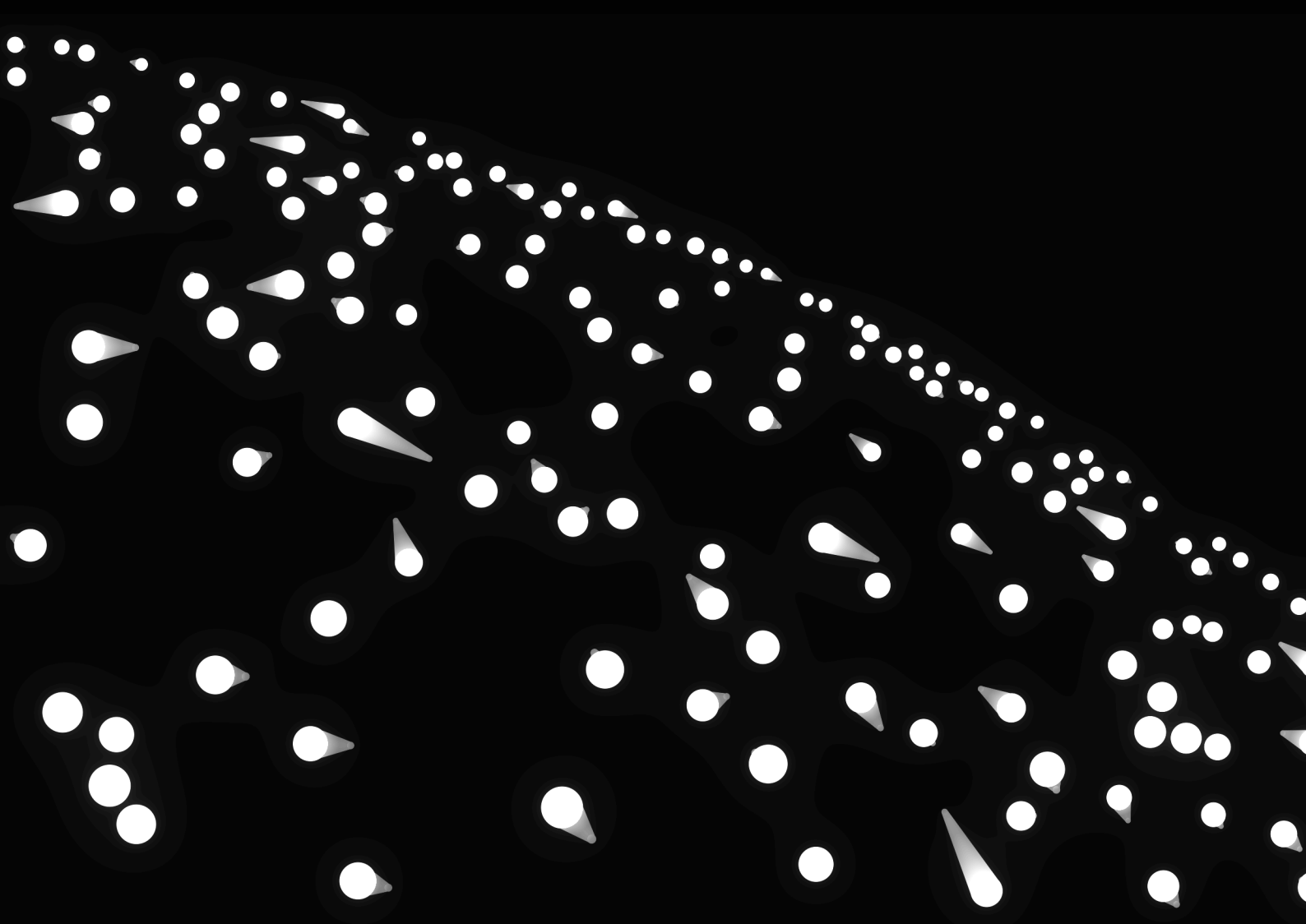


Evolutionary Brachytherapy Treatment Plan Optimisation using Low-Resolution Dose Calculations

Wouter van den Boer



Evolutionary Brachytherapy Treatment Plan Optimisation using Low-Resolution Dose Calculations

by

Wouter van den Boer

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday October 5, 2026 at 10:00.

Student number: 4963911

Project duration: January 2026 – October 2026

Daily supervisor: Dr. Anton Bouter CWI
Thesis committee: Prof. dr. Peter A.N. Bosman CWI, TU Delft, chair and supervisor
Prof. dr. Tanja Alderliesten LUMC, TU Delft, supervisor
Dr. Jasmijn A. Baaijens TU Delft

An electronic version of this thesis is available at <https://repository.tudelft.nl>.

Preface

This thesis marks the conclusion of my Master's degree in Computer Science at Delft University of Technology. I am deeply grateful for all the experiences and lessons of the last two years. This thesis gave me the opportunity to conduct research in the fascinating world of evolutionary algorithms. Along the way I learned how rigorous good research needs to be, especially before it can be trusted to influence how patients are treated. I will keep drawing upon these experiences long after completing this thesis.

I want to thank Prof. dr. Peter A.N. Bosman, Prof. dr. Tanja Alderliesten and Dr. Anton Bouter, along with the other members of the Evolutionary Intelligence group at CWI, for many good discussions and for their invaluable guidance that shaped this work.

Lastly, I would like to thank my friends and family for their encouragement and support throughout this journey. I couldn't have done it without you.

Abstract

Radiotherapy treats cancer by irradiating tumour cells so that they stop dividing and die off. High-dose-rate (HDR) brachytherapy is a form of radiotherapy that delivers radiation from inside the body, allowing a high dose in the tumour while sparing surrounding tissue. The effectiveness of HDR brachytherapy depends on how radiation is distributed in a patient's body which is determined by a treatment plan. HDR brachytherapy treatment planning for cervical cancer can be automated using evolutionary algorithms (EAs), as is done by the BRIGHT treatment planning toolkit, which uses the EA MO-RV-GOMEA. Treatment plans found by BRIGHT, after several minutes of optimisation using a graphics processing unit (GPU), have been shown to match or outperform current clinical practice. Even faster treatment planning could lower hardware requirements, shorten the time a patient spends with the applicator in place, or yield better treatment plans in the same optimisation time. Treatment planning using BRIGHT requires many evaluations of candidate plans, each computing the dose at many points, which dominate optimisation time. Using a lower dose calculation resolution can enable faster treatment planning, yet allows the EA to overfit to the resulting imprecise approximation, which risks producing worse treatment plans. This thesis studies whether and how treatment planning using BRIGHT can be made faster, without loss of plan quality, by using a lower dose calculation resolution.

MO-RV-GOMEA is first studied in benchmark problems with artificial uncertainty. Later, two modifications to BRIGHT for reducing overfitting are proposed. Using a lower dose calculation resolution is shown to yield better plans than the baseline during the first 10 seconds of optimisation, but never improves on the final plan quality in terms of sparing, coverage and shape metrics. Independent of the dose calculation resolution, rounding errors stemming from partial evaluations are shown to negatively impact optimisation more than previously anticipated, and using full evaluations is shown to be at least as fast as partial evaluations in the GPU-parallelised version of BRIGHT.

Disclosure Use of Generative LLMs

Generative large language models (LLMs) were used during the work reported in this thesis to search literature, automate workflows and write software. Additionally, LLMs were used to support the writing of this thesis by rephrasing sentences and scanning for language mistakes.

All outputs produced by LLMs were verified by the author. All insights, research ideas, interpretations and conclusions are those of the author, arrived at together with the supervisors of this thesis.

Contents

Preface	iii
Abstract	iv
Disclosure Use of Generative LLMs	v
List of Abbreviations	viii
1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	1
1.3 Research Questions	2
1.4 Thesis Outline	2
2 Background	3
2.1 Optimisation	3
2.1.1 Multi-Objective Optimisation	4
2.2 Evolutionary Algorithms	6
2.2.1 AMaLGaM	7
2.2.2 GOMEA	8
2.2.2.1 RV-GOMEA	9
2.2.2.2 MO-RV-GOMEA	10
2.3 Uncertainty in Evolutionary Optimisation	12
2.3.1 Noisy Objective Functions	12
2.3.2 Approximate Objective Functions	12
2.4 Brachytherapy Treatment Planning	13
2.4.1 Treatment Plan Quality	13
2.4.2 Dose Calculation	15
2.4.3 BRIGHT	16
2.4.4 Uncertainty in BRIGHT	17
3 Uncertainty in Benchmark Problems	19
3.1 Noisy Fitness Functions	19
3.1.1 Convergence under Noise	19
3.1.2 Enforcing a Minimum Sampling Variance	21
3.1.3 Multiple Objectives	22
3.1.4 Conclusion	24
3.2 Low-Fidelity Fitness Evaluations	24
3.2.1 Convergence When Using a Low-Fidelity Fitness Function	26
3.2.2 Conclusion	28
3.3 Conclusion	28
4 Reduced Dose Calculation Resolution in BRIGHT	29
4.1 Experimental Setup	29
4.2 Overfitting	29

4.3	Reduced Dose Calculation Resolution	34
4.4	Conclusion	37
5	Mitigating Overfitting	38
5.1	Rotating Multiple Sets of Dose Calculation Points	38
5.2	Different Dose Calculation Resolution Per Region	42
5.3	Comparing Dose Distributions	44
5.4	Conclusion	48
6	Discussion	49
7	Conclusion	51
7.1	Future Work	52
	References	54
	Appendix	57
A.1	Normalisation of Dose-Volume Indices Stated in EQD2	57
A.2	Approximation Fronts over the Full Time-Limit Sweep	59

List of Abbreviations

AMaLGaM	Adapted Maximum-Likelihood Gaussian Model iterated density-estimation evolutionary algorithm
AMS	Anticipated Mean Shift
AVS	Adaptive Variance Scaling
BBO	Black-Box Optimisation
CTV	Clinical Target Volume
CTV _{HR}	High-Risk Clinical Target Volume
CTV _{IR}	Intermediate-Risk Clinical Target Volume
DCP	Dose Calculation Point
DVI	Dose-Volume Index
EA	Evolutionary Algorithm
EBRT	External Beam Radiotherapy
EDA	Estimation of Distribution Algorithm
EQD2	Equivalent Dose in 2 Gy fractions
FOS	Family Of Subsets
GBO	Grey-Box Optimisation
GOM	Gene-pool Optimal Mixing
GOMEA	Gene-pool Optimal Mixing Evolutionary Algorithm
GTV	Gross Tumour Volume
HDR	High-Dose-Rate
HV	Hypervolume
LCI	Least Coverage Index
LSI	Least Sparing Index
MO	Multi-Objective
NIS	No Improvement Stretch
OAR	Organ At Risk
ROI	Region Of Interest
RV	Real-Valued
SDR	Standard Deviation Ratio
SO	Single-Objective
WBO	White-Box Optimisation
ZDT	Zitzler-Deb-Thiele benchmark problems

Introduction

1.1 Motivation

Optimisation problems arise throughout science, industry, and everyday life [1], [2], [3]. Modern computers often solve them effectively, yet the choice of algorithm remains important, as no single algorithm performs best on every problem [4], and algorithmic improvements can provide large speed-ups [5], [6]. Evolutionary algorithms (EAs) are a class of metaheuristics originally inspired by natural evolution [7]. They maintain a population of candidate solutions and iteratively improve it using techniques such as selection and variation. The quality of a candidate solution, its *fitness*, is determined by evaluating the objective function of the optimisation problem. EAs are well suited to complex real-world optimisation problems, but such problems commonly come with computationally expensive objective functions. When each evaluation requires running a physical simulation or a lengthy computation, the cost of optimisation is dominated by fitness evaluations. One strategy to reduce this cost is to use an approximation of the true function that is faster to evaluate. Some solutions may appear promising on the approximation function, but are bad in reality. An optimiser which focusses on such seemingly good solutions spends time and computational resources without making real progress. This phenomenon is called overfitting. The use of low-resolution evaluations can effectively speed up optimisation only if overfitting does not occur.

High-dose-rate (HDR) brachytherapy is a form of cancer treatment in which a radioactive source is temporarily placed close to or inside a tumour [8]. The source is guided through an applicator and can stop at a number of fixed positions within it, called *dwell positions*. Choosing how much radiation to deliver from each dwell position is a difficult problem with multiple objectives. A tradeoff needs to be found between damaging the tumour and sparing neighbouring healthy tissue. The BRIGHT treatment planning toolkit [9], [10] optimises brachytherapy treatment plans using a multi-objective (MO) EA called MO-RV-GOMEA [11]. BRIGHT evaluates candidate plans using a simulation that computes the planned radiation dose at many dose calculation points (DCPs) in 3D space. These evaluations are computationally expensive and dominate the optimisation time. Faster optimisation could lower hardware requirements, shorten the time a patient spends with the applicator in place, or yield better treatment plans in the same optimisation time. Reducing the number of DCPs provides a natural approximation of the true objective function: evaluating the dose at fewer points is cheaper, but less accurate. This thesis investigates how a reduced dose calculation resolution can be used to speed up brachytherapy treatment planning using BRIGHT for cervical cancer.

1.2 Problem Statement

Evaluating a candidate HDR brachytherapy treatment plan for cervical cancer in BRIGHT is computationally expensive. These evaluations account for a large part of the total treatment plan optimisation runtime. When evaluating a plan, calculating the dose at fewer points yields a cheaper low-resolution approximation of the true objective function. MO-RV-GOMEA might overfit to the low-resolution approximation, and converge to plans that appear strong under the approximation, but are inferior under the true objective function. Bouter et al. [12] showed that overfitting indeed occurs and impairs final plan quality when using few DCPs for prostate HDR brachytherapy treatment plan optimisation. The central challenge is to make fitness evaluations cheaper by using fewer DCPs while avoiding

overfitting, in order to obtain the same quality of solutions faster, or better solutions in the same timespan.

1.3 Research Questions

This thesis addresses research question **RQ**, which follows from the problem statement, and is addressed through four sub-questions. **SRQ1** studies the behaviour of MO-RV-GOMEA on benchmark problems in a setting where solution evaluations are stochastic and thus noisy, which models the use of a different set of DCPs for each evaluation. **SRQ2** on the other hand studies the behaviour of MO-RV-GOMEA on benchmark problems in a setting where a random perturbation of the true fitness function is never or infrequently redrawn, which models the re-use of DCPs. **SRQ3** then asks what happens in BRIGHT when the number of DCPs used for evaluating treatment plan quality is reduced. Finally, **SRQ4** asks what strategies can be used to counteract the resulting overfitting in BRIGHT.

RQ How can the use of low-resolution dose calculations speed up the optimisation of brachytherapy treatment planning using BRIGHT without loss of plan quality?

SRQ1 How does the use of noisy fitness functions affect MO-RV-GOMEA’s convergence, computational cost and final solution quality?

SRQ2 How does the number of distinct low-fidelity fitness functions used during a run affect MO-RV-GOMEA’s convergence, computational cost and final solution quality?

SRQ3 How does reducing the number of dose-calculation points affect BRIGHT’s convergence, computation time and final plan quality?

SRQ4 What strategies can be used to speed up optimisation using low-resolution dose calculations in BRIGHT, without loss of plan quality?

