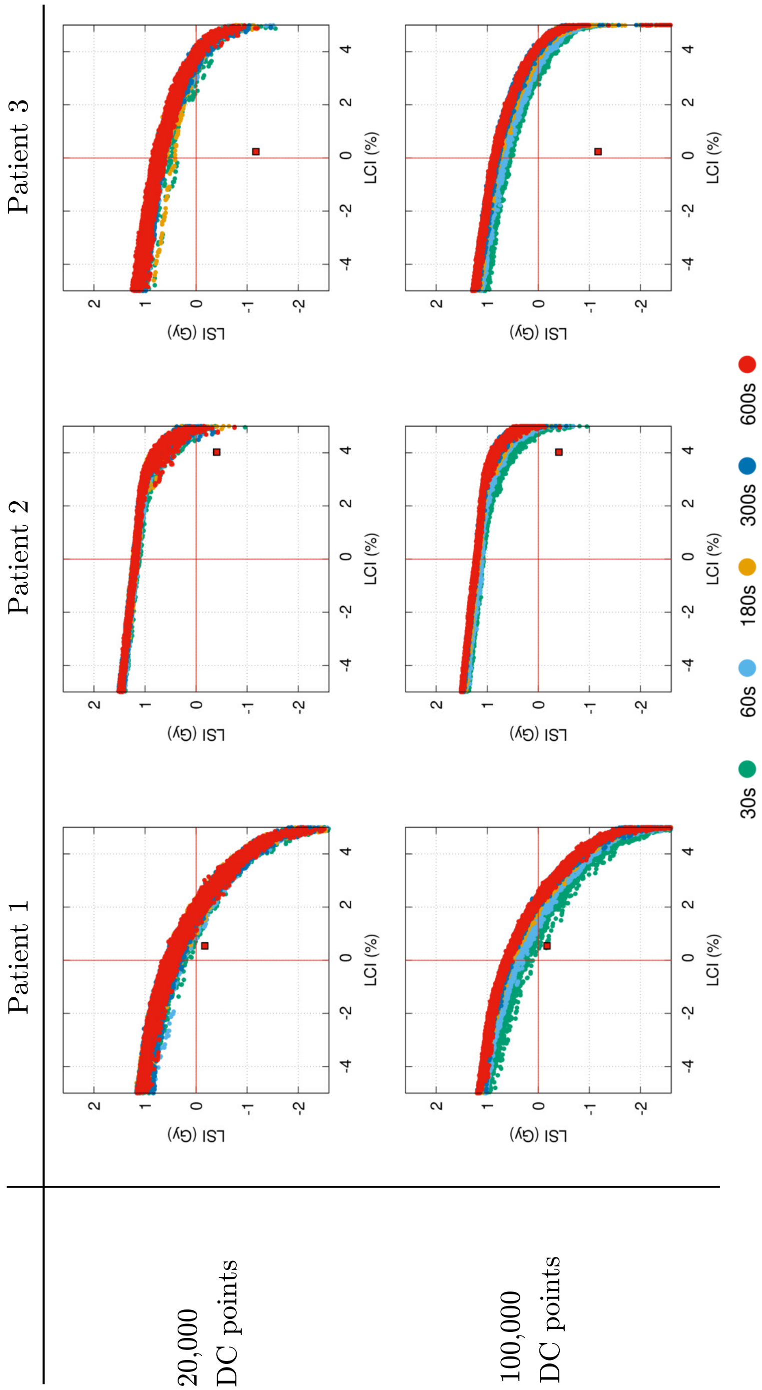


Figure 5.3: Bi-objectively optimized plans for patients 1, 2, and 3, obtained after optimizing for the specified number of seconds, with 20,000 or 100,000 DC points. Results shown here are re-evaluated on 500,000 DC points. To show the joint variation of the optimization and sampling of DC points, all treatment plans from 30 independent optimization runs are shown. The clinical plan is displayed as a red square.

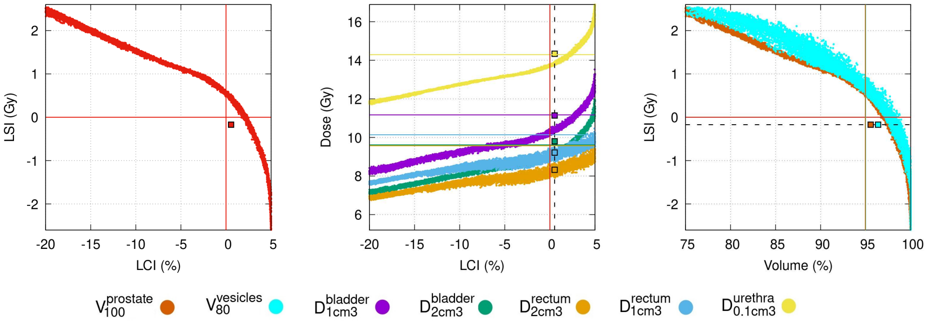


Figure 5.4: Combined results of 30 optimization runs for patient 1, optimized for 3 minutes on 100,000 DC points with LCI$_w$ and LSI$_w$ as objectives, showing corresponding values of all DV indices for each value of the LCI and LSI.

clinical protocol, because DV indices that are more difficult to satisfy are assigned a larger weight in the LCI$_w$ and LSI$_w$. Similar to Figure 5.2, all DV indices of each treatment plan are displayed in a separate color according to the legend, and the clinical plan is displayed as a square. For each value of the LCI and LSI, Figure 5.4 shows the corresponding values for all DV indices. Plots for all remaining patients are included in Appendix 5.B.

5.4. DISCUSSION

In this chapter, we introduced a GPU-based acceleration of a previously introduced bi-objective evolutionary optimization approach for treatment planning for HDR prostate brachytherapy. This bi-objective treatment planning method directly optimizes DV indices, whereas conventional treatment planning methods, e.g., IPSA [16] or HIPO [14], use a penalty-based optimization model that weakly correlates with the quality of DV indices [11].

In contrast to a more recently introduced multi-criteria optimization approach [6] that requires pre-trained regression models to guide the optimization towards clinically acceptable plans, our bi-objective evolutionary optimization approach requires no prior training, because it directly calculates all relevant DV indices throughout optimization. As our bi-objective evolutionary optimization approach requires between 30 and 300 seconds, we have shown that it is indeed possible to approximate a surface of trade-off plans within a clinically acceptable time frame.

Increasing the accuracy of DV-index calculation by increasing the number of DC points further improves the DV indices achieved by the bi-objective optimization approach, with very good results obtainable in 3 minutes on 100,000 DC points. Even this more time-consuming approach is arguably still within the time limit of clinical practice, because planning in our clinic currently takes 30 to 60 minutes, which is mainly spent on graphical optimization [20]. Increasing the number of DC points beyond 100,000 would further increase the required computation time for a negligible increase in plan quality.

Table 5.2: DV indices of the clinical plans, alongside means and standard deviations (30 runs) of plans optimized on 20,000 (GOMEA20k) or 100,000 (GOMEA100k) DC points by the bi-objective optimization approach, for all 18 patients used in this study. Plans were selected from the trade-off curve to have maximum LCI while satisfying all sparing criteria. Volume indices and LCI values are shown as a percentage of ROI volume. Dose indices and LSI values are shown as a percentage of the PD of 13 Gy. Records shaded in gray do not satisfy the clinical protocol. Optimized results shaded in gray did not satisfy the clinical protocol in at least 1 out of 30 runs. All underlined records are statistically significantly different from the corresponding index of the clinical plan (Wilcoxon signed-rank test, $p < 0.05$; Bonferroni correction, $n = 324$).

Patient	Plan	$V_{100}^{\text{prostate}}$	V_{80}^{vesicles}	$D_{1cm3}^{\text{bladder}}$	$D_{2cm3}^{\text{bladder}}$	D_{1cm3}^{rectum}	D_{2cm3}^{rectum}	$D_{0.1cm3}^{\text{urethra}}$	$V_{150}^{\text{prostate}}$	$V_{200}^{\text{prostate}}$	LCI	LSI
1	Clinical	95.5	96.4	85.8	75.3	70.9	64.0	110.4	34.3	13.4	0.5	-1.3
	GOMEA20k	96.4±0.3	97.7±0.6	83.3±0.8	73.6±0.5	70.3±1.0	64.1±1.0	108.9±0.9	27.8±1.4	11.2±0.6	1.4±0.3	0.2±0.2
	GOMEA100k	97.0±0.2	97.7±0.4	83.5±0.5	73.7±0.3	71.8±1.3	65.5±1.2	109.7±0.3	28.5±1.3	11.5±0.6	2.0±0.2	0.1±0.1
2	Clinical	99.0	99.7	80.2	69.3	50.6	43.9	113.1	26.3	9.9	4.0	-3.1
	GOMEA20k	99.8±0.1	100.0±0.0	85.0±1.1	72.6±1.0	55.7±3.1	48.3±2.5	108.8±1.0	19.0±1.0	7.2±0.4	4.8±0.1	0.5±0.5
	GOMEA100k	99.9±0.0	100.0±0.0	85.3±0.6	72.8±0.6	56.1±3.1	48.7±2.4	109.4±0.4	19.3±0.9	7.4±0.4	4.9±0.0	0.4±0.4
3	Clinical	98.3	95.2	92.5	83.0	78.6	69.3	109.0	28.1	9.8	0.2	-9.0
	GOMEA20k	98.5±0.2	98.9±0.3	81.4±0.6	73.7±0.4	76.5±1.2	67.4±1.1	108.7±0.9	22.8±1.7	8.5±0.5	3.5±0.2	0.1±0.1
	GOMEA100k	99.0±0.1	99.1±0.1	81.5±0.3	73.9±0.1	77.1±0.6	67.7±0.6	109.5±0.3	24.0±1.4	9.1±0.3	4.0±0.1	0.1±0.1
4	Clinical	99.3	86.2	89.2	80.9	67.8	59.8	114.4	37.4	17.6	-8.8	-6.9
	GOMEA20k	93.1±0.5	93.3±0.4	82.0±0.5	73.8±0.2	72.6±2.0	63.2±1.7	109.0±0.8	25.6±1.3	10.9±0.7	-2.0±0.4	0.2±0.2
	GOMEA100k	93.7±0.2	93.7±0.2	81.9±0.4	73.9±0.1	72.9±1.8	63.4±1.3	109.4±0.3	26.3±0.7	11.4±0.5	-1.4±0.2	0.1±0.1
5	Clinical	96.1	88.6	86.3	74.8	70.8	61.3	110.4	36.1	16.0	-6.4	-0.8
	GOMEA20k	97.0±0.5	98.6±0.5	83.0±1.3	73.3±0.8	75.4±1.8	66.8±1.8	108.8±1.2	28.1±2.2	10.4±0.9	2.0±0.5	0.1±0.1
	GOMEA100k	97.8±0.2	98.7±0.4	83.5±0.5	73.7±0.3	76.3±1.1	67.7±1.1	109.6±0.5	27.8±2.0	10.2±0.8	2.8±0.2	0.1±0.1
6	Clinical	97.8	96.2	80.0	68.3	80.3	70.8	121.1	35.4	14.2	1.2	-11.1
	GOMEA20k	96.0±0.4	100.0±0.1	82.4±1.8	71.7±1.5	77.4±0.8	68.4±0.6	109.4±0.6	27.5±1.1	10.3±0.4	1.0±0.4	0.2±0.2
	GOMEA100k	96.7±0.1	99.8±0.2	83.7±0.8	72.7±0.7	77.7±0.3	68.7±0.4	109.6±0.3	27.2±1.0	10.3±0.4	1.7±0.1	0.2±0.2
7	Clinical	97.5	78.2	84.2	76.9	64.7	57.0	113.9	31.4	14.7	-16.8	-3.9
	GOMEA20k	95.2±0.3	95.5±0.4	81.1±0.4	73.8±0.2	70.5±2.3	62.5±1.8	108.5±0.9	26.3±1.1	10.9±0.6	0.1±0.3	0.2±0.1
	GOMEA100k	95.9±0.2	96.1±0.2	81.0±0.4	73.9±0.1	70.5±1.9	62.5±1.6	109.3±0.5	25.8±1.0	10.7±0.6	0.9±0.2	0.1±0.1
8	Clinical	98.3	98.8	75.7	67.8	81.1	73.9	127.1	28.9	11.3	3.3	-17.1
	GOMEA20k	96.3±0.4	99.0±0.6	81.1±2.0	72.4±1.3	77.5±0.7	70.5±0.7	109.2±0.8	17.9±0.8	6.7±0.3	1.3±0.4	0.2±0.3
	GOMEA100k	97.0±0.2	98.9±0.4	82.2±0.9	73.3±0.6	77.7±0.3	70.5±0.3	109.8±0.3	17.9±0.7	6.8±0.3	2.0±0.2	0.1±0.1
9	Clinical	97.8	98.0	80.5	73.8	53.1	47.5	113.9	28.2	12.3	2.8	-3.9
	GOMEA20k	97.7±0.2	99.3±0.4	80.2±0.5	73.8±0.3	58.4±1.9	52.4±1.6	108.2±1.1	22.7±1.7	9.0±0.8	2.7±0.2	0.1±0.2
	GOMEA100k	98.3±0.1	99.5±0.2	80.0±0.3	73.9±0.1	58.8±1.7	52.5±1.3	109.3±0.5	22.8±1.1	9.0±0.4	3.3±0.1	0.1±0.1
10	Clinical	96.7	95.5	76.5	65.9	67.1	56.0	109.6	28.5	10.5	0.5	0.4
	GOMEA20k	98.8±0.1	99.9±0.1	84.2±1.0	73.0±0.8	73.4±2.2	61.3±1.8	109.7±0.3	21.5±0.9	7.7±0.3	3.8±0.1	0.2±0.1
	GOMEA100k	99.2±0.1	99.9±0.1	84.8±0.5	73.5±0.4	74.8±1.9	62.3±1.5	109.9±0.1	21.4±0.8	7.7±0.3	4.2±0.1	0.1±0.0
11	Clinical	98.3	100.0	82.8	72.5	64.5	56.9	109.7	34.8	15.1	3.3	0.3
	GOMEA20k	99.2±0.1	100.0±0.1	82.7±1.4	71.9±1.2	64.7±1.6	56.7±1.2	109.6±0.4	30.6±0.8	13.4±0.5	4.2±0.1	0.3±0.4
	GOMEA100k	99.5±0.1	100.0±0.1	84.4±0.9	73.2±0.6	65.6±1.3	57.6±1.1	109.9±0.2	30.5±0.9	13.4±0.5	4.5±0.1	0.1±0.1
12	Clinical	98.8	95.0	76.0	66.8	63.0	56.9	116.8	28.6	11.2	0.0	-6.8
	GOMEA20k	98.7±0.1	99.8±0.1	82.5±1.0	72.8±0.9	69.3±1.4	62.3±1.2	109.6±0.4	23.9±1.5	9.0±0.7	3.7±0.1	0.3±0.3
	GOMEA100k	98.9±0.1	99.8±0.1	83.1±0.7	73.2±0.6	68.9±1.5	62.0±1.3	109.9±0.1	23.2±1.0	8.9±0.4	3.9±0.1	0.1±0.1
13	Clinical	97.5	95.4	81.5	73.4	62.7	56.0	111.5	20.1	8.5	0.4	-1.5
	GOMEA20k	97.2±0.4	97.8±0.5	81.3±0.9	73.5±0.6	69.0±2.2	61.9±1.9	108.8±1.2	17.8±1.1	6.5±0.3	2.2±0.4	0.2±0.2
	GOMEA100k	97.9±0.2	98.0±0.2	81.6±0.4	73.9±0.1	70.9±1.6	63.6±1.4	109.4±0.4	17.8±0.6	6.5±0.2	2.9±0.2	0.1±0.1
14	Clinical	93.4	85.9	65.9	59.3	70.0	61.5	122.7	30.1	11.8	-9.1	-12.7
	GOMEA20k	94.4±0.3	99.9±0.2	80.6±1.6	72.8±1.1	73.9±1.7	66.3±1.3	109.5±0.6	27.1±1.1	11.1±0.5	-0.6±0.3	0.2±0.2
	GOMEA100k	94.9±0.1	99.9±0.1	81.2±0.8	73.3±0.6	74.4±1.4	66.5±1.1	109.9±0.1	26.6±1.2	10.9±0.4	-0.1±0.1	0.1±0.1
15	Clinical	95.7	99.9	84.7	76.2	76.2	65.8	111.9	29.0	12.8	0.7	-2.2
	GOMEA20k	98.0±0.5	99.8±0.1	82.2±0.9	73.5±0.6	75.6±1.4	66.5±1.3	108.7±1.4	25.0±1.2	10.0±0.7	3.0±0.5	0.2±0.2
	GOMEA100k	98.7±0.2	99.8±0.2	82.5±0.5	73.6±0.4	76.3±0.9	67.2±0.9	109.7±0.5	25.2±1.2	10.3±0.5	3.7±0.2	0.1±0.1
16	Clinical	96.2	98.1	75.2	65.9	53.6	46.9	107.1	28.1	9.4	1.2	2.9
	GOMEA20k	99.9±0.1	100.0±0.0	81.8±1.5	71.2±1.2	60.6±1.6	53.1±1.4	107.4±1.0	21.6±1.8	6.6±0.5	4.9±0.1	2.2±0.9
	GOMEA100k	100.0±0.0	100.0±0.0	82.0±1.1	71.4±0.9	63.0±1.8	55.1±1.5	107.5±0.9	21.3±1.2	6.8±0.6	5.0±0.0	2.3±0.9
17	Clinical	95.3	95.5	72.2	65.5	67.9	61.1	109.3	25.2	8.6	0.3	0.7
	GOMEA20k	99.6±0.1	99.8±0.1	81.8±0.7	73.7±0.3	75.1±1.7	68.1±1.6	108.8±0.8	21.2±1.6	6.5±0.8	4.5±0.1	0.2±0.2
	GOMEA100k	99.8±0.1	99.9±0.1	82.1±0.6	73.8±0.2	75.9±1.2	68.9±1.1	109.7±0.3	21.1±1.6	6.5±0.6	4.8±0.1	0.1±0.1
18	Clinical	96.8	98.5	100.7	92.5	69.4	62.6	127.8	38.6	15.5	1.8	-18.5
	GOMEA20k	93.7±0.3	95.7±0.8	78.8±0.5	73.7±0.3	73.8±1.9	65.8±1.6	109.2±0.7	26.4±1.7	8.9±0.6	-1.3±0.3	0.2±0.1
	GOMEA100k	94.4±0.2	95.5±0.5	78.8±0.2	73.9±0.1	73.7±1.7	65.8±1.4	109.6±0.3	26.0±1.2	8.8±0.5	-0.6±0.2	0.1±0.1

From Section 5.3.1 we conclude that a sufficient number of DC points should be used to prevent the optimization from overfitting and thus to make sure the most benefit is achieved from using inverse planning. For sets of at least 100,000 DC points, the effect of overfitting was found to be negligible. These results were found using points that were sampled uniformly at random in each ROI, because random sampling was previously found to be a better estimator of DV indices than sampling on a regular grid [22]. Other sampling techniques have previously been used [16, 28], which could prove to require a smaller number of DC points for the accurate calculation of certain DV indices.

In this chapter, an equal number of DC points was sampled in each ROI, regardless of ROI volume. Preliminary experiments showed that distributing the DC points according to ROI volume resulted in plans with visibly worse LCI and LSI values. This is caused by the fact that LCI_w and LSI_w values mostly depend on the most violated DV index, and hence the number of DC points in the corresponding ROI. It is therefore valuable to have a relatively high number of DC points in the corresponding ROI of the most violated DV index. However, the ROI with the most violated DV index changes dynamically throughout optimization, and depends on many factors, including patient anatomy and clinical aspiration values. Further research is required to find a better distribution of DC points among ROIs, while avoiding a loss of obtained plan quality. The efficiency of the calculation of the values of the DV indices could also be improved by only using DC points in the rectum and bladder that are closest to the dwell positions, as these points are likely to receive the highest dose [28].

As the use of multi-objective EAs results in a set of trade-off solutions, this inevitably leads to a decision-making process where a clinician must still decide which of the provided treatment plans is most suitable to actually use. Plots as in Figure 5.4 can prove to be an aid for this decision-making process, as these plots help to visualize the trade-offs between all criteria of the clinical protocol, which should be complemented by the inspection of the dose distribution and dose-volume histograms for the evaluation of single plans. The plot showing only LCI and LSI can only give a rough indication of the trade-off between coverage and sparing, while the remaining plots tell the exact values of each DV index for a given LCI or LSI. Moreover, interactive software could further increase the intuitiveness of these plots by highlighting a selected treatment plan in all three plots. Note that the plots displayed in Figure 5.4 display the combined results of 30 runs, for the sake of displaying the variation in the results of the optimization and the sampling of DC points, whereas a decision maker would only see the results of one optimization run. Each run however covers the entire range of possible trade-offs shown in Figure 5.4.

Based on previous work [18] and the results in Section 5.3, we conclude that MO-RV-GOMEA is an algorithm that is excellently suited for direct DV-index based bi-objective treatment plan optimization. The best results were found when the optimization objectives were the LCI_w and the LSI_w, and a set of 100,000 DC points, i.e., 20,000 DC points per ROI, was used. For this configuration, up to 3 minutes were required for the optimization, as no substantial improvements were found for time limits up to 10 minutes.

Topics of future research include generalization to optimization on different clinical protocols, e.g., the American Brachytherapy Society consensus guidelines [31], as the

LCI and LSI could be constructed based on any clinical protocol consisting of a set of DV criteria. This also includes the addition of the $V_{150}^{\text{prostate}}$ and $V_{200}^{\text{prostate}}$ to the LSI, because these indices are currently included as hard constraints.

5.5. CONCLUSIONS

In general, bi-objective treatment planning provides a clinician with intuitive insight into the treatment planning process by being able to inspect a wide range of potential treatment plans having different trade-offs between coverage of the target organs and sparing of the OARs. We have developed a GPU-based parallel version of a previously introduced bi-objective treatment planning method [18] that optimizes for the worst target and sparing criterion. Furthermore, the optimization model has been adjusted to provide better results, particularly improving values of DV indices that are not the worst criterion. This bi-objective treatment planning method is now capable of producing a set of high-quality solutions, each with a different trade-off between the coverage of the target organs and sparing of the OARs, optimized for 100,000 DC points, in approximately 3 minutes, making it efficient enough to be used in a time-constrained clinical setting.

REFERENCES

[1] Alterovitz, R., Lessard, E., Pouliot, J., Hsu, I.-C. J., O'Brien, J. F., and Goldberg, K. (2006). Optimization of HDR brachytherapy dose distributions using linear programming with penalty costs. *Medical Physics*, 33(11):4012–4019.

[2] Bouter, A., Alderliesten, T., Bel, A., Witteveen, C., and Bosman, P. A. N. (2018). Large-scale parallelization of partial evaluations in evolutionary algorithms for real-world problems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1199–1206. ACM.

[3] Bouter, A., Alderliesten, T., Pieters, B. R., Bel, A., Niatsetski, Y., and Bosman, P. A. N. (2019). GPU-accelerated bi-objective treatment planning for prostate high-dose-rate brachytherapy. *Medical Physics*, 46(9):3776–3787.

[4] Bouter, A., Luong, N. H., Witteveen, C., Alderliesten, T., and Bosman, P. A. N. (2017). The multi-objective real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 537–544. ACM.

[5] Bell, N. and Hoberock, J. (2011). Thrust: A productivity-oriented library for CUDA. *GPU Computing Gems Jade Edition*, 2:359–371.

[6] Cui, S., Després, P., and Beaulieu, L. (2018). A multi-criteria optimization approach for HDR prostate brachytherapy: I. Pareto surface approximation. *Physics in Medicine & Biology*, 63(20):205004.

[7] D'Agostino, D., Pasquale, G., and Merelli, I. (2014). A fine-grained CUDA implementation of the multi-objective evolutionary approach NSGA-II: Potential impact

for computational and systems biology applications. In *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, pages 273–284. Springer.

[8] De Greef, M., Crezee, J., Van Eijk, J., Pool, R., and Bel, A. (2009). Accelerated ray tracing for radiotherapy dose calculations on a GPU. *Medical Physics*, 36(9Part1):4095–4102.

[9] Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc.

[10] Deist, T. M. and Gorissen, B. L. (2016). High-dose-rate prostate brachytherapy inverse planning on dose-volume criteria by simulated annealing. *Physics in Medicine & Biology*, 61(3):1155.

[11] Gorissen, B. L., Den Hertog, D., and Hoffmann, A. L. (2013). Mixed integer programming improves comprehensibility and plan quality in inverse optimization of prostate HDR brachytherapy. *Physics in Medicine & Biology*, 58(4):1041.

[12] Hissoiny, S., Ozell, B., Bouchard, H., and Després, P. (2011). GPUMCD: A new GPU-oriented Monte Carlo dose calculation platform. *Medical Physics*, 38(2):754–764.

[13] Holland, J. (1975). Adaptation in natural and artificial systems. *Ann Arbor: University of Michigan Press.*

[14] Karabis, A., Giannouli, S., and Baltas, D. (2005). HIPO: A hybrid inverse treatment planning optimization algorithm in HDR brachytherapy. *Radiotherapy and Oncology*, 76(2):S29.

[15] Kirisits, C., Siebert, F.-A., Baltas, D., De Brabandere, M., Hellebust, T. P., Berger, D., and Venselaar, J. (2007). Accuracy of volume and DVH parameters determined with different brachytherapy treatment planning systems. *Radiotherapy and Oncology*, 84(3):290–297.

[16] Lessard, E. and Pouliot, J. (2001). Inverse planning anatomy-based dose optimization for HDR-brachytherapy of the prostate using fast simulated annealing algorithm and dedicated objective function. *Medical Physics*, 28(5):773–779.

[17] Li, S.-C. and Yu, T.-L. (2017). Speeding up DSMGA-II on CUDA platform. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 809–816. ACM.

[18] Luong, N. H., Alderliesten, T., Bel, A., Niatsetski, Y., and Bosman, P. A. N. (2018a). Application and benchmarking of multi-objective evolutionary algorithms on high-dose-rate brachytherapy planning for prostate cancer treatment. *Swarm and Evolutionary Computation*, 40:37–52.

[19] Luong, N. H., Alderliesten, T., Pieters, B. R., Bel, A., Niatsetski, Y., and Bosman, P. A. N. (2018b). Fast and insightful bi-objective HDR prostate brachytherapy planning. *Radiotherapy and Oncology*, 127:S130.

[20] Maree, S. C., Luong, N. H., Kooreman, E. S., van Wieringen, N., Bel, A., Hinnen, K. A., Westerveld, H., Pieters, B. R., Bosman, P. A. N., and Alderliesten, T. (2019). Evaluation of bi-objective treatment planning for high-dose-rate prostate brachytherapy – A retrospective observer study. *Brachytherapy*, 18(3):396–403.

[21] Meigooni, A. S., Hoskin, P., Baltas, D., and Venselaar, J. L. M. (2012). Introduction and innovations in brachytherapy. In *Comprehensive Brachytherapy*, pages 30–35. CRC Press.

[22] Niemierko, A. and Goitein, M. (1990). Random sampling for evaluating treatment plans. *Medical Physics*, 17(5):753–762.

[23] NVIDIA Corporation (2018). CUDA C programming guide v10.0.130.

[24] Ortega, G., Filatovas, E., Garzón, E. M., and Casado, L. G. (2017). Non-dominated sorting procedure for Pareto dominance ranking on multicore CPU and/or GPU. *Journal of Global Optimization*, 69(3):607–627.

[25] Pratx, G. and Xing, L. (2011). GPU computing in medical physics: A review. *Medical Physics*, 38(5):2685–2697.

[26] Rivard, M. J., Coursey, B. M., DeWerd, L. A., Hanson, W. F., Saiful Huq, M., Ibbott, G. S., Mitch, M. G., Nath, R., and Williamson, J. F. (2004). Update of AAPM task group no. 43 report: A revised AAPM protocol for brachytherapy dose calculations. *Medical Physics*, 31(3):633–674.

[27] Ruciński, A., Gajewski, J., Olko, P., Rinaldi, I., Patera, V., and Schiavi, A. (2016). GPU-accelerated Monte Carlo code for fast dose recalculation in proton beam therapy. *Acta Physica Polonica B*, 48(10):1625.

[28] Siauw, T., Cunha, A., Atamtürk, A., Hsu, I.-C., Pouliot, J., and Goldberg, K. (2011). IPIP: A new approach to inverse planning for HDR brachytherapy by directly optimizing dosimetric indices. *Medical Physics*, 38(7):4045–4051.

[29] Tian, Z., Li, Y., Hassan-Rezaeian, N., Jiang, S. B., and Jia, X. (2017). Moving GPU-OpenCL-based Monte Carlo dose calculation toward clinical use: Automatic beam commissioning and source sampling for treatment plan dose calculation. *Journal of Applied Clinical Medical Physics*, 18(2):69–84.

[30] Wong, M. L. (2009). Parallel multi-objective evolutionary algorithms on graphics processing units. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion: Late Breaking Papers*, pages 2515–2522. ACM.

[31] Yamada, Y., Rogers, L., Demanes, D. J., Morton, G., Prestidge, B. R., Pouliot, J., Gil'ad, N. C., Zaider, M., Ghilezan, M., and Hsu, I.-C. (2012). American brachytherapy society consensus guidelines for high-dose-rate prostate brachytherapy. *Brachytherapy*, 11(1):20–32.

Appendices

5.A. MO-RV-GOMEA

Unless stated otherwise, all default parameters are used for MO-RV-GOMEA [4].

A population size of 96 was used, based on preliminary experiments. To prevent having idle threads during the dose calculation, the population size should be divisible by the height of a thread block. As we use 16×16 thread blocks, the population size is divisible by 16.

The linkage model used by MO-RV-GOMEA is learned based on the Euclidean distance between dwell positions [18]. In this work we omit linkage sets of all sizes smaller than 5, which is a value that was empirically determined based on preliminary experiments, because these linkage sets only cause small changes to a solution, but take a relatively long time to evaluate. This is caused by the fact that partial evaluations can be applied to the dose calculation, but not to the calculation of the DV-indices, making the DV-index calculation relatively computationally expensive for linkage sets with few elements. Furthermore, the majority of the linkage sets in a linkage tree consists of relatively small elements, meaning that the removal of these linkage sets can save a substantial amount of time. For instance, given ℓ problem variables, as many as ℓ out of $2\ell - 1$ linkage sets of a full linkage tree are univariate elements.

5.B. GPU-Accelerated Bi-Objective Treatment Planning

Figures 5.B.1 to 5.B.4 show all bi-objectively optimized treatment plans for all 18 patients used in this study. To show variance of treatment plan outcomes, the combined set of outcomes from 30 runs of the optimization are shown, where each run used 100,000 DC points and had a time limit of 3 minutes.

In the plots in the middle column, all DV indices of each treatment plan are displayed in a separate color according to the legend, meaning that each treatment plan is represented by five different data points in different colors, by definition aligned vertically, as each plan is defined by a single value for its LCI. In the plots in the rightmost column, the coverage of the prostate and vesicles are shown for each treatment plan. Each treatment plan is represented by two data points in different colors, by definition aligned horizontally.

Tables 5.B.1 and 5.B.2 show means and variances of plans optimized by GOMEA for different settings for the time limit and the number of DC points. As the variance of a large set of plans is difficult to quantify in a meaningful way, we list the means and variances of plans selected from the resulting set of plans. To report the variance of the LCI, we select the plan with LSI > 0 and maximum LCI value. To report the variance of the LSI, we select the plan with LCI > 0 and maximum LSI value.

Table 5.B.1 shows that the required number of DC points to consistently find plans satisfying the clinical protocol is patient specific.

The most striking result in Table 5.B.2 is the fact that, for patients 6 and 8, using a set of 20,000 DC points did not always lead to a plan that satisfied all clinical criteria. This was the case for patient 6 using time limits of 60 and 180 seconds, and for patient 8 using time limits of 300 and 600 seconds. In all these 4 cases, after reevaluation of all DV indices on 500,000 DC points, 1 out of 30 runs did not include a plan satisfying all clinical criteria. This was no longer the case when a set of 100,000 DC points was used, given enough time, so we can safely contribute this effect to variance caused by the limited number of DC points. As indicated in the discussion of the main chapter, the accuracy of DV index calculations could also be improved by only sampling DC points in the regions nearby active dwell positions.

Table 5.B.1: Means and variances of selected plans resulting from 30 runs of GOMEA with a time limit of 10 minutes, and an increasing number of DC points. The means and variances of the LCI (%) were computed for the plan with LSI > 0 and maximum LCI value. The means and variances of the LSI (% of planning-aim dose) were computed for the plan with LCI > 0 and maximum LSI value. Values violating the clinical protocol for at least 1 out of 30 runs are shaded in gray.

Patient	5,000 DC pts		10,000 DC pts		20,000 DC pts		50,000 DC pts		100,000 DC pts		200,000 DC pts	
	LCI	LSI	LCI	LSI	LCI	LSI	LCI	LSI	LCI	LSI	LCI	LSI
1	0.3±0.8	0.6±1.8	1.0±0.6	2.1±1.3	1.8±0.4	3.7±0.7	2.1±0.2	4.5±0.4	2.3±0.1	4.6±0.3	2.3±0.1	4.7±0.3
2	3.9±0.5	8.6±0.3	4.5±0.2	8.9±0.2	4.8±0.1	9.1±0.2	4.9±0.0	9.3±0.1	5.0±0.0	9.3±0.1	5.0±0.0	9.3±0.1
3	2.4±0.5	3.2±0.9	3.3±0.3	4.7±1.0	3.7±0.2	5.6±0.7	4.1±0.1	6.4±0.2	4.2±0.1	6.7±0.2	4.2±0.1	6.8±0.2
4	-3.1±0.7	-5.7±1.0	-2.0±0.5	-4.3±1.0	-1.6±0.3	-3.4±0.6	-1.3±0.2	-2.7±0.4	-1.2±0.2	-2.5±0.4	-1.2±0.2	-2.4±0.5
5	0.9±0.8	1.6±1.2	1.6±0.8	2.6±1.4	2.5±0.2	4.2±0.6	2.8±0.2	5.0±0.5	2.9±0.1	5.3±0.3	3.1±0.1	5.6±0.2
6	-0.6±0.7	-1.0±1.2	0.8±0.5	1.3±1.1	1.4±0.3	2.5±0.5	1.8±0.2	3.1±0.4	2.0±0.1	3.5±0.3	2.1±0.1	3.5±0.2
7	-1.1±0.7	-2.1±1.6	-0.1±0.4	-0.1±0.8	0.4±0.3	0.9±0.6	0.9±0.2	1.8±0.4	1.0±0.2	2.1±0.3	1.1±0.2	2.3±0.3
8	0.2±0.5	0.1±1.1	1.0±0.6	1.7±0.9	1.5±0.6	2.7±0.9	2.0±0.3	3.3±0.7	2.2±0.2	3.5±0.4	2.3±0.2	3.8±0.3
9	1.3±0.9	2.9±2.1	2.4±0.5	4.8±1.2	3.0±0.2	5.9±0.7	3.2±0.2	6.7±0.7	3.4±0.1	7.3±0.3	3.5±0.1	7.4±0.2
10	2.8±0.5	7.0±0.6	3.6±0.2	7.8±0.3	4.0±0.1	8.2±0.1	4.2±0.1	8.3±0.1	4.3±0.1	8.3±0.1	4.3±0.0	8.3±0.1
11	3.2±0.4	6.9±1.3	4.0±0.2	8.4±0.7	4.3±0.1	8.9±0.4	4.5±0.0	9.2±0.1	4.6±0.0	9.2±0.1	4.6±0.0	9.2±0.1
12	2.8±0.2	7.4±0.3	3.4±0.2	8.0±0.2	3.7±0.1	8.2±0.1	3.9±0.1	8.4±0.1	4.0±0.1	8.4±0.1	4.0±0.0	8.4±0.1
13	1.1±0.7	2.3±1.3	2.0±0.6	4.0±1.3	2.6±0.3	4.9±0.6	3.0±0.1	5.7±0.4	3.1±0.1	5.9±0.3	3.2±0.1	6.0±0.3
14	-1.7±0.4	-5.5±1.7	-0.8±0.4	-2.5±1.1	-0.3±0.2	-1.2±0.8	0.1±0.1	0.2±0.3	0.1±0.1	0.4±0.3	0.2±0.1	0.5±0.2
15	1.0±3.3	2.2±2.3	2.6±0.7	3.8±1.3	3.1±0.4	4.7±1.1	3.6±0.3	5.6±0.8	3.8±0.2	5.9±0.5	3.9±0.1	6.2±0.2
16	3.8±0.4	8.3±0.3	4.5±0.1	8.8±0.2	4.8±0.1	9.1±0.1	4.9±0.0	9.3±0.1	5.0±0.0	9.3±0.1	5.0±0.0	9.3±0.1
17	3.8±0.4	6.5±0.9	4.3±0.3	6.8±1.2	4.6±0.1	7.3±0.5	4.8±0.1	7.8±0.3	4.8±0.0	8.1±0.1	4.9±0.0	8.1±0.1
18	-2.6±0.7	-6.0±1.7	-1.5±0.4	-3.7±0.9	-0.9±0.2	-2.1±0.7	-0.5±0.1	-1.2±0.3	-0.4±0.1	-0.9±0.2	-0.3±0.1	-0.6±0.2

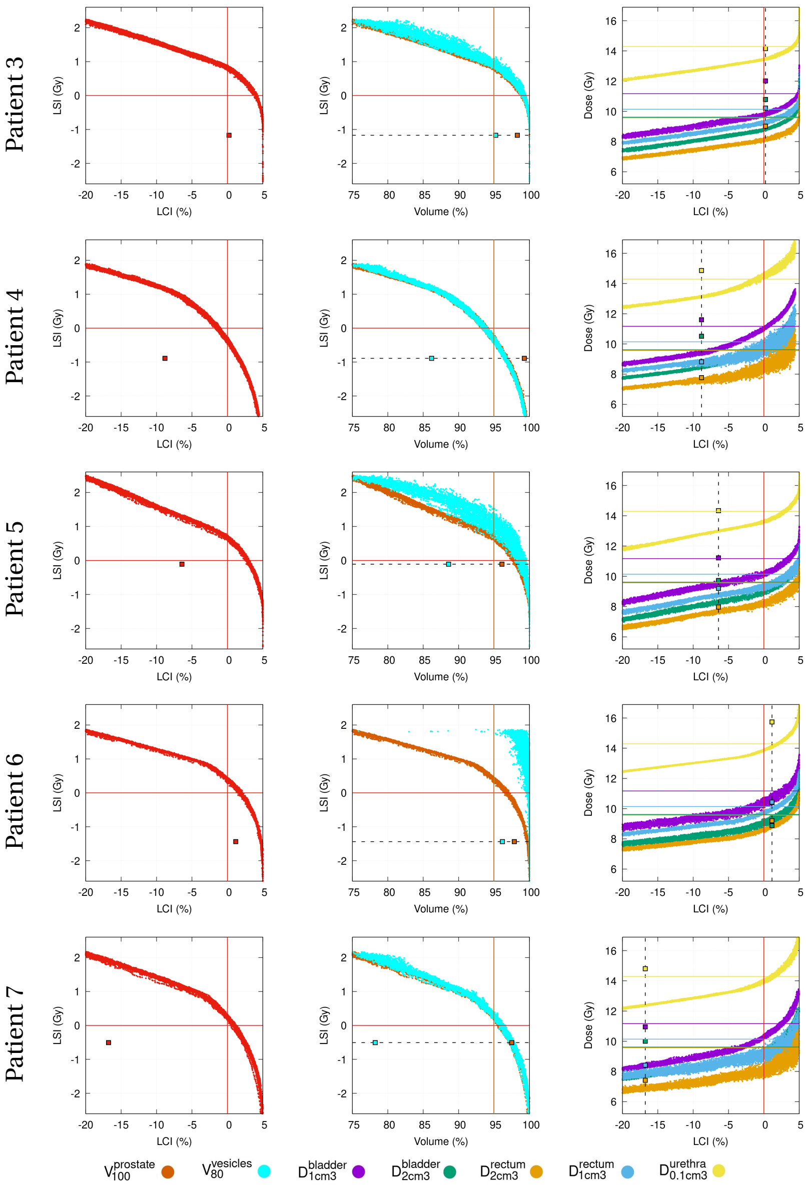


Figure 5.B.1: Combined results of 30 optimization runs for patients 3 to 7, optimized for 3 minutes on 100,000 DC points, showing corresponding values of all DV indices for each value of the LCI and LSI. DV indices of clinical plans are shown as squares in colors corresponding to the respective index.

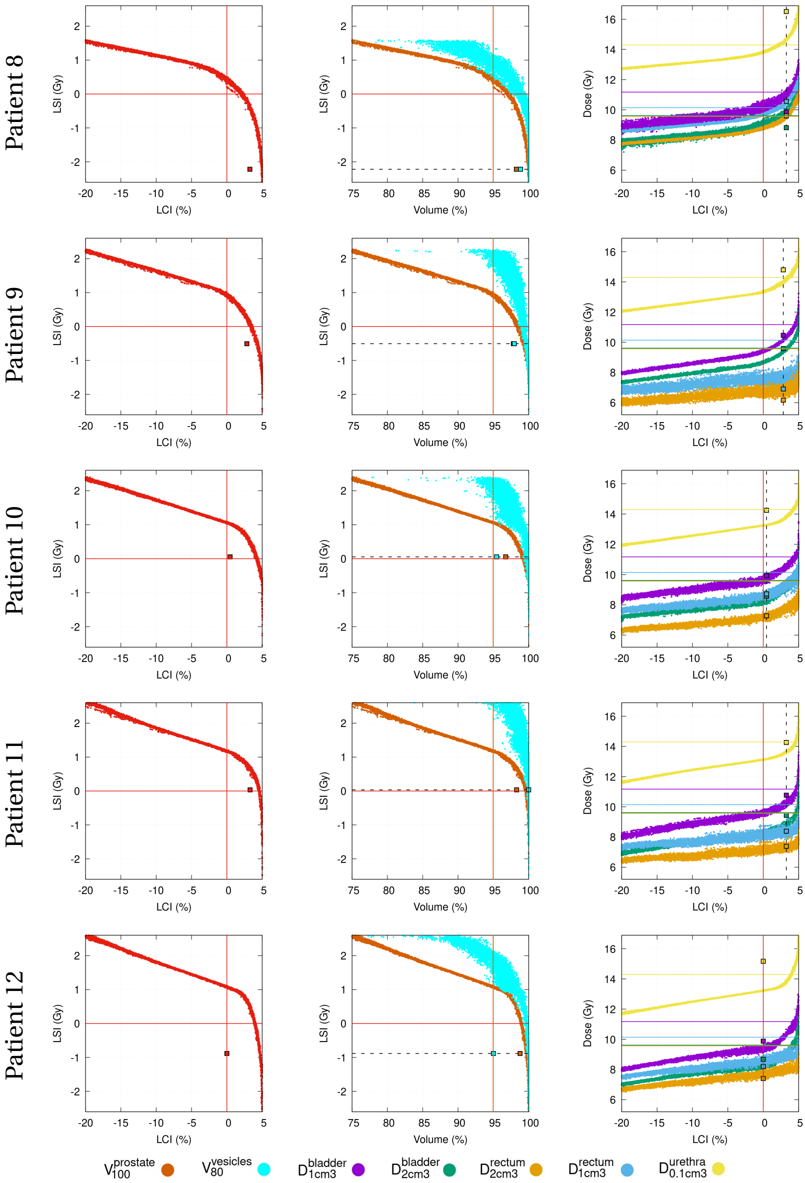


Figure 5.B.2: Combined results of 30 optimization runs for patients 8 to 12, optimized for 3 minutes on 100,000 DC points, showing corresponding values of all DV indices for each value of the LCI and LSI. DV indices of clinical plans are shown as squares in colors corresponding to the respective index.

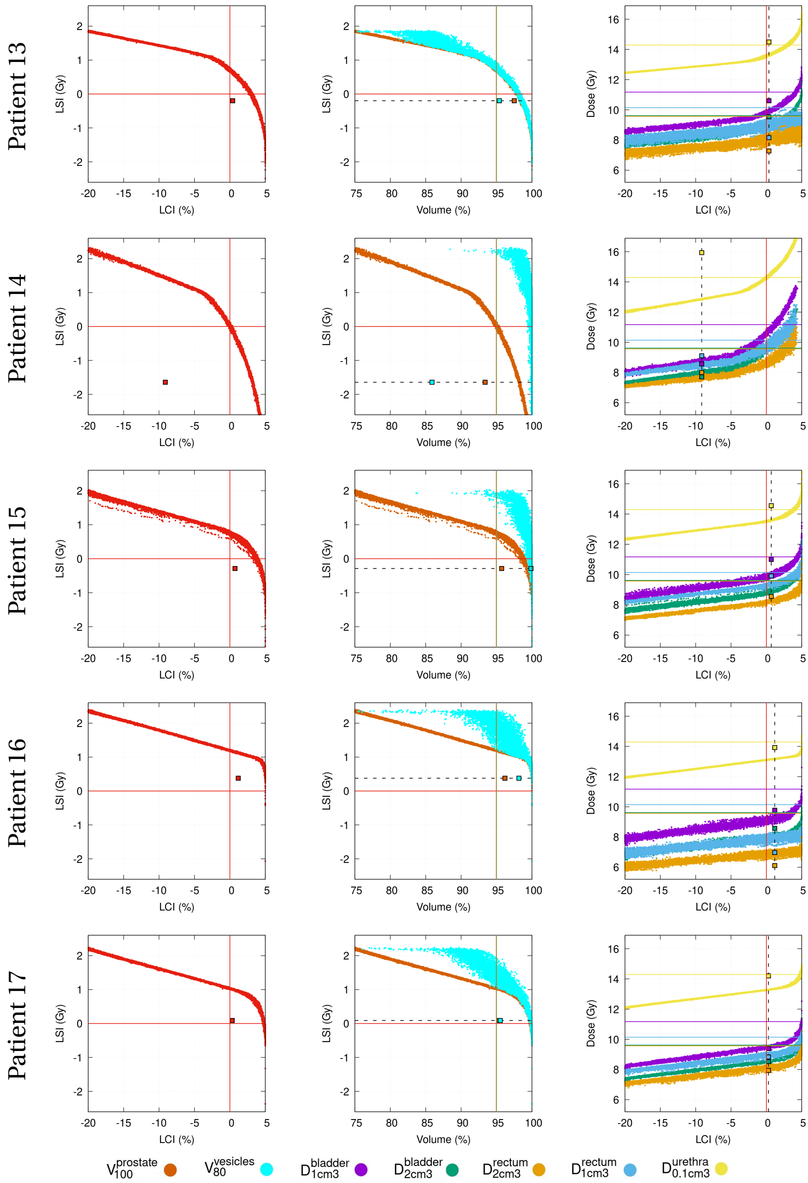


Figure 5.B.3: Combined results of 30 optimization runs for patients 13 to 17, optimized for 3 minutes on 100,000 DC points, showing corresponding values of all DV indices for each value of the LCI and LSI. DV indices of clinical plans are shown as squares in colors corresponding to the respective index.

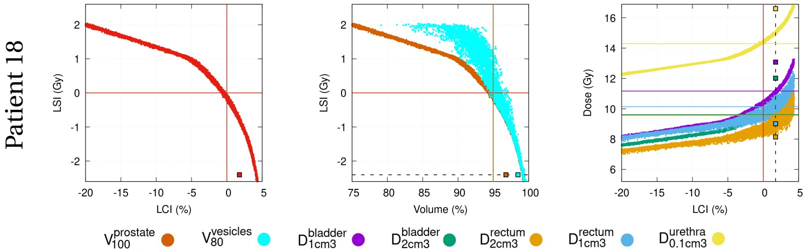


Figure 5.B.4: Combined results of 30 optimization runs for patient 18, optimized for 3 minutes on 100,000 DC points, showing corresponding values of all DV indices for each value of the LCI and LSI. DV indices of clinical plans are shown as squares in colors corresponding to the respective index.

Table 5.B.2: Means and variances of selected plans resulting from 30 runs of GOMEA with the specified number of DC points, for an increasing time limit in seconds. The means and variances of the LCI (%) were computed for the plan with LSI > 0 and maximum LCI value. The means and variances of the LSI (% of planning-aim dose) were computed for the plan with LCI > 0 and maximum LSI value. Values violating the clinical protocol for at least 1 out of 30 runs are shaded in gray.

Patient	DC pts	30 s		60 s		180 s		300 s		600 s	
		LCI	LSI	LCI	LSI	LCI	LSI	LCI	LSI	LCI	LSI
1	20k	1.4±0.3	2.7±0.7	1.5±0.2	3.1±0.4	1.6±0.4	3.4±0.8	1.8±0.2	3.6±0.5	1.8±0.4	3.7±0.7
	100k	1.0±0.3	1.7±0.4	1.5±0.2	2.8±0.4	2.0±0.2	4.0±0.3	2.1±0.1	4.3±0.3	2.3±0.1	4.6±0.3
2	20k	4.8±0.1	8.9±0.2	4.8±0.1	9.1±0.2	4.8±0.1	9.1±0.2	4.8±0.1	9.0±0.2	4.8±0.1	9.1±0.2
	100k	4.8±0.1	8.6±0.2	4.9±0.0	8.9±0.1	4.9±0.0	9.2±0.2	5.0±0.0	9.3±0.1	5.0±0.0	9.3±0.1
3	20k	3.5±0.2	5.0±0.6	3.7±0.2	5.4±0.6	3.7±0.2	5.5±0.5	3.8±0.2	5.5±0.5	3.7±0.2	5.6±0.7
	100k	3.1±0.2	4.3±0.4	3.6±0.1	5.3±0.3	4.0±0.1	6.2±0.2	4.1±0.1	6.4±0.2	4.2±0.1	6.7±0.2
4	20k	-2.0±0.4	-4.0±0.8	-1.9±0.3	-3.7±0.8	-1.7±0.3	-3.6±0.8	-1.8±0.4	-3.7±0.8	-1.6±0.3	-3.4±0.6
	100k	-2.5±0.3	-4.8±0.8	-1.9±0.3	-3.8±0.6	-1.4±0.2	-2.8±0.3	-1.3±0.2	-2.7±0.4	-1.2±0.2	-2.5±0.4
5	20k	2.0±0.5	3.5±0.8	2.3±0.4	4.0±0.7	2.3±0.3	4.2±0.7	2.3±0.4	4.0±0.9	2.5±0.2	4.2±0.6
	100k	2.0±0.3	3.3±0.5	2.4±0.2	4.2±0.4	2.8±0.2	5.0±0.3	2.9±0.1	5.1±0.4	2.9±0.1	5.3±0.3
6	20k	1.0±0.4	1.6±0.6	1.2±0.6	1.9±0.8	1.3±0.4	2.2±0.7	1.3±0.4	2.3±0.6	1.4±0.3	2.5±0.5
	100k	0.3±0.3	0.4±0.5	1.2±0.2	2.0±0.3	1.7±0.1	2.8±0.3	1.9±0.1	3.2±0.3	2.0±0.1	3.5±0.3
7	20k	0.1±0.3	0.3±0.7	0.3±0.4	0.5±0.7	0.3±0.3	0.7±0.6	0.5±0.3	1.0±0.6	0.4±0.3	0.9±0.6
	100k	-0.4±0.4	-0.8±0.7	0.5±0.2	1.0±0.4	0.9±0.2	1.8±0.3	1.0±0.2	1.9±0.3	1.0±0.2	2.1±0.3
8	20k	1.3±0.4	2.1±0.5	1.4±0.4	2.3±0.6	1.7±0.4	2.6±0.8	1.7±0.4	2.7±0.7	1.5±0.6	2.7±0.9
	100k	0.3±0.4	0.5±0.6	1.4±0.2	2.1±0.3	2.0±0.2	3.1±0.3	2.1±0.2	3.4±0.4	2.2±0.2	3.5±0.4
9	20k	2.7±0.2	5.6±0.5	2.9±0.2	6.0±0.6	2.9±0.2	6.2±0.5	2.9±0.2	5.9±1.0	3.0±0.2	5.9±0.7
	100k	2.8±0.1	5.6±0.4	3.1±0.1	6.3±0.4	3.3±0.1	6.9±0.3	3.3±0.1	7.1±0.3	3.4±0.1	7.3±0.3
10	20k	3.8±0.1	7.7±0.2	4.0±0.1	7.9±0.1	4.0±0.1	8.1±0.1	4.0±0.1	8.1±0.1	4.0±0.1	8.2±0.1
	100k	2.9±0.4	6.3±0.5	3.8±0.1	7.6±0.2	4.2±0.1	8.0±0.1	4.3±0.1	8.2±0.1	4.3±0.1	8.3±0.1
11	20k	4.2±0.1	8.4±0.3	4.3±0.2	8.6±0.3	4.3±0.1	8.9±0.2	4.3±0.1	8.8±0.3	4.3±0.1	8.9±0.4
	100k	3.8±0.2	7.4±0.4	4.3±0.1	8.4±0.2	4.5±0.1	9.0±0.1	4.5±0.0	9.1±0.1	4.6±0.0	9.2±0.1
12	20k	3.7±0.1	8.0±0.1	3.7±0.1	8.1±0.1	3.7±0.1	8.2±0.1	3.7±0.1	8.2±0.1	3.7±0.1	8.2±0.1
	100k	3.6±0.1	7.7±0.1	3.8±0.1	7.9±0.1	3.9±0.1	8.3±0.1	3.9±0.0	8.3±0.1	4.0±0.1	8.4±0.1
13	20k	2.2±0.4	3.9±0.8	2.4±0.4	4.3±0.8	2.6±0.3	4.8±0.7	2.6±0.4	5.0±0.8	2.6±0.3	4.9±0.6
	100k	1.5±0.3	2.4±0.6	2.4±0.2	4.0±0.4	2.9±0.2	5.2±0.4	3.0±0.1	5.6±0.4	3.1±0.1	5.9±0.3
14	20k	-0.6±0.3	-1.7±0.5	-0.4±0.2	-1.3±0.7	-0.3±0.2	-1.0±0.9	-0.4±0.2	-1.1±0.7	-0.3±0.2	-1.2±0.8
	100k	-2.1±0.5	-5.8±1.6	-0.6±0.2	-1.6±0.4	-0.1±0.1	-0.2±0.3	0.0±0.1	0.1±0.3	0.1±0.1	0.4±0.3
15	20k	3.0±0.5	4.4±1.0	3.0±0.6	4.2±1.3	3.0±0.5	4.3±1.1	3.2±0.4	4.6±0.9	3.1±0.4	4.7±1.1
	100k	2.5±0.2	3.6±0.6	3.2±0.2	4.9±0.3	3.7±0.2	5.7±0.2	3.8±0.1	5.9±0.3	3.8±0.2	5.9±0.5
16	20k	4.9±0.1	8.8±0.1	4.8±0.1	8.9±0.2	4.8±0.1	9.1±0.1	4.8±0.1	9.1±0.1	4.8±0.1	9.1±0.1
	100k	4.9±0.1	8.0±0.3	5.0±0.0	8.7±0.1	5.0±0.0	9.1±0.1	5.0±0.0	9.2±0.1	5.0±0.0	9.3±0.1
17	20k	4.5±0.1	7.3±0.4	4.6±0.1	7.5±0.3	4.6±0.1	7.5±0.6	4.6±0.1	7.5±0.3	4.6±0.1	7.3±0.5
	100k	4.5±0.1	7.0±0.2	4.6±0.1	7.4±0.1	4.8±0.1	7.9±0.1	4.8±0.0	7.9±0.1	4.8±0.0	8.1±0.1
18	20k	-1.3±0.3	-2.7±0.5	-1.0±0.3	-2.4±0.6	-1.0±0.2	-2.3±0.5	-1.0±0.2	-2.3±0.6	-0.9±0.2	-2.1±0.7
	100k	-2.9±0.6	-6.9±1.5	-1.1±0.2	-2.4±0.4	-0.6±0.2	-1.3±0.4	-0.5±0.2	-1.1±0.4	-0.4±0.1	-0.9±0.2