1.4 Thesis Outline

Chapter 2 introduces the concepts which this thesis builds on: optimisation, EAs, uncertainty in evolutionary optimisation, and brachytherapy treatment planning. Chapter 3 addresses **SRQ1** and **SRQ2** by adding uncertainty to the objective functions of benchmark problems. Chapter 4 studies what happens to optimisation in BRIGHT when a lower dose calculation resolution is used, and by doing so addresses **SRQ3**. Chapter 5 addresses **SRQ4** by evaluating strategies designed to reduce the overfitting observed in Chapter 4. Chapter 6 reflects on the results and their limitations, and Chapter 7 answers the main research question **RQ** and outlines directions for future work.

Background

2.1 Optimisation

A real-valued single-objective (SO) optimisation problem can be defined by two functions; the *objective function* f and the *constraint function* g :

$$f : \mathbb{R}^l \rightarrow \mathbb{R}, \quad (2.1)$$

$$g : \mathbb{R}^l \rightarrow \mathbb{R}. \quad (2.2)$$

A solution $\mathbf{x} \in \mathbb{R}^l$ for an optimisation problem is composed of l *decision variables* $[x_0, \dots, x_{l-1}]$. A solution $\mathbf{x}$ is feasible if $g(\mathbf{x}) \leq 0$. Unless stated otherwise all optimisation problems in this work are minimisation problems and thus solutions with lower objective values are more desirable. A solution $\mathbf{x}$ is optimal if there exists no other feasible solution $\mathbf{x}'$ such that $f(\mathbf{x}') < f(\mathbf{x})$. The goal of an optimiser is to find a (near) optimal solution.

In a Black-Box Optimisation (BBO) setting, an optimiser has only the ability to evaluate $f(\mathbf{x})$ and $g(\mathbf{x})$, and has no additional knowledge about $f(\mathbf{x})$ or $g(\mathbf{x})$. In contrast, a White-Box Optimisation (WBO) setting is one where the optimiser has access to the full analytical form of $f(\mathbf{x})$ and $g(\mathbf{x})$, and can fully exploit this information. Such information can be used by an optimiser to find good solutions faster. Grey-Box Optimisation (GBO) defines a middle ground, where only some additional knowledge about the optimisation problem can be exploited by the optimiser.

Dependence between variables and decomposability of the objective function are examples of knowledge that can be valuable to an optimiser. Two decision variables are *dependent* when the effect of one on the objective function depends on the value of the other. For example, in the objective function $f(\mathbf{x}) = (x_0 + x_1)^2$, x_0 and x_1 are dependent, because $f(\mathbf{x})$ is minimal only if $x_0 = -x_1$. An optimiser aware of such dependencies can vary the dependent variables together rather than independently such that it is more likely to find good solutions sooner. An objective function is *decomposable* if $f(\mathbf{x}) = \sum_j f_j^{(\text{subf})}(\mathbf{x}_{S_j})$, where each subfunction $f_j^{(\text{subf})} : \mathbb{R}^{|S_j|} \rightarrow \mathbb{R}$ acts only on the variables indexed by a subset $S_j \subseteq \{0, \dots, l-1\}$. If the objective value of a solution is already known, then changing the variables indexed by a subset $T \subseteq \{0, \dots, l-1\}$, requires only every $f_j^{(\text{subf})}$ where $S_j \cap T \neq \emptyset$ to be recomputed in order to find the new objective value. Such a *partial evaluation* can obtain the new objective value at a smaller computational cost than the cost of evaluating the whole objective function. Partial evaluations can make great contributions to the efficiency of optimisation algorithms that vary few decision variables at a time.

There exists a plethora of optimisation methods. These methods fall into two broad categories: exact and heuristic. Exact optimisation methods, such as dynamic programming [13] and linear programming [14], guarantee an optimal solution. These exact methods typically require full knowledge of the problem structure, which aligns with WBO. Heuristic methods, such as simulated annealing [15] and EAs [7], do not guarantee an optimal solution, but often scale better to large and complex problems, and work well in BBO and GBO settings.

2.1.1 Multi-Objective Optimisation

In real-world optimisation problems, the goal may be to optimise multiple conflicting objectives. Where SO maps a solution to a scalar in the objective space $\mathbb{R}$, a multi-objective (MO) problem maps it to a point in the k -dimensional objective space $\mathbb{R}^k$ via $\mathbf{f}$:

$$\mathbf{f} : \mathbb{R}^l \rightarrow \mathbb{R}^k, \quad (2.3)$$

where

$$f_i : \mathbb{R}^l \rightarrow \mathbb{R}, \quad (2.4)$$

$$\mathbf{f}(\mathbf{x}) = [f_0(\mathbf{x}), f_1(\mathbf{x}), \dots, f_{k-1}(\mathbf{x})]. \quad (2.5)$$

Selecting the best solution in an MO space is not straightforward, as the preference over the different objectives often is subjective. We illustrate this subjectivity with an example. Imagine we are choosing a computer to purchase and aim to minimise cost and maximise performance. We cannot, context-independently, determine whether a computer with performance 100 at cost 20 is preferable to one with performance 50 at cost 10. The two computers are considered *incomparable*. In contrast, in an SO setting objective values are a scalar in $\mathbb{R}$ and we always know whether a solution is better, worse or precisely as good as some other solution (assuming no uncertainty in solution evaluations). More formally, $\mathbb{R}$ induces a natural *total order*:

$$\forall_{a,b \in \mathbb{R}} [a < b \vee a = b \vee a > b]. \quad (2.6)$$

The objective space $\mathbb{R}^k$ clearly induces no natural total order, illustrated by the incomparability of the two computers in our example. Without a natural total order there is (in general) no single best solution. One way to define a total order in an MO setting is to collapse the objectives into a single scalar, encoding a preference over them as weights:

$$f_{\text{scalar}}(\mathbf{x}) = w_0 f_0(\mathbf{x}) + w_1 f_1(\mathbf{x}) + \dots + w_{k-1} f_{k-1}(\mathbf{x}). \quad (2.7)$$

Figure 2.1(a) and Figure 2.1(b) illustrate how, for a set of candidate solutions, a weight vector picks out a single “best” solution. Optimising using such a scalarisation of the objectives with an SO optimiser returns only a single solution. A surrounding set of solutions, which would reveal how much one objective value must worsen to improve another, is never produced. The weights for the scalarisation must be chosen before optimisation, but this is difficult without knowing the tradeoff structure [16].

Although $\mathbb{R}^k$ induces no natural total order, it does induce a natural *partial order* referred to as *Pareto dominance*. A solution $\mathbf{x}$ (Pareto) dominates a solution $\mathbf{x}'$, written $\mathbf{x} \succ \mathbf{x}'$, when it is at least as good in every objective and better in at least one:

$$\mathbf{x} \succ \mathbf{x}' \iff \forall_{i \in \{0, \dots, k-1\}} [f_i(\mathbf{x}) \leq f_i(\mathbf{x}')] \wedge \exists_{j \in \{0, \dots, k-1\}} [f_j(\mathbf{x}) < f_j(\mathbf{x}')]. \quad (2.8)$$

Two solutions for which neither $\mathbf{x} \succ \mathbf{x}'$ nor $\mathbf{x}' \succ \mathbf{x}$ holds are incomparable. Solutions which are not dominated by any other solution are referred to as non-dominated solutions. The *Pareto front* consists of all such non-dominated solutions. Inspecting this front directly reveals the full tradeoff structure, allowing a domain expert to select a solution with a suitable tradeoff. A Pareto front is illustrated in Figure 2.1(c). Pareto fronts are only practical for a limited number of objectives, because they are hard to visualise in more than 3 dimensions. An optimiser generally does not find the exact Pareto front, but only finds a set of mutually non-dominated solutions that approximate it, which is called the *approximation set*.

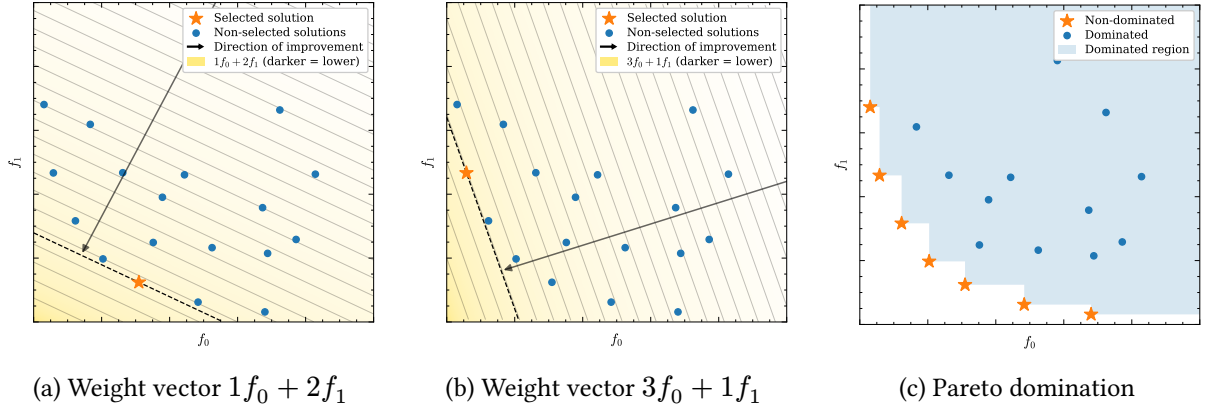


Figure 2.1: Illustrations of two different strategies for selecting solutions in an MO setting. Both objectives are minimised. Figure 2.1(a) and Figure 2.1(b) show the weighted sum scalarisation $w_0f_0 + w_1f_1$ with different weight vectors, which reduces the problem to a single objective and selects a single solution. Figure 2.1(c) marks all non-dominated solutions, which form the Pareto front, as stars.

In SO optimisation, measuring the performance of an optimiser is simple: it suffices to report the best objective value found. In an MO setting, measuring performance becomes more difficult, as we are not interested in a single value, but in the tradeoffs that the approximation set presents. The quality of an approximation set can be quantified with a *performance indicator*, which is a function that maps an approximation set to a scalar reflecting its quality. Many performance indicators can be conceived [17]. Broadly, performance indicators aim to measure two things: proximity to the Pareto front and diversity of tradeoffs. In this work we use two generally applicable performance indicators, *hypervolume* (HV) and the *distance from the Pareto front to the approximation set* ($D_{P_f \rightarrow S}$), and performance indicators specific to brachytherapy. We introduce HV and $D_{P_f \rightarrow S}$ in this section. The brachytherapy-specific performance indicators are introduced in Subsection 2.4.3.

The HV of an approximation set S is the volume of objective space that is dominated by S , bounded by a reference point $\mathbf{r}$, as illustrated in Figure 2.2(a). A larger HV indicates a better approximation set. HV captures both proximity and diversity: an approximation set closer to the Pareto front dominates a larger volume, while spreading its solutions across the front adds volume that clustered solutions would leave uncovered. Computing the HV requires no knowledge of the true Pareto front, only a reference point $\mathbf{r}$ needs to be chosen, which makes it applicable to real-world problems whose Pareto fronts are unknown, such as our brachytherapy problem. The reference point $\mathbf{r}$ must however be chosen carefully. If $\mathbf{r}$ is placed too close, the outer solutions of the approximation set fall outside the region bounded by $\mathbf{r}$ and do not contribute to the HV. If $\mathbf{r}$ is placed too far, the HV is dominated by a large fixed volume, and a strong bias is placed on extreme solutions on the outer edges of the front.

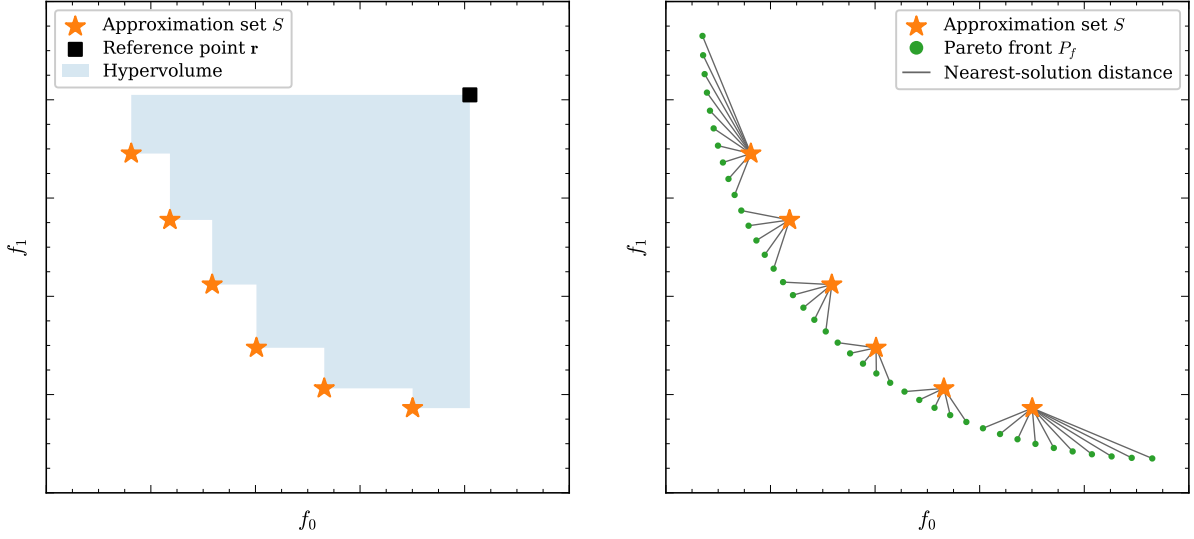
The distance from the Pareto front to the approximation set, $D_{P_f \rightarrow S}$, is the average distance from the Pareto front to the nearest solution in the approximation set, as illustrated in Figure 2.2(b). Given a reference set P_f of Pareto-optimal solutions, it is defined as

$$D_{P_f \rightarrow S}(S) = \frac{1}{|P_f|} \sum_{\mathbf{x}' \in P_f} \min_{\mathbf{x} \in S} d(\mathbf{x}', \mathbf{x}), \quad (2.9)$$

where

$$d(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_{i=0}^{k-1} (f_i(\mathbf{x}) - f_i(\mathbf{x}'))^2} \quad (2.10)$$

is the distance between two solutions in objective space. $D_{P_f \rightarrow S}$ captures both proximity and diversity: any region of the front that the approximation set fails to cover contributes a large distance. A smaller $D_{P_f \rightarrow S}$ indicates a better approximation set. Unlike HV, computing $D_{P_f \rightarrow S}$ requires a known Pareto front, which restricts its use to benchmark problems.



(a) HV: the volume of objective space dominated by S and bounded by the reference point $\mathbf{r}$.

(b) $D_{P_f \rightarrow S}$: each point on the Pareto front P_f contributes the distance to its nearest solution in S .

Figure 2.2: illustrations of the two generally applicable performance indicators which are used in this work to quantify the quality of an approximation set S in an MO setting.

2.2 Evolutionary Algorithms

EAs are a class of metaheuristics originally inspired by natural evolution [7]. In an EA, a *population* of candidate solutions for an optimisation problem is improved iteratively, over multiple *generations*. The candidate solutions are referred to as *individuals*. *Variation* is applied to the individuals in order to create *offspring*. The offspring with the best objective values, or *fitness*, become part of the next generation; this is referred to as selection. Under *elitism*, a number of the best individuals of the current generation are carried over as well, guaranteeing that the best solutions found so far are not lost. A good EA balances evaluating solutions in unseen regions of the search space (*exploration*), and evaluating solutions in already seen, promising, regions of the search space (*exploitation*) [18]. Too much exploitation risks early convergence to local optima, while too much exploration risks wasting evaluations and taking longer than needed to find good solutions. Algorithm 2.1 provides pseudocode of a general EA.