6

GPU-Accelerated Parallel Gene-Pool Optimal Mixing Applied to Multi-Objective Deformable Image Registration

This chapter is based on the following publication: Bouter, A., Alderliesten, T., and Bosman, P. A. N. (2021b). GPU-accelerated parallel gene-pool optimal mixing applied to multi-objective deformable image registration. In *IEEE Congress on Evolutionary Computation*, pages 2539–2548. IEEE

Abstract

The Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm (RV-GOMEA) has previously been successfully used to achieve highly scalable optimization of various real-world problems in a gray-box optimization setting. Deformable Image Registration (DIR) is a multi-objective problem, aimed at finding the most likely non-rigid deformation of a given source image so that it matches a given target image. We specifically consider the case where the deformation model allows for finite-element-type modeling of tissue properties. This optimization problem is non-smooth, necessitating techniques like EAs to get good results. Though the objectives of DIR are non-separable, non-neighboring regions of the deformation grid are conditionally independent. We show that GOMEA allows to exploit such knowledge through the large-scale parallel application of variation steps, where each is only accepted when leading to an improvement, on a Graphics Processing Unit (GPU). On various 2-dimensional DIR problems, we find that this way, similar results can be achieved as when sequential processing is performed, while allowing for substantial speed-ups (up to a factor of 111) for the highest-dimensional problems (i.e., the highest deformation-grid resolution). This work opens the door to the extension of this type of DIR to larger (3-dimensional) deformation grids, and its application to other real-world problems.

6.1. INTRODUCTION

Evolutionary Algorithms (EAs) are used for the optimization of many real-world problems [11, 12]. Contributing to this is the fact that often there are multiple objectives of importance in practice and EAs are considered to be among the state of the art in the field of multi-objective optimization [14]. Though many optimization problems to which EAs are applied are considered from a Black-Box Optimization (BBO) perspective, meaning any problem-specific knowledge is ignored, it is possible in various real-world problems to use problem knowledge to the advantage of the EA, allowing for substantially improved performance and scalability. Such a setting, where partial problem knowledge is used to the advantage of the optimization algorithm, is considered to be a Gray-Box Optimization (GBO) setting. Various GBO settings that allow different kinds of problem-specific information to be used have previously been shown to allow for large improvements in efficiency [10, 15]. Here, we consider the GBO setting that allows for the use of partial evaluations, i.e., the efficient evaluation of the impact on fitness of partial modifications of solutions. Such a GBO setting is possible in various real-world problems [4, 18] and is by no means an indication that the problem is trivial. Problems for which partial evaluations are possible are potentially multi-objective, multi-modal, non-smooth, and non-separable, thus still warranting the use of evolutionary optimization methods.

It is known that with the Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA) excellent performance and scalability can be achieved in a GBO setting [5], as partial evaluations can then be efficiently exploited. Consequently, GOMEA has previously been applied to the real-world optimization of treatment plans for internal radiotherapy [18] and medical Deformable Image Registration (DIR) [4] in a GBO setting with clinically desirable results and superior performance over black-box EAs. The excellent performance of GOMEA is mainly caused by the fact that variation is applied to (small) sub-sets of variables and the variation step is only accepted if it results in an improvement of the fitness. This results in a high selection pressure for local improvements, allowing GOMEA to scale to problems of thousands of variables within reasonable computation times [5].

As the efficiency of optimization is often critically important for real-world applications, further efficiency enhancements may be necessary, especially for very large-scale problems. Especially for EAs, parallelization is a frequently used technique to speed up the optimization process, as it is straightforward to perform the evaluation of the population in parallel. GOMEA holds the potential for an even larger degree of parallelizability due to the local nature of the Gene-pool Optimal Mixing (GOM) variation operator, because variation may be applied to different subsets of variables in parallel as long as these subsets of variables are conditionally independent. This improvement in parallelizability may arise even in non-separable problems, as conditional independence is a sufficient condition for the parallelization of the GOM variation steps. The parallelizability of such variation steps is mainly of interest for large-scale problems with overlapping strong dependencies, because such problems inhibit a large number of conditional dependencies.

Because a very large degree of parallelization may be obtained, due to combined effects of parallel evaluation of individuals in the population and different steps of variation within each individual, using the large-scale parallel computing power of Graphics Processing Units (GPUs) becomes a very interesting possibility. GPUs are frequently used for high-performance computing, have been a driving force behind the popularity of deep learning, and already have seen various different applications in the field of evolutionary computation [13, 19].

In this chapter, we show how a large number of variation operations of GOM may be applied in parallel to the same individual in the population, leading to an unprecedented degree of parallelizability of an EA. We demonstrate the effects of our approach through the real-world application of DIR [21], an important problem in the field of medical imaging, where the optimization goal is to find an anatomically feasible deformation from a given source image to a given target image.

6.2. Gray-Box Optimization

We consider real-valued optimization functions $f(x) : \mathbb{R}^\ell \rightarrow \mathbb{R}$ subject to minimization. The variables to be optimized are denoted $X = \{X_1, X_2, \ldots, X_\ell\}$, which are indexed through $\mathbb{I} = [1, 2, \ldots, \ell]$, with $X_i \in \mathbb{R}$. A realization of the problem variables, i.e., a candidate solution to the optimization problem, is denoted $\boldsymbol{x} = [x_1, x_2, \ldots, x_\ell]$.

In this chapter, we focus on a GBO setting that allows for partial evaluations, i.e., a more efficient fitness evaluation following the modification of a subset of variables. As such, it is known that the optimization function $f(\boldsymbol{x})$ is constructed using a set of subfunctions $F = \{f_1, f_2, \ldots, f_q\}$. Each subfunction $f_i \in F$ takes a subset of problem variables, with indices $\mathcal{I}_i \subseteq \mathbb{I}$, as its input. A subset of the variables of a solution $\boldsymbol{x}$, restricted to the indices in $\mathcal{I}_j$, is denoted $\boldsymbol{x}_{\mathcal{I}_j}$. The outputs of the subfunctions in F are aggregated using a binary operator $\oplus$ that has a known inverse $\ominus$, e.g., addition or multiplication [3]. In some GBO problems, the aggregated result of the subfunctions is passed through a (potentially non-linear) function $g : \mathbb{R} \rightarrow \mathbb{R}$. When this is the case, in order to perform partial evaluations, the intermediate result of each subfunction $f_i \in F$ should be stored in memory for each solution.

An optimization problem in a GBO setting can therefore be formulated in the following way, as previously stated in [7]:

$$f(\boldsymbol{x}) = g\left(f_1(\boldsymbol{x}_{\mathcal{I}_1}) \oplus f_2(\boldsymbol{x}_{\mathcal{I}_2}) \oplus \cdots \oplus f_q(\boldsymbol{x}_{\mathcal{I}_q})\right). \tag{6.1}$$

To describe dependencies in an optimization problem, a Variable Interaction Graph (VIG) [25] can be constructed. In a VIG $G = (V, E)$, each vertex $u \in V$ refers to a problem variable $X_u \in X$, and each edge $(u, v) \in E$ denotes that a dependency exists between variables X_u and X_v. When the function g in Equation 6.1 is the identity function, a VIG can be straightforwardly constructed by adding an edge between each pair of variables that jointly occur in the required input variables of at least one subfunction, i.e., the set of edges E consists of all (u, v) such that a subfunction $f_j \in F$ exists with $\{u, v\} \subseteq \mathcal{I}_j$. When the function g in Equation 6.1 is not the identity function, the VIG is a fully connected graph. It may however be possible to filter edges from this VIG based on dependency strength, indicated by either problem information or learned during optimization. In this chapter, we focus on the case where g is the identity function, as this case corresponds to the objectives of the DIR application introduced in Section 6.4.

6.3. GENE-POOL OPTIMAL MIXING

GOMEA was first introduced for the domain of discrete optimization [23], and later adapted to the domain of real-valued optimization [5]. This work is focused on the multi-objective real-valued variant of GOMEA, named Multi-Objective Real-Valued GOMEA (MO-RV-GOMEA) [5].

The main strength of GOMEA is the Gene-pool Optimal Mixing (GOM) variation operator. When GOM is applied to a solution $\boldsymbol{x}$, variation is iteratively applied to subsets of variables of $\boldsymbol{x}$. Each variation step is only accepted if it results in the improvement of the fitness of $\boldsymbol{x}$. Otherwise, $\boldsymbol{x}$ is reverted to its state before this variation step. In the multi-objective variant of GOMEA, an iteration of GOM is accepted when it leads to domination of the parent, or to a solution that is non-dominated by the elitist archive. The subsets of variables to which GOM is applied are explicitly defined by a linkage model, which describes the dependencies between variables.

Clustering is used to spread the population across the different regions of the Pareto front. Elitist solutions are stored in an elitist archive with a specified target size, which uses an adaptive gridding technique [17]. A separate linkage model is learned for each cluster. When the elitist archive reaches a size of more than 20% over the target size, the size of the archive is dynamically reduced by discretizing the objective space and filtering solutions such that each cell of the discretized objective space contains no more than one solution of the archive. As we make no changes to clustering and elitist archiving in MO-RV-GOMEA, we refer the interested reader for more details to [5] and below focus on the variation operator.

6.3.1. LINKAGE MODEL

To which subsets of variables GOM is applied, is governed by a linkage model that describes dependencies between problem variables. The linkage model is an explicitly defined Family Of Subsets (FOS) $\mathcal{F}$ that is a subset of the powerset of $\mathcal{I}$. Each FOS element $\mathcal{F}_j \in \mathcal{F}$, also called a linkage set, describes a subset of variables to which a variation step of GOM is applied. For the most effective variation, dependent variables should be within the same FOS element, and independent variables should be in different FOS elements. Often, in a BBO setting, the FOS is learned during a run, whereas in a GBO setting, the FOS is pre-specified [18].

For each FOS element $\mathcal{F}_j \in \mathcal{F}$, a sampling model $\mathcal{M}_j$ is maintained from which new variables are sampled. In the case of real-valued variables, i.e., RV-GOMEA, each sampling model is essentially a multivariate Gaussian probability distribution that is estimated with maximum likelihood. Details are described in [5].

6.3.2. PARALLEL GENE-POOL OPTIMAL MIXING

We consider a GBO problem with a known VIG, constructed as described in Section 6.2. When GOM is applied to a solution $\boldsymbol{x}$ using a FOS element $\mathcal{F}_j$, the effect of this variation step on the fitness is evaluated using one or more partial evaluations. Such partial evaluations depend only on a limited number of variables. Specifically, this depends on all variables X_u with $u \in \mathcal{F}_j$, and all variables X_v for which an edge $(u, v) \in E$ exists to a variable X_u. A consequence of this is that several iterations of GOM, in which different linkage sets are used, but applied to the same solution, may be completely independent, because their respective partial evaluations are dependent on disjoint sets of variables. Therefore, the decision to accept or reject these iterations of GOM may be made independently, and in parallel.

In Figure 6.1 we show a possible VIG, where the nodes in blue indicate that GOM is applied to these variables, i.e., $\mathcal{F}_i = \{1,2,3,4,5\}$. All variables marked in red are dependent on at least one variable in blue, and remaining variables are independent. Therefore, an independent iteration of GOM may be applied to a subset of the unmarked variables, e.g., $\{10,11,12,13\}$. Note that when g in Equation 6.1 is not the identity function, there may be a dependence between variables not connected in the VIG. Because dependencies introduced at this level are likely weak, we argue that the decisions to accept or reject the parallel iterations of GOM can likely still be done in parallel without large repercussions. However, an additional (full) evaluation may be required to calculate the final objective value of the solution to which parallel GOM has been applied to ensure that the joint effect of variation is accounted for correctly. Further investigation into such problems is necessary to fully understand the conditions under which parallel GOM can be safely performed, but outside of the scope of this chapter, as this chapter is focused on the DIR application.

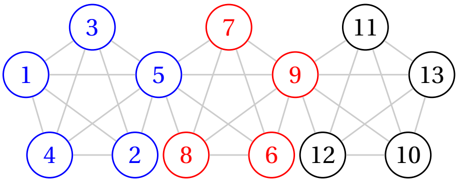


Figure 6.1: A potential VIG showing which variables would be dependent (in red) if GOM were to be applied to $\mathcal{F}_j = \{1,2,3,4,5\}$ (in blue). Independent variables are shown in black. A different iteration of GOM may be applied to a subset of independent variables in parallel.

To formalize the parallel application of GOM, we define a set $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{|\mathcal{G}|}\}$ that defines for which sets of FOS elements GOM is applied in parallel. Given any $\mathcal{G}_i \in \mathcal{G}$, GOM is applied in parallel to all $\mathcal{F}_u$ with $u \in \mathcal{G}_i$. The requirement to apply GOM to two FOS elements $\mathcal{F}_i$ and $\mathcal{F}_j$ in parallel is that $\mathcal{F}_i$ and $\mathcal{F}_j$ have no variables in common, and no dependency exists between any variable in $\mathcal{F}_i$ and any variable in $\mathcal{F}_j$, i.e., $\mathcal{F}_i \cap \mathcal{F}_j = \emptyset$, and for any $i, j \in \mathcal{G}_k$ there exists no $u \in \mathcal{F}_i$ and $v \in \mathcal{F}_j$ such that $(u, v) \in E$. Given the order $\mathcal{O}$ in which GOM is normally (i.e., in serial) applied, i.e., a permutation of $[1, 2, \ldots, |\mathcal{F}|]$, iterations of GOM that are consecutive and independent (i.e., $\mathcal{F}_i$ and $\mathcal{F}_j$ satisfy the conditions to be in the same element of $\mathcal{G}$) may be applied in parallel without having an impact on the algorithm.

Constructing $\mathcal{G}$ from the sequence of $\mathcal{F}_i$ in this way, the associated parallelizability of GOM however largely depends on the order $\mathcal{O}$ in which FOS elements are considered. In GOMEA, $\mathcal{O}$ is generated uniformly at random during each generation to avoid introducing a bias in the optimization process. Such a randomly generated order is however unlikely to achieve high parallelizability.

Instead, $\mathcal{G}$ could be constructed with the purpose of achieving a high degree of parallelizability. This may however introduce a bias on the optimization, because the order in which FOS elements are considered is no longer randomly generated, but governed by $\mathcal{G}$. To maximize the degree of parallelizability of $\mathcal{G}$, the size of each $\mathcal{G}_i \in \mathcal{G}$ should be maximized, and $|\mathcal{G}|$ should consequently be minimized. Maximizing the size of $\mathcal{G}$ is equivalent to the well-known NP-complete graph coloring problem [16], when applied to the dependency graph of FOS elements, i.e., the graph with one vertex per FOS element and edges denoting dependencies between FOS elements. Each FOS element with the same color can then be included in the same subset of $\mathcal{G}$. To find a graph coloring in a reasonable amount of time, a greedy algorithm may be used [24].

6.3.3. Multi-Objective Parallel Gene-Pool Optimal Mixing

An additional potential issue arises when the multi-objective variant of GOMEA is used, as is done here, because fewer solutions are added to the elitist archive using parallel GOM instead of serial GOM. This is because, when serial GOM is applied, a solution may be added to the elitist archive after every iteration of GOM. Consequently, various near-identical, but subsequently improving, copies of a solution may be added to the elitist archive in subsequent iterations of GOM. When GOM is applied in parallel, various iterations of GOM are evaluated and, if leading to an improvement, applied in parallel. Only this new state of the parent solution, for which the variables corresponding to many FOS elements may have been modified, is considered for uptake into the elitist archive. This may have an effect on the optimization process, because the modification of a solution is accepted when it leads to a solution that may be added to the elitist archive. In serial GOM, these decisions are sequential and therefore cumulative, whereas in parallel GOM they are all based on comparisons and changes to a single state of a parent. Furthermore, the addition of a solution to the elitist archive may influence the acceptation of next iterations of GOM, regardless of the fact that such iterations may be evaluated in parallel.

Pseudocode for a version of MO-RV-GOMEA that uses parallel GOM, using population size n and k clusters, is shown in Algorithms 6.8 and 6.9. Algorithm 6.8 shows how GOM is applied to each solution in the population for different FOS elements in parallel. Algorithm 6.9 shows that, for each step of GOM that is accepted, the resulting difference in objective values is added to the parent objective values $\boldsymbol{f}_x$ using a parallel reduction, as this requires the synchronization between all threads. Further details of MO-RV-GOMEA, e.g., the anticipated mean shift and forced improvement procedures, are described in [5].

6.4. Deformable Image Registration

Deformable Image Registration (DIR) [21] is a problem aimed at finding a non-rigid deformation that deforms a source image to match a target image as well as possible.

Finding such a deformation is of interest for various applications in the medical domain, including that of radiotherapy, for example when a pre-operative image of a patient needs to be matched with a post-operative image to enable providing accurate post-operative radiotherapy [20, 22].

The objective of DIR is not only to maximize the similarity of the deformed source image to the target image, but also to minimize the magnitude of the deformation, as (very) large deformations are frequently not anatomically possible and could result in

Algorithm 6.8 Parallel MO-RV-GOMEA

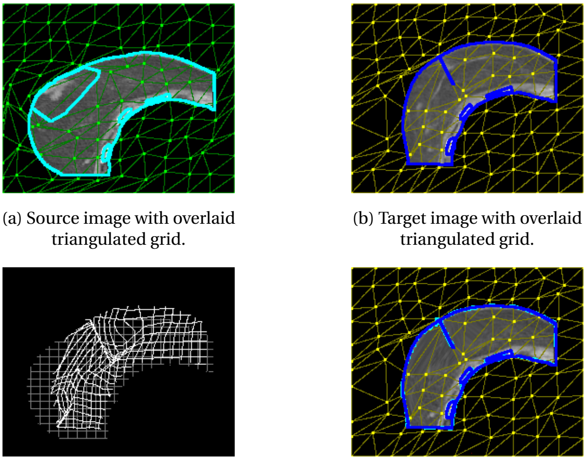
```
procedure PARALLEL MO-RV-GOMEA($n, k, \mathcal{G}$)
    $\mathcal{P} \leftarrow$ Initialization($n$)
    while $\neg$TerminationCriterionSatisfied() do
        $\mathcal{S} \leftarrow$ Selection($\mathcal{P}$)
        $\mathcal{C}^{\mathcal{P}}, \mathcal{C}^{\mathcal{S}} \leftarrow$ ClusterPopulationAndSelection($\mathcal{P}, \mathcal{S}, k$)
        $\mathcal{M} \leftarrow$ EstimateSamplingModels($\mathcal{S}, \mathcal{C}^{\mathcal{S}}$)
        for $\mathcal{G}_i \in \mathcal{G}$ do                                   ▷ Random order
            parfor $j \in \mathcal{G}_i$ do                          ▷ Iterate over FOS elements
                parfor $\mathcal{C}_u^{\mathcal{P}} \in \mathcal{C}^{\mathcal{P}}$ do                     ▷ Iterate over clusters
                    parfor $\boldsymbol{x} \in \mathcal{C}_u^{\mathcal{P}}$ do                  ▷ Iterate over solutions
                        ParGOM($\boldsymbol{x}, \mathcal{F}_j, \mathcal{M}_{j,u}, \mathcal{A}$)
            for $\boldsymbol{x} \in \mathcal{P}$ do
                if NonDominatedByArchive($\boldsymbol{x}, \mathcal{A}$) then
                    $\mathcal{A} \leftarrow \mathcal{A} \cup \boldsymbol{x}$                             ▷ Add to archive
        $\mathcal{P} \leftarrow$ AnticipatedMeanShift($\mathcal{P}$)
        $\mathcal{P} \leftarrow$ ForcedImprovements($\mathcal{P}$)
        $\mathcal{M} \leftarrow$ UpdateSamplingModels($\mathcal{M}$)
```

Algorithm 6.9 Parallel Gene-pool Optimal Mixing

```
procedure PARGOM($\boldsymbol{x}, \mathcal{F}_j, \mathcal{M}_{j,u}, \mathcal{A}$)
    $\boldsymbol{o} \leftarrow$ SamplePartialSolution($\mathcal{M}_{j,u}$)
    $\Delta \boldsymbol{f_o} \leftarrow$ PartialEvaluation($\boldsymbol{o}, \boldsymbol{x}, \boldsymbol{f_x}, \mathcal{F}_j$)
    $\boldsymbol{f_o} \leftarrow \boldsymbol{f_x} + \Delta \boldsymbol{f_o}$
    if $\boldsymbol{f_o}$ dominates $\boldsymbol{f_x}$ or ImprovesArchive($\boldsymbol{f_o}, \mathcal{A}$) then
        $\boldsymbol{f_x} \leftarrow \boldsymbol{f_x} + \Delta \boldsymbol{f_o}$                              ▷ Parallel reduction
        for $i \in \{1, \ldots, |\mathcal{F}_j|\}$ do
            $\boldsymbol{x}[\mathcal{F}_j[i]] \leftarrow \boldsymbol{o}[i]$
```

overfitting. Furthermore, in some cases, additional guidance information may be supplied in the form of landmark points or contours on both the source and target image. Such points or contours are generally delineated by a clinical expert, and should also match between the deformed source image and the target image. There often is however uncertainty in the quality of expert guidance information, making this an objective, not a hard constraint. Due to these multiple, possibly conflicting, objectives of interest, DIR has previously been approached as a multi-objective optimization problem [1, 8].

In this multi-objective approach to DIR, currently working on 2-dimensional images, a dual-dynamic transformation model is used [1], because such a model can effectively deal with large deformations and appearing and disappearing structures, as is the case in, e.g., matching pre- and post-operative images. When using a dual-dynamic transformation model, both the source and the target image are overlaid with a dynamic triangulated grid with the same topology. Note that this does not necessarily mean identical grid point locations at initialization [9], even though this is how we initialize the grids in this chapter. The source and target grids are each denoted as a list of triangles, Δ^s and Δ^t, respectively, where each triangle is defined by three 2-dimensional points, i.e., six real-valued numbers. The list of problem variables then consists of the complete list of unique coordinates of these points. Note that points are shared between neighboring triangles, and points located on the boundary of the image in the initial topology of the grid are restricted to remain on the boundary.



(a) Source image with overlaid triangulated grid.

(b) Target image with overlaid triangulated grid.

(c) Deformation applied to a uniform grid.

(d) Deformation applied to source image.

Figure 6.2: Example of an 11 × 11 dual-dynamic grid applied to the pre/post-operative case, including guidance contours.

Since the topologies of the triangulation grids are the same, each triangle $\delta_i^s \in \Delta^s$ in the triangulated source grid has a corresponding triangle $\delta_i^t \in \Delta^t$ in the target grid, which defines the transformation, as displayed in Figure 6.2. The deformation of a disappearing structure can therefore be achieved by encapsulating the disappearing structure within a number of triangles in the source grid, and shrinking the encapsulating triangles to a point-like size in the target grid. The deformation of an appearing structure can be modeled vice-versa.

A multi-resolution strategy is generally used in DIR. This means that a coarse, low-resolution, grid is used at the start of the optimization process, and the outcome of this low-resolution run is used to initialize (the population of) a run with a higher resolution grid. This is done by subdividing each triangle of the triangulation into 4 smaller triangles, thereby increasing the number of optimization variables by approximately a factor of 4. A multi-resolution strategy greatly helps the optimization process, because large changes in image content would require a vast amount of triangles to be changed substantially if a high-resolution grid is used right away. Hence there are many more higher-order strong dependencies in that case. Achieving this is very difficult, especially considering that the grid may not fold.