Algorithm 2.1: Pseudocode for a general EA

```

1:  $P \leftarrow \text{randomInitialPopulation}()$ 
2: while  $\neg \text{terminationCriterionMet}(P)$  do
3:    $O \leftarrow \text{createAndEvaluateOffspring}(P)$ 
4:    $P \leftarrow \text{selectSurvivingIndividuals}(P, O)$ 
5: end
6: return  $\text{best}(P)$ 

```

Traditional variation operators, such as crossover, which recombines parts of two or more parent solutions, and mutation, which randomly perturbs a single solution, act directly on the individuals themselves. Estimation of distribution algorithms (EDAs) instead replace these operators with sampling new offspring from a probability distribution, which is estimated from previous generation(s). In some sense, EDAs use a special stochastic variation operator that uses all selected individuals as parents. The strength of EDAs is that they provide a principled way of exploiting dependencies between decision variables [19]. The remainder of this section introduces the EAs used in this thesis.

2.2.1 AMaLGaM

The *Adapted Maximum-Likelihood Gaussian Model iterated density-estimation evolutionary algorithm* (AMaLGaM) is a real-valued EDA [19]. Each generation, the mean and covariance matrix of a Gaussian distribution are estimated from the τn best individuals, where n is the population size and τ is the selection ratio. The covariance matrix can be estimated in full or restricted to a *univariate* factorisation. A univariate factorisation does not model dependencies, or covariances, between decision variables, whereas a full covariance matrix captures all pairwise dependencies. The offspring are sampled from the resulting distribution.

AMaLGaM extends the basic EDA scheme with two mechanisms. The *Anticipated Mean Shift* (AMS) shifts a fraction of the offspring along the direction in which the mean moved in the previous generation, adding momentum that helps the population traverse (long) slopes of the search space. While improvements are found far from the Gaussian's mean *Adaptive Variance Scaling* (AVS) increases the estimated variance. The variance is shrunk otherwise. The *standard deviation ratio* (SDR) is the difference between the mean of the Gaussian and the solutions that are an improvement, referred to as $x_{\text{avg-imp}}$. θ_{SDR} is a threshold. If $\text{SDR} > \theta_{\text{SDR}}$ and the current population improved upon the previous population then the variance of the Gaussian is increased. Pseudocode of AMaLGaM is provided in Algorithm 2.2. NIS represents the *no improvement stretch* which counts the number of consecutive generations in which no improvements have been found. α_{AMS} is the fraction of new solutions to apply AMS to. δ_{AMS} is the magnitude of the mean shift. η_{INC} and η_{DEC} are the factors by which AVS increases and decreases the distribution multiplier. Note that in AMaLGaM the best, or elitist, solution is carried over to the next generation.

Algorithm 2.2: Pseudocode for AMaLGaM, reproduced from [19]

```

1:  $c_{\text{Multiplier}} \leftarrow 1$ 
2:  $n_{\text{AMS}} \leftarrow \alpha_{\text{AMS}}(n - 1)$ 
3:  $\text{NIS} \leftarrow 0$ 
4:  $t \leftarrow 0$ 
5: do
6:    $S \leftarrow$  the best  $\tau n$  solutions in  $P$  (truncation selection)
7:   estimateDistributionParameters()
8:    $P_0 \leftarrow$  the best solution in  $S$ 
9:   for  $i \leftarrow 1 \dots n - 1$  do
10:     $P_i \leftarrow \text{drawRandomSample}()$ 
11:   end
12:   for  $n_{\text{AMS}}$  randomly chosen solutions  $P_i, 1 \leq i \leq n - 1$  do
13:     $P_i \leftarrow P_i + c_{\text{Multiplier}} \delta_{\text{AMS}} \hat{\mu}_{\text{Shift}}(t)$ 
14:   end
15:   if any  $P_i$  better than  $P_0, 1 \leq i \leq n - 1$  then
16:      $\text{NIS} \leftarrow 0$ 
17:     if  $c_{\text{Multiplier}} < 1$  then
18:        $c_{\text{Multiplier}} \leftarrow 1$ 
19:     end
20:      $x_{\text{avg-imp}} \leftarrow$  average of all  $P_i$  better than  $P_0, 1 \leq i \leq n - 1$ 
21:     computeSDR()
22:     if  $\text{SDR} > \theta_{\text{SDR}}$  then
23:        $c_{\text{Multiplier}} \leftarrow \eta_{\text{INC}} c_{\text{Multiplier}}$ 
24:     end
25:   else
26:     if  $c_{\text{Multiplier}} \leq 1$  then
27:        $\text{NIS} \leftarrow \text{NIS} + 1$ 
28:     end
29:     if  $c_{\text{Multiplier}} > 1$  or  $\text{NIS} \geq \text{NIS}_{\text{MAX}}$  then
30:        $c_{\text{Multiplier}} \leftarrow \eta_{\text{DEC}} c_{\text{Multiplier}}$ 
31:     end
32:     if  $c_{\text{Multiplier}} < 1$  and  $\text{NIS} < \text{NIS}_{\text{MAX}}$  then
33:        $c_{\text{Multiplier}} \leftarrow 1$ 
34:     end
35:   end
36:    $t \leftarrow t + 1$ 
37: while maximum number of evaluations not reached

```

2.2.2 GOMEA

The *Gene-pool Optimal Mixing Evolutionary Algorithm* (GOMEA) is the first in a family of EAs that share two things: an explicit *linkage model* and the use of *Gene-pool Optimal Mixing* (GOM) for variation and selection [20]. The linkage model is represented by a *Family Of Subsets* (FOS) $\mathcal{F} = \{F_0, \dots, F_{|\mathcal{F}|-1}\}$, where each $F_j \subseteq \{0, \dots, l - 1\}$ is a set of indices of decision variables that are thought to be dependent and to be varied jointly. The *univariate* FOS $\mathcal{F} = \{\{0\}, \{1\}, \dots, \{l - 1\}\}$ treats all variables as independent, and is the linkage model which an EDA with a univariate factorisation implicitly uses. A *marginal product* FOS has pairwise disjoint elements, and fits problems that decompose into independent blocks of variables. A *linkage tree* (LT) FOS is a hierarchical FOS: it starts from the l single decision variables and repeatedly merges the two most similar elements until an element containing all variables is created. When problem-specific knowledge about the dependence structure is available, as in GBO, the FOS can be fixed a priori; otherwise it is re-determined each generation. The GOM function performs variation of the variables of a single FOS element of one individual. A change

is only kept if the fitness of the individual was improved. GOM combines variation and selection into one operator, because changes which do not improve an individual's fitness are immediately rejected. In doing so, elitism is immediately achieved, as an individual cannot worsen. GOM also is a natural fit for partial evaluations, as most elements in the FOS are merely small subsets of all decision variables, allowing for the use of the faster partial evaluations when performing the GOM procedure.

2.2.2.1 RV-GOMEA

RV-GOMEA is the real-valued variant of GOMEA [21]. Here every FOS element F_j has a *sampling model* M_j , from which new partial solutions are drawn. Each sampling model is a Gaussian distribution with AMS, AVS and SDR borrowed from AMaLGaM. Algorithm 2.3 shows the GOM procedure, which takes as input an individual x , a FOS element F_j and the sampling model M_j . In RV-GOMEA, changes which do not improve the fitness can still be accepted with a small probability p_{accept} , which helps the population escape local optima. Solutions which have not improved for NIS_{MAX} generations undergo the *forced improvement* procedure, in which the variables of a FOS element are replaced by a linear combination of their current values and those of the elitist solution, until an improvement is found. Algorithm 2.4 gives the overall structure of RV-GOMEA.

Algorithm 2.3: GOM(x, F_j, M_j) Gene-pool Optimal Mixing in RV-GOMEA, adapted from [22]

```

1: for  $i \in \{0, \dots, |F_j| - 1\}$  do
2:    $\text{backup}[i] \leftarrow x[F_j[i]]$ 
3: end
4:  $\text{sample} \leftarrow \text{samplePartialSolution}(M_j)$ 
5: for  $i \in \{0, \dots, |F_j| - 1\}$  do
6:    $x[F_j[i]] \leftarrow \text{sample}[i]$ 
7: end
8:  $\text{fitness}_{\text{new}} \leftarrow \text{partialEvaluation}(x, \text{fitness}_x, F_j)$ 
9: if  $\text{fitness}_{\text{new}} < \text{fitness}_x$  or  $U(0, 1) < p_{\text{accept}}$  then
10:   $\text{fitness}_x \leftarrow \text{fitness}_{\text{new}}$ 
11: else
12:   for  $i \in \{0, \dots, |F_j| - 1\}$  do
13:     $x[F_j[i]] \leftarrow \text{backup}[i]$ 
14:   end
15: end

```

Algorithm 2.4: Pseudocode for RV-GOMEA, reproduced from [22]

```

1:  $P \leftarrow \text{initialiseAndEvaluatePopulation}(n)$ 
2:  $\mathcal{F} \leftarrow \text{initialiseLinkageModel}()$ 
3: for  $j \in \{0, \dots, |\mathcal{F}| - 1\}$  do
4:    $M_j \leftarrow \text{initialiseSamplingModel}(F_j)$ 
5: end
6: while  $\neg \text{terminationCriterionSatisfied}()$  do
7:    $P_0 \leftarrow \mathbf{x}_{\text{elitist}}$ 
8:   for  $j \in \{0, \dots, |\mathcal{F}| - 1\}$ , in random order do
9:     for  $x \in P[1 \dots n - 1]$  do
10:       $\text{GOM}(x, F_j, M_j)$  ▷ Algorithm 2.3
11:    end
12:     $\text{intermediateUpdate}(M_j)$ 
13:  end
14:  for  $x \in P[1 \dots n_{\text{AMS}}]$  do
15:     $\text{anticipatedMeanShift}(x)$ 
16:  end
17:  for  $x \in P[1 \dots n - 1]$  do
18:    if  $\text{NIS}(x) > \text{NIS}_{\text{MAX}}$  then
19:       $\text{forcedImprovement}(x)$ 
20:    end
21:  end
22:  for  $j \in \{0, \dots, |\mathcal{F}| - 1\}$  do
23:     $\text{updateSamplingModel}(M_j)$ 
24:  end
25: end

```

2.2.2.2 MO-RV-GOMEA

MO-RV-GOMEA is the MO version of RV-GOMEA [11], and is the optimiser used by BRIGHT. A single Gaussian per FOS element cannot always cover an entire Pareto set, because the Pareto set is often curved in the search space. *MO-RV-GOMEA* therefore *clusters* the population in objective space, and maintains a separate sampling model M_{ij} for every FOS element F_j in every cluster C_i . The selection is clustered into q possibly overlapping clusters using Euclidean distances in objective space. One SO cluster is created per objective, which contains the best solutions in that objective. The remaining clusters are grown around cluster leaders which are chosen to lie far apart. Every solution in the population is subsequently assigned to exactly one cluster, so that it can be sampled from the models of that cluster.

The mutually non-dominated solutions are kept in the *elitist archive*. Acceptance inside GOM changes now that we are in an MO setting: a modification is kept if the modified solution dominates its previous state or if it is not dominated by any solution in the elitist archive. Algorithm 2.5 gives the overall structure of MO-RV-GOMEA, Algorithm 2.6 presents the MO version of GOM and Algorithm 2.7 shows how the elitist archive is updated.

A candidate for the elitist archive is rejected outright if any archive member dominates it, and every archive member which the candidate dominates is discarded when the candidate is inserted. A target size for the archive is defined. If the archive size exceeds 125% of the target size, the archive is pruned in a way such that diversity is preserved. After clustering, at most a fraction τ of each cluster is replaced by archive members, so elitist solutions re-enter the population.

Algorithm 2.5: Pseudocode for parallel MO-RV-GOMEA, adapted from [22] and [11]

```

1:  $P \leftarrow \text{initialiseAndEvaluatePopulation}(n)$ 
2:  $\mathcal{F} \leftarrow \text{initialiseLinkageModel}()$ 
3:  $A \leftarrow \emptyset$  ▷ start with empty elitist archive
4: while  $\neg \text{terminationCriterionSatisfied}()$  do
5:    $S \leftarrow \text{selection}(P)$ 
6:    $C \leftarrow \text{clusterPopulationAndSelection}(P, S, q)$ 
7:    $\text{copyElitesToPopulation}(C, A)$ 
8:   for  $C_i \in C, j \in \{0, \dots, |\mathcal{F}| - 1\}$  do
9:      $M_{ij} \leftarrow \text{estimateSamplingModel}(C_i, F_j)$ 
10:  end
11:  for  $j \in \{0, \dots, |\mathcal{F}| - 1\}$ , in random order do ▷ iterate over FOS elements
12:    for  $C_i \in C$  do ▷ iterate over clusters, possibly in parallel
13:      for  $x \in C_i$  do ▷ iterate over solutions, possibly in parallel
14:         $\text{MO-GOM}(x, F_j, M_{ij}, A)$  ▷ Algorithm 2.6
15:      end
16:    end
17:    for  $x \in P$  do
18:       $\text{updateElitistArchive}(x, A)$  ▷ Algorithm 2.7
19:    end
20:  end
21:  for  $x \in P$  do
22:     $\text{anticipatedMeanShift}(x)$ 
23:  end
24:  for  $x \in P$  do
25:    if  $\text{NIS}(x) > \text{NIS}_{\text{MAX}}$  then
26:       $\text{forcedImprovement}(x)$ 
27:    end
28:  end
29:  for  $C_i \in C, j \in \{0, \dots, |\mathcal{F}| - 1\}$  do
30:     $\text{updateSamplingModel}(M_{ij})$ 
31:  end
32: end
33: return  $A$ 

```

Algorithm 2.6: MO-GOM(x, F_j, M_{ij}, A), reproduced from [22]

```

1:  $\text{sample} \leftarrow \text{samplePartialSolution}(M_{ij})$ 
2:  $\Delta f \leftarrow \text{partialEvaluation}(\text{sample}, x, f_x, F_j)$ 
3:  $f_{\text{new}} \leftarrow f_x + \Delta f$ 
4: if  $f_{\text{new}} \succ f_x$  or  $\text{improvesArchive}(f_{\text{new}}, A)$  then
5:    $f_x \leftarrow f_{\text{new}}$ 
6:   for  $i \in \{0, \dots, |F_j| - 1\}$  do
7:      $x[F_j[i]] \leftarrow \text{sample}[i]$ 
8:   end
9: end

```

Algorithm 2.7: updateElitistArchive(x, A), describing the adaptive elitist archive of [22]

```

1: if  $\exists y \in A : y \succ x$  then
2:   return ▷  $x$  is dominated, so reject it
3: end
4:  $A \leftarrow (A \setminus \{y \in A : x \succ y\}) \cup \{x\}$  ▷ add  $x$ , discarding what it dominates
5: if  $|A|$  exceeds  $1.25 \times \text{target size}$  then
6:    $\text{pruneArchive}(A)$  ▷ prune to  $|A| = 0.75 \times \text{target size}$ 
7: end

```

2.3 Uncertainty in Evolutionary Optimisation

Jin and Branke [23] divide the uncertainty an EA may face into four categories: noisy objective functions, approximated objective functions, objective functions that change over time, and the setting in which selected solutions cannot be realised exactly. Two of these categories are relevant to this work, and are covered below: noisy objective functions and approximated, or surrogate, objective functions. The difficulty in both of these settings is that the fitness function is inaccurate, so the fitness an individual appears to have differs from its true fitness.