6.4.1. Definition

DIR involves the optimization of 2 required objectives, $f_{\text{dissimilarity}}$ and $f_{\text{deformation}}$, and 1 optional objective, f_{guidance}. All objectives are subject to minimization. A deformation is infeasible when a triangle of the deformation grid is folded onto a different triangle, which is the case when edges of the deformation grid are intersecting.

The purpose of the $f_{\text{dissimilarity}}$ objective is to maximize the similarity between the deformed source image and the target image, or, equivalently, minimize the dissimilarity between this pair of images. The $f_{\text{dissimilarity}}$ we use here is the squared difference in image intensities I, compared between each pixel in the source image p^s and its deformed location in the target image p_c^s, or vice versa. This is computed for each triangle δ by iterating over the pixels inside the triangle, denoted $px(\delta)$, and normalized by dividing by the total number of pixels. Thus, the $f_{\text{dissimilarity}}$ objective is defined as [4]:

$$f_{\text{dissimilarity}} = \frac{1}{|px(\Delta^s)| + |px(\Delta^t)|}\left[\sum_{\delta_i^s \in \Delta^s}\left[\sum_{p^s \in px(\delta_i^s)} \left(I^s(p^s) - I^t(p_c^s)\right)^2\right] + \sum_{\delta_i^t \in \Delta^t}\left[\sum_{p^t \in px(\delta_i^t)} \left(I^t(p^t) - I^s(p_c^t)\right)^2\right]\right]$$

Note that this definition of $f_{\text{dissimilarity}}$ only compares gray values of images and is not effective at finding deformations between images of different modalities (e.g., CT and MRI). For this purpose, more sophisticated methods, such as cross-correlation, could be used.

The purpose of the $f_{\text{deformation}}$ objective is to minimize the magnitude of the deformation, because large deformations are likely to be physically or anatomically impossible. For this purpose, Hooke's law [2] is used, comparing the length of each edge e_s in the source grid to the corresponding edge e_s^c in the target grid, and vice versa. The $f_{\text{deformation}}$ objective is defined as follows [4]:

$$f_{\text{deformation}} = \sum_{\delta_s \in \Delta_s}\left[\sum_{e_s \in edges(\delta_s)} \left(\|e_s\| - \|e_s^c\|\right)^2\right] + \sum_{\delta_t \in \Delta_t}\left[\sum_{e_t \in edges(\delta_t)} \left(\|e_t\| - \|e_t^c\|\right)^2\right]$$

The optional objective f_{guidance} requires the minimization of the matching error between landmarks or contours on the deformed source image, and those on the target image. This objective is only used when guidance information is available for the problem instance at hand. Such guidance information is supplied as a set G consisting of tuples with contour points or landmark points on the source and target image, i.e., $G = \{(G_s, G_t)_1, (G_s, G_t)_2, \dots, (G_s, G_t)_k\}$. Each tuple consists of a set of points G_s on the source image and a set of points G_t on the target image, with each point $p_s \in G_s$ having a corresponding point in G_t, denoted p_s^c, and vice versa. The points in G_s contained within a triangle δ of the deformation grid are denoted $G_s(\delta)$, and the Euclidean distance between two points p_i and p_j is denoted $d(p_i, p_j)$. The f_{guidance} objective is then defined as follows [4]:

$$
\begin{aligned}
f_{\text{guidance}} = & \sum_{\delta_s \in \Delta_s} \left[\sum_{(G_s, G_t) \in G} \frac{1}{|G_s| + |G_t|} \sum_{p_s \in G_s(\delta_s)} \min_{p_t \in G_t} \left\{ d(p_s^c, p_t) \right\} \right] \\
& + \sum_{\delta_t \in \Delta_t} \left[\sum_{(G_s, G_t) \in G} \frac{1}{|G_s| + |G_t|} \sum_{p_t \in G_t(\delta_t)} \min_{p_s \in G_s} \left\{ d(p_t^c, p_s) \right\} \right]
\end{aligned}
$$

Given the fact that each of these objectives is composed of a sum of subfunctions, partial evaluations may be applied to each of these to efficiently evaluate modifications to a small number of problem variables. Despite this, none of these objectives are separable. For these reasons, DIR is a very suitable problem to be optimized in a GBO setting.

6.4.2. Gene-Pool Optimal Mixing

It was previously shown that the optimization of DIR in a GBO setting with GOMEA results in large speed-ups and improvements in scalability compared to an EA in a BBO setting [4]. A linkage model was used where almost all linkage sets consisted of the coordinates of the three 2-dimensional corner points belonging to one triangle in either the source or target grid, i.e., 6 problem variables (see Figure 6.3. Not every triangle in the grid is however included in the linkage model, but a selection of triangles was made to cover all problem variables, i.e., have each problem variable be contained in at least 1 linkage set. This selection is identical for both the source and the target grid. In this work, the same linkage model is used as in [4], as this linkage model was shown to be effective. Potentially finding more effective linkage models, or linkage models that synergize well with the application of parallel GOM, is an interesting topic of future research.

When GOM is applied to the variables in a linkage set, it requires the (partial) evaluation of all subfunctions using these variables as input. For the linkage model used here, this means that the application of GOM to a linkage set containing the coordinates of a triangle δ_i^s, requires the evaluation of triangle δ_i^s itself and all its neighboring triangles $N(\delta_i^s)$. Note that the evaluation of a triangle δ_i^s also depends on the coordinates of the corresponding triangle in the opposing grid δ_i^t, and vice versa.

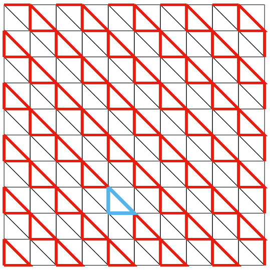


Figure 6.3: The triangulated grid in case of grid dimensions 11 × 11 and the linkage model (in red) used for this instance. Each FOS element $\mathcal{F}_i$ consists of the coordinates (up to three 2-dimensional coordinates) belonging to (part of) one triangle, e.g., the triangle highlighted in blue.

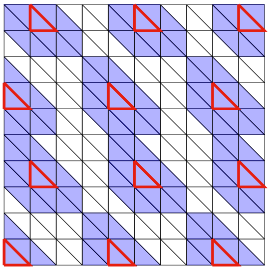


Figure 6.4: A set of triangles, on an 11 × 11 grid, to which GOM is applied in parallel is marked in red. All triangles for which the objective values are then recalculated are shaded in blue.

6.4.3. Parallel Gene-Pool Optimal Mixing

A further speed-up can be obtained by applying GOM in a large-scale parallel fashion, as discussed in Section 6.3.2.

For the parallel application of multiple iterations of GOM to the same solution, it is required that each iteration requires the evaluation of a disjoint set of subfunctions. Given that the application of GOM to the variables in triangle δ_i^s requires the evaluation of all neighboring triangles $N(\delta_i^s)$, and evaluating δ_i^s is dependent on δ_i^t, this application of GOM is dependent on all variables describing the coordinates of δ_i^s, δ_i^t, $N(\delta_i^s)$, and $N(\delta_i^t)$. Therefore, applying GOM to two triangles δ_i^s and δ_j^s in parallel requires that $N(\delta_i^s)$ and $N(\delta_j^s)$ are disjoint. By definition, $N(\delta_i^t)$ and $N(\delta_j^t)$ are also disjoint if $N(\delta_i^s)$ and $N(\delta_j^s)$ are disjoint.

All FOS elements, as displayed in Figure 6.3, can be divided into 12 sets (independent of grid resolution) of FOS elements such that each FOS element is independent of all other FOS elements within the same group. In Figure 6.4 we show (in red) the set of triangles within one group $\mathcal{G}_i \in \mathcal{G}$, and all dependent triangles (in blue). Whenever GOM is applied to the FOS elements in group $\mathcal{G}_i$, the objective values of all triangles in blue are required to be evaluated.

Due to the large (but constant) number of dependent variables in the application of GOM to a single triangle, the potential speed-up of applying parallel GOM is not very large when considering low-dimensional grids, but increases linearly with the number of points on the grid. For this reason, the acceleration techniques discussed here would be of much interest for the extension of the dual-dynamic transformation model to 3-dimensional DIR, as this would lead to much higher-dimensional problems, and even more potential for parallelization.

6.4.4. Experiments

In this section, we first test whether the application of parallel GOM has an effect on the achievable objective values, because the application of parallel GOM requires the elements of the FOS to be iterated through in an order that is efficiently parallelizable, which may introduce a bias to the optimization process. Secondly, we analyze the speed-ups obtained by applying only population-wise parallelization, or also applying parallelization to different iterations of GOM, as discussed in Section 6.4.3. Thirdly, we show resulting approximation fronts and selected deformations obtained by running the multi-resolution scheme.

First, we compare the hypervolumes [26] of runs with and without parallel GOM to analyze whether the potential bias introduced by enforcing an order for GOM has an impact on the performance. This is done for grid resolutions of 6×6, 11×11, 21×21, and 41×41, corresponding to problem dimensionalities of 144, 484, 1764, and 6724, respectively. As the reference point for comparing hypervolumes, we use the worst value of each objective found in 200 randomly generated solutions, initialized in the same way as any initial population.

In practice, because optimizing high-resolution deformation grids is challenging, a multi-resolution scheme is used, as described in Section 6.4. Each run is initialized with a 6×6 dual-dynamic grid, requiring 144 problem variables. After $g^{\max}$ generations, the final population of this run is used to initialize the initial population of a run using an 11×11 dual-dynamic grid, requiring 484 problem variables. This is repeated for grid sizes 21×21 and 41×41, requiring 1764 and 6724 problem variables, respectively. Because the use of a multi-resolution scheme is recommended in practice, we analyze the difference in hypervolume achieved between serial and parallel GOM in the multi-resolution scheme. A population size of 192 is used for the lowest resolution grid, because this is divisible by 16, which is beneficial for the GPU implementations, and results in a reasonable number of approximately 20 solutions per cluster. For each consecutive higher resolution, the population size is increased by a factor of 2. Each resolution is run for $g^{\max} = 200$ generations. For the computation of the hypervolume, the reference point initialized at the start of the lowest resolution run is used for all higher resolution runs.

We then compare the run times of the baseline, single-core CPU-powered, version of GOMEA to two GPU-accelerated versions. In both GPU-accelerated versions of GOMEA, GOM is performed to all solutions in the population in parallel. Furthermore, in both GPU-accelerated versions, the estimation of the sampling models, the sampling of new (partial) solutions, and non-domination comparisons to the elitist archive are performed on the GPU. Only in the parallel GOM version of GOMEA, GOM is applied to different FOS elements in parallel, as described in Section 6.4.3.

For grid resolutions of 6 × 6, 11 × 11, 21 × 21, and 41 × 41, corresponding to problem dimensionalities of 144, 484, 1764, and 6724, respectively, all versions of GOMEA are run for 200 generations, and the total computation time is compared. Each run was performed 30 times.

Finally, we assess the solution quality for the prone/supine problem instance by using 5 landmark points that were not used during the optimization process as guidance information. For this set of landmark points, the mean Target Registration Error (TRE) is the mean distance (in mm) from the deformed source landmark location to the target landmark location. As we mainly focused on increasing the efficiency of the existing optimization method, and the same type of problems are used as in previous work [1], qualitative results are only shown for one of the problem instances.

Data

Experiments are performed on three different pairs of source and target images. These images are displayed in Figure 6.5, with the source images in the top row and the target images in the bottom row.

The images displayed in Figures 6.5a and 6.5d are breast MRI scans of a healthy volunteer in different positions. The source image was captured with the volunteer lying on the stomach (prone position), while the target image was captured with the patient lying on the back (supine). Due to the change in position of the volunteer, large deformations are expected between the source and the target image. Furthermore, because these images are 2-dimensional slices, tissue could have moved in or out of the imaging plane. Therefore, content mismatch could be present in this pair of images.

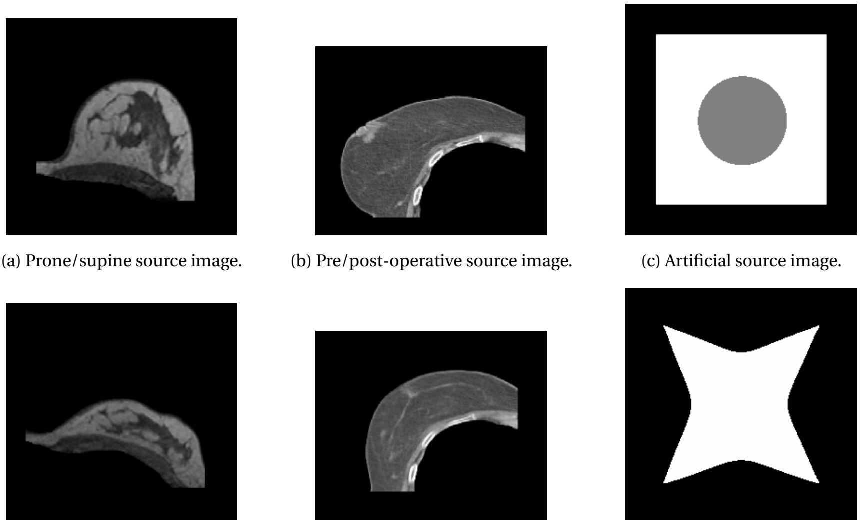
The images displayed in Figures 6.5b and 6.5e are breast CT scans of the same patient acquired before and after surgical treatment, during which cancerous tissue was removed. As such, this pair of images is a perfect example of disappearing structure. Guidance information, i.e., organ contours and a delineation of the area of the removed tissue, is available for this pair of images.

The images displayed in Figures 6.5c and 6.5f are artificial images showing an outer square that is deformed to a star-like shape, and an inner circle that has disappeared in the target image. Contours of both inner and outer shapes are available as guidance information for this pair of images.

Experimental Set-Up

All experiments for the GPU-accelerated GOMEA are performed using a single Nvidia Geforce RTX 2080 Ti GPU, consisting of 4352 CUDA cores and 11 GiB of memory, and an Intel(R) Xeon(R) Silver 4110 CPU @ 2.10 GHz. All experiments for the CPU-based implementation of GOMEA are run on a machine using an Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20 GHz.

Populations sizes used are noted for each experiment. Population sizes used are multiples of 16, as this is beneficial for the GPU implementations. Furthermore, 20 clusters are used to spread the population towards the different regions of the Pareto front, and a target size of 1000 is used for the elitist archive. Unless stated otherwise, default parameters are used for GOMEA [5].



(a) Prone/supine source image. (b) Pre/post-operative source image. (c) Artificial source image.

(d) Prone/supine target image. (e) Pre/post-operative target image. (f) Artificial target image.

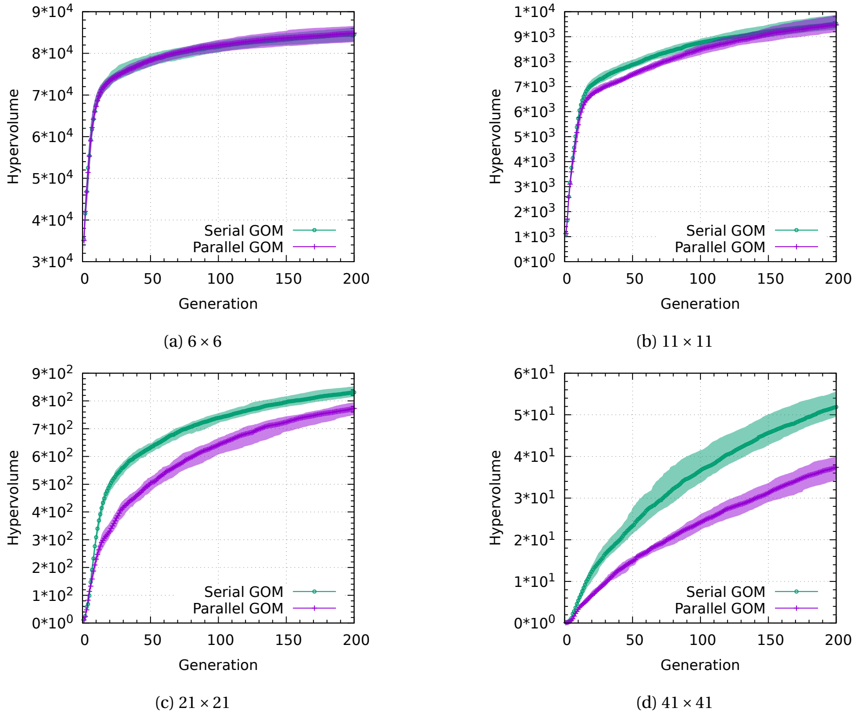
Figure 6.5: Pairs of source and target images used. Breast CT and MRI images are provided through the courtesy of The Netherlands Cancer Institute - Antoni van Leeuwenhoek Hospital.

RESULTS

In Figure 6.6 we show a comparison of the hypervolume progress per generation for the serial and parallel GOM, both using GPU acceleration. This Figure shows that, for low-dimensional grids, there is no discernible difference between serial and parallel GOM in terms of hypervolume. For higher-dimensional grids, however, the hypervolume of serial GOM is substantially better than that of parallel GOM. This is likely caused by the change in frequency by which solutions are added to the elitist archive, as discussed in Section 6.3.2. This effect is only apparent for high-dimensional grids, because the number of iterations of GOM that are done in parallel is much higher than for low-dimensional grids.

In Figure 6.7 we show a comparison of the hypervolume progress in a multi-resolution scheme for the serial and parallel GOM, both using GPU acceleration. This Figure shows that, when using the multi-resolution scheme, the differences in hypervolume between serial and parallel GOM become much smaller than for the single-resolution runs. We argue that this is caused by the fact that runs in the multi-resolution scheme start with an initial population from which many solutions can be inserted into the elitist archive, because it was not randomly generated, but optimized for a lower-resolution grid. The persisting difference in hypervolume is then likely caused by the fact that different iterations of GOM can result in improvements towards different regions of the Pareto front. The isolated results of such iterations will be lost when applying parallel GOM, but could have been valuable contributions to the elitist archive.

The median and interdecile ranges of the computation times are shown in Figure 6.8 for the baseline CPU version, and the GPU-accelerated versions using serial or



(a) 6 × 6

(b) 11 × 11

(c) 21 × 21

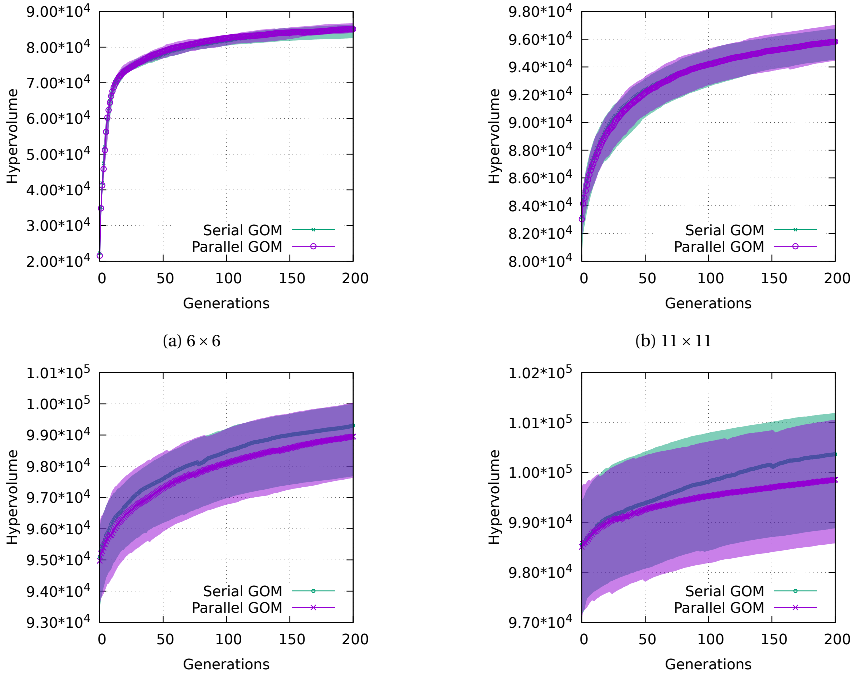
(d) 41 × 41

Figure 6.6: Medians and interdecile ranges (30 runs) of hypervolumes for serial and parallel GOM for the pre/post-operative problem for single-resolution grid of different dimensionalities. Note: different scales on vertical axis.

parallel GOM. All versions of GOMEA use population sizes starting from 192 and increasing exponentially. Similarly, median and interdecile ranges of the speed-ups of the two parallel approaches compared to the baseline are displayed in Figure 6.8.

In Figure 6.8 we see that, as expected, speed-ups achieved by applying parallel GOM increase for larger problem dimensionalities, resulting in speed-ups up to a median factor of 111. For low-dimensional problems, using a larger population size increases the speed-up, as this increases the potential for parallelizability. For high-dimensional problems, however, increasing the population size leads to smaller increases in efficiency, because a high degree of parallelizability was already caused by the high-dimensionality of the problem. When serial GOM is used, achieved speed-ups decrease with an increase of the problem dimensionality. This is likely caused by the overhead of the GPU implementation, and the low utilization of the GPU, though software optimization for this specific scenario would likely lead to higher speed-ups.

In Figure 6.9a we show the resulting 2-dimensional front found after running each resolution in the multi-resolution scheme for 200 generations on the prone/supine problem. Solutions in Figure 6.9a are colored according to their mean TRE, and the solution with the minimum mean TRE (2.8 mm) is marked with a red circle. The deformed source image and the deformation applied to a regular grid are displayed in



(a) 6 × 6

(b) 11 × 11

(c) 21 × 21

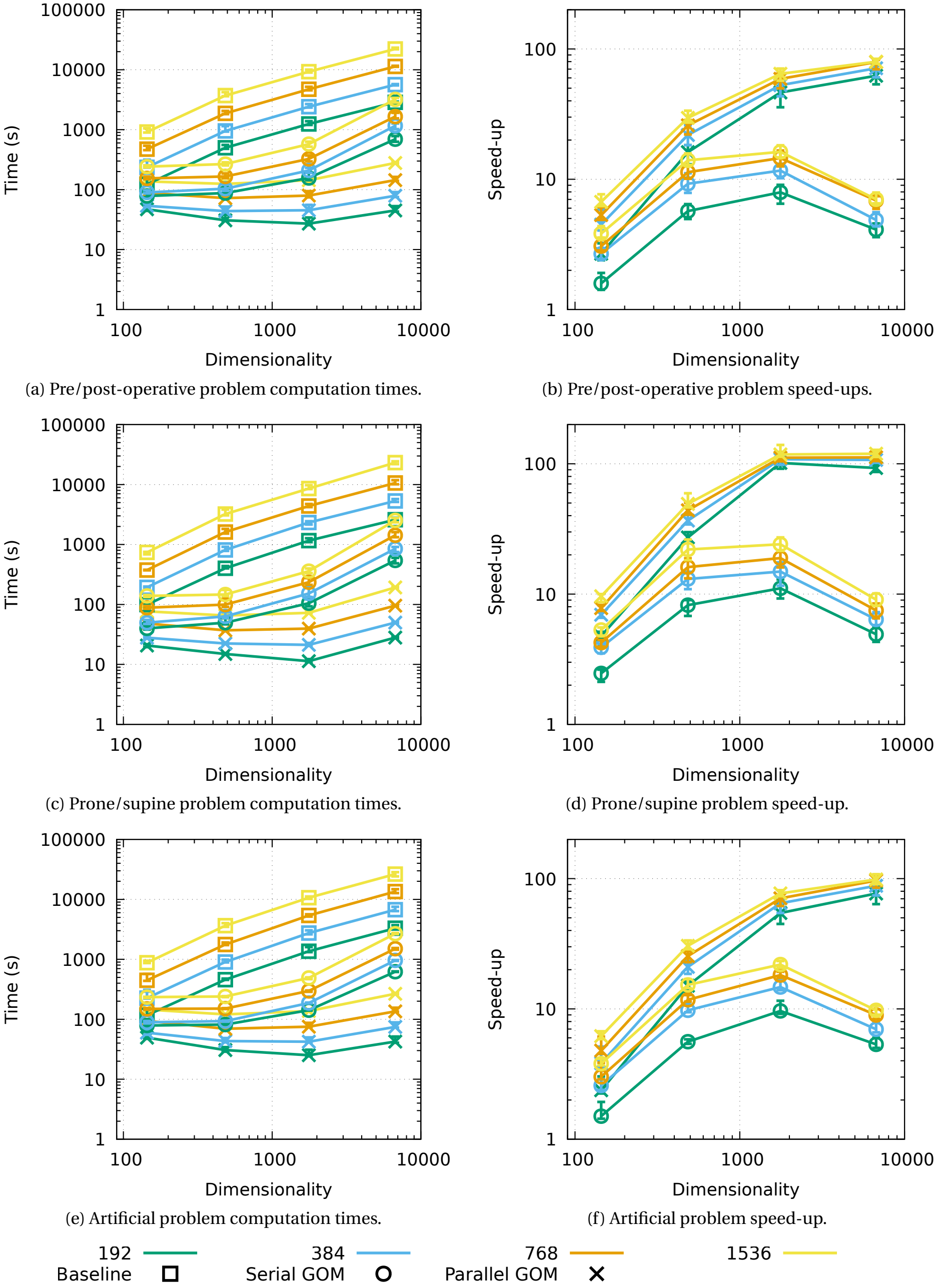
(d) 41 × 41

Figure 6.7: Medians and interdecile ranges (30 runs) of hypervolumes for serial and parallel GOM for the pre/post-operative problem using a multi-resolution scheme, where the final population of the lower resolution is used to initialize the higher resolution runs. Note: different scales on vertical axis.

Figures 6.9b and 6.9c for this solution with the minimum mean TRE. Figure 6.9a confirms that solutions with large deformations are often not the most clinically desirable, as the mean TRE tends to increase for the solutions with the highest amount of deformation.

6.5. DISCUSSION

A challenge of the dual-dynamic grid optimization model for DIR is to apply it to 3-dimensional (medical) images rather than 2-dimensional images, as is done in this chapter and in previous work. One of the aspects of this challenge is to make the required computation time of this optimization problem manageable, because the number of problem variables will greatly increase when a 3-dimensional deformation grid is used. Furthermore, the number of variables will increase more rapidly when the resolution of the grid is increased. As shown in Section 6.4.4, the speed-up achieved by applying parallel GOM largely depends on the number of problem variables, or similarly, the resolution of the deformation grid. Therefore, we argue that the work presented in this chapter is a necessary step to allow for the optimization of 3-dimensional DIR using the dual-dynamic transformation model.



(a) Pre/post-operative problem computation times.

(b) Pre/post-operative problem speed-ups.

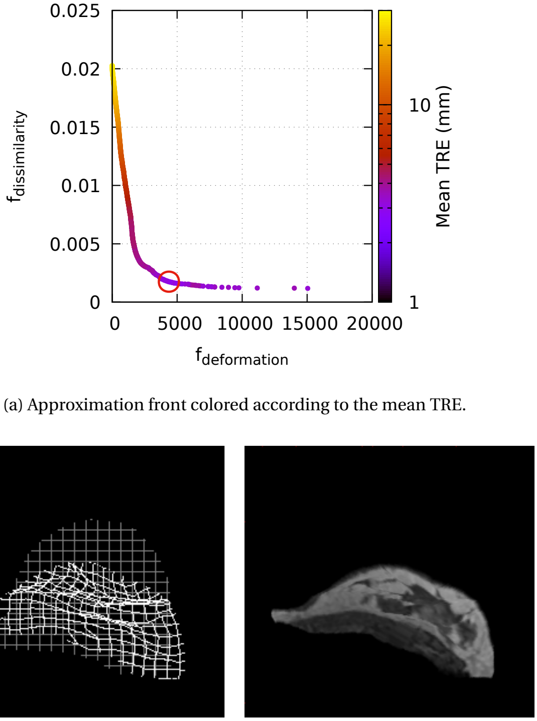
(c) Prone/supine problem computation times.

(d) Prone/supine problem speed-up.

(e) Artificial problem computation times.

(f) Artificial problem speed-up.

Figure 6.8: Computation times and speed-ups of serial and parallel GOM compared to the baseline CPU version of RV-GOMEA for different population sizes.



(a) Approximation front colored according to the mean TRE.

(b) Deformation of the selected solution.

(c) Deformed source image of the selected solution.

Figure 6.9: Resulting 2-dimensional front after optimizing each grid resolution for 200 generations in the multi-resolution scheme. The deformation is shown for the solution with the minimum mean TRE (2.8 mm), marked by a red circle, visualized by using a regular grid (b) as well as the resulting deformed source image (c).

The application of parallel GOM was in this chapter limited to the DIR application, though we argue in Section 6.3.2 how it may also be applied to other problems. Also for problems for which distant groups of variables are not strictly (conditionally) independent, but only weakly dependent, it may be possible to apply GOM in parallel. The VIG of such a problem would be fully connected, but edges between weakly connected variables could be filtered out to obtain a sparse VIG. Finding the best way to filter out weak dependencies from the VIG, and finding whether parallel GOM would benefit problems where this is the case, is an interesting direction of future research.

6.6. CONCLUSION

In this chapter we showed how the GOM variation operator may be used in a large-scale parallel way to apply a large number of variation steps, each only accepted when leading to an improvement in fitness, to different sets of conditionally independent variables in parallel. This results in a high degree of parallelizability that can be exploited by using the parallel computing power of GPUs.

Parallel GOM was applied to the real-world problem of DIR, which is a problem in the field of medical imaging that requires finding a transformation from a given source image to a given target image. In particular, we consider a multi-objective formulation of DIR that uses a dual-dynamic transformation model, which is capable of dealing with large deformations, appearing and disappearing structures, but results in a difficult non-smooth optimization problem. We find that applying parallel GOM to DIR can lead to a reduction in hypervolume gained per generation, because fewer intermediate results are saved in the elitist archive. Parallel GOM does however lead to large speed-ups of up to a factor of 111 compared to the serial version of RV-GOMEA. These results show that the application of parallel GOM leads to a large increase in efficiency for large-scale instances of DIR, opening the door to the extension of the dual-dynamic transformation model of DIR to 3-dimensional deformation grids, and other real-world problems.

REFERENCES

[1] Alderliesten, T., Bosman, P. A. N., and Bel, A. (2015). Getting the most out of additional guidance information in deformable image registration by leveraging multi-objective optimization. In Ourselin, S. and Styner, M. A., editors, *Medical Imaging 2015: Image Processing*, volume 9413, pages 469 – 475. International Society for Optics and Photonics, SPIE.

[2] Arfken, G. B. and Weber, H. J. (1999). Mathematical methods for physicists.

[3] Bouter, A., Alderliesten, T., Bel, A., Witteveen, C., and Bosman, P. A. N. (2018). Large-scale parallelization of partial evaluations in evolutionary algorithms for real-world problems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1199–1206. ACM.

[4] Bouter, A., Alderliesten, T., and Bosman, P. A. N. (2017). A novel model-based evolutionary algorithm for multi-objective deformable image registration with content mismatch and large deformations: benchmarking efficiency and quality. In Styner,

M. A. and Angelini, E. D., editors, *Medical Imaging 2017: Image Processing*, volume 10133, pages 304 – 311. International Society for Optics and Photonics, SPIE.

[5] Bouter, A., Alderliesten, T., and Bosman, P. A. N. (2021a). Achieving highly scalable evolutionary real-valued optimization by exploiting partial evaluations. *Evolutionary Computation*, 29(1):129–155.

[6] Bouter, A., Alderliesten, T., and Bosman, P. A. N. (2021b). GPU-accelerated parallel gene-pool optimal mixing applied to multi-objective deformable image registration. In *IEEE Congress on Evolutionary Computation*, pages 2539–2548. IEEE.

[7] Bouter, A., Maree, S. C., Alderliesten, T., and Bosman, P. A. N. (2020). Leveraging conditional linkage models in gray-box optimization with the real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 603–611.

[8] Bosman, P. A. N. and Alderliesten, T. (2012). Incremental Gaussian model-building in multi-objective EDAs with an application to deformable image registration. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 241–248. ACM.

[9] Bosman, P. A. N. and Alderliesten, T. (2016). Smart grid initialization reduces the computational complexity of multi-objective image registration based on a dual-dynamic transformation model to account for large anatomical differences. In Styner, M. A. and Angelini, E. D., editors, *Medical Imaging 2016: Image Processing*, volume 9784, pages 1108 – 1114. International Society for Optics and Photonics, SPIE.

[10] Chen, W., Whitley, D., Tinós, R., and Chicano, F. (2018). Tunneling between plateaus: Improving on a state-of-the-art MAXSAT solver using partition crossover. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 921–928. ACM.

[11] Chiong, R., Weise, T., and Michalewicz, Z. (2012). *Variants of evolutionary algorithms for real-world applications*. Springer.

[12] Coello Coello, C. A. and Lamont, G. B. (2004). *Applications of multi-objective evolutionary algorithms*, volume 1. World Scientific.

[13] de Souza, M. L. and Pozo, A. T. R. (2014). A GPU implementation of MOEA/D-ACO for the multiobjective traveling salesman problem. In *Brazilian Conference on Intelligent Systems (BRACIS)*, pages 324–329. IEEE.

[14] Deb, K. (1999). Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3):205–230.

[15] Deb, K. and Myburgh, C. (2016). Breaking the billion-variable barrier in real-world optimization using a customized evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 653–660. ACM.

[16] Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer.

[17] Luong, H. N. and Bosman, P. A. N. (2012). Elitist archiving for multi-objective evolutionary algorithms: To adapt or not to adapt. In *International Conference on Parallel Problem Solving from Nature*, pages 72–81. Springer.

[18] Luong, N. H., Alderliesten, T., Bel, A., Niatsetski, Y., and Bosman, P. A. N. (2018). Application and benchmarking of multi-objective evolutionary algorithms on high-dose-rate brachytherapy planning for prostate cancer treatment. *Swarm and Evolutionary Computation*, 40:37–52.

[19] Ortega, G., Filatovas, E., Garzón, E. M., and Casado, L. G. (2017). Non-dominated sorting procedure for Pareto dominance ranking on multicore CPU and/or GPU. *Journal of Global Optimization*, 69(3):607–627.

[20] Petersen, R. P., Truong, P. T., Kader, H. A., Berthelet, E., Lee, J. C., Hilts, M. L., Kader, A. S., Beckham, W. A., and Olivotto, I. A. (2007). Target volume delineation for partial breast radiotherapy planning: Clinical characteristics associated with low interobserver concordance. *International Journal of Radiation Oncology* Biology* Physics*, 69(1):41–48.

[21] Sotiras, A., Davatzikos, C., and Paragios, N. (2013). Deformable medical image registration: A survey. *IEEE Transactions on Medical Imaging*, 32(7):1153–1190.

[22] Struikmans, H., Wárlám-Rodenhuis, C., Stam, T., Stapper, G., Tersteeg, R. J., Bol, G. H., and Raaijmakers, C. P. (2005). Interobserver variability of clinical target volume delineation of glandular breast tissue and of boost volume in tangential breast irradiation. *Radiotherapy and Oncology*, 76(3):293–299.

[23] Thierens, D. and Bosman, P. A. N. (2011). Optimal mixing evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 617–624. ACM.

[24] Thulasiraman, K., Arumugam, S., Nishizeki, T., Brandstädt, A., et al. (2016). *Handbook of Graph Theory, Combinatorial Optimization, and Algorithms*. Taylor & Francis.

[25] Whitley, D., Hains, D., and Howe, A. (2009). Tunneling between optima: Partition crossover for the traveling salesman problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 915–922. ACM.

[26] Zitzler, E. and Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms — A comparative case study. In *International Conference on Parallel Problem Solving from Nature*, pages 292–301. Springer.

7

Spatial Redistribution of Irregularly-Spaced Pareto-Approximation Fronts for More Intuitive Navigation and Solution Selection

This chapter is based on the following publication: Bouter, A., Pirpinia, K., Alderliesten, T., and Bosman, P. A. N. (2017b). Spatial redistribution of irregularly-spaced Pareto fronts for more intuitive navigation and solution selection. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1697–1704

ABSTRACT

A multi-objective optimization approach is often followed by an a posteriori decision-making process, during which the most appropriate solution of the Pareto-approximation set is selected by a professional in the field. Conventional visualization methods do not correct for Pareto-approximation fronts with irregularly-spaced solutions. However, achieving a uniform spread of solutions can make the decision-making process more intuitive when decision tools such as sliders, which represent the preference for each objective, are used. We propose a method that maps an m-dimensional Pareto-approximation front to an $(m-1)$-simplex and spreads out points to achieve a more uniform distribution of these points in the simplex while maintaining the local neighborhood structure of the solutions as much as possible. This set of points can then more intuitively be navigated due to the more uniform distribution. We test our approach on a set of non-uniformly spaced three-dimensional Pareto-approximation fronts of a real-world problem: deformable image registration of medical images. The results of these experiments are visualized as points in a triangle, showing that we indeed achieve a representation of the Pareto-approximation front with a near-uniform distribution of points where these are still positioned as expected, i.e., according to their quality in each of the objectives of interest.

7.1. INTRODUCTION

Many real-world problems are actually multi-objective, meaning that multiple, often conflicting, objectives of interest arise and decisions need to be taken in the presence of trade-offs between the objectives. Without having a clear notion of preferences among these objectives a priori, the most commonly adopted approach is multi-objective optimization [9]. Solving a multi-objective optimization problem results in a set of solutions that can be considered equally good, since they represent optimal trade-offs between the objectives. This set of solutions in the parameter space of the optimization problem is called the Pareto-approximation set, and the set of solutions in the objective space is called the Pareto-approximation front. Typically, a user then has to select a solution from the acquired Pareto-approximation front that best represents his/her preferences in terms of the objectives' trade-offs.

Naturally it follows that the way the Pareto-approximation front is presented to the user plays an essential role in enabling the final decision-making process. An appropriate Pareto-approximation front visualization should provide the user the ability to explore efficiently the entire objective space, while capturing the structure of the Pareto-approximation front and the (local) ordering between all solutions. Such a visualization can be straightforward for problems with two objectives, but presents challenges for higher dimensions. Many interesting visualization techniques have focused on visualization of Pareto-approximation fronts consisting of more than three dimensions in order to enable insightful decision making, using, e.g., level diagrams [4], self-organizing maps [8], parallel coordinates [12], the prosection method [20], or a projection to a 2-dimensional [21] or 3-dimensional surface [11]. Nonetheless, these methods do not address the challenge of solutions on the Pareto-approximation front being indistinguishable, which is mostly a problem for Pareto-approximation fronts that have a non-uniform distribution of solutions across their surface. An example of a uniformly spaced Pareto-approximation front (obtained from a benchmark problem), as opposed to a non-uniformly spaced Pareto-approximation front (obtained from a real-world problem), can be seen in Figure 7.1. Selecting a solution from the latter can be challenging, due to the fact that many solutions are virtually indistinguishable from each other. This can be a problem, because solutions that are very close in objective space could represent very different solutions in parameter space. Moreover, the remapping of a Pareto-approximation front for visualization purposes can even lead to a larger number of points being indistinguishable, even if such points were not close to each other in the original Pareto-approximation front. Therefore, an ideal remapping of a Pareto-approximation front to a 2-dimensional space should result in a representation where the solutions are better (i.e., more uniformly) distributed, while maintaining the characteristics of the initial topology of the Pareto-approximation front and the (local) ordering relations. This remapped set of points could then be easily navigated by using a set of trade-off sliders, one for each objective, which represent the relative preference for each objective. Since, however, there is no mapping from a higher dimensional space to a lower dimensional space that preserves 100% of local ordering relations [14], this problem becomes a multi-objective optimization problem itself. In this work, we therefore formulate this as a problem with two objectives. To solve it, we use a Multi-Objective Real-Valued Gene-pool Optimal Mixing Evolutionary

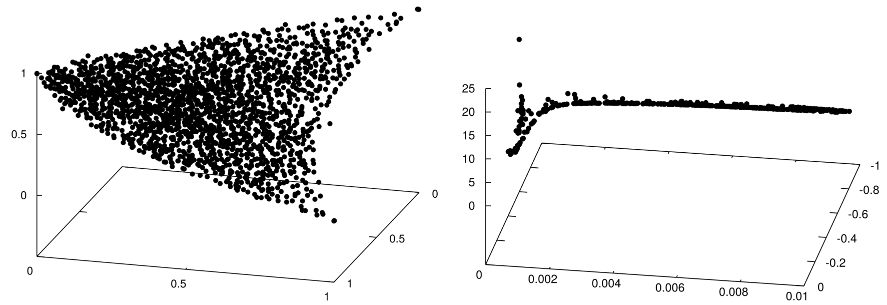


Figure 7.1: Example of a uniformly spaced Pareto-approximation front of a benchmark problem (left) and a non-uniformly spaced Pareto-approximation front from a real-world problem (right). Exploring the local structure of the real-world Pareto-approximation front via an interactive decision tool is potentially quite challenging.

Algorithm (MO-RV-GOMEA) [2]. Although the proposed approach generalizes to more than three dimensions, in this work we focus on the 3-dimensional case, for which the output of the algorithm also results in an intuitively sensible 2-dimensional visualization. Moreover, we are intrinsically motivated by data obtained from a real-world, three-objective optimization problem: Deformable Image Registration (DIR).

The remainder of this chapter is organized as follows. In Section 7.2, we illustrate the challenges related to the uniformly distributed mapping of non-uniformly spaced sets of solutions on the Pareto-approximation front. In Section 7.3, we describe our methods of addressing these challenges, including the optimization objectives and MO-RV-GOMEA. In Section 7.4, we describe our experimental setup, as well as our benchmark problems and the problem of DIR. In Section 7.4.3, we present our results, and in Section 7.5 we discuss our findings and present our conclusions.

7.2. Problem Description

We use the example displayed in Figures 7.2 and 7.3 to illustrate the difficulties involved in achieving a sufficiently spread mapping of a Pareto-approximation front while preserving the structure of the original Pareto-approximation front as best as possible. This better illustrates the reasoning behind the use of a multi-objective approach to this problem. Figure 7.2 displays an example of a projection of a 3-dimensional Pareto-approximation front onto a triangle. This front contains a number of points that are largely overlapping, making the selection of one of these individual points very difficult. Figure 7.3 then displays a fairly intuitive attempt at achieving a larger spread between points while preserving the structure of the original Pareto-approximation front. Because preservation of the original structure is not clearly defined, we look at the following three measures that are relevant for displaying the quality of solutions in regard to other solutions that are visualized. These measures of the mapped Pareto-approximation front should ideally be identical to those of the original front:

1. For each point, the ordering of Euclidean distances to each other point.
2. For each point, the set of k nearest neighbors, where k is a small constant.
3. For each vertex of the simplex, corresponding to the extremum of one objective, the ordering of the points based on the distances from this vertex to these points.

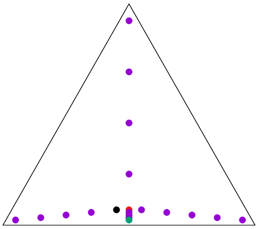


Figure 7.2: Projection of an arbitrary 3-dimensional Pareto-approximation front onto a triangle.

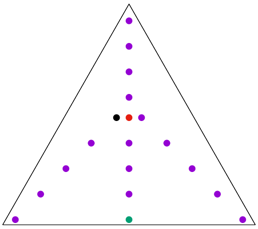


Figure 7.3: Example of a redistribution of the points in Figure 7.2.

Despite the seemingly intuitive spreading of points, all three previously listed measures are in some way different for Figure 7.2 and Figure 7.3, whereas they should ideally be similar. The first two measures have changed due to, among others, the point marked in red having two different nearest neighbors in the two figures. The third measure has changed due to, among others, the green point being further away from the bottom left vertex of the triangle than the black point in Figure 7.2, but closer in Figure 7.3.

Figures 7.2 and 7.3 only show one example, but it is likely that no sufficient spreading of the points is possible without the violation of all three measures. Moreover, it seems very difficult to define a different meaningful measure of Pareto-approximation front structure that remains constant when points are spread out. This would allow the measure to be used as a constraint while the spread between pairs of points is maximized. Instead, violations of the rankings are seemingly necessary to achieve a sufficient spread of points in the mapped Pareto-approximation front, meaning that there is an inherent trade-off between the spread of points and the preservation of the Pareto-approximation front structure. We therefore avoid using a hard constraint or penalty value for the preservation of the structure of the original Pareto-approximation front. Alternatively, a weighted sum of objectives can be optimized, but this will require the manual tuning of the weights, which is unintuitive and problem specific.

7.3. METHODS

An m-dimensional Pareto-approximation front $\mathcal{P}_F$ of size n, normalized to the range $[0,1]^m$ is given as input. We assume, without loss of generality, that in the original optimization problem all objectives need to be minimized. We then construct a mapping of the points on the input Pareto-approximation front to an $(m-1)$-simplex. We consider two different representations: one where we aim to map the maximum of each dimension to the region near one of the vertices of the simplex, and one where we aim to map the minimum of each dimension to the region near one of the vertices of the simplex. Both representations are valid and could be useful in different scenarios, depending on the shape of the input Pareto-approximation front, and possible preferences regarding the decision-making process.

Points on the mapped surface (or in the mapped hypervolume for high-dimensional problems) of the simplex must be sufficiently spread in order to easily be individually selectable. At the same time, the structure of the original Pareto-approximation front must be preserved as much as possible. These two objectives are contradictive and no weights can intuitively be defined, leading us to solve this problem using a multi-objective optimization approach. The two objectives of interest are defined as a function of the set of points $\mathcal{P}_{\boldsymbol{F}}$ of the input Pareto-approximation front. Each point $p_i \in \mathcal{P}_{\boldsymbol{F}}$ is mapped to the point $p'_i \in [0,1]^{m-1}$. The set of parameters of the optimization problem consists of the $m-1$ Cartesian coordinates of each p'_i. More specifically, any solution to this problem is represented by a vector $\boldsymbol{x}$ of length $\ell = n(m-1)$.

7.3.1. Multi-Objective Optimization

For the multi-objective approach used in this chapter, the two objective functions of interest F_{spread} and F_{stress} are defined in this section, which both need to be minimized. The first objective function, F_{spread}, is aimed at the maximization of the distance between each point and its nearest neighbor. This problem is in general known as the Circles-In-A-Square (CIAS) packing problem, which is a quite difficult continuous optimization problem when solved to optimality [5]. We are however not interested in achieving optimal spread of all points, because we only aim to achieve a spread that is sufficient to spread out large clumps of points and prevents large empty spaces in the mapped representation of the input Pareto-approximation front. For this reason, we use a relaxed definition of the original CIAS objective function, which is derived in Equation 7.1, based on the p-norm, with $\boldsymbol{x}$ the vector of points and $d(i,j)$ the Euclidean distance between points i and j.

$$\begin{aligned} F_{\text{CIAS}}(\boldsymbol{x}) &= \operatorname*{arg\,max}_{\boldsymbol{x}} \left\{ \min_{i,j} d\left(i,j\right) \right\} \\ &= \operatorname*{arg\,min}_{\boldsymbol{x}} \left\{ \max_{i,j} \frac{1}{d\left(i,j\right)} \right\} \\ &= \operatorname*{arg\,min}_{\boldsymbol{x}} \left\{ \lim_{p\to\infty} \sqrt[p]{\sum_{i=0}^{|\boldsymbol{x}|-1} \sum_{j=0}^{i-1} \left| \frac{1}{d\left(i,j\right)} \right|^p} \right\} \\ &= \operatorname*{arg\,min}_{\boldsymbol{x}} \left\{ \lim_{p\to\infty} \sum_{i=0}^{|\boldsymbol{x}|-1} \sum_{j=0}^{i-1} d\left(i,j\right)^{-p} \right\} \end{aligned} \tag{7.1}$$

The objective function that is used in our optimization approach is defined as F_{spread} in Equation 7.2, again with $d(a,b)$ the Euclidean distance between a and b. The sum of pairwise distances with $p = 4$ is used, because the smoothness of this objective makes it easier to optimize than an objective with a flat landscape, such as the original definition of F_{CIAS}.

$$F_{\text{spread}} = \sum_{i=0}^{n-1} \sum_{j=0}^{i-1} d(p'_i, p'_j)^{-4} \tag{7.2}$$

Although any of the measures introduced in Section 7.2 could be used for optimization of the preservation of the original Pareto-approximation front structure, all these measures are discrete, which will increase the difficulty of their optimization. Instead we use the objective function defined as F_{stress} in Equation 7.3, with $d(a, b)$ again the Euclidean distance between a and b. This objective is continuous, has a smooth landscape, and is based on the Sammon stress [18]. The second term of Equation 7.3 has the purpose of mapping the extreme points of the input Pareto-approximation front onto the space near the corresponding vertex of the simplex. For this, we define $\boldsymbol{V}$ as a set of hyperplanes, where each $\boldsymbol{V}_j \in \boldsymbol{V}$ defines a hyperplane. The hyperplane $\boldsymbol{V}_j^{\text{min}}$ has the coordinate 0 in dimension j, and $\boldsymbol{V}_j^{\text{max}}$ is the hyperplane that has the coordinate 1 in dimension j. In Equation 7.3, $\boldsymbol{V}_j^{\text{min}}$ and $\boldsymbol{V}_j^{\text{max}}$ can be used for $\boldsymbol{V}_j$, depending on which of the two representations discussed in Section 7.3 is used. The hyperplane $\boldsymbol{V}_j$ is mapped onto the point $\boldsymbol{V}'_j$, which uniquely corresponds to one of m vertices of the simplex.

$$F_{\text{stress}} = \frac{2}{n-1} \sum_{i=0}^{n-1} \sum_{j=0}^{i-1} \Big(d(p_i, p_j) - d(p'_i, p'_j) \Big)^2 + \frac{1}{m} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \Big(d(p_i, \boldsymbol{V}_j) - d(p'_i, \boldsymbol{V}'_j) \Big)^2 \tag{7.3}$$

To ensure that each point is mapped to a point inside the simplex, a constraint value is used that is equal to the number of points that is not inside the simplex. The constraint domination [10] technique is used for constraint handling. This means that a solution is considered to dominate any solution that has a higher constraint value.

7.3.2. MO-RV-GOMEA

For the optimization of the parameters that define a mapping, we use the recently introduced MO-RV-GOMEA [2], because Evolutionary Algorithms (EAs) are known to be among the state of the art for multi-objective optimization [9]. MO-RV-GOMEA has been shown to perform better than well-known state-of-the-art EAs [2], such as NSGA-II [10], especially when partial evaluations are possible, which is the case here.

MO-RV-GOMEA is a model-based EA for the optimization of real-valued variables. An adaptive elitist archive [15] is maintained to keep track of non-dominated solutions. The population is clustered, because this is known to be highly beneficial to find a good spread of solutions across the entire Pareto-optimal front [6, 16]. A fraction of the solutions in the population that rank the best according to non-domination sorting [10] is selected. The variation operator is based on a linkage model that explicitly defines subsets of problem variables, so-called linkage sets, that are considered to be dependent. A linkage model can either be learned during optimization, in which case a model is learned for each cluster separately, or it can be defined a priori, in which case it is fixed throughout the optimization process and each cluster uses the same linkage model. For each linkage set of each cluster, a multivariate normal probability distribution is estimated with maximum likelihood based on the selection. The estimated probability distribution of a linkage set is used to sample new values for the parameters included in this linkage set, which are inserted into existing solutions in the population. Only if such a modification of a subset of variables is considered to be an improvement, the modification is accepted. Otherwise, the solution is returned to its previous state. Partial modifications of solutions can be evaluated efficiently through so-called partial evaluations, which efficiently evaluate the modified objective values of a solution based on the modification of the variables. The probability distribution of a linkage set is newly estimated each generation, but it is scaled by a factor that is adapted based on improvements that are found and where they are found. We refer the interested reader to the relevant literature for further details [2].

To efficiently solve the objectives defined in Equations 7.2 and 7.3, we use a linkage model where each linkage set consists of all $(m-1)$ Cartesian coordinates of a single point $p_i \in \mathcal{P}_{\boldsymbol{F}}$. This linkage model allows for efficient partial evaluations, because any objective value can efficiently be updated after the coordinates of one point are modified. Half of the population of MO-RV-GOMEA is initialized uniformly random in the $(m-1)$-simplex, while the other half of the population is initialized based on the coordinates of each point in the input Pareto-approximation front. Specifically, each solution in the latter half of the population is initialized such that each point $p_i \in \mathcal{P}_{\boldsymbol{F}}$ maps to the point p'_i for which the barycentric weight of the j^{th} vertex of the simplex is $p_i[j] \cdot U(0,1)$, where $p_i[j]$ is the coordinate of p_i in dimension j. This latter half of the population can improve the speed of optimization due to more targeted initialization, while the former half of the population prevents premature convergence in rare cases where the targeted initialization procedure is unsuccessful at finding reasonable initial solutions.

7.4. Experiments

We test the performance of our method by applying it to a set of 3-dimensional Pareto-approximation fronts. The results of these experiments can easily be visualized, because each 3-dimensional Pareto-approximation front is mapped to a 2-simplex, i.e., a triangle. Each point on the resulting mapping of a Pareto-approximation front is then color coded according to its normalized values in each of the dimensions of the original Pareto-approximation front. Specifically, each point is assigned an RGB color with the intensity of red corresponding to its coordinate in the x-dimension, the intensity of green corresponding to its coordinate in the y-dimension, and the intensity of blue corresponding to its coordinate in the z-dimension. This simultaneously allows for a visual representation of the degree by which the structure of the original Pareto-approximation front is preserved, and the degree by which points are spread out.