2.3.1 Noisy Objective Functions

An objective function is *noisy* when evaluating the same solution twice may yield different values. We write

$$\tilde{f}(\mathbf{x}) = f_{\text{true}}(\mathbf{x}) + \varepsilon \quad (2.11)$$

where f_{true} is the true objective function and ε is drawn independently from a noise distribution at every evaluation. Some EAs rely on the ranking induced by the objective function, for instance, GOM of Algorithm 2.3 only determines which solution to keep based on which is better and nowhere considers the magnitude of difference in fitness. Optimisation then only is impaired if the noise is great enough for an inferior solution to be wrongfully thought to be superior. Rakshit et al. [24] survey strategies for optimisation using EAs in noisy settings such as sampling a solution multiple times to average out the noise, estimating objective values from neighbouring solutions, and adapting the selection operator. Resampling is an effective yet costly approach. Its cost is why it is usually applied adaptively, allocating more samples only where the ordering is actually in doubt [25]. Selection can be made more robust to noise for instance by assigning probabilities to a dominance relation.

Pareto dominance is a relation over multiple objectives, and a noise realisation in any single objective can already change the perceived relation. Several authors therefore replace the Pareto dominance relation of Eq. (2.8) with one that considers uncertainty. Limbourg [26] represents for each uncertain objective i its fitness as an interval $[f_i(\mathbf{x}), \bar{f}_i(\mathbf{x})]$, and uses it to define a certain and uncertain version of the Pareto dominance relation. Teich [27] proposes a similar approach, but instead assigning each comparison of solutions a probability of dominance.

2.3.2 Approximate Objective Functions

When objective evaluations are computationally expensive, the objective evaluations can account for nearly all of the optimisation cost, and the use of a cheaper approximation of the true objective function becomes attractive. Such an approximation is called a *surrogate*, or a *metamodel* [28], [29]. We write $\hat{f}(\mathbf{x}) \approx f_{\text{true}}(\mathbf{x})$, and define the approximation error as

$$e(\mathbf{x}) = \hat{f}(\mathbf{x}) - f_{\text{true}}(\mathbf{x}). \quad (2.12)$$

Surrogates are obtained either by learning a model from previously evaluated solutions, or by simplifying the true objective function itself. The latter are known as *low-fidelity* objective functions, where accuracy is traded for speed [30], [31]. The crucial difference with noisy objective functions is that $e(\mathbf{x})$ is a deterministic function of $\mathbf{x}$. Evaluating the same solution twice returns the same value, so the true objective value cannot be approximated by performing reevaluations. When relying only on the ranking induced by the objective function, as is for instance done in GOM (Algorithm 2.3), a surrogate is harmful only where it changes the ordering of solutions. The ordering of two solutions $\mathbf{x}$ and $\mathbf{x}'$ is reversed when $\hat{f}$ and f_{true} disagree on the sign of the difference between the two:

$$\text{sign}(\hat{f}(\mathbf{x}) - \hat{f}(\mathbf{x}')) \neq \text{sign}(f_{\text{true}}(\mathbf{x}) - f_{\text{true}}(\mathbf{x}')). \quad (2.13)$$

Written out in terms of the approximation error, the ordering is flipped when

$$(f_{\text{true}}(\mathbf{x}') < f_{\text{true}}(\mathbf{x})) \wedge (e(\mathbf{x}') - e(\mathbf{x}) > f_{\text{true}}(\mathbf{x}) - f_{\text{true}}(\mathbf{x}')) \quad \text{or} \\ (f_{\text{true}}(\mathbf{x}) < f_{\text{true}}(\mathbf{x}')) \wedge (e(\mathbf{x}) - e(\mathbf{x}') > f_{\text{true}}(\mathbf{x}') - f_{\text{true}}(\mathbf{x})). \quad (2.14)$$

A large approximation error is thus harmless as long as it is nearly the same for both solutions, whereas a small one suffices to reverse an ordering once the two solutions are close in true objective value.

Depending on e , the minimum of $\hat{f}$ may not align with that of f_{true} . An optimiser applied to $\hat{f}$ then wrongfully converges to a *false optimum*: a solution which scores well on the surrogate but not on the true objective function. We call this process overfitting (to the surrogate fitness function).

2.4 Brachytherapy Treatment Planning

Radiotherapy is used to treat cancer by delivering radiation to a tumour. Radiation damages tumour cells such that they stop dividing and die off. Brachytherapy is a form of radiotherapy in which a radioactive source is temporarily placed inside the body, directly in or next to the tumour [8]. The received radiation dose falls off steeply with distance, allowing surrounding healthy tissue to be spared. Brachytherapy treatment planning decides where the source is placed and how long it stays there, which together determine the dose distribution. Treatment planning is inherently an MO problem, because there is a tradeoff between damaging tumour cells and not damaging healthy tissue. This work considers high-dose-rate (HDR) brachytherapy for cervical cancer. In HDR brachytherapy the dose is delivered within minutes by a radioactive source that is driven through the channels of an applicator and/or catheters. The radioactive source stops at a discrete sequence of *dwell positions* in each channel. The time the source stops at a given dwell position is called its *dwell time*. All dwell positions and their dwell times constitute a *treatment plan*. The dwell positions lie along the channels of the applicator and catheters, which are already placed when the BRIGHT toolkit is used for optimising the dwell times. BRIGHT uses MO-RV-GOMEA to find an approximation front of treatment plans, each representing a different tradeoff between damaging tumour cells and not damaging healthy tissue. Subsection 2.4.1 describes how the quality of a treatment plan is quantified, Subsection 2.4.2 how the planned dose distribution of a treatment plan is computed, and Subsection 2.4.3 how BRIGHT aggregates the planned dose into the objectives which are optimised for. Lastly, Subsection 2.4.4 explains where uncertainty enters the computed objective values of treatment plans.

2.4.1 Treatment Plan Quality

Treatment plans are judged on the planned dose in (subvolumes of) regions of interest (ROIs). ROIs include the gross tumour volume (GTV) and the clinical target volumes (CTVs), in which a high dose is desired, and the organs at risk (OARs), in which a low dose is desired. Treatment plan quality is measured using *dose-volume indices* (DVIs). Two types of DVIs exist: *dose indices* and *volume indices*. The dose index D_v of an ROI is the lowest planned dose in the subvolume of size v with the highest planned dose in the ROI. v can either be a fraction of the total ROI's volume or an absolute volume. For example, $D_{2\text{cm}^3}$ of the bladder is the minimum dose in the 2cm^3 of bladder tissue with the highest planned dose. The volume index V_d is the volume of an ROI with at least d planned dose. Dose can be measured in *gray* (Gy). However, a radiotherapy treatment is often not delivered in a single session but split into several *fractions*, and the same total dose has a different biological effect depending on how it is fractionated. Dose is therefore expressed as an equivalent dose in 2 Gy fractions (EQD2) [8].

HDR brachytherapy for cervical cancer is generally delivered in multiple brachytherapy fractions and preceded by external beam radiotherapy (EBRT) fractions, which apply radiation more uniformly.

A *clinical protocol*, such as EMBRACE-II, states aims for the DVIs of a treatment plan. Translating the aims of EMBRACE-II directly into objectives for BRIGHT to optimise for does not yield the desirable 3D dose distributions for cervical cancer. EMBRACE-II+SHAPE was therefore defined, which extends EMBRACE-II with the two additional DVI aims *bot-normal-tissue* and *mid-top-normal-tissue* and shape aims. The DVI aims of EMBRACE-II+SHAPE are listed in Table 2.1. Each entry states an ROI, a dose index, a comparison, and an aim. Entries with a $>$ comparison are *coverage aims*, which define a minimum dose a subvolume of a target volume should receive. Entries with a $<$ comparison are *sparing aims*, which define a maximum dose a subvolume of an ROI should receive. A protocol may state both a lower and an upper bound for the same DVI, as is the case here for $D_{90\%}$ of the *high risk clinical target volume* (CTV_{HR}). Shape targets concern the geometry of the dose distribution, which DVIs do not capture. An *isodose surface* is the surface formed between points that do and do not receive some dose, so the 100% isodose surface bounds the region which receives at least 100% of the prescribed dose. Each shape aim states that the isodose surface at a given dose should closely follow the contour of a given ROI. EMBRACE-II+SHAPE contains shape targets for the bladder, bowel, rectum and sigmoid, each stating that where the organ borders the CTV_{HR}, the 100% isodose surface should follow the contour of the CTV_{HR} (instead of extending into the organ).

ROI	DVI		Aim
<i>Coverage</i>			
GTV _{res}	$D_{98\%}$	$>$	95 Gy(EQD2)
CTV _{HR}	$D_{90\%}$	$>$	90 Gy(EQD2)
CTV _{HR}	$D_{98\%}$	$>$	75 Gy(EQD2)
CTV _{IR}	$D_{98\%}$	$>$	60 Gy(EQD2)
<i>Sparing</i>			
CTV _{HR}	$D_{90\%}$	$<$	95 Gy(EQD2)
Bladder	$D_{2\text{cm}^3}$	$<$	80 Gy(EQD2)
Rectum	$D_{2\text{cm}^3}$	$<$	65 Gy(EQD2)
Recto-vaginal point	D	$<$	65 Gy(EQD2)
Sigmoid	$D_{2\text{cm}^3}$	$<$	70 Gy(EQD2)
Bowel	$D_{2\text{cm}^3}$	$<$	70 Gy(EQD2)
bot-normal-tissue	$D_{90\%}$	$<$	35 %
mid-top-normal-tissue	$D_{2\text{cm}^3}$	$<$	100 %

Table 2.1: The DVI aims of EMBRACE-II+SHAPE. Each row states an ROI, a DVI of that ROI, a comparison operator, and the aim value. Coverage aims set a lower bound and sparing aims set an upper bound. The shape aims are not listed here. The EMBRACE-II protocol states these same DVI aims except the two on *bot-normal-tissue* and *mid-top-normal-tissue*, and has no shape aims. GTV_{res} is the residual gross tumour volume, CTV_{HR} the high risk clinical target volume and CTV_{IR} the intermediate risk clinical target volume; *bot-normal-tissue* and *mid-top-normal-tissue* are healthy tissue regions below and around the applicator. Aims in Gy(EQD2) are in terms of the dose over all fractions, and aims in percent are percentages of the prescribed dose per single fraction.

2.4.2 Dose Calculation

DVIs can be estimated from the planned dose at a finite set of points inside the ROI, called dose calculation points (DCPs). A dose index D_v can be computed by sorting the planned dose at the DCPs in the ROI, and reading off the value at the rank corresponding to v . A volume index V_d can simply be calculated by counting how many of the DCPs receive a dose greater than d . The more DCPs are used, the more accurate, yet computationally expensive DVI estimates become. Volume indices are not included in the clinical protocols used in this work and thus play no further role.

Treatment plans can be evaluated using a simulation which requires the dwell positions and the *delineated* ROIs to be known. A clinician marks, or delineates, the ROIs on a 3D image of the patient, which consists of 2D slices. The dwell positions are also reconstructed from the same image. The TG-43 formalism [32] defines how much dose a point in the patient's body can be assumed to receive from a given source based on their relative locations. A *dose rate* R_{ij} can be defined as the dose which DCP j receives per unit of dwell time at dwell position i . Since the dwell positions contribute independently to the total dose at each point, the dose d_j at the DCP indexed by j is computed as

$$d_j = \sum_i t_i R_{ij}, \quad (2.15)$$

where t_i is the dwell time at the dwell position indexed by i . DCPs are sampled randomly for the dose calculation: more DCPs constitute a higher resolution or higher fidelity level. By default BRIGHT divides the DCPs equally among all ROIs. The dose rates for all pairs of the l dwell positions and m DCPs are collected into a *dose-rate matrix* $\mathbf{R} \in \mathbb{R}^{l \times m}$, which BRIGHT precomputes per patient. The dwell times of a plan form a vector $\mathbf{t}$. The dose at every DCP then follows from a single product:

$$\mathbf{d}(\mathbf{t}) = \mathbf{t}\mathbf{R}. \quad (2.16)$$

Two properties follow directly, both of which BRIGHT uses, which allow fitness evaluations to be performed faster. First, the dwell times of a whole population of plans can be stacked into a matrix, such that the dose at every DCP of every plan follows from a single matrix-matrix multiplication, which BRIGHT performs on a graphics processing unit (GPU). Second, when only a subset of the dwell times is changed, only the rows of $\mathbf{R}$ belonging to those dwell times contribute to the change in dose. This enables partial evaluations by performing Eq. (2.16) with $\mathbf{R}$ substituted by a matrix containing only a subset of its rows. Figure 2.3 summarises how BRIGHT computes dose index values.

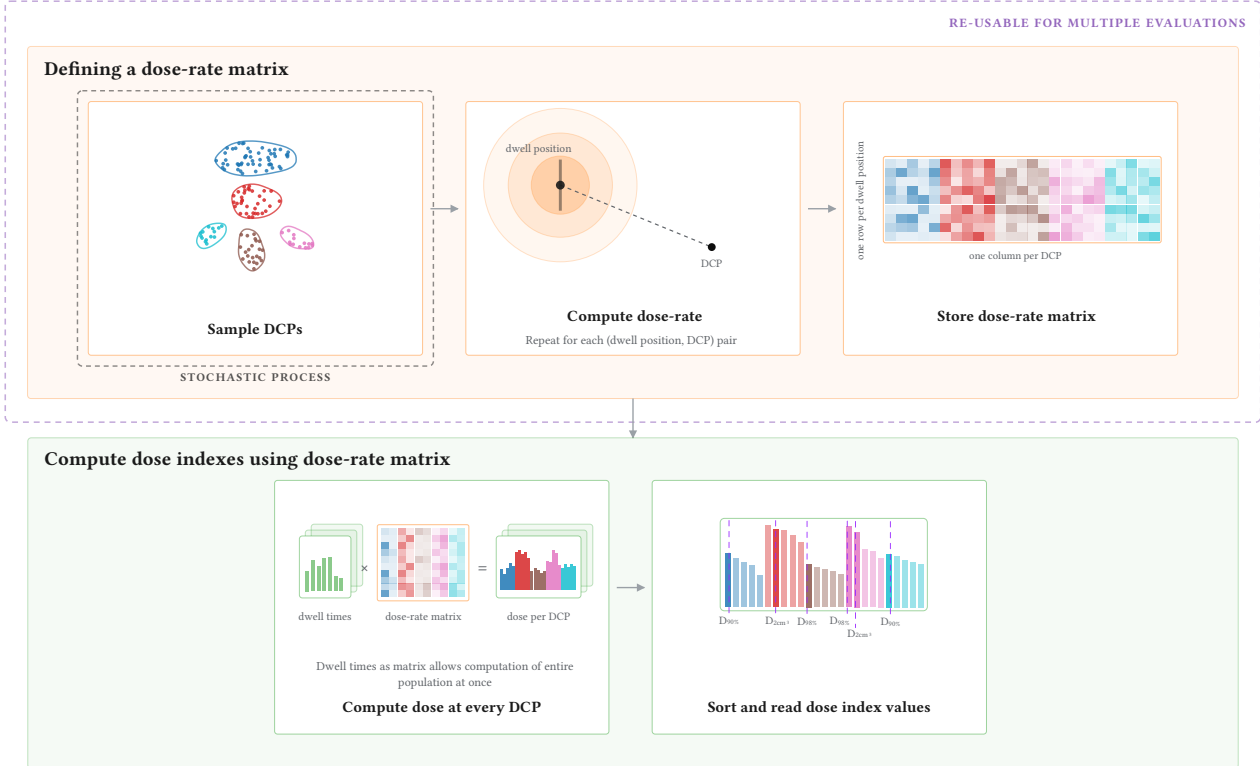


Figure 2.3: Illustration of how BRIGHT arrives at the DVI values of a treatment plan. Everything inside the dashed purple frame can be computed once and be reused by multiple fitness evaluations. Only the sampling of the DCPs is stochastic and every step after it is deterministic. Each evaluation multiplies the dwell times of one or more treatment plans with the dose-rate matrix in a single product, after which the doses at the DCPs of an ROI are sorted, so that every dose index of that ROI can be read off at its own rank.