We perform two experiments for each benchmark problem introduced in Section 7.4.1, one where the maximum of each dimension is mapped to a vertex of the triangle and one where the minimum of each dimension is mapped to a vertex of the triangle. A time limit of one hour is used for each experiment, and each experiment is performed on a desktop computer with an Intel Core i7-2600 CPU @ 3.40GHz.

7.4.1. BENCHMARK PROBLEMS

A number of irregularly-spaced Pareto-approximation fronts were previously obtained by taking a multi-objective approach to the real-world problem known as DIR, which is discussed in more detail in Section 7.4.2. These benchmark problems are referred to as DIR_1 through DIR_{10}. A 3-dimensional scatter plot of each DIR benchmark problem is displayed in the leftmost column of Figures 7.6 and 7.7, where each point is color coded according to the scheme introduced in Section 7.4. The benchmark problems consist of a number of points between approximately 400 and 1200 points, but a random subset of 100 points was used for the purpose of clearer visualization and faster optimization.

7.4.2. DEFORMABLE IMAGE REGISTRATION

DIR [19] is a key tool in several medical processes, e.g., in radiotherapy [13]. DIR has a lot of potential since it can be used for radiotherapy planning as well as surgical planning and treatment response assessment [7]. However, DIR presents several challenges which limit its wider application in clinical practice.

Solving a DIR problem entails finding the optimal *non-linear* transformation to align two images, i.e., the optimal spatial correspondence between points in a so-called moving image and the reference image. For most general-purpose registration methods, DIR is formulated as a single-objective optimization problem, where the cost function to be optimized is a linear combination of terms that express objectives of interest. These objectives most often describe the dissimilarity between the images that needs to be minimized, but also the deformation magnitude. Although in general a certain amount of deformation is necessary in order to achieve a good match, and thereby a low value for the dissimilarity, too much deformation can result in physically incorrect deformations. Therefore, penalizing the deformation magnitude ensures avoiding such unwanted deformations. Depending on the type of registration problem, more objectives can be added; for example, for the most challenging registration problems which involve large anatomical changes, guidance information in the form of a third objective can help the registration algorithm; further, objectives which enforce local rigidity (i.e., not allowing deformation of certain regions such as bony anatomy) can be added.

However, this single-objective formulation presents a challenge, as the weights associated with the objectives that define the linear combination need to be determined beforehand, along with multiple other registration-specific parameters. This results in a time-consuming process, since the interplay between parameters, objectives of interest, and registration outcome is very complex for challenging registration problems, and the optimal configuration of parameters can be very problem-specific, leading to multiple trial-and-error attempts for each problem separately. Recently, multi-objective optimization approaches for DIR have been introduced, either by directly finding the optimal transformation that aligns the two images [1], or by optimizing the parameters (including the weights) of existing single-objective registration software [17]. In either case, these multi-objective

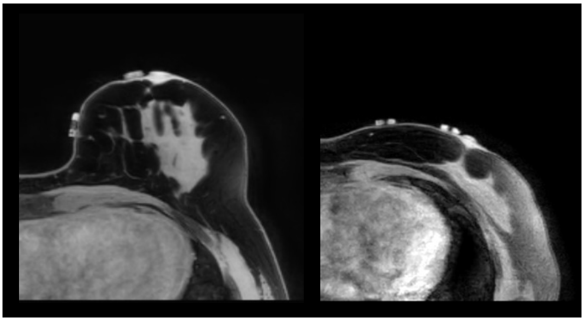


Figure 7.4: An example of a DIR problem. Left: slice of 3-dimensional prone MRI (moving image), right: 3-dimensional supine MRI (reference image).

approaches remove the need for predefining sets of parameters, resulting in a Pareto-approximation front of registration outcomes, which is then potentially navigable by an expert a posteriori, while immediately seeing the associated DIR outcomes.

In this work, using the aforementioned multi-objective approach that optimizes the weights associated with the objectives of an existing registration method, we acquired ten Pareto-approximation fronts of ten instances of a very challenging DIR problem: aligning breast Magnetic Resonance Imaging (MRI) scans of patients (in this study, healthy volunteers) acquired in a prone (i.e., lying face down) to a supine position (i.e., lying face up). Solving this registration problem can be useful in surgical planning for breast cancer patients that undergo breast-conserving surgery. In standard clinical practice, contrast-enhanced prone MRI is acquired to aid diagnosis, since supine MRI suffers from breathing motion artifacts which do not allow acquisition with contrast enhancement. By registering, however, the prone MRI to a non-contrast-enhanced MRI of a patient in supine position, pre-operative information is related to the intra-operative setting, where the patient is also in a supine position. This can lead to better tumor localization during surgery, reducing the chances of local recurrence and improving cosmetic outcome. However, the large deformation that the breast undergoes between prone and supine positioning makes this DIR problem very hard. For this reason, for this DIR problem not only dissimilarity and deformation magnitude were optimized, but also guidance information is exploited by minimizing the distance between marker locations in the moving and reference image that were attached on the breast of the volunteer. This resulted in ten 3-dimensional Pareto-approximation fronts. An example of a DIR problem can be seen in Figure 7.4.

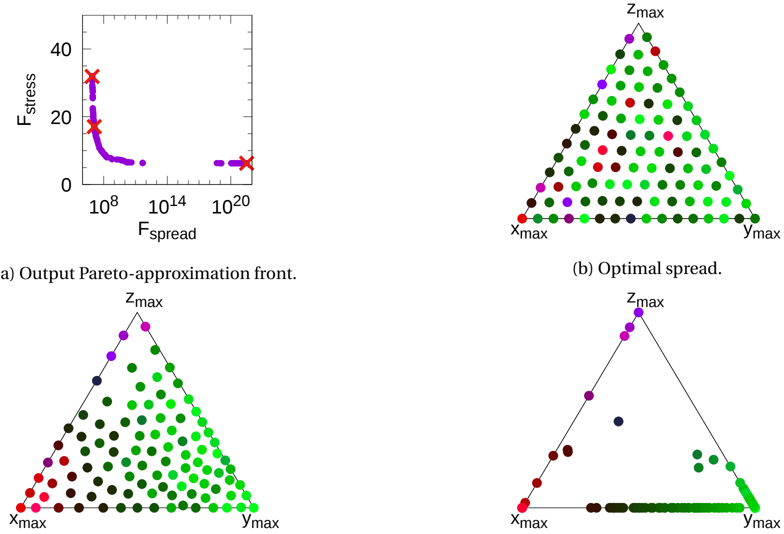
The very different scale of these three objectives, as well as the fact that dissimilarity and guidance error are not always necessarily conflicting, results in highly non-uniform Pareto-approximation fronts (see, e.g., Figure 7.1 right) . Another reason for an uneven distribution is possibly the difference in optimization difficulty of the three objectives. The dissimilarity objective is a highly non-convex function, with a lot of local minima, whereas the objectives that describe deformation and guidance information are convex. Each solution on the Pareto-approximation front represents a differently deformed image. Therefore, the differences between the deformed images, albeit probably small between solutions which are very close to each other, are potentially clinically relevant depending on the DIR application, and therefore, they should be distinguishable and navigable for the user.

7.4.3. Experimental Results

Because we solve our problem using a multi-objective approach, the result of the optimization is itself actually a Pareto-approximation front of solutions with different trade-offs for F_{spread} and F_{stress}. In Figure 7.5 we show the output Pareto-approximation front for the input Pareto-approximation front of DIR_1 (shown in Figure 7.6), along with visualized solutions from different regions of the output Pareto-approximation front. Figures 7.5b and 7.5d are the best solutions in terms of F_{spread} and F_{stress}, respectively. The fact that these solutions are optimal in one objective also means that they are the worst solution of the Pareto-approximation front in terms of the other objective. Figures 7.5b and 7.5d make it evident that these solutions are both not desirable, i.e., Figure 7.5b shows a distribution of points where similarly colored points are not mapped to the same region of the triangle, and Figure 7.5d shows large clumps of indistinguishable points. Rather, a solution from a non-extreme region of the Pareto-approximation front should be selected, as shown in Figure 7.5c. Automatic selection of a solution from the Pareto-approximation front could be performed based on, e.g., a linear combination of weights of the two objectives, or a region of interest on the Pareto-approximation front, which would also mean that the problem could be solved single-objectively. Automatic selection is however difficult to tune such that it selects a proper solution for each benchmark problem, and is moreover not the focus of this work. Solutions presented in this section are therefore manually selected from the Pareto-approximation set based on visual preference.

In Figures 7.6 and 7.7, we display the results of the experiments. The leftmost column shows 3-dimensional scatter plots of the input Pareto-approximation fronts. The second column from the left shows the output Pareto-approximation fronts of both optimization approaches, i.e., where either the maximum or the minimum of of each dimension is directed towards one vertex of the simplex. In case the maximum of each dimension is directed to the vertices, $\boldsymbol{V}_j^{\max}$ is used for $\boldsymbol{V}_j$ in Equation 7.3. In case the minimum of each dimension is directed to the vertices, $\boldsymbol{V}_j^{\min}$ is used for $\boldsymbol{V}_j$ in Equation 7.3. The position of the selected solutions, displayed in the two rightmost columns, are denoted on the output Pareto-approximation fronts.

We see that the points in Figures 7.6 and 7.7 are all spread out near-uniformly, and the points are clearly positioned in regions near points with similar colors, i.e., similar objective values. Moreover, points are correctly placed in the region of the triangle that corresponds to their coordinates in the original Pareto-approximation front, e.g., green points were originally located near $(0,1,0)$ and should therefore be mapped to the region near $y_{\max}$, or a region opposing $y_{\min}$.



(a) Output Pareto-approximation front.

(b) Optimal spread.

(c) Selected solution.

(d) Optimal stress.

Figure 7.5: Output Pareto-approximation front in Figure 7.5a, with the marked solutions from left to right displayed in Figures 7.5b to 7.5d.

7.5. Discussion and Conclusions

We have introduced a method for the mapping of any m-dimensional Pareto-approximation front to an $(m-1)$-simplex, achieving a more uniform spread of points that still accurately represents the partial ordering of quality of the associated solutions in each of the objectives of interest. A uniform spread of points is beneficial to the decision-making process when solutions are initially assumed to be of equal preference, because this makes each point individually distinguishable. Moreover, if a decision-making tool such as a set of sliders to control the relative preference of each objective is used, a uniform spread of points will avoid navigation through very dense or sparse regions of the solution space. Sparse regions are difficult to navigate through, because many possible objective weights in such regions will correspond to the same solution. Dense regions of the solution space make it difficult to distinguish each solution while in the parameter space these solutions may well be significantly different.

A time limit of one hour was used for the experiments in this chapter, but the key contribution of this chapter is the general idea of solving the remapping problem multi-objectively rather than a perfectly stream-lined algorithm. A variety of methods could be applied to improve the efficiency of the optimization. For instance, a multi-resolution approach could be used that starts by performing the optimization on a small, diverse, subset of points on the original Pareto-approximation front. Remaining points are then incrementally added at a later stage of the optimization procedure, and these points can be initialized around already mapped points that are close in the original Pareto-approximation front. Secondly, more advanced methods for

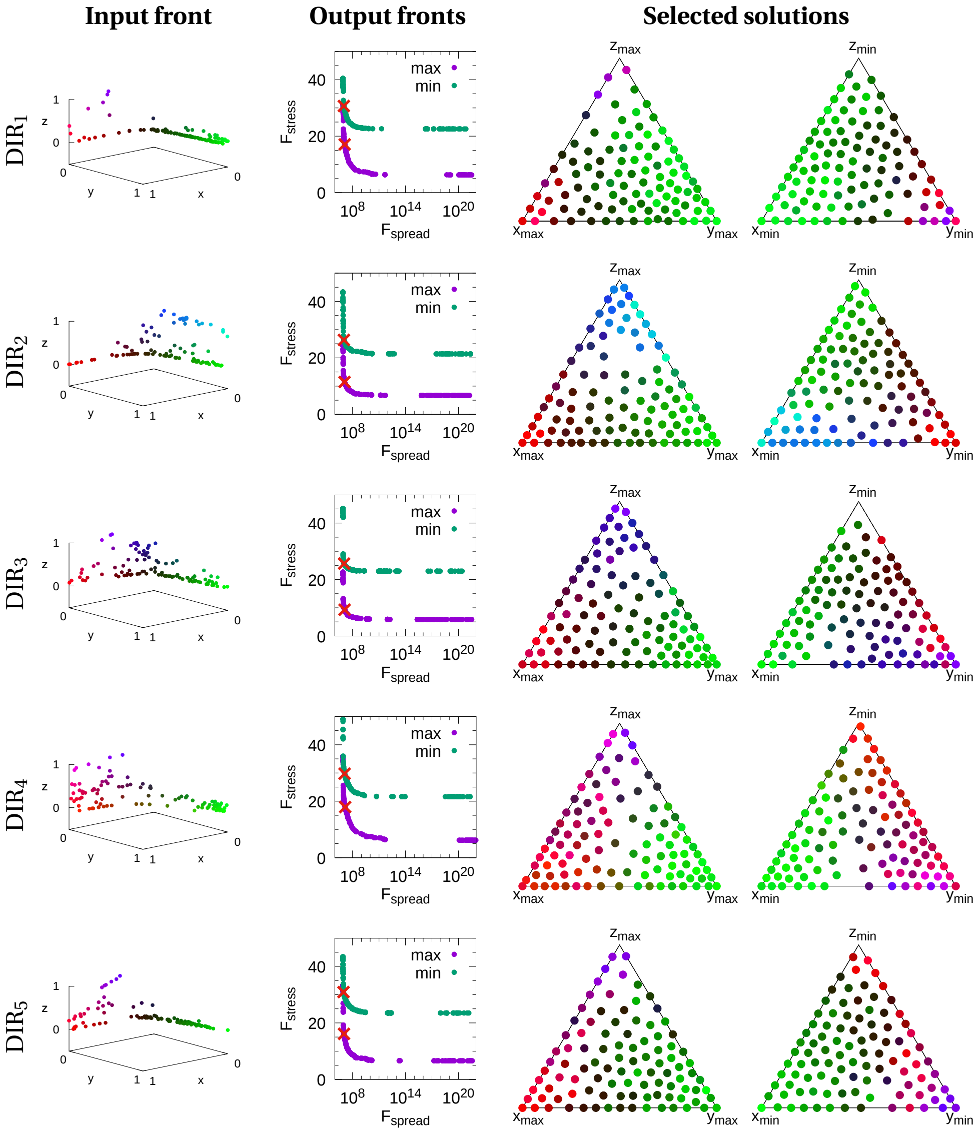


Figure 7.6: Results of benchmark problems DIR$_1$ through DIR$_5$. The leftmost column shows a 3-dimensional scatter plot of the input Pareto-approximation front. The second column shows output Pareto-approximation fronts of the approach with maximum values mapped to the vertices of the triangle (in purple) and the approach with minimum values mapped to the vertices of the triangle (in green). In this plot, the solutions displayed in the two rightmost columns are marked by a red cross.

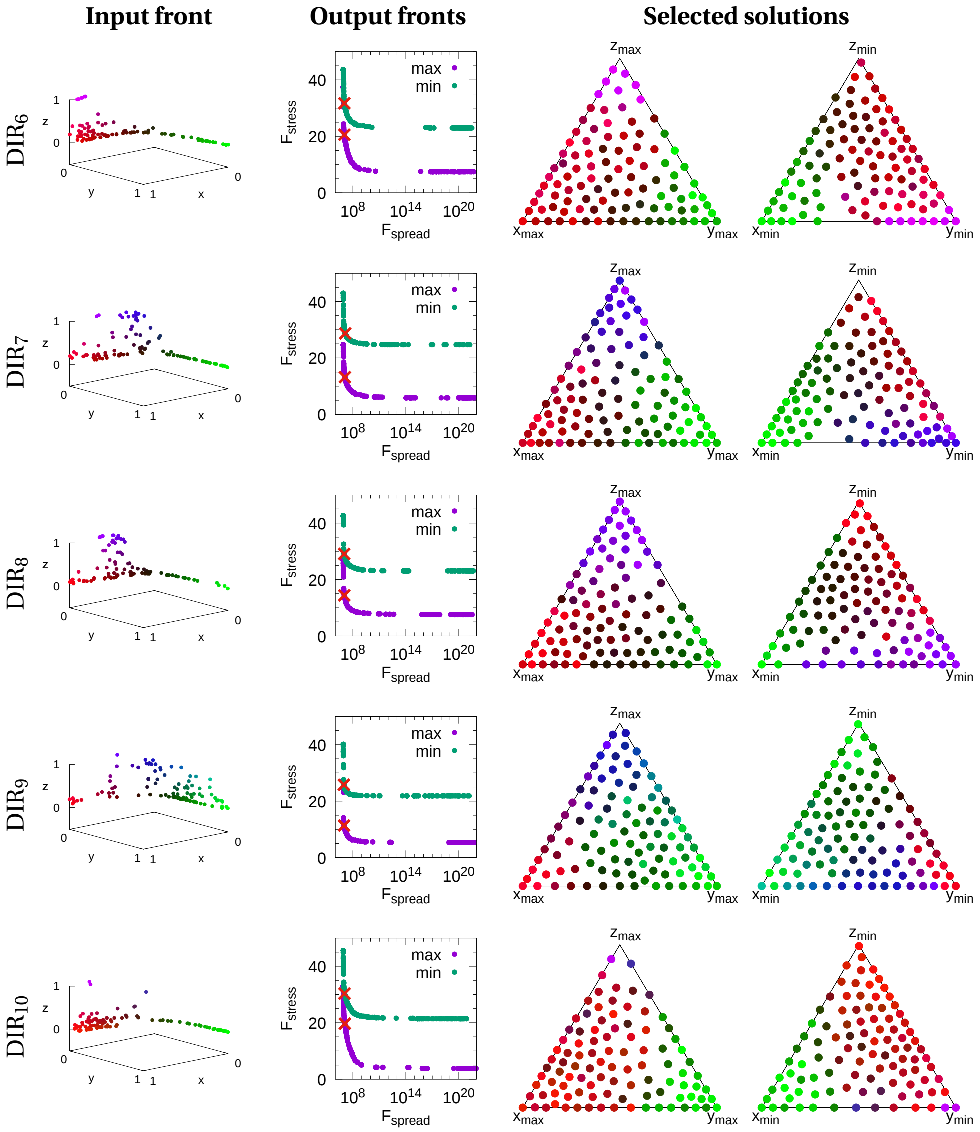


Figure 7.7: Results of benchmark problems DIR$_6$ through DIR$_{10}$. The leftmost column shows a 3-dimensional scatter plot of the input Pareto-approximation front. The second column shows output Pareto-approximation fronts of the approach with maximum values mapped to the vertices of the triangle (in purple) and the approach with minimum values mapped to the vertices of the triangle (in green). In this plot, the solutions displayed in the two rightmost columns are marked by a red cross.

the initialization of the population might be possible, which would speed up the optimization by providing better initial solutions.

An alternative approach to the redistribution of Pareto-approximation fronts could entail the formulation of the problem as a discrete permutation problem. With such an approach, a Pareto-approximation front would be mapped to a set of uniformly distributed locations on the simplex, one for each point on the input Pareto-approximation front. The problem then consists of finding the best one-to-one mapping of points on the input Pareto-approximation fronts to locations on the simplex. Using this approach, a single-objective optimization method can be used, but it is not known whether this approach is capable of producing good results for any possible input Pareto-approximation front.

We have tested our approach on a set of 3-dimensional Pareto-approximation fronts from the real-world problem known as DIR. Results show that we are well able to find 2-dimensional representations of these Pareto-approximation fronts where each point is easily distinguishable and points are located at an intuitive position with respect to their quality in each of the objectives of interest, enabling the design and use of effective decision support tools, e.g., in the case of the DIR problem in this chapter.

REFERENCES

[1] Alderliesten, T., Bosman, P. A. N., and Bel, A. (2015). Getting the most out of additional guidance information in deformable image registration by leveraging multi-objective optimization. In Ourselin, S. and Styner, M. A., editors, *Medical Imaging 2015: Image Processing*, volume 9413, pages 469 – 475. International Society for Optics and Photonics, SPIE.

[2] Bouter, A., Luong, N. H., Witteveen, C., Alderliesten, T., and Bosman, P. A. N. (2017a). The multi-objective real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 537–544. ACM.

[3] Bouter, A., Pirpinia, K., Alderliesten, T., and Bosman, P. A. N. (2017b). Spatial redistribution of irregularly-spaced Pareto fronts for more intuitive navigation and solution selection. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1697–1704.

[4] Blasco, X., Herrero, J. M., Sanchis, J., and Martínez, M. (2008). A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Information Sciences*, 178(20):3908–3924.

[5] Bosman, P. A. N. and Gallagher, M. (2016). The importance of implementation details and parameter settings in black-box optimization: A case study on Gaussian estimation-of-distribution algorithms and circles-in-a-square packing problems. *Soft Computing*, pages 1–15.

[6] Bosman, P. A. N. and Thierens, D. (2003). The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):174–188.

[7] Brock, K. K., Dawson, L. A., Sharpe, M. B., Moseley, D. J., and Jaffray, D. A. (2006). Feasibility of a novel deformable image registration technique to facilitate classification, targeting, and monitoring of tumor and normal tissue. *International Journal of Radiation Oncology* Biology* Physics*, 64(4):1245–1254.

[8] Chen, S., Amid, D., Shir, O. M., Limonad, L., Boaz, D., Anaby-Tavor, A., and Schreck, T. (2013). Self-organizing maps for multi-objective Pareto frontiers. In *2013 IEEE Pacific Visualization Symposium (PacificVis)*, pages 153–160.

[9] Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc.

[10] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

[11] Ibrahim, A., Rahnamayan, S., Martin, M. V., and Deb, K. (2016). 3D-RadVis: Visualization of Pareto front in many-objective optimization. In *IEEE Congress on Evolutionary Computation*, pages 736–745.

[12] Inselberg, A. and Dimsdale, B. (1990). Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proceedings of the First IEEE Conference on Visualization: Visualization ‘90*, pages 361–378.

[13] Kessler, M. L. (2006). Image registration and data fusion in radiation therapy. *The British Journal of Radiology*, 79(1):S99–S108.

[14] Koppen, M. and Yoshida, K. (2007). Visualization of Pareto-sets in evolutionary multi-objective optimization. In *Hybrid Intelligent Systems, 2007. HIS 2007. 7th International Conference on*, pages 156–161. IEEE.

[15] Luong, H. N. and Bosman, P. A. N. (2012). Elitist archiving for multi-objective evolutionary algorithms: To adapt or not to adapt. In *International Conference on Parallel Problem Solving from Nature*, pages 72–81. Springer.

[16] Pelikan, M., Sastry, K., and Goldberg, D. E. (2005). Multiobjective hBOA, clustering, and scalability. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 663–670. ACM.

[17] Pirpinia, K., Bosman, P. A. N., Sonke, J. J., van Herk, M., and Alderliesten, T. (2016). A first step toward uncovering the truth about weight tuning in deformable image registration. In Styner, M. A. and Angelini, E. D., editors, *Medical Imaging 2016: Image Processing*, volume 9784, pages 1099 – 1107. International Society for Optics and Photonics, SPIE.

[18] Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 100(5):401–409.

[19] Sotiras, A., Davatzikos, C., and Paragios, N. (2013). Deformable medical image registration: A survey. *IEEE Transactions on Medical Imaging*, 32(7):1153–1190.

[20] Tušar, T. and Filipič, B. (2015). Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. *IEEE Transactions on Evolutionary Computation*, 19(2):225–245.

[21] Walker, D. J., Everson, R. M., and Fieldsend, J. E. (2013). Visualising mutually nondominating solution sets in many-objective optimisation. *IEEE Transactions on Evolutionary Computation*, 17(2):165–184.

8

Discussion and Conclusions

The purpose of this chapter is to discuss the findings of this thesis, alongside its shortcomings and the future areas of research that it has uncovered. This is done by discussing the research question of each chapter and whether it has been answered.

Chapter 2

In Chapter 2 we stated the following research question:

> **Research Question: Chapter 2**
>
> How can the key strengths of GOMEA be exploited in real-valued optimization to achieve highly scalable GBO?

This research question was answered by the definitive introduction of the real-valued variant of GOMEA, named RV-GOMEA. In Chapter 2, a novel version of RV-GOMEA is introduced leveraging principles from CMA-ES [7, 8], a state-of-the-art EA for real-valued BBO, similar to the previously introduced version of RV-GOMEA [1], which was leveraging principles from AMaLGaM [3, 4]. These two versions of RV-GOMEA achieved very similar performance, with only minor differences on a select few benchmark problems, and both were capable of exploiting the GBO setting by using partial evaluations. We can therefore conclude that the key concepts of GOMEA do indeed transcend the binary domain and translate to the domain of real-valued optimization. Furthermore, these concepts can be applied in various ways, based on different real-valued optimizers, as demonstrated by the successful versions of RV-GOMEA. Also in different areas of EC and for different types of encodings or problem variables, including in the field of genetic programming and for permutation problems, these concepts have been successfully applied [5, 16].

A limiting factor of the GBO setting we consider is the fact that it relies on partial evaluations, which requires the suitability of the definition of the optimization function to a certain degree, i.e., as an aggregation of a number of subfunctions, as discussed in Section 1.2. Such partial evaluations have been used to vastly speed up the evolutionary optimization process in case of the real-world problems of treatment planning for prostate HDR brachytherapy, and deformable image registration, in Chapters 4 to 6. Furthermore, partial evaluations are also applied to the point scattering problem that is optimized for the purpose of visualizing high-dimensional approximation fronts in Chapter 7. For all of these applications, the optimization problem is challenging due to a combination of a multi-objective problem formulation with non-linear, non-separable objectives. Still, partial evaluations have a different effect on the complexity of evaluating the quality of a solution for each of these applications. In the brachytherapy treatment planning problem, the complexity of an evaluation is $\mathcal{O}(d\ell + d\log(d))$ for ℓ dwell times and d dose calculation points, and partial evaluations only reduce the first term of this complexity, i.e., the term caused by the dose calculation. For a partial evaluation of k variables, this complexity is reduced to $\mathcal{O}(dk + d\log(d))$, which does not look like a large improvement considering the fact that d is various orders of magnitude larger than ℓ, but certainly helps in a clinical setting where computation time is essential. For the DIR problem, a partial evaluation of k variables has a complexity of $\mathcal{O}(k)$, because a partial evaluation requires the evaluation of a constant number of triangles in the deformation grid. This constant, however, depends on the topology of the transformation grid, and is around a factor of 10 for a two-dimensional grid. Therefore, the speed-up of using partial evaluations becomes much less than a factor of ℓ, but this speed-up does increase for larger dimensional grids, which is essential for the future use of DIR to find three-dimensional deformations, because this involves substantially larger grid sizes. Finally, for the point scattering problem, the partial evaluation of k variables has a complexity of $\mathcal{O}(k\ell)$, while a full evaluation has a complexity of $\mathcal{O}(\ell^2)$.