2.4.3 BRIGHT

BRIGHT is a toolkit used for optimising HDR brachytherapy treatment plans using MO-RV-GOMEA [9], [10]. BRIGHT contains a graphical user interface which makes clear at a glance what tradeoffs are available between treatment plans. The graphical user interface lets a clinician navigate, compare and adjust plans while inspecting their dose distributions on the 3D image of the patient. For optimisation BRIGHT aggregates the many aims defined in a clinical protocol into two or three weighted sums. The weights of the shape aims are pre-defined and are aggregated into the SHAPE objective. The DVI aims are aggregated into two objective functions: the *least coverage index* (LCI) and the *least sparing index* (LSI), which loosely capture “how much target volume is damaged” and “how much healthy tissue is damaged” according to the DVIs.

$D_a(t)$ is the value of the DVI of aim a for plan t and T_a is the aim’s target value. The violation of each aim is normalised such that violations of different aims can be compared:

$$\delta_a(t) = \begin{cases} \frac{T_a - D_a(t)}{0.85T_a} & \text{if } a \text{ is a coverage aim} \\ \frac{D_a(t) - T_a}{T_a} & \text{if } a \text{ is a sparing aim.} \end{cases} \quad (2.17)$$

A positive $\delta_a(t)$ means an aim a is violated.

We define LCI_w and LSI_w as follows:

$$\text{LCI}_w(t) = \sum_{a \in \text{coverage aims}} w_a \delta_a(t), \quad (2.18)$$

$$\text{LSI}_w(\mathbf{t}) = \sum_{a \in \text{sparing aims}} w_a \delta_a(\mathbf{t}), \quad (2.19)$$

where exponentially greater weights are assigned to aims with bigger relative violations of their DVI aims for each respective plan. The fitness function used by BRIGHT for protocols which include shape aims is:

$$\mathbf{f}(\mathbf{t}) = [\text{LCI}_w(\mathbf{t}), \text{LSI}_w(\mathbf{t}), \text{SHAPE}(\mathbf{t})]. \quad (2.20)$$

Instead of reporting LCI_w and LSI_w of the found plans, BRIGHT reports LCI and LSI which are considered more interpretable:

$$\text{LCI}(\mathbf{t}) = - \max_{a \in \text{coverage aims}} \delta_a(\mathbf{t}), \quad (2.21)$$

$$\text{LSI}(\mathbf{t}) = - \max_{a \in \text{sparing aims}} \delta_a(\mathbf{t}), \quad (2.22)$$

where only the greatest aim violation is included and the problem is rephrased as a maximisation problem. All DVI aims are met when LCI and LSI are greater than 0. This region is referred to as the *golden corner*. BRIGHT regards a plan as infeasible when its LCI or LSI is below -0.3 . Only feasible plans are considered in the performance indicators of this thesis. Of the plans in an approximation set S , the *L-plan* is the one for which LCI and LSI are most balanced, in the sense that the lowest of the two is greatest [33]:

$$\text{argmax}_{\mathbf{t} \in S} \min(\text{LCI}(\mathbf{t}), \text{LSI}(\mathbf{t})), \quad (2.23)$$

and the *L-score* is $\max_{\mathbf{t} \in S} \min(\text{LCI}(\mathbf{t}), \text{LSI}(\mathbf{t}))$. The L-score, the best SHAPE score, and the number of solutions found which lie in the golden corner are, in addition to HV of all feasible plans, used as performance metrics in this thesis. The FOS is a linkage tree and is defined once by hierarchical clustering using the Euclidean distance between the dwell positions [22]. This is one form of grey-box knowledge which the problem offers, as dwell positions which lie close together affect similar areas and are thus more dependent, whereas distant dwell positions barely interact.

2.4.4 Uncertainty in BRIGHT

Evaluating a treatment plan with a dose calculation simulation only approximates the true objective values, because the dose is computed at a finite set of DCPs, and the approximation becomes more accurate as more DCPs are used [34]. Using fewer DCPs is however attractive: evaluations become cheaper, and optimisation can cover more generations in the same amount of time. Bouter et al. [12] show that overfitting occurs when fewer DCPs are used when optimising brachytherapy treatment plans for prostate cancer using BRIGHT.

The sampling of the DCPs is the only stochastic process in the fitness evaluation (top left in Figure 2.3) and every step after it is a deterministic function of the drawn DCP set. A set of DCPs thus induces a function that is an approximation of the true fitness function. If for each evaluation a new set of DCPs are drawn, the optimiser will face noise of the form introduced in Subsection 2.3.1. If a set of DCPs is reused for multiple evaluations, the fitness function is an unchanging approximation of the true fitness function in the sense of Subsection 2.3.2 and Eq. (2.12). Which kind of uncertainty BRIGHT faces is thus not a property of the fitness function, but a choice between redrawing or reusing a DCP set. By default a multi-resolution scheme is used where three sets of different sizes are drawn at the start of each run. The smallest set is used for the first third of the optimisation run, the largest for the last third of the run. Redrawing a new DCP set is costly, but leaves an optimiser without a static approximation of the true fitness function that it can overfit to. Redrawing within a generation would be especially infeasible, as the use of a single dose-rate matrix is what allows the whole population

to be evaluated in one matrix-matrix multiplication on the GPU (Eq. (2.16)), and plans evaluated on different sets of DCPs could no longer be evaluated in parallel.

How far the number of DCPs can be reduced, and how the uncertainty which that introduces should be handled, is what this thesis investigates.

Uncertainty in Benchmark Problems

3

This chapter aims to answer [SRQ1](#) and [SRQ2](#).

SRQ1 How does the use of noisy fitness functions affect MO-RV-GOMEA’s convergence, computational cost and final solution quality?

SRQ2 How does the number of distinct low-fidelity fitness functions used during a run affect MO-RV-GOMEA’s convergence, computational cost and final solution quality?

We do so by studying (MO-)RV-GOMEA and AMaLGaM when used to optimise benchmark problems. In our experiments, either a noisy or a low-fidelity approximation of the fitness function is used. Benchmark problems help us gain insight into the algorithms’ behaviour in a generic (uncertain) setting, which can help us later form a strategy on how to use low-resolution fitness evaluations in BRIGHT. A run on a benchmark problem takes seconds on a single CPU core whereas a BRIGHT run takes minutes on an expensive GPU. Using benchmark problems thus allows us to iterate faster and any insights might be transferable to other work on other real-world problems that display uncertainty. We modify the fitness functions of the benchmark problems such that they are modelled after the noisy setting of Eq. (2.11) and low-fidelity setting of Eq. (2.12).

3.1 Noisy Fitness Functions

We start with studying noisy fitness functions, as defined by Eq. (2.11). The use of noisy fitness functions models how the fitness function of BRIGHT behaves when a new distinct set of DCPs is used for each evaluation. We study the SO case first, as progress is easier to measure there with quality being measured by the best objective value found so far. The multi-objective case follows and brings us closer to the setting of BRIGHT. All experiments in this section use the implementations published by Bosman [35].

3.1.1 Convergence under Noise

We start with asking two questions in an SO setting: First, *does RV-GOMEA still converge to the optimum when the fitness function is noisy?* Second, *does elitism affect whether, and how quickly, RV-GOMEA converges to the optimum when the fitness function is noisy?* Elitist solutions might be exactly the solutions which have a beneficial noise realisation, making their fitness seem better than they are in reality on the true fitness function. GOM in RV-GOMEA is inherently elitist, and thus we use RV-GOMEA’s sibling AMaLGaM to study the effects of elitism. AMaLGaM itself carries only the single best solution over from one generation to the next and can thus already be considered less elitist than RV-GOMEA. We also use a modified variant of AMaLGaM where the elitist solution is not copied to the next generation. In terms of Algorithm 2.2, this removes line 8, in which the best selected solution is copied into the new population, so that the loop of line 9 draws all n solutions from the estimated distribution rather than $n - 1$.

All runs minimise the Sphere benchmark function:

$$f(\mathbf{x}) = \sum_{i=1}^l x_i^2, \quad (3.1)$$

whose optimum lies at $\mathbf{x} = \mathbf{0}$. At every evaluation the value returned to the optimiser is $\tilde{f}(\mathbf{x}) = f_{\text{true}}(\mathbf{x}) + \varepsilon$, with ε drawn independently from $\mathcal{N}(0, \sigma_{\text{noise}}^2)$. We examine six noise levels: $\sigma_{\text{noise}} \in \{10^{-7}, 10^{-5}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$. We use $l = 100$ decision variables, and the population is initialised uniformly in $[-115, -100]^l$ following [21], which does not contain the optimum to prevent biased initialisation around the optimum. A univariate linkage model is used in all runs. Each configuration is run for a budget of 10^7 evaluations, with a population size of 100 for AMaLGaM and 30 for RV-GOMEA, and is repeated over 30 independent runs.

Figure 3.1 shows the true fitness of the solution that is best in noisy space over evaluations for all six noise levels. Lines show the median over the runs and shaded bands show the interdecile range. Within each algorithm the six noise levels trace a common descent for most of the run. The two AMaLGaM variants share that descent with one another, whereas RV-GOMEA follows a trajectory of its own: it holds at the initial fitness of roughly 10^6 for longer than AMaLGaM. Each curve leaves the descent of its own algorithm and flattens, at a fitness ordered by noise magnitude: the largest σ_{noise} stalls earliest and at the highest true objective value, and each reduction in σ_{noise} extends the descent further before the same stall occurs. After the stall no further progress is made for the remainder of the budget.

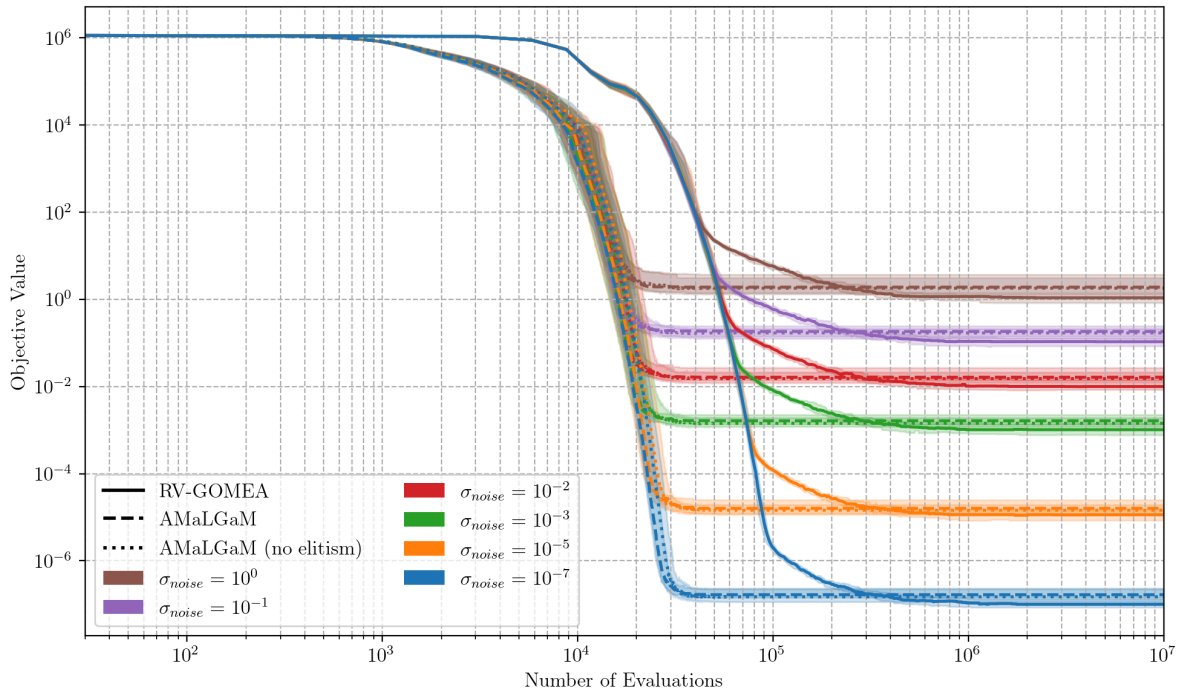


Figure 3.1: True fitness of the best, in noisy space, objective value over time on the Sphere benchmark problem with added Gaussian noise. Colours correspond to different noise standard deviations σ_{noise} . For each setting, results are aggregated over 30 independent runs. Lines show the median, shaded regions indicate the interdecile range.

Figure 3.2 separates the noise levels into individual panels and zooms to the interval from 10^6 to 10^7 evaluations on a linear scale. Lines show the median over the runs and the interdecile range is not shown. The tails of the convergence plots are flat, with no downward drift. Convergence happens between 1 and 2 standard deviations of the noise from the optimum, with RV-GOMEA achieving slightly better final solution quality than AMaLGaM.

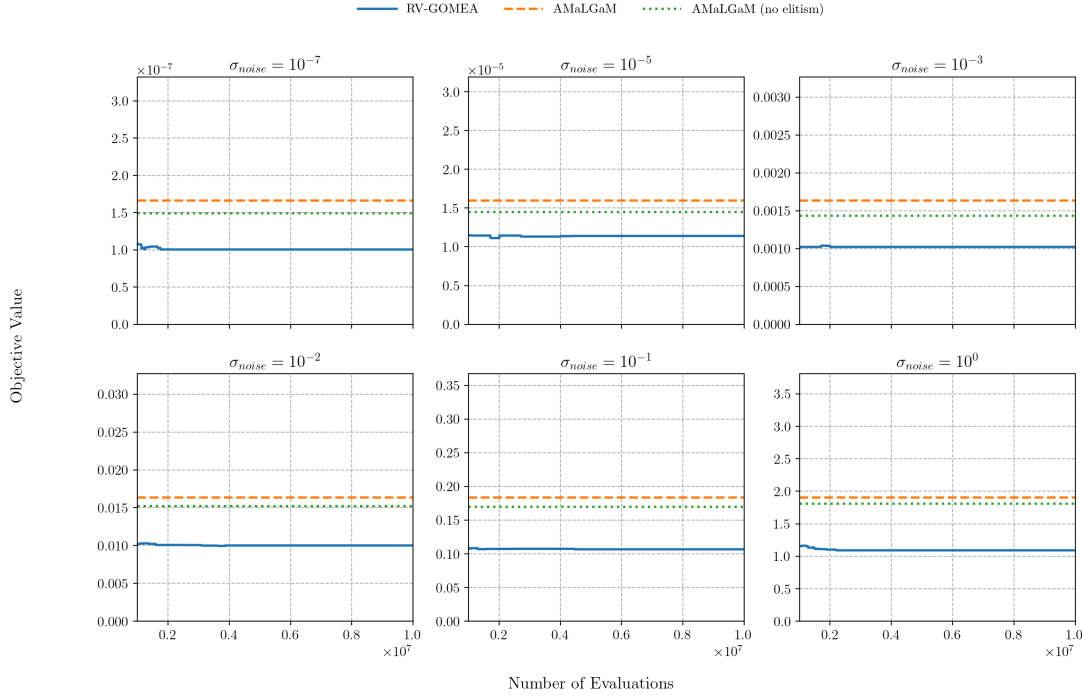


Figure 3.2: True fitness of the in noisy space best objective value over time on the Sphere benchmark problem with added Gaussian noise. Each subplot corresponds to a different noise standard deviation σ_{noise} . For each setting, results are aggregated over 30 independent runs. Lines show the median, plotted on a linear scale between 0.1×10^7 and 1.0×10^7 evaluations.

Before progress stalls, we see that the rate of improvement is the same for all noise levels. The higher the noise level, the earlier the stall happens, but before this point, the progress is as fast. This implies, assuming evaluations with higher noise levels are computationally cheaper in a real-world problem, that a method of switching between noise levels might be a faster approach. The optimiser fully stalls at a noise-dependent floor rather than reaching the optimum. Once solutions approach the optimum closely, ε is relatively large, and it becomes impossible to distinguish actually good solutions from bad ones. A majority of the best solutions would still be found closer to the true optimum, when the best solution so far is between σ_{noise} and $2\sigma_{\text{noise}}$ from the optimum, and progress towards the optimum would, at least slowly, continue. It seems that from generation to generation, all algorithms either keep sampling from the same distribution that doesn't include (with reasonable probability) the true optimum, or carry over falsely optimal solutions from previous generations. The former must be true, at least for AMaLGaM without elitism, because it also halts between $1.5\sigma_{\text{noise}}$ and $2\sigma_{\text{noise}}$ from the optimum. A plausible explanation is that the distribution from which solutions are sampled remains the same because it collapses to a single point, such that it later never finds solutions that are closer to the true optimum.

3.1.2 Enforcing a Minimum Sampling Variance

In AMaLGaM and RV-GOMEA, the distribution multiplier is decreased when no improvements are found, and once NIS_{MAX} is reached it keeps decreasing towards 0. The stall seen in the results of the previous experiment could thus be caused by a collapse of the distribution from which new (partial) solutions are sampled. If this is the case, then applying a lower bound could restore progress. We test this hypothesis with another experiment.

The setup is that of the previous experiment, but with the variance of the distribution from which new solutions are sampled floored at $k\sigma_{\text{noise}}$ and $c_{\text{Multiplier}}$ floored at 1, such that there always is some diversity in newly sampled solutions. We use $k \in \{0, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$, where $k = 0$ invokes the same behaviour as the previous experiment of Figure 3.1.

Figure 3.3 shows one panel per factor k . Movement towards the optimum slows down for $k \neq 0$ and both AMaLGaM variants converge further from the optimum than with $k = 0$. Instead the plateau rises with k , so that larger floors arrest progress earlier and at a higher fitness, and no combination of σ_{noise} and k reaches the optimum within the budget. AMaLGaM halts completely while RV-GOMEA does move towards the optimum after the initial stall, but does not reach a point better than when using $k = 0$ within the given evaluation budget.

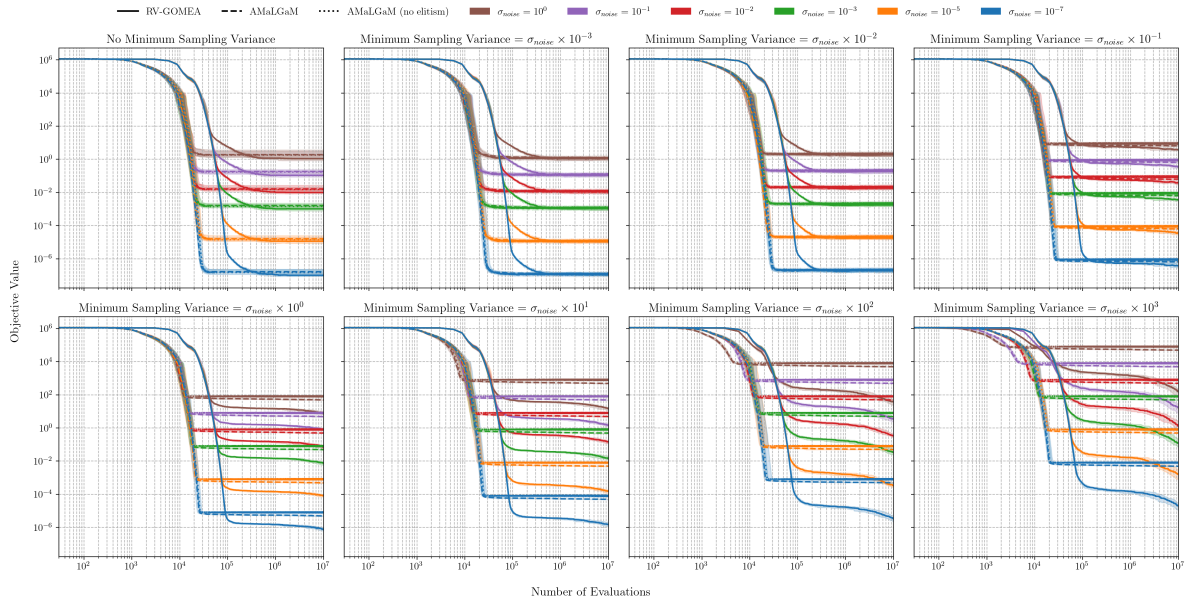


Figure 3.3: True fitness of the best, in noisy space, objective value over time on the Sphere benchmark problem with added Gaussian noise. Colours correspond to different noise standard deviations σ_{noise} . In each subplot, there is a different minimum variance for the distribution from which new solutions are sampled each generation. For each setting, results are aggregated over 30 independent runs. Lines show the median, shaded regions indicate the interdecile range.

The stall happens earlier when enforcing a minimum sampling variance. Both AMaLGaM versions fully stall, while RV-GOMEA does still show progress after its stall, but performance only degrades with higher k between 0 and 10^7 evaluations. The difference between AMaLGaM and RV-GOMEA may be explained by the fact that RV-GOMEA considers, specifically in a univariate setting, differences in fitness per changed variable, whereas AMaLGaM draws all $l = 100$ variables at the same time. Possibly, due to changes to variables being tested one by one in GOM (with a univariate FOS), RV-GOMEA is still able to make slow progress in the true objective space. Nonetheless, this direction does not look promising for either algorithm unless one can detect when a stall happens, and only then set k higher.

3.1.3 Multiple Objectives

BRIGHT uses MO-RV-GOMEA, so the SO result must be checked in the setting which actually applies. Noise now corrupts the Pareto dominance relation rather than a scalar ordering. An elitist archive moreover stores solutions on their noisy values and by default never re-evaluates them. The question

is whether the SO conclusions, a stall at a noise-dependent floor and a rate of progress unaffected until that floor, carry over to MO-RV-GOMEA.

The experiment used to answer said question is set up as follows. MO-RV-GOMEA is used and run on four different bi-objective benchmark problems: ZDT1, ZDT2, ZDT4 and ZDT6 [36]. $l = 20$ decision variables and a population initialised uniformly in $[0, 1]^l$, which contains the optimum. Note that range $[0, 1]^l$ is part of the definition of the ZDT problems. For ZDT1 and ZDT2, the objective values would be unbounded outside of $[0, 1]^l$, and thus no Pareto front would exist. ZDT4 and ZDT6 are undefined for some values outside their range. At every evaluation the value returned to the optimiser for objective i is $\tilde{f}_i(\mathbf{x}) = f_i(\mathbf{x}) + \varepsilon_i$, with each ε_i drawn independently from $\mathcal{N}(0, \sigma_{\text{noise}}^2)$. We examine the same six noise levels, $\sigma_{\text{noise}} \in \{10^{-7}, 10^{-5}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$. The optimiser uses a single population of 200 individuals, a univariate linkage model, 10 clusters, and an elitist archive with a target size of 1 000, under a budget of 10^7 evaluations. Quality is measured by $D_{P_f \rightarrow S}$, computed on true fitness values against a reference set of 5 000 points from the optimal Pareto front. Each configuration is repeated over 30 independent runs.

Figure 3.4 shows the true $D_{P_f \rightarrow S}$ of the approximation set found by the algorithm over the number of evaluations. Lines show the median $D_{P_f \rightarrow S}$ and the shaded bands show the interdecile range. We observe that for all noise levels the runs start on the same $D_{P_f \rightarrow S}$. The runs do not track a common curve early on and immediately depart from each other. Some settings have a great variance between runs, specifically ZDT4 with $\sigma_{\text{noise}} = 10^{-7}$ and ZDT2 with $\sigma_{\text{noise}} = 10^{-1}$ and $\sigma_{\text{noise}} = 10^{-2}$. Progress of $D_{P_f \rightarrow S}$ eventually halts and does not improve further. The runs generally halt at values clearly ranked by the σ_{noise} used.

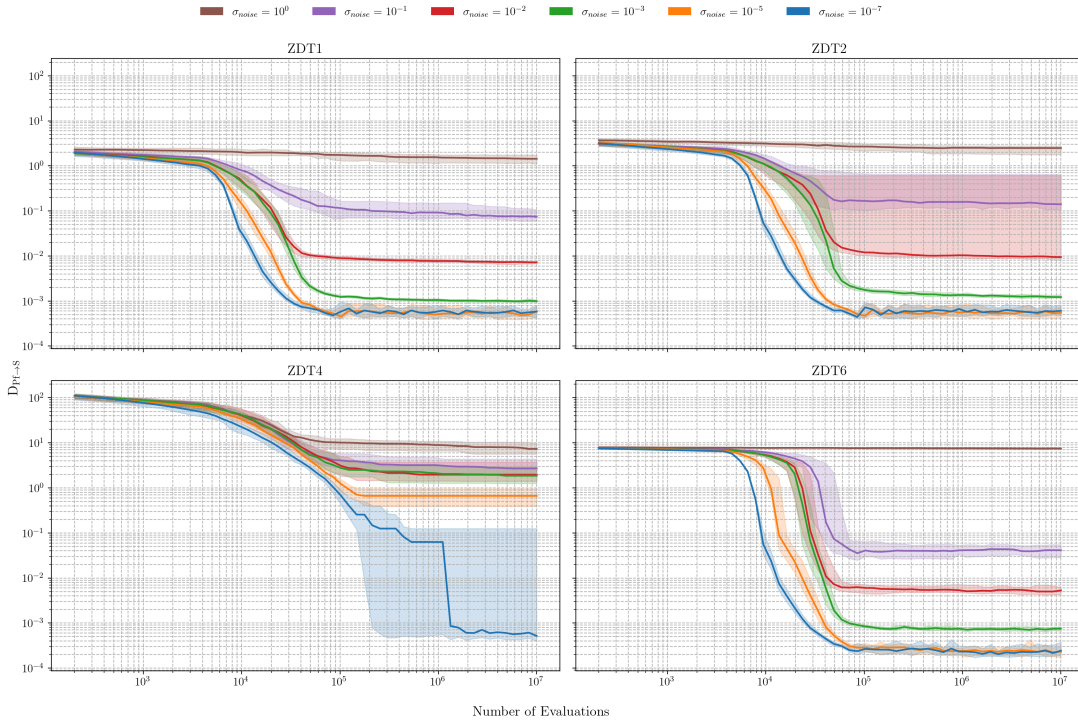


Figure 3.4:

$D_{P_f \rightarrow S}$ from a 5 000-point sample of the Pareto front to the approximation set found by MO-RV-GOMEA on ZDT benchmark problems with added Gaussian noise, computed on true objective values. Colours correspond to noise standard deviations σ_{noise} . Results are aggregated over 30 independent runs; lines show the median, shaded regions the interdecile range.

ZDT4 with $\sigma_{\text{noise}} = 10^{-7}$ shows great variance, which can be explained by the multi-modal nature of ZDT4, where multiple local Pareto fronts exist. An optimiser might get stuck on such a local Pareto front and never find a better one. The $D_{P_f \rightarrow S}$ found is better for a lower σ_{noise} . The fact that better solutions are found for lower σ_{noise} directly transfers from our SO experiments. What we do not observe in this MO setting, as opposed to the SO setting of Subsection 3.1.1, is the same approximate rate of improvement for all noise levels until progress stalls suddenly. Possibly, this is due to the solutions not being further from the true Pareto front, but rather because the approximation set is smaller. It might happen that due to a single noise realisation, a number of points that would be in the approximation set are now dominated. Another possible explanation is that the initialisation range used for the SO Sphere benchmark problem was far from the optimum, whereas this was not the case here for the ZDT problems. In the SO Sphere setting, it might be that where Figure 3.1 shows the same rate of improvement for low and high noise levels, the algorithm is simply traversing the large section of search space between the initial population and the optimum, where ε is extremely small compared to the true fitness differences between individuals on the slope. The computational cost difference of low and high noise fitness functions needs to be great enough (instead of non-zero) for a strategy where we use high noise early in the run and later low noise to be viable.

3.1.4 Conclusion

In order to answer SRQ1, we first considered an SO setting and found that in the setting tested, the rate of improvement is the same for different noise levels, until a sudden halt of progress. Progress halts at a distance from the optimum which increases with the noise level. Elitism seems to have no effect on this behaviour, and convergence still stalls for non-elitist algorithms, even when the distribution from which new solutions are sampled does not degrade to a single point. In the evaluated MO setting we do find that there is a difference in the rate of improvement between noise levels, even before a clear stagnation. This observed difference between MO and SO behaviour may also be explained by other factors such as the different problem types and the difference between quality metrics. Computational cost cannot be expressed unless it is known by how much noisy evaluations are cheaper. To converge to the (near) optimum, noise levels need to be zero or insignificantly small. However, given a significant enough saving in computational cost when using noisy evaluations it seems that a speed-up can be achieved by using noisy evaluations early in the optimisation.

3.2 Low-Fidelity Fitness Evaluations

The first part of this chapter explored one type of uncertainty found in BRIGHT: noisy uncertainty, which is observed when a new DCP set is sampled for each evaluation. In this section we study what happens when we do not draw a new DCP set each evaluation. A fitness function defined by a single DCP set constitutes an approximation of the true fitness function as introduced in Subsection 2.3.2, i.e., a low-fidelity function. We attempt to answer SRQ2, and also explore a middle-ground between the two extremes, where a low-fidelity function is redrawn for each generation, as opposed to never or each evaluation. In order to perform experiments on benchmark functions we introduce a method of creating artificial low-fidelity fitness functions of the form Eq. (2.12) from preexisting benchmark functions. We know little about the approximation error of low-DCP functions in BRIGHT and thus define one that can be altered, allowing us to evaluate different kinds of $e(\mathbf{x})$.

We define the low-fidelity fitness function such that the value returned to the optimiser for objective i is

$$\hat{f}_i(\mathbf{x}) = f_{i,\text{true}}(\mathbf{x}) + e_i(\mathbf{x}) \quad (3.2)$$

with

$$e_i(\mathbf{x}) = \sum_{j=1}^{|\mathbf{x}|} [A \sin(\omega x_j + \varphi_i)] \quad (3.3)$$

where A is the amplitude, ω the frequency, and φ_i a phase offset. φ_i is drawn uniformly from $[0, 2\pi)$, such that each approximation of $f_{i,\text{true}}$ is different. A and ω need to be defined beforehand, where a large A constitutes a greater average error, and ω changes the shape of the error. A small ω would resemble a global shift of the true fitness function, and a large ω would constitute the introduction of many false local optima. Figure 3.5 illustrates what this e_i does to a fitness function, specifically the one-dimensional Sphere function, for different φ and ω at $A = 0.1$.