It is important to note that not every scenario in which partial evaluations are possible, warrants their use. Firstly, when an optimization problem is known to be separable, optimizing each of these separated sub-problems in a parallel or interleaved fashion using BBO solvers is a simpler, potentially more effective, method than using partial evaluations. This is mainly the case for single-objective optimization, but also holds for multi-objective problems as long as a supposed separation of variables leads to blocks of variables that are independent with respect to each objective. In other scenarios, partial evaluations may be warranted, but their most effective use is not necessarily in combination with an EA. For example, the gradient-based L-BFGS [10] can benefit from partial evaluations, and will perform better than an EA on non-separable single-objective problems with weak dependencies between problem variables and smooth problem landscapes. However, on problems with non-smooth landscapes, L-BFGS will inevitably get stuck in a local minimum, as shown in Chapter 2. Multi-objective problems will require scalarization of the objective functions into a single objective, and running L-BFGS a large number of times with different weights. Since a scalarized approach makes it difficult to reach all areas of the Pareto front in cases of concave Pareto fronts or differently scaled objectives, and using restarts to find solutions with different trade-offs is not very efficient, the use of an EA for GBO becomes more interesting in this scenario. Given these different scenarios, we argue that the main use-case for partial evaluations is for multi-objective problems with non-separable objective functions.

A possible direction of future work for MO-RV-GOMEA is the multi-objective acceptation criterion that is used in GOM. Currently, iterations of GOM are accepted when resulting in a dominating solution, or a solution that is non-dominated by the elitist archive. The complexity of this step therefore scales with the target size of the elitist archive, and is not expected to perform well for many-objective, i.e., more than 3 conflicting objectives, optimization problems. An alternative acceptation criterion could be based on a scalarization of the objective values, e.g., linear or Tchebycheff scalarization. To spread the population toward different regions of the Pareto front, the weights of the scalarization would be different for different solutions in the population. Research in this direction has already been done, and has resulted in improved performance for various many-objective benchmark problems [12]. It has, however, not led to sufficiently good results for a many-objective version of treatment planning for HDR brachytherapy, because the many-objective results have a lower quality than the bi-objective results when mapped back to the bi-objective space. A more appropriate, problem-specific, selection of vectors may be necessary to push the population to the most interesting area of the Pareto front.

A different potential direction of research for MO-RV-GOMEA involves the fact that, in the current version of MO-RV-GOMEA, elitist solutions may be copied back to the population to replace low-quality solutions, with the purpose of replacing solutions that are hard to improve through GOM. If this purpose can be achieved in a different way, without interacting with the archive, it would not be necessary to save the partial objective values for the entire elitist archive, which could save a large amount of memory for various applications.

Chapter 3

A complication of GOM for non-separable problems with strong dependencies is that, when variation is applied to a small subset of problem variables, dependencies of these problem variables to other problem variables are not considered. This complication led to the statement of the following research question:

Research Question: Chapter 3

To what extent is RV-GOMEA limited in its effectiveness to exploit partial evaluations in a GBO setting in case of optimization problems with overlapping strong dependencies, and how could we improve this?

In Chapter 3, it was shown that RV-GOMEA only achieves good performance on problems with overlapping strong dependencies when a full linkage model is used. Such linkage models do, however, not scale to high-dimensional problems with more than a few hundreds of variables. For the purpose of improving on this, conditional linkage models for RV-GOMEA are introduced. These linkage models were shown to be more effective than non-conditional linkage models for problems with overlapping strong dependencies. Experimental results were, however, still restricted to a limited number of possible conditional linkage models.

The use of conditional linkage models does come with the cost of an increase in required computation time for the estimation of the sampling distributions and the sampling of the random variables. As long as each variable is dependent on a constant number of other variables, this does not increase the worst-case time complexity of the EA, but the inquired overhead may still have a large effect on the execution time. In a setting where computation time is important, a non-conditional linkage model may therefore still be preferable.

Even when using conditional linkage models, GOM is still applied to a limited subset of variables, because this is required to effectively exploit partial evaluations. It is therefore possible that, with respect to all non-modified variables, the variables to be modified are in a local optimum, meaning that no possible variation step will be accepted. Though this effect seems unavoidable, it may be alleviated by hierarchical conditional linkage models, because such models apply variation to differently-sized groups of variables. In particular, due to the good performance of the hybrid conditional linkage models, a conditional version of the linkage tree is a promising candidate for a well-performing model, as it was already shown to be successful in the domain of discrete optimization [6].

CHAPTER 4

Besides algorithmic improvements, the scalability and performance of EAs can also be greatly improved by the application of techniques from high-performance computing, in particular parallelization. The following research question was therefore stated in Chapter 4:

Research Question: Chapter 4

Which parallelization techniques are the most suitable for large-scale optimization in a GBO setting, particularly for GOMEA, and in which scenarios?

The main finding of Chapter 4 is that the most suitable parallelization technique for large-scale optimization in a GBO setting depends on the number of subfunctions that can be evaluated in parallel, and their computational complexity. In scenarios with at least a moderate number and moderate complexity of subfunctions, e.g., for at least 16 subfunctions that require 0.1 ms of computation time, Graphics Processing Units (GPUs) are the most suitable, including for the application of treatment planning for prostate HDR brachytherapy. This is caused by the fact that the evaluation of subfunctions can be done in parallel in a GBO setting, in addition to the parallel evaluation of the population. However, in Chapter 4, only the potential speed-up of the evaluation function was considered, regardless of the optimization algorithm used. In most real-valued EAs targeted at BBO, e.g., CMA-ES, this full potential for parallelization can be exploited, because each variation step leads to a modification of all variables, meaning that all subfunctions can be evaluated in parallel.

Compared to this, it is more difficult to exploit the full parallelization potential in RV-GOMEA, because the main strength of RV-GOMEA comes from the fact that partial modifications are made, and only improvements are accepted. To determine whether such a partial modification leads to an improvement, a partial evaluation is done. However, if such partial evaluations of subfunctions have problem variables in common, their contributions to the objective value are dependent, and cannot be done in parallel. Instead, only partial evaluations that are independent of each other may be evaluated in parallel. This means that there is a trade-off in a GBO setting, because the local improvements of RV-GOMEA have excellent synergy with a GBO setting, but EAs aimed at a BBO setting can exploit a higher degree of parallelization. The expectation is, however, that RV-GOMEA will perform better in a GBO setting, because the GOM operator leads to an improvement in scalability. Furthermore, as shown in Chapter 6, it is possible to find groups of independent variables to which GOM may be applied in parallel. Therefore, the observations in Chapter 4 show that, for both RV-GOMEA and for other EAs aimed at a BBO setting, a GBO setting that allows for partial evaluations is an excellent use-case for high-performance computing methods. Therefore, the use of a GBO setting may lead to substantial improvements in the efficiency of the optimization with an EA, though the use of a GBO setting is not very widespread in the EC community, least of all in the domain of real-valued optimization, despite their potential shown in Chapter 4.

CHAPTER 5

The findings of Chapter 4, lead to the statement of the following research question:

Research Question: Chapter 5

How can bi-objective treatment planning with MO-RV-GOMEA be accelerated so that it can be performed in a clinically acceptable amount of time?

In Chapter 5, we confirm that it is indeed possible to perform bi-objective treatment planning for HDR prostate brachytherapy with MO-RV-GOMEA in a clinically acceptable amount of time. This is achieved by using the large-scale parallel computing power of GPUs, mainly applied to accelerate dose calculation, and various algorithmic enhancements, such as filtering small linkage sets from the linkage model. We find that the setting using 20,000 dose calculation points per organ, as used in previous work [11], requires only 30 seconds of computation time. A more accurate setting using 100,000 dose calculation points per organ, which was never before reported in literature to have been achieved for clinical purposes, requires only 3 minutes, which was deemed clinically acceptable by medical experts.

We argue that the computation time of the treatment plan optimization can be cut down even more in the near future, due to a number of factors. The first of these factors is the rapid technological advancement of GPUs, because the most recently introduced NVIDIA GPUs, the NVIDIA RTX 3000 series, have not yet been used in this research, and promise a large increase in performance over their preceding series. This promise in an increase of performance is mainly caused by an increase in the number of cores, the clock frequency of each of the cores, and the memory bandwidth. Beside an increase in performance, the larger amount of GPU memory supplied by the new generation of GPUs may enable even more precise computation of DVIs. A second factor that may result in reduced computation time of the treatment planning procedure is the fact that the currently used random sampling of dose calculation points does not take into account that only the subvolumes of the OARs nearby the target volumes are of importance to the optimization. Random sampling therefore leads to a large number of redundant points inside the OARs.

One of the difficulties of treatment planning for prostate HDR brachytherapy, and arguably treatment planning in a general sense, is the fact that, despite a well-defined clinical protocol, many aspects of plan quality are not well-defined, but judged by the expertise of a medical expert. This mainly became clear in a study where catheter positions were optimized, which resulted in plans satisfying all clinical planning criteria specified in the clinical protocol. However, these plans were not clinically acceptable according to clinical experts [15]. One example of a feature that is not sufficiently well-defined in the clinical protocol is plan homogeneity, as clinical experience has shown that the optimization of the $V^{\text{prostate}}_{150\%}$ and the $V^{\text{prostate}}_{200\%}$, as specified in the clinical protocol, is not sufficient to produce sufficiently homogeneous treatment plans [2]. Furthermore, no planning aims for the normal-tissue dose are included in the protocol, meaning that the optimization suffers no penalty from creating large hotspots just outside the target volumes.

An additional difficulty that is specific to bi-objective treatment planning, is the fact that treatment plans that are adjacent in objective space, i.e., on the approximation front, may be very different in parameter space, i.e., in terms of dwell times. Therefore, a clinical expert that is analyzing the treatment plans on the front may see very sudden changes in the dose distribution when scrolling through the approximation front, which makes plan selection less intuitive. Initial research on the navigability of approximation fronts used Bézier curves to model smooth approximation fronts, where the dwell times of each treatment plan on the approximation front are a linear combination of the dwell times associated with the plans represented by the two end-points, i.e., control points of the Bézier curve, of the approximation front [13]. Future research on this topic includes the use of more than two control points for the Bézier curves, and showing not only one, but multiple smoothly navigable approximation fronts with different properties.

To enhance the quality of the optimization procedure, additional clinical planning criteria may be introduced. If such criteria can be defined in terms of DVIs, they may logically be included in either the LCI or the LSI. If this is not the case, various options may be considered. Firstly, an additional objective may be introduced. This does, however, greatly increase the difficulty of the optimization problem and thereby its time complexity, and increase the required target size of the elitist archive to obtain an approximation front that covers a sufficient proportion of the objective space. Secondly, additional optimization criteria may be introduced as a soft or hard constraint. When introduced as a soft constraint, a penalty term would be added to each objective value. A downside to this is the fact that the golden corner, i.e., the part of the objective space where all objectives are non-negative, no longer includes all solutions that satisfy all clinical planning criteria as specified in the clinical protocol. Instead, some solutions satisfying all clinical planning criteria are outside of the golden corner. When introduced as a hard constraint, the value of the metric is required to be non-zero for the solution to be considered feasible. It may however be difficult to find an appropriate threshold for the additionally introduced optimization metric, because this threshold may be patient-specific, and have a large influence on the optimization process. The reason for this is that setting the threshold too low will not result in sufficiently good solutions, and setting the threshold too high will result in a large number of sufficiently good solutions being infeasible.

CHAPTER 6

As discussed in regard to the research question of Chapter 4, GOM may simultaneously be applied to several independent groups of variables. Because we observed that the DIR application has a structure that may be distributed into a large number of independent groups of variables, we hypothesize that large-scale parallelization may greatly improve the efficiency of this application. We therefore stated the following research question:

Research Question: Chapter 6

Can RV-GOMEA be accelerated by leveraging the potential for large-scale parallelization caused by the dependency structure and large dimensionality of problems such as DIR?

Applying GOM to a large number of FOS elements in parallel, i.e., parallel GOM, is very much suited for the DIR application, which is mainly caused by the fact that the GBO definition, as described in Chapter 4, of the DIR application, has the identity function as its mapping function. This means that the contributions to the objective values of independent partial evaluations may simply be summed together to obtain their joint contributions.

In the application of treatment planning for HDR prostate brachytherapy, there is a non-linear mapping function, making the application of large-scale parallel GOM more difficult, because all variables are dependent on all other variables, with a variable dependency strength. To apply parallel GOM to such an application, defining a threshold for the interaction strength between variables, below which they may be considered independent for the purpose of parallel GOM, is required. However, the parallel partial evaluations only determine whether each of the GOM iterations is accepted or rejected. Subsequently, due to the actual overlap in dependencies, calculating the joint contribution of the accepted partial evaluations requires a full evaluation, which may even lead to a rejection of the accepted partial improvements, because they jointly have a negative effect on the fitness. Furthermore, due to the relatively small number of variables compared to DIR and the non-linear mapping function, parallel GOM is not expected to result in a large increase in efficiency for the HDR prostate brachytherapy application.

For maximum efficiency of parallel GOM, all FOS elements need to be distributed into the minimum number of sets that each contain FOS elements that are all independent of each other (and the fitness function at hand). Finding the optimal distribution is equivalent to finding the minimal graph coloring on the graph where each node represents a FOS element, and each edge represents a dependency between

FOS elements, because at least one variable in each of these elements is dependent on at least one variable in the other FOS element in the fitness function. Because graph coloring is an NP-hard problem [9], a heuristic is required to find a sufficiently good graph coloring while ensuring that run times remain acceptable. However, a linkage model with a lower capacity, i.e., one that leads to a higher number of required evaluations in order to reach sufficiently good solutions, may have a higher parallelizability than a linkage model with a higher capacity. Therefore, a linkage model with a lower capacity may be more efficient, in terms of computation time, when parallel GOM is applied, but less efficient when it is not. Due to this dependency between the capacity and the parallelizability of linkage models, combined with the fact that it may not be computationally feasible to find the optimal graph coloring, it is difficult to determine what the optimal linkage model is when applying GPU parallelization to GOM.

CHAPTER 7

When optimizing a multi-objective problem, it is generally necessary to have an expert with domain knowledge make a final selection of the most suitable solution from the approximation front. For problems with more than two objectives, it starts becoming difficult to clearly view all the possibilities in the approximation front and their trade-offs. This is, for example, the case for the DIR problem, which may result in three-dimensional approximation fronts. Furthermore, the approximation fronts found for this problem are generally irregular, meaning that solutions are not uniformly spread across the objective space, making navigation among the solutions on the approximation front even more difficult. The following research question was therefore addressed in Chapter 7:

Research Question: Chapter 7

Can irregularly distributed multi-dimensional approximation fronts be intuitively mapped to a lower-dimensional space to help navigation and solution selection?

In Chapter 7, it is shown that this is indeed possible, through the introduction of a bi-objective problem formulation that is optimized with MO-RV-GOMEA in a GBO setting that allows for partial evaluations. For non-regular three-dimensional fronts, mapping them to regularly spaced points on a two-dimensional simplex greatly increased the usability in a real-world use case of DIR [14].

Due to the fact that the introduced problem formulation excellently lends itself for optimization in a GBO setting that allows for partial evaluations, this chapter furthermore showcases an additional real-world application of RV-GOMEA. As stated previously, the partial evaluations for this application have linear time complexity, compared to full evaluations that have a quadratic time complexity.

In this work, the dimensionality of the space was only reduced by a single dimension, but this method may potentially be applied to reduce the dimensionality of higher-dimensional approximation fronts to two dimensions.

CONCLUSIONS

The main research goal of this thesis was to design real-valued MBEAs capable of efficiently scaling to large-scale optimization problems with thousands or even millions of variables. For this purpose, the real-valued variant of GOMEA, named RV-GOMEA, has been introduced in this thesis, which has been shown to achieve excellent performance and scalability. In a GBO setting where partial evaluations are possible, RV-GOMEA has achieved unprecedented scalability, and was able to achieve near-optimal values on the 5 million-dimensional sphere function within one hour on a single CPU core present in a normal desktop computer. With the introduction of conditional linkage models, improvements over the original RV-GOMEA were made on one of the more difficult types of problems, i.e., those with overlapping strong dependencies. RV-GOMEA has also been successfully applied to the problem of treatment plan optimization for prostate HDR brachytherapy, leading to its clinical introduction in March 2020 at the Amsterdam UMC, location AMC. Finally, the application of RV-GOMEA to the DIR problem in a GBO setting has resulted in a speed-up of several orders of magnitude compared to previous results using MAMaLGaM. These results show that RV-GOMEA has great potential for large-scale optimization, in particular on problems where a GBO setting can be exploited, opening the door to many future applications.

REFERENCES

[1] Bouter, A., Alderliesten, T., Witteveen, C., and Bosman, P. A. N. (2017). Exploiting linkage information in real-valued optimization with the real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 705–712. ACM.

[2] Barten, D. L. J., Bouter, A., van Wieringen, N., Pieters, B. R., Hinnen, K. A., Westerveld, H., Maree, S. C., van der Meer, M. C., Alderliesten, T., Bosman, P. A. N., Niatsetski, Y., and Bel, A. (2021). Artificial intelligence based planning of HDR prostate brachytherapy: First clinical experience. *Radiotherapy and Oncology*, 161:S653–S655.

[3] Bosman, P. A. N., Grahl, J., and Thierens, D. (2009). AMaLGaM IDEAs in noiseless black-box optimization benchmarking. In *Proceedings of the Genetic and Evolutionary Computation Conference: Late Breaking Papers*, pages 2247–2254. ACM.

[4] Bosman, P. A. N., Grahl, J., and Thierens, D. (2013). Benchmarking parameter-free AMaLGaM on functions with and without noise. *Evolutionary Computation*, 21(3):445–469.

[5] Bosman, P. A. N., Luong, N. H., and Thierens, D. (2016). Expanding from discrete Cartesian to permutation gene-pool optimal mixing evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 637–644. ACM.

[6] Dushatskiy, A., Virgolin, M., Bouter, A., Thierens, D., and Bosman, P. A. N. (2021). Parameterless gene-pool optimal mixing evolutionary algorithms. *arXiv preprint arXiv:2109.05259*.

[7] Hansen, N., Müller, S., and Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18.

[8] Hansen, N. and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195.

[9] Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer.

[10] Liu, D. C. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528.

[11] Luong, N. H., Alderliesten, T., Bel, A., Niatsetski, Y., and Bosman, P. A. N. (2018a). Application and benchmarking of multi-objective evolutionary algorithms on high-dose-rate brachytherapy planning for prostate cancer treatment. *Swarm and Evolutionary Computation*, 40:37–52.

[12] Luong, N. H., Alderliesten, T., and Bosman, P. A. N. (2018b). Improving the performance of MO-RV-GOMEA on problems with many objectives using Tchebycheff scalarizations. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 705–712.

[13] Maree, S. C., Alderliesten, T., and Bosman, P. A. N. (2020). Ensuring smoothly navigable approximation sets by Bézier curve parameterizations in evolutionary bi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 215–228. Springer.

[14] Pirpinia, K., Bosman, P. A. N., Loo, C. E., Russell, N. S., van Herk, M. B., and Alderliesten, T. (2018). Simplex-based navigation tool for a posteriori selection of the preferred deformable image registration outcome from a set of trade-off solutions obtained with multiobjective optimization for the case of breast MRI. *Journal of Medical Imaging*, 5(4):1 – 10.

[15] van der Meer, M. C., Pieters, B. R., Niatsetski, Y., Alderliesten, T., Bel, A., and Bosman, P. A. N. (2018). Better and faster catheter position optimization in HDR brachytherapy for prostate cancer using multi-objective real-valued GOMEA. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1387–1394.

[16] Virgolin, M., Alderliesten, T., Witteveen, C., and Bosman, P. A. N. (2017). Scalable genetic programming by gene-pool optimal mixing and input-space entropy-based building-block learning. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1041–1048. ACM.

ACKNOWLEDGMENTS

Now that I've arrived at the end of the journey that was my Ph.D., I would like to thank each and every person who helped me get this far. This is not limited to those mentioned here, but also includes everyone else who has supported me in any way during the time of my Ph.D.

First and foremost, I owe a lot to my supervisors Peter and Tanja, who always made me strive for excellence. I can confidently say that Tanja's eye for perfection, and Peter's endless expertise on evolutionary computation, had an immensely positive impact on my work. A word of thanks also goes out to Cees, who was shortly part of my supervisory team during the early days of my Ph.D.

This project would not have been possible without the required funding from NWO and Elekta, which I am very grateful for. Furthermore, I gratefully acknowledge the support of NVIDIA Corporation with the donation of a Titan X GPU used for this research.

In particular, I am grateful for project partner Elekta, who was always been very involved in our research project, e.g., by consistently attending project meetings and regularly welcoming me and my project partners at their office in Veenendaal. Thank you, Yury, Joost, Bob, and Roel, for a great collaboration, and the many lunch walks through Veenendaal.

Though some would say that a Ph.D. is a lonely journey, this was certainly not the case for me. I was very lucky to have the best imaginable project partners in Hoang, Stef, and Marjolein, with whom there was never a dull moment. I am very proud of what we as a team have achieved, and I could not have wished for smarter or more passionate project partners.

I have very much felt at home at CWI during the years of my Ph.D., and I have met many memorable people there that I owe this to. A special word goes out to Marco who had just started his Ph.D. at CWI on the day that I started my M.Sc. thesis there, becoming office mates for years to come. Thank you, Marco, for being a great friend for all these years. I miss the days of hearing Disney songs, and the occasional growling, echoing through the office.

The results that our project team achieved would not have been possible without the expertise of many medical professionals working at the AMC, for example Arjan, Niek, Danique, Kees, Bradley, Henrike, Karel, and Lukas. Thank you, Bradley, for being our clinical guiding light during many project meetings. Moreover, A special word of thanks goes out to Danique, who really got the ball rolling for BRIGHT to be clinically introduced. I can confidently say that this would have never happened without your guidance. Thank you for an excellent collaboration.

Next I would like to thank my parents and other family members. Mainly my parents have always cared for me and supported me, and have guided me in becoming what I am today.

Last but not least, I would like to thank the sweetest and most caring girlfriend I could have wished for, Carmen, who has constantly been by my side during the final stretch of the writing of this thesis. I am very grateful and happy to be with you and I love all our moments together. Furthermore, I hope that the journey of your Ph.D. will be as exciting as mine. Te quiero.

CURRICULUM VITÆ

Anton Bouter was born on the 7th of November of 1993 in Capelle aan den IJssel, the Netherlands. After getting his *Gymnasium* diploma from the Emmauscollege, Rotterdam in 2011, he enrolled in the Bachelor's studies Computer Science at the Delft University of Technology. In 2014, he graduated *cum laude* and with the completion of the honours programme, for which he competed in various editions of the ACM International Collegiate Programming Contest (ICPC). For his Bachelor's thesis, he worked on resource constrained project scheduling for the optimization of flexible schedules for train maintenance. This was done as an internship at the company NedTrain, responsible for maintenance of the trains used on the Dutch railway network.

Anton then started to pursue a Master's degree in Computer Science at the Delft University of Technology, and graduated in 2016. He selected the Software Technology track, with a focus on algorithmics, completing courses such as Advanced Algorithms, Parallel Algorithms and Parallel Computers, Randomized Algorithms, and Game Theory. His Master's thesis was done at the Dutch National Centre for Mathematics and Computer Science (CWI), supervised by prof. dr. Peter A.N. Bosman, on the design and application of model-based evolutionary algorithms for the multi-objective deformable image registration problem, which is also one of the medical applications described in this dissertation. This resulted in the initial version of the Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm (RV-GOMEA).

After his graduation in 2016, Anton accepted a Ph.D. position at CWI, supervised by prof. dr. Peter A.N. Bosman and dr. Tanja Alderliesten, which culminated with the release of this dissertation.

Finally, Anton has been a Scientific Software developer in the Evolutionary Intelligence group of CWI since 2020, trying to bridge the gap between academia and real-world applications.