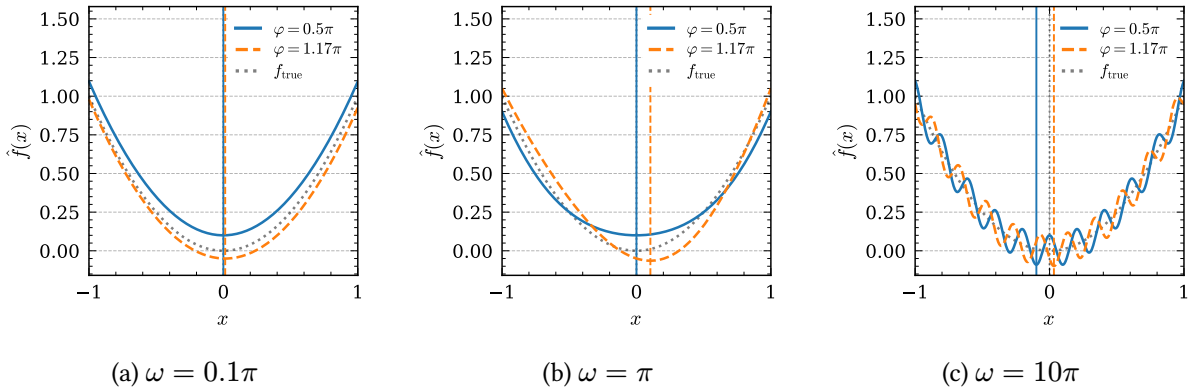


Figure 3.5: The one-dimensional Sphere function with e_i of Eq. (3.3) added, at an amplitude of $A = 0.1$. Each subplot uses a different frequency ω , and within a subplot each colour shows one of two phase offsets φ . The true fitness function is shown in grey. Thin vertical lines mark the global minimum of the function drawn in the same colour and line style.

The expected value of our $e_i(\mathbf{x})$, for φ_i uniformly randomly sampled from $[0, 2\pi)$ is 0, much like for $\mathcal{N}(0, \sigma_{\text{noise}}^2)$ which we used for studying [SRQ1](#). The distribution is however not normal, but resembles a U, or more specifically an arcsine distribution, as can be seen in Figure 3.6.

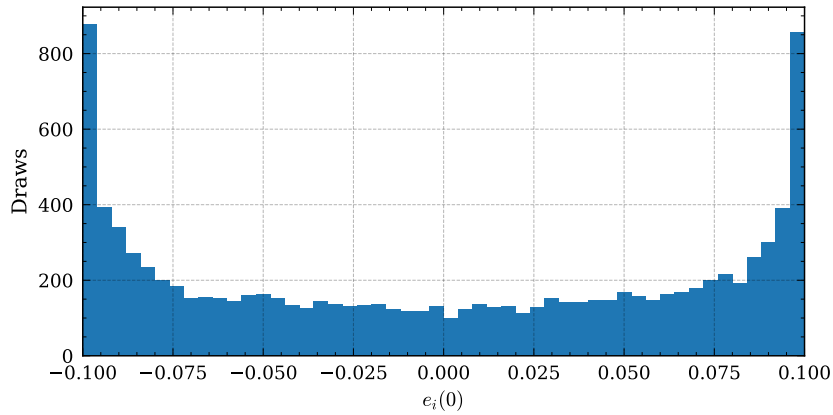


Figure 3.6: The value of the one-dimensional error $e_i(\mathbf{x}) = A \sin(\omega x + \varphi_i)$ at $x = 0$, over 10 000 phase offsets φ_i drawn uniformly from $[0, 2\pi)$, with $A = 0.1$ and $\omega = \pi$.

3.2.1 Convergence When Using a Low-Fidelity Fitness Function

We add $e_i(\mathbf{x})$ of Eq. (3.3) to the ZDT1, ZDT2, ZDT4, ZDT6 benchmark problems [36]. We investigate three types of runs: one with never changing offsets φ_i , one where they are redrawn every generation, and one where they are resampled for each evaluation. Note that the last of the three settings resembles that of Section 3.1, as the random variable, φ_i in our case, is re-drawn each evaluation.

We examine four amplitudes, $A \in \{0.001, 0.01, 0.05, 0.1\}$, and three frequencies, $\omega \in \{0.1\pi, 1\pi, 10\pi\}$. The optimiser settings are those of the MO experiment of Section 3.1: $l = 20$ decision variables, a population initialised uniformly in $[0, 1]^l$, a population size of 200, a univariate linkage model, 10 clusters, and an elitist archive with a target size of 1 000, under a budget of 10^7 evaluations. Quality is again measured by $D_{P_f \rightarrow S}$, computed on true fitness values against a reference set of 5 000 points from the Pareto front. Each configuration is repeated over 10 independent runs.

Figure 3.7 shows the true $D_{P_f \rightarrow S}$ over evaluations on ZDT1, with one panel per redraw frequency. The setting in which the offsets φ_i are never resampled converges fastest, and finds better solutions earlier on most median runs. The interdecile ranges are very large for the run with fixed offsets φ_i , and are smaller for the generationally changing φ_i and even smaller for the always different φ_i . $\omega = 10\pi$ for $A = 0.05$ and $A = 0.1$ performs poorly, with large interdecile ranges, on all three redraw schedules.

Figure 3.8 shows the same experiment for ZDT2, ZDT4 and ZDT6. Here we also observe that the interdecile ranges are larger for the setting with fixed offsets φ_i compared to the other redraw schedules. The median runs for the setting with the fixed offsets φ_i achieve better results early in the runs. On ZDT4, the median runs with the fixed offsets φ_i also converge to significantly better $D_{P_f \rightarrow S}$ than those with changing φ_i , about 3 orders of magnitude better.

Generally, the less often the low-fidelity fitness function is redrawn, the faster the median run converges. ZDT4 is a multi-modal problem, where an optimiser might get stuck on a false optimum. A draw of φ_i might be such that a local optimum appears as the true optimum. If $\hat{f}(\mathbf{x})$ changes each generation or evaluation, it becomes more likely that at least some solutions will end up in a false optimum, and nudge the optimiser to focus search in this false optimum. This is a possible explanation for why runs using a single $\hat{f}(\mathbf{x})$ do not get stuck early, as there only is one opportunity for an unfavourable draw of φ_i .

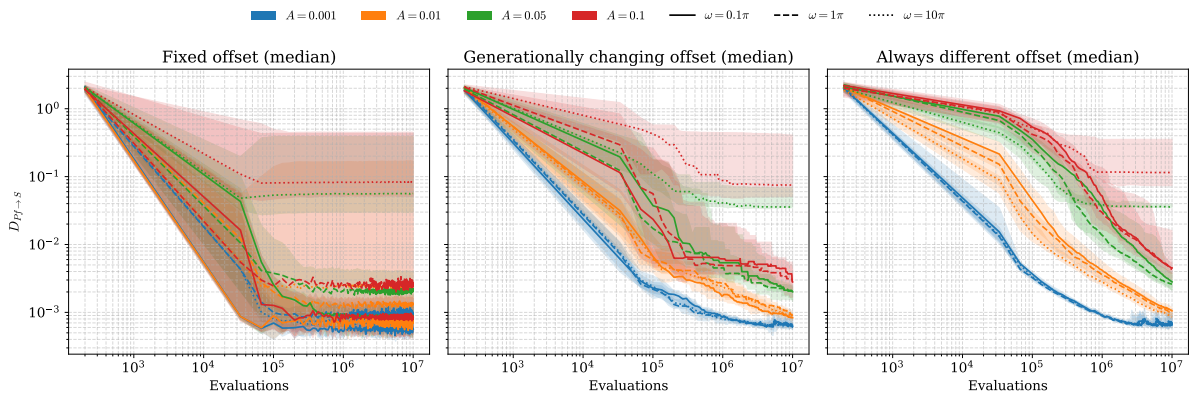


Figure 3.7: True $D_{P_f \rightarrow S}$ over evaluations for MO-RV-GOMEA on ZDT1 with $e(\mathbf{x})$ of Eq. (3.3) as added approximation error. Colours indicate amplitude A ; line style indicates frequency ω . Lines show the median and shaded bands show the interdecile range over 10 runs. Subplots each show how often the phase offsets φ_i are redrawn, which is never, once per generation, or at every evaluation.

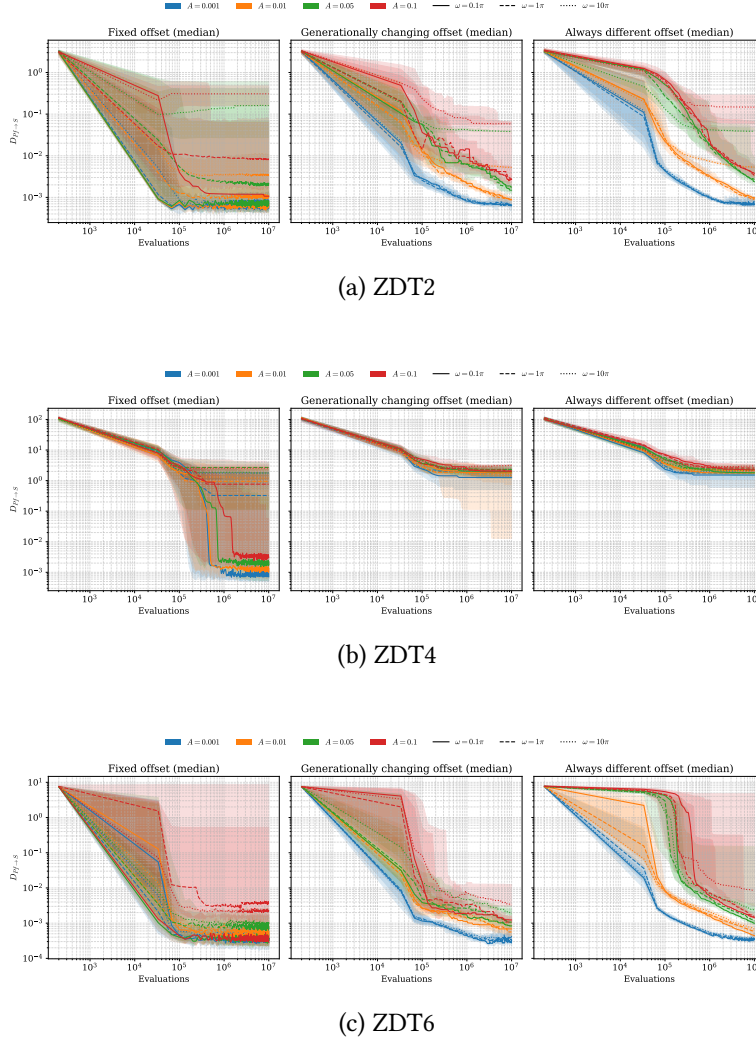


Figure 3.8: True $D_{P_f \rightarrow S}$ over evaluations for MO-RV-GOMEA on ZDT2, ZDT4, and ZDT6 with $e(\mathbf{x})$ of Eq. (3.3) as added approximation error. Colours indicate amplitude A ; line style indicates frequency ω . Lines show the median and shaded bands show the interdecile range over 10 runs. Subplots each show how often the phase offsets φ_i are redrawn, which is never, once per generation, or at every evaluation.

The optimiser sees no noise when the offsets are not redrawn, so the ordering which the low-fidelity fitness function induces is consistent between generations. Meaning, the optimiser is not misled from one generation to the next, which is why it can converge fastest in the setting where e_i is never redrawn. The price is that it converges to the optimum of that particular low-fidelity fitness function rather than to the true optimum, and which optimum that is depends on the offsets φ_i which happen to be drawn. This is what widens the variance between runs: the spread between the distance of the false optima to the true optimum.

ZDT4 is multi-modal and has many local Pareto fronts of its own, so a low-fidelity fitness function which changes gives the optimiser multiple chances to draw an approximation error that might make the optimiser get stuck in a local optimum. It seems that on the multi-modal ZDT4 problem, it is better to have one wrong low-fidelity fitness function, rather than a changing low-fidelity or noisy fitness function.

3.2.2 Conclusion

In order to answer [SRQ2](#), we added an approximation error $e(\boldsymbol{x})$ to the ZDT problems. We performed this experiment with different redraw schedules of $e(\boldsymbol{x})$, where it is drawn only once, every generation, or for every evaluation. Convergence is faster the less often $e(\boldsymbol{x})$ is redrawn, and reusing a single low-fidelity fitness function can reach better solutions faster. Runs where the low-fidelity fitness function is redrawn infrequently saw greater variance between runs. If consistency between runs is desired, changing the low-fidelity fitness function frequently is preferable. We can however make no statements about the computational cost, as this depends entirely on how expensive it is to redraw a low-fidelity fitness function frequently.

3.3 Conclusion

This chapter studied the two forms of uncertainty found in BRIGHT, in a controlled setting of benchmark problems. Section [3.1](#) modelled the use of a new set of DCPs for each evaluation with added noise. Section [3.2](#) modelled the reuse of DCPs for multiple evaluations with a function of $\boldsymbol{x}$ as an added approximation error. The two settings are the extremes of a single design choice, namely how often a new set of DCPs is drawn, and the second experiment of Section [3.2](#) also covered the middle-ground, where low-fidelity functions are redrawn but only after multiple evaluation. Under noise, progress halts at a distance from the optimum which increases with the level of noise. Before the halt of progress, the rate of improvement was the same for all noise levels in the SO setting, but not in the MO setting. The use of a small number of low-fidelity fitness functions allows for results as good as when the low-fidelity fitness function is redrawn more frequently, but variance between runs can be larger. Strategies such as reevaluations of the same solution to gain more certainty for fitness values which lie close to each other may have potential for non-parallelised settings, but are not feasible for BRIGHT because of its GPU-parallelised nature. We also have seen the multi-modal ZDT4 problem which showed that convergence happens at much worse solution qualities when using multiple low-fidelity fitness functions throughout the run, compared to using only one.

Two limitations should be kept in mind when carrying these results forward. The first is that the errors used here were constructed rather than measured, and the error which a finite set of DCPs induces in BRIGHT need not resemble a sinusoid in any way. The second is that computational cost was never part of the comparison, as every setting was given the same number of evaluations. In BRIGHT, resampling DCPs carries a significant cost. We therefore turn to BRIGHT in the next chapters.

Reduced Dose Calculation Resolution in BRIGHT

4

This chapter studies BRIGHT when treatment plans are optimised at different dose-calculation resolutions. In doing so, we answer [SRQ3](#).

SRQ3 How does reducing the number of dose-calculation points affect BRIGHT’s convergence, computation time and final plan quality?

Section [4.1](#) lays out the experimental setup used in this chapter. Section [4.2](#) takes an analytical approach to understand what happens when the number of DCPs is reduced. Section [4.3](#) then asks [SRQ3](#) directly, by giving runs at three DCP counts the same time budget. This chapter also sets the baseline which Chapter [5](#) later uses for its experiments.