LIST OF PUBLICATIONS

18. Scholman, R. J., **Bouter, A.**, Dickhoff, L. R. M., Alderliesten, T., and Bosman, P. A. N. (2022). Obtaining smoothly navigable approximation sets in bi-objective multimodal optimization. In *Parallel Problem Solving from Nature – PPSN XVII*, pages 247–262, Cham. Springer International Publishing

17. **Bouter, A.** and Bosman, P. A. N. (2022). GPU-accelerated parallel gene-pool optimal mixing in a gray-box optimization setting. In *Proceedings of the Genetic and Evolutionary Computation Conference*, page 675–683, New York, NY, USA. Association for Computing Machinery

16. Commandeur, J. L., **Bouter, A.**, Dickhoff, L. R., Barten, D. L., Westerveld, H., Pieters, B. R., Alderliesten, T., and Bosman, P. A. N. (2022). Incorporating control of contiguous high-dose volumes in automated optimization for prostate BT. *Radiotherapy and Oncology*, 170:S232–S234

15. Barten, D. L. J., **Bouter, A.**, van Wieringen, N., Pieters, B. R., Hinnen, K. A., Westerveld, H., Maree, S. C., van der Meer, M. C., Alderliesten, T., Bosman, P. A. N., Niatsetski, Y., and Bel, A. (2021). Artificial intelligence based planning of HDR prostate brachytherapy: First clinical experience. *Radiotherapy and Oncology*, 161:S653–S655

14. Dushatskiy, A., Virgolin, M., **Bouter, A.**, Thierens, D., and Bosman, P. A. N. (2021). Parameterless gene-pool optimal mixing evolutionary algorithms. *arXiv preprint arXiv:2109.05259*

13. **Bouter, A.**, Alderliesten, T., and Bosman, P. A. N. (2021b). GPU-accelerated parallel gene-pool optimal mixing applied to multi-objective deformable image registration. In *IEEE Congress on Evolutionary Computation*, pages 2539–2548. IEEE

12. **Bouter, A.**, Alderliesten, T., Pieters, B. R., Buus, S., Niatsetski, Y., and Bosman, P. A. N. (2021c). A multi-protocol validation study of automated bi-objective planning for HDR prostate brachytherapy. *Radiotherapy and Oncology*, 158:S195–S197

11. **Bouter, A.**, Alderliesten, T., and Bosman, P. A. N. (2021a). Achieving highly scalable evolutionary real-valued optimization by exploiting partial evaluations. *Evolutionary Computation*, 29(1):129–155

10. Olieman, C., **Bouter, A.**, and Bosman, P. A. N. (2020). Fitness-based linkage learning in the real-valued gene-pool optimal mixing evolutionary algorithm. *IEEE Transactions on Evolutionary Computation*

9. **Bouter, A.**, Maree, S. C., Alderliesten, T., and Bosman, P. A. N. (2020). Leveraging conditional linkage models in gray-box optimization with the real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 603–611

8. **Bouter, A.**, Alderliesten, T., Pieters, B. R., Bel, A., Niatsetski, Y., and Bosman, P. A. N. (2019b). GPU-accelerated bi-objective treatment planning for prostate high-dose-rate brachytherapy. *Medical Physics*, 46(9):3776–3787

7. **Bouter, A.**, Alderliesten, T., Pieters, B., Bel, A., Niatsetski, Y., and Bosman, P. A. N. (2019a). Bi-objective optimization of dosimetric indices for HDR prostate brachytherapy within 30 seconds. *Radiotherapy and Oncology*, 133:S199–S200

6. **Bouter, A.**, Alderliesten, T., Bel, A., Witteveen, C., and Bosman, P. A. N. (2018). Large-scale parallelization of partial evaluations in evolutionary algorithms for real-world problems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1199–1206. ACM

5. **Bouter, A.**, Pirpinia, K., Alderliesten, T., and Bosman, P. A. N. (2017d). Spatial redistribution of irregularly-spaced Pareto fronts for more intuitive navigation and solution selection. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1697–1704

4. Luong, N. H., **Bouter, A.**, van der Meer, M. C., Niatsetski, Y., Witteveen, C., Bel, A., Alderliesten, T., and Bosman, P. A. N. (2017). Efficient, effective, and insightful tackling of the high-dose-rate brachytherapy treatment planning problem for prostate cancer using evolutionary multi-objective optimization algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1372–1379. ACM

3. **Bouter, A.**, Luong, N. H., Witteveen, C., Alderliesten, T., and Bosman, P. A. N. (2017c). The multi-objective real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 537–544. ACM

2. **Bouter, A.**, Alderliesten, T., Witteveen, C., and Bosman, P. A. N. (2017b). Exploiting linkage information in real-valued optimization with the real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 705–712. ACM

1. **Bouter, A.**, Alderliesten, T., and Bosman, P. A. N. (2017a). A novel model-based evolutionary algorithm for multi-objective deformable image registration with content mismatch and large deformations: benchmarking efficiency and quality. In Styner, M. A. and Angelini, E. D., editors, *Medical Imaging 2017: Image Processing*, volume 10133, pages 304 – 311. International Society for Optics and Photonics, SPIE

SIKS Dissertation Series

2016 01 Syed Saiden Abbas (RUN), Recognition of Shapes by Humans and Machines

02 Michiel Christiaan Meulendijk (UU), Optimizing medication reviews through decision support: prescribing a better pill to swallow

03 Maya Sappelli (RUN), Knowledge Work in Context: User Centered Knowledge Worker Support

04 Laurens Rietveld (VU), Publishing and Consuming Linked Data

05 Evgeny Sherkhonov (UVA), Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers

06 Michel Wilson (TUD), Robust scheduling in an uncertain environment

07 Jeroen de Man (VU), Measuring and modeling negative emotions for virtual training

08 Matje van de Camp (TiU), A Link to the Past: Constructing Historical Social Networks from Unstructured Data

09 Archana Nottamkandath (VU), Trusting Crowdsourced Information on Cultural Artefacts

10 George Karafotias (VUA), Parameter Control for Evolutionary Algorithms

11 Anne Schuth (UVA), Search Engines that Learn from Their Users

12 Max Knobbout (UU), Logics for Modelling and Verifying Normative Multi-Agent Systems

13 Nana Baah Gyan (VU), The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach

14 Ravi Khadka (UU), Revisiting Legacy Software System Modernization

15 Steffen Michels (RUN), Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments

16 Guangliang Li (UVA), Socially Intelligent Autonomous Agents that Learn from Human Reward

17 Berend Weel (VU), Towards Embodied Evolution of Robot Organisms

18 Albert Meroño Peñuela (VU), Refining Statistical Data on the Web

19 Julia Efremova (Tu/e), Mining Social Structures from Genealogical Data

20 Daan Odijk (UVA), Context & Semantics in News & Web Search

21 Alejandro Moreno Célleri (UT), From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground

22 Grace Lewis (VU), Software Architecture Strategies for Cyber-Foraging Systems

23 Fei Cai (UVA), Query Auto Completion in Information Retrieval

24 Brend Wanders (UT), Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach

25 Julia Kiseleva (TU/e), Using Contextual Information to Understand Searching and Browsing Behavior
26 Dilhan Thilakarathne (VU), In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains
27 Wen Li (TUD), Understanding Geo-spatial Information on Social Media
28 Mingxin Zhang (TUD), Large-scale Agent-based Social Simulation - A study on epidemic prediction and control
29 Nicolas Höning (TUD), Peak reduction in decentralised electricity systems - Markets and prices for flexible planning
30 Ruud Mattheij (UvT), The Eyes Have It
31 Mohammad Khelghati (UT), Deep web content monitoring
32 Eelco Vriezekolk (UT), Assessing Telecommunication Service Availability Risks for Crisis Organisations
33 Peter Bloem (UVA), Single Sample Statistics, exercises in learning from just one example
34 Dennis Schunselaar (TUE), Configurable Process Trees: Elicitation, Analysis, and Enactment
35 Zhaochun Ren (UVA), Monitoring Social Media: Summarization, Classification and Recommendation
36 Daphne Karreman (UT), Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies
37 Giovanni Sileno (UvA), Aligning Law and Action - a conceptual and computational inquiry
38 Andrea Minuto (UT), Materials that Matter - Smart Materials meet Art & Interaction Design
39 Merijn Bruijnes (UT), Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect
40 Christian Detweiler (TUD), Accounting for Values in Design
41 Thomas King (TUD), Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance
42 Spyros Martzoukos (UVA), Combinatorial and Compositional Aspects of Bilingual Aligned Corpora
43 Saskia Koldijk (RUN), Context-Aware Support for Stress Self-Management: From Theory to Practice
44 Thibault Sellam (UVA), Automatic Assistants for Database Exploration
45 Bram van de Laar (UT), Experiencing Brain-Computer Interface Control
46 Jorge Gallego Perez (UT), Robots to Make you Happy
47 Christina Weber (UL), Real-time foresight - Preparedness for dynamic innovation networks
48 Tanja Buttler (TUD), Collecting Lessons Learned
49 Gleb Polevoy (TUD), Participation and Interaction in Projects. A Game-Theoretic Analysis
50 Yan Wang (UVT), The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

2017 01 Jan-Jaap Oerlemans (UL), Investigating Cybercrime

02 Sjoerd Timmer (UU), Designing and Understanding Forensic Bayesian Networks using Argumentation

03 Daniël Harold Telgen (UU), Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines

04 Mrunal Gawade (CWI), Multi-core Parallelism in a Column-store

05 Mahdieh Shadi (UVA), Collaboration Behavior

06 Damir Vandic (EUR), Intelligent Information Systems for Web Product Search

07 Roel Bertens (UU), Insight in Information: from Abstract to Anomaly

08 Rob Konijn (VU) , Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery

09 Dong Nguyen (UT), Text as Social and Cultural Data: A Computational Perspective on Variation in Text

10 Robby van Delden (UT), (Steering) Interactive Play Behavior

11 Florian Kunneman (RUN), Modelling patterns of time and emotion in Twitter #anticipointment

12 Sander Leemans (TUE), Robust Process Mining with Guarantees

13 Gijs Huisman (UT), Social Touch Technology - Extending the reach of social touch through haptic technology

14 Shoshannah Tekofsky (UvT), You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior

15 Peter Berck (RUN), Memory-Based Text Correction

16 Aleksandr Chuklin (UVA), Understanding and Modeling Users of Modern Search Engines

17 Daniel Dimov (UL), Crowdsourced Online Dispute Resolution

18 Ridho Reinanda (UVA), Entity Associations for Search

19 Jeroen Vuurens (UT), Proximity of Terms, Texts and Semantic Vectors in Information Retrieval

20 Mohammadbashir Sedighi (TUD), Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility

21 Jeroen Linssen (UT), Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)

22 Sara Magliacane (VU), Logics for causal inference under uncertainty

23 David Graus (UVA), Entities of Interest — Discovery in Digital Traces

24 Chang Wang (TUD), Use of Affordances for Efficient Robot Learning

25 Veruska Zamborlini (VU), Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search

26 Merel Jung (UT), Socially intelligent robots that understand and respond to human touch

27 Michiel Joosse (UT), Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors

28 John Klein (VU), Architecture Practices for Complex Contexts

29 Adel Alhuraibi (UvT), From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT"

30 Wilma Latuny (UvT), The Power of Facial Expressions

31 Ben Ruijl (UL), Advances in computational methods for QFT calculations

32 Thaer Samar (RUN), Access to and Retrievability of Content in Web Archives

33 Brigit van Loggem (OU), Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity

34 Maren Scheffel (OU), The Evaluation Framework for Learning Analytics

35 Martine de Vos (VU), Interpreting natural science spreadsheets

36 Yuanhao Guo (UL), Shape Analysis for Phenotype Characterisation from High-throughput Imaging

37 Alejandro Montes Garcia (TUE), WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy

38 Alex Kayal (TUD), Normative Social Applications

39 Sara Ahmadi (RUN), Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR

40 Altaf Hussain Abro (VUA), Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems

41 Adnan Manzoor (VUA), Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle

42 Elena Sokolova (RUN), Causal discovery from mixed and missing data with applications on ADHD datasets

43 Maaike de Boer (RUN), Semantic Mapping in Video Retrieval

44 Garm Lucassen (UU), Understanding User Stories - Computational Linguistics in Agile Requirements Engineering

45 Bas Testerink (UU), Decentralized Runtime Norm Enforcement

46 Jan Schneider (OU), Sensor-based Learning Support

47 Jie Yang (TUD), Crowd Knowledge Creation Acceleration

48 Angel Suarez (OU), Collaborative inquiry-based learning

2018 01 Han van der Aa (VUA), Comparing and Aligning Process Representations

02 Felix Mannhardt (TUE), Multi-perspective Process Mining

03 Steven Bosems (UT), Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction

04 Jordan Janeiro (TUD), Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks

05 Hugo Huurdeman (UVA), Supporting the Complex Dynamics of the Information Seeking Process

06 Dan Ionita (UT), Model-Driven Information Security Risk Assessment of Socio-Technical Systems

07 Jieting Luo (UU), A formal account of opportunism in multi-agent systems

08 Rick Smetsers (RUN), Advances in Model Learning for Software Systems

09 Xu Xie (TUD), Data Assimilation in Discrete Event Simulations

10 Julienka Mollee (VUA), Moving forward: supporting physical activity behavior change through intelligent technology

11 Mahdi Sargolzaei (UVA), Enabling Framework for Service-oriented Collaborative Networks

12 Xixi Lu (TUE), Using behavioral context in process mining

13 Seyed Amin Tabatabaei (VUA), Computing a Sustainable Future

14 Bart Joosten (UVT), Detecting Social Signals with Spatiotemporal Gabor Filters

15 Naser Davarzani (UM), Biomarker discovery in heart failure

16 Jaebok Kim (UT), Automatic recognition of engagement and emotion in a group of children

17 Jianpeng Zhang (TUE), On Graph Sample Clustering

18 Henriette Nakad (UL), De Notaris en Private Rechtspraak

19 Minh Duc Pham (VUA), Emergent relational schemas for RDF

20 Manxia Liu (RUN), Time and Bayesian Networks

21 Aad Slootmaker (OUN), EMERGO: a generic platform for authoring and playing scenario-based serious games

22 Eric Fernandes de Mello Araújo (VUA), Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks

23 Kim Schouten (EUR), Semantics-driven Aspect-Based Sentiment Analysis

24 Jered Vroon (UT), Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots

25 Riste Gligorov (VUA), Serious Games in Audio-Visual Collections

26 Roelof Anne Jelle de Vries (UT),Theory-Based and Tailor-Made: Motivational Messages for Behavior Change Technology

27 Maikel Leemans (TUE), Hierarchical Process Mining for Scalable Software Analysis

28 Christian Willemse (UT), Social Touch Technologies: How they feel and how they make you feel

29 Yu Gu (UVT), Emotion Recognition from Mandarin Speech

30 Wouter Beek, The "K" in "semantic web" stands for "knowledge": scaling semantics to the web

---

2019 01 Rob van Eijk (UL),Web privacy measurement in real-time bidding systems. A graph-based approach to RTB system classification

02 Emmanuelle Beauxis Aussalet (CWI, UU), Statistics and Visualizations for Assessing Class Size Uncertainty

03 Eduardo Gonzalez Lopez de Murillas (TUE), Process Mining on Databases: Extracting Event Data from Real Life Data Sources

04 Ridho Rahmadi (RUN), Finding stable causal structures from clinical data

05 Sebastiaan van Zelst (TUE), Process Mining with Streaming Data

06 Chris Dijkshoorn (VU), Nichesourcing for Improving Access to Linked Cultural Heritage Datasets

07 Soude Fazeli (TUD), Recommender Systems in Social Learning Platforms

08 Frits de Nijs (TUD), Resource-constrained Multi-agent Markov Decision Processes

09 Fahimeh Alizadeh Moghaddam (UVA), Self-adaptation for energy efficiency in software systems

10 Qing Chuan Ye (EUR), Multi-objective Optimization Methods for Allocation and Prediction
11 Yue Zhao (TUD), Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs
12 Jacqueline Heinerman (VU), Better Together
13 Guanliang Chen (TUD), MOOC Analytics: Learner Modeling and Content Generation
14 Daniel Davis (TUD), Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses
15 Erwin Walraven (TUD), Planning under Uncertainty in Constrained and Partially Observable Environments
16 Guangming Li (TUE), Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models
17 Ali Hurriyetoglu (RUN),Extracting actionable information from microtexts
18 Gerard Wagenaar (UU), Artefacts in Agile Team Communication
19 Vincent Koeman (TUD), Tools for Developing Cognitive Agents
20 Chide Groenouwe (UU), Fostering technically augmented human collective intelligence
21 Cong Liu (TUE), Software Data Analytics: Architectural Model Discovery and Design Pattern Detection
22 Martin van den Berg (VU),Improving IT Decisions with Enterprise Architecture
23 Qin Liu (TUD), Intelligent Control Systems: Learning, Interpreting, Verification
24 Anca Dumitrache (VU), Truth in Disagreement - Crowdsourcing Labeled Data for Natural Language Processing
25 Emiel van Miltenburg (VU), Pragmatic factors in (automatic) image description
26 Prince Singh (UT), An Integration Platform for Synchromodal Transport
27 Alessandra Antonaci (OUN), The Gamification Design Process applied to (Massive) Open Online Courses
28 Esther Kuindersma (UL), Cleared for take-off: Game-based learning to prepare airline pilots for critical situations
29 Daniel Formolo (VU), Using virtual agents for simulation and training of social skills in safety-critical circumstances
30 Vahid Yazdanpanah (UT), Multiagent Industrial Symbiosis Systems
31 Milan Jelisavcic (VU), Alive and Kicking: Baby Steps in Robotics
32 Chiara Sironi (UM), Monte-Carlo Tree Search for Artificial General Intelligence in Games
33 Anil Yaman (TUE), Evolution of Biologically Inspired Learning in Artificial Neural Networks
34 Negar Ahmadi (TUE), EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES
35 Lisa Facey-Shaw (OUN), Gamification with digital badges in learning programming

36 Kevin Ackermans (OUN), Designing Video-Enhanced Rubrics to Master Complex Skills

37 Jian Fang (TUD), Database Acceleration on FPGAs

38 Akos Kadar (OUN), Learning visually grounded and multilingual representations

2020 01 Armon Toubman (UL), Calculated Moves: Generating Air Combat Behaviour

02 Marcos de Paula Bueno (UL), Unraveling Temporal Processes using Probabilistic Graphical Models

03 Mostafa Deghani (UvA), Learning with Imperfect Supervision for Language Understanding

04 Maarten van Gompel (RUN), Context as Linguistic Bridges

05 Yulong Pei (TUE), On local and global structure mining

06 Preethu Rose Anish (UT), Stimulation Architectural Thinking during Requirements Elicitation - An Approach and Tool Support

07 Wim van der Vegt (OUN), Towards a software architecture for reusable game components

08 Ali Mirsoleimani (UL),Structured Parallel Programming for Monte Carlo Tree Search

09 Myriam Traub (UU), Measuring Tool Bias and Improving Data Quality for Digital Humanities Research

10 Alifah Syamsiyah (TUE), In-database Preprocessing for Process Mining

11 Sepideh Mesbah (TUD), Semantic-Enhanced Training Data Augmentation-Methods for Long-Tail Entity Recognition Models

12 Ward van Breda (VU), Predictive Modeling in E-Mental Health: Exploring Applicability in Personalised Depression Treatment

13 Marco Virgolin (CWI), Design and Application of Gene-pool Optimal Mixing Evolutionary Algorithms for Genetic Programming

14 Mark Raasveldt (CWI/UL), Integrating Analytics with Relational Databases

15 Konstantinos Georgiadis (OUN), Smart CAT: Machine Learning for Configurable Assessments in Serious Games

16 Ilona Wilmont (RUN), Cognitive Aspects of Conceptual Modelling

17 Daniele Di Mitri (OUN), The Multimodal Tutor: Adaptive Feedback from Multimodal Experiences

18 Georgios Methenitis (TUD), Agent Interactions & Mechanisms in Markets with Uncertainties: Electricity Markets in Renewable Energy Systems

19 Guido van Capelleveen (UT), Industrial Symbiosis Recommender Systems

20 Albert Hankel (VU), Embedding Green ICT Maturity in Organisations

21 Karine da Silva Miras de Araujo (VU), Where is the robot?: Life as it could be

22 Maryam Masoud Khamis (RUN), Understanding complex systems implementation through a modeling approach: the case of e-government in Zanzibar

23 Rianne Conijn (UT), The Keys to Writing: A writing analytics approach to studying writing processes using keystroke logging

24 Lenin da Nóbrega Medeiros (VUA/RUN), How are you feeling, human? Towards emotionally supportive chatbots

25 Xin Du (TUE), The Uncertainty in Exceptional Model Mining

26 Krzysztof Leszek Sadowski (UU), GAMBIT: Genetic Algorithm for Model-Based mixed-Integer opTimization

27 Ekaterina Muravyeva (TUD), Personal data and informed consent in an educational context

28 Bibeg Limbu (TUD), Multimodal interaction for deliberate practice: Training complex skills with augmented reality

29 Ioan Gabriel Bucur (RUN), Being Bayesian about Causal Inference

30 Bob Zadok Blok (UL), Creatief, Creatieve, Creatiefst

31 Gongjin Lan (VU), Learning better – From Baby to Better

32 Jason Rhuggenaath (TUE), Revenue management in online markets: pricing and online advertising

33 Rick Gilsing (TUE), Supporting service-dominant business model evaluation in the context of business model innovation

34 Anna Bon (MU), Intervention or Collaboration? Redesigning Information and Communication Technologies for Development

35 Siamak Farshidi (UU), Multi-Criteria Decision-Making in Software Production

---

2021 01 Francisco Xavier Dos Santos Fonseca (TUD),Location-based Games for Social Interaction in Public Space

02 Rijk Mercuur (TUD), Simulating Human Routines: Integrating Social Practice Theory in Agent-Based Models

03 Seyyed Hadi Hashemi (UVA), Modeling Users Interacting with Smart Devices

04 Ioana Jivet (OU), The Dashboard That Loved Me: Designing adaptive learning analytics for self-regulated learning

05 Davide Dell’Anna (UU), Data-Driven Supervision of Autonomous Systems

06 Daniel Davison (UT), "Hey robot, what do you think?" How children learn with a social robot

07 Armel Lefebvre (UU), Research data management for open science

08 Nardie Fanchamps (OU), The Influence of Sense-Reason-Act Programming on Computational Thinking

09 Cristina Zaga (UT), The Design of Robothings. Non-Anthropomorphic and Non-Verbal Robots to Promote Children’s Collaboration Through Play

10 Quinten Meertens (UvA), Misclassification Bias in Statistical Learning

11 Anne van Rossum (UL), Nonparametric Bayesian Methods in Robotic Vision

12 Lei Pi (UL), External Knowledge Absorption in Chinese SMEs

13 Bob R. Schadenberg (UT), Robots for Autistic Children: Understanding and Facilitating Predictability for Engagement in Learning

14 Negin Samaeemofrad (UL), Business Incubators: The Impact of Their Support

15 Onat Ege Adali (TU/e), Transformation of Value Propositions into Resource Re-Configurations through the Business Services Paradigm

16 Esam A. H. Ghaleb (UM), Bimodal emotion recognition from audio-visual cues

17 Dario Dotti (UM), Human Behavior Understanding from motion and bodily cues using deep neural networks

18 Remi Wieten (UU), Bridging the Gap Between Informal Sense-Making Tools and Formal Systems - Facilitating the Construction of Bayesian Networks and Argumentation Frameworks

19 Roberto Verdecchia (VU), Architectural Technical Debt: Identification and Management

20 Masoud Mansoury (TU/e), Understanding and Mitigating Multi-Sided Exposure Bias in Recommender Systems

21 Pedro Thiago Timbó Holanda (CWI), Progressive Indexes

22 Sihang Qiu (TUD), Conversational Crowdsourcing

23 Hugo Manuel Proença (LIACS), Robust rules for prediction and description

24 Kaijie Zhu (TUE), On Efficient Temporal Subgraph Query Processing

25 Eoin Martino Grua (VUA), The Future of E-Health is Mobile: Combining AI and Self-Adaptation to Create Adaptive E-Health Mobile Applications

26 Benno Kruit (CWI & VUA), Reading the Grid: Extending Knowledge Bases from Human-readable Tables

27 Jelte van Waterschoot (UT), Personalized and Personal Conversations: Designing Agents Who Want to Connect With You

28 Christoph Selig (UL), Understanding the Heterogeneity of Corporate Entrepreneurship Programs

2022 01 Judith van Stegeren (UT), Flavor text generation for role-playing video games

02 Paulo da Costa (TU/e), Data-driven Prognostics and Logistics Optimisation: A Deep Learning Journey

03 Ali el Hassouni (VUA), A Model A Day Keeps The Doctor Away: Reinforcement Learning For Personalized Healthcare

04 Ünal Aksu (UU), A Cross-Organizational Process Mining Framework

05 Shiwei Liu (TU/e), Sparse Neural Network Training with In-Time Over-Parameterization

06 Reza Refaei Afshar (TU/e), Machine Learning for Ad Publishers in Real Time Bidding

07 Sambit Praharaj (OU), Measuring the Unmeasurable? Towards Automatic Co-located Collaboration Analytics

08 Maikel L. van Eck (TU/e), Process Mining for Smart Product Design

09 Oana Andreea Inel (VUA), Understanding Events: A Diversity-driven Human-Machine Approach

10 Felipe Moraes Gomes (TUD), Examining the Effectiveness of Collaborative Search Engines

11 Mirjam de Haas (UT), Staying engaged in child-robot interaction, a quantitative approach to studying preschoolers' engagement with robots and tasks during second-language tutoring

12 Guanyi Chen (UU), Computational Generation of Chinese Noun Phrases

13 Xander Wilcke (VUA), Machine Learning on Multimodal Knowledge Graphs: Opportunities, Challenges, and Methods for Learning on Real-World Heterogeneous and Spatially-Oriented Knowledge

14 Michiel Overeem (UU), Evolution of Low-Code Platforms

15 Jelmer Jan Koorn (UU), Work in Process: Unearthing Meaning using Process Mining

16 Pieter Gijsbers (TU/e), Systems for AutoML Research

17 Laura van der Lubbe (VUA), Empowering vulnerable people with serious games and gamification

18 Paris Mavromoustakos Blom (TiU), Player Affect Modelling and Video Game Personalisation

19 Bilge Yigit Ozkan (UU), Cybersecurity Maturity Assessment and Standardisation

20 Fakhra Jabeen (VUA), Dark Side of the Digital Media - Computational Analysis of Negative Human Behaviors on Social Media

21 Seethu Mariyam Christopher (UM), Intelligent Toys for Physical and Cognitive Assessments

22 Alexandra Sierra Rativa (TiU), Virtual Character Design and its potential to foster Empathy, Immersion, and Collaboration Skills in Video Games and Virtual Reality Simulations

23 Ilir Kola (TUD), Enabling Social Situation Awareness in Support Agents

24 Samaneh Heidari (UU), Agents with Social Norms and Values - A framework for agent based social simulations with social norms and personal values

25 Anna L.D. Latour (LU), Optimal decision-making under constraints and uncertainty

26 Anne Dirkson (LU), Knowledge Discovery from Patient Forums: Gaining novel medical insights from patient experiences

27 Christos Athanasiadis (UM), Emotion-aware cross-modal domain adaptation in video sequences

28 Onuralp Ulusoy (UU), Privacy in Collaborative Systems

29 Jan Kolkmeier (UT), From Head Transform to Mind Transplant: Social Interactions in Mixed Reality

30 Dean De Leo (CWI), Analysis of Dynamic Graphs on Sparse Arrays

31 Konstantinos Traganos (TU/e), Tackling Complexity in Smart Manufacturing with Advanced Manufacturing Process Management

32 Cezara Pastrav (UU), Social simulation for socio-ecological systems

33 Brinn Hekkelman (CWI/TUD), Fair Mechanisms for Smart Grid Congestion Management

34 Nimat Ullah (VUA), Mind Your Behaviour: Computational Modelling of Emotion & Desire Regulation for Behaviour Change

35 Mike E.U. Ligthart (VUA), Shaping the Child-Robot Relationship: Interaction Design Patterns for a Sustainable Interaction

---

2023 01 Bojan Simoski (VUA), Untangling the Puzzle of Digital Health Interventions

02 Mariana Rachel Dias da Silva (TiU), Grounded or in flight? What our bodies can tell us about the whereabouts of our thoughts

03 Shabnam Najafian (TUD), User Modeling for Privacy-preserving Explanations in Group Recommendations

04 Gineke Wiggers (UL), The Relevance of Impact: bibliometric-enhanced legal information retrieval