4.1 Experimental Setup

Much of the experimental setup is shared by the BRIGHT experiments of this chapter and of Chapter [5](#): the patient data and clinical protocol, the configuration of BRIGHT, how results are reevaluated, the performance indicators used, and the hardware. Deviations from it are stated with the individual experiment. All experiments except those of Section [4.2](#) optimise Eq. (2.20) under the EMBRACE-II+SHAPE version. The cohort is the five patients labelled A to E, except in Section [5.3](#), which uses unseen patients labelled F to M for a final comparison.

BRIGHT is run with the multi-resolution scheme disabled throughout this thesis, so the number of DCPs is fixed within a run and is only changed to a higher value for a final reevaluation. The linkage model is a static linkage tree, set once by hierarchical clustering by the Euclidean distance between dwell positions, filtered to sets of size ≥ 5 . The elitist archive has a target size of 7 500. All runs use 12 clusters and a population size of 288. Runs are terminated on wall-clock time, except in Section [4.2](#) where an evaluation budget is used. The initialisation, which draws the DCPs and builds the dose-rate matrix of Eq. (2.16), and the reevaluation described below are both excluded from that budget, so a time limit only concerns the optimisation itself.

Four indicators are reported. These are all based on objective values that are (re)evaluated on 500 000 DCPs: the HV over the three objectives LCI_w , LSI_w and SHAPE (Eq. (2.20)) with reference point $r = [0.3, 0.3, 10]$, the L-score, the number of solutions in the golden corner, and the best SHAPE value in the golden corner. For patients which do not reliably reach the golden corner, the latter two performance indicators are replaced by the number of points where $LCI > -0.1 \wedge LSI > -0.1$, and the best SHAPE value among the plans in the same region.

Every BRIGHT run reported in this thesis was performed on a single NVIDIA TITAN Xp with 12 GiB of memory, in a machine with two Intel Xeon E5-2630 v4 processors and 125 GiB of system memory, under CUDA 12.9.

4.2 Overfitting

We start with an analysis with the goal of building more understanding of any noise and approximation errors found in BRIGHT. We record the difference between the fitnesses found at fewer DCPs versus a reference DCP set, we study the correlation between rankings induced by the small DCP set and the reference DCP set, and which DVIs decide LCI and LSI. Recording these statistics is expensive.

This experiment only considers one patient and performs runs with the EMBRACE-II protocol rather than EMBRACE-II+SHAPE. Van der Meer et al. [34] measure inaccuracy stemming from a reduced number of DCPs by the absolute difference between the DVIs computed at that resolution and those computed at a reference resolution. This is the natural quantity when the DVIs themselves are of interest, but, as Eq. (2.13) states, the magnitude of the approximation error is not a perfect measure for when algorithms like MO-RV-GOMEA start to overfit. This experiment therefore measures two things in a run: the difference between perceived and reference objective values, and the correlation between the ranking of the population induced by the perceived fitness versus the ranking of the population induced by reference objective values. The latter is measured in Kendall's τ . Kendall's τ counts the pairs of solutions two orderings agree on and subtracts those they disagree on, normalised by the number of pairs, so $\tau = 1$ means the orderings are identical. Kendall's τ can thus be used to measure how often Eq. (2.13) holds within a population. The number of DCPs is varied between runs and the reference fitness of plans is always computed using 1 000 000 DCPs.

Preliminary experiments also showed that uncertainty of the fitness values in BRIGHT stems from another source which is a rounding error that accumulates through the use of partial evaluations. Fitness values are updated, using partial evaluations, as follows:

$$\mathbf{f}_{\text{new}}(\mathbf{t}') = \mathbf{f}_{\text{old}}(\mathbf{t}) + \mathbf{f}_{\Delta}(\mathbf{t}, \mathbf{t}'). \quad (4.1)$$

Naturally, if $\mathbf{f}(\mathbf{t})$ is updated many times in this manner, any floating point rounding errors will accumulate. BRIGHT by default recomputes, without partial evaluations, all fitness values every 10 generations. We increase this frequency to observe whether this introduced noise has an effect on optimisation, and present the results in this chapter, rather than in the appendix, because a higher frequency of full recomputation is used in all later experiments. Our experiment includes the default frequency of every 10 generations, once every generation, and a complete black-box approach where partial evaluations are never used.

This experiment uses the general two-objective EMBRACE-II protocol, under which the SHAPE objective is not optimised for, and `bot-normal-tissue` and `mid-top-normal-tissue` of Table 2.1 are absent. The optimisation problem here is $\mathbf{f}(\mathbf{t}) = [\text{LCI}(\mathbf{t}), \text{LSI}(\mathbf{t})]$ and every front is two-dimensional. Only patient A is used. Runs terminate after 300 generations rather than on wall-clock budget, so all configurations perform the same amount of search and their wall-clock times can be compared. Two DCP set sizes, 20 000 and 200 000, and three frequencies of recalculating objective values are used. The black-box setting is one in which no partial evaluations are used, and all presented fitness values are based on a full (re)computation of the fitness of a solution at 1 000 000 DCPs.

Per generation and per objective the median of $f_n - f_{1M}$ over the population is computed, where n is the total number of DCPs used. Kendall's τ is computed between the orderings induced by f_n and f_{1M} . We only compute Kendall's τ for each population, and do not consider what happens in GOM for each individual comparison, as this is computationally very expensive to do for multiple runs. 30 runs are performed for each configuration.

Figure 4.1 shows the median of $\Delta f = f_n - f_{1M}$ against the generation for both LCI and LSI. Lines show the median of the median $f_n - f_{1M}$ per run, shaded regions show the interdecile ranges. All 20 000 DCP configurations settle between 0.002 and 0.005 in LCI and at roughly 0.010 in LSI, the 200 000 DCP runs at values much closer to 0. There is a clear tendency for $f_n - f_{1M}$ to be positive, especially when n is 20 000 instead of 200 000. This means that the median perceived fitness of individuals in each run is better than the fitness on the reference function f_{1M} . The interdecile ranges

are wider for fewer DCPs. Differences between the various reevaluation frequencies and the black-box setting are minimal in terms of median Δf .

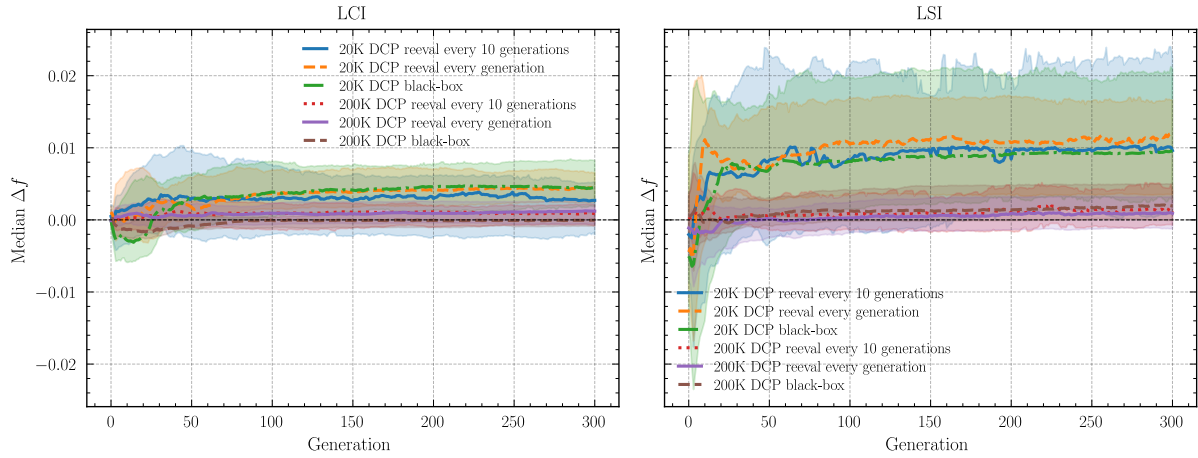


Figure 4.1: Median difference between the perceived objective value f_n , evaluated at n DCPs, and the reference value f_{1M} of the same plans, i.e. $f_n - f_{1M}$, over the generations of a run on patient A. Lines show the median over the population and shaded regions the interdecile range.

Figure 4.2 shows Kendall's τ between the orderings induced by f_n and f_{1M} . The settings which reevaluate every generation and those which use no partial evaluations (black-box) rise to $\tau \approx 0.995$ and stay there. Of these, the 200 000 DCP settings show higher Kendall's τ in the first 50 generations than 20 000 DCP configurations. The two settings which reevaluate every ten generations show a sawtooth pattern instead: Kendall's τ falls over the ten generations following a recomputation, to between roughly 0.90 and 0.94 for the median runs and jumps back up at each recomputation. The sawtooth pattern is similar for the 20 000 and 200 000 DCP configurations.

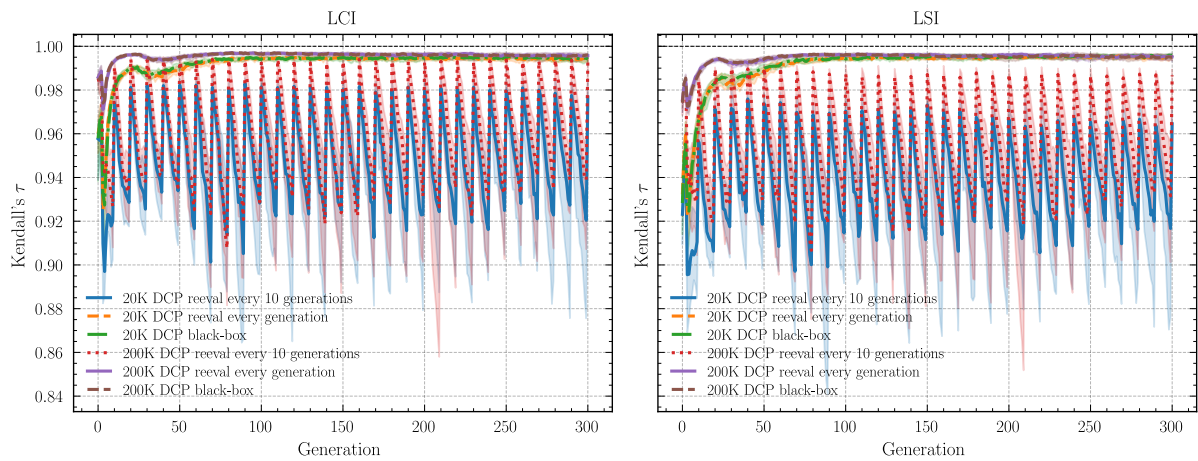


Figure 4.2: Median Kendall's τ between the perceived objective values f_n and the reference values f_{1M} over the population, per generation, for patient A. Lines show the median over the runs and shaded regions the interdecile range. A value of 1 means the run orders its population exactly as the reference does.

Figure 4.3 shows all solutions in all final approximation sets, reevaluated at 1 000 000 DCPs. The fronts for the configurations with more DCPs reach better solutions more consistently, whereas at 20 000 DCPs the scatter points form a thicker band, indicating that some runs produce fronts containing worse treatment plans in terms of LSI and LCI. A similar pattern is observed when comparing the

approximation fronts found by runs with the different full-reevaluation frequencies. The run with one reevaluation every 10 generations produces fronts containing treatment plans that are worse in terms of LSI and LCI. The impact seems to be so significant that the configuration using 20 000 DCPs without partial evaluations seems to perform as well as the configuration using 200 000 DCPs and the default reevaluation frequency of once every 10 generations. It is hard to judge whether there is a significant difference between reevaluating every generation versus not using partial evaluations at all.

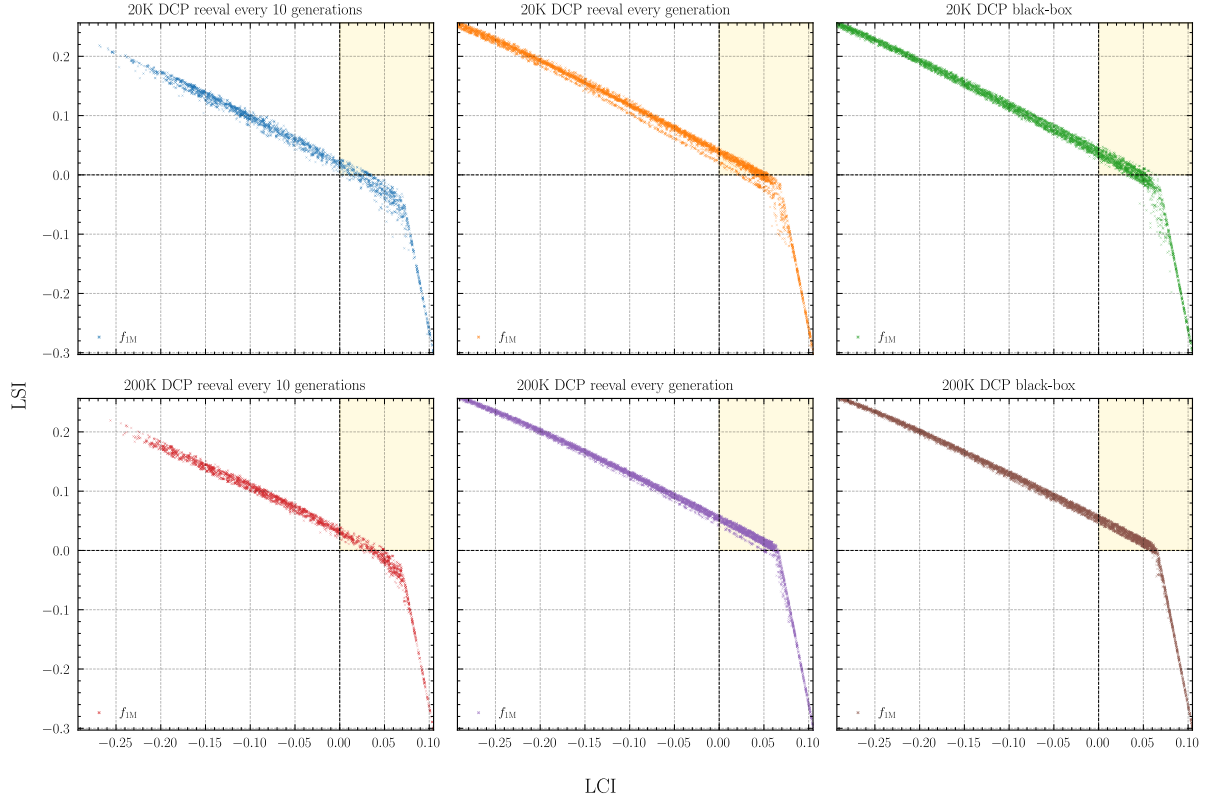


Figure 4.3: Final approximation sets, reevaluated at 1 000 000 DCPs, under the two-objective EMBRACE-II protocol on patient A. One panel per configuration, with all of its 30 final approximation fronts plotted.

Figure 4.4 reports the number of plans in the golden corner after reevaluation, the elitist archive size, and the duration of a run. Run durations are excluding initialisation time and reevaluation time. The configurations which reevaluate every 10 generations find a median of 3 and 5 plans in the golden corner at 20 000 and 200 000 DCPs, against 22 and 45 when recomputing every generation and 18 and 43 without partial evaluations; archive sizes follow the same pattern. A run of 300 generations takes about 1.3 min at 20 000 DCPs and 3.3 min at 200 000 DCPs. We again observe here that performance is worse for lower dose calculation resolutions and worse when plans are reevaluated only every 10 generations. Notable is that the black-box setting is not slower than the configurations that utilise partial evaluations.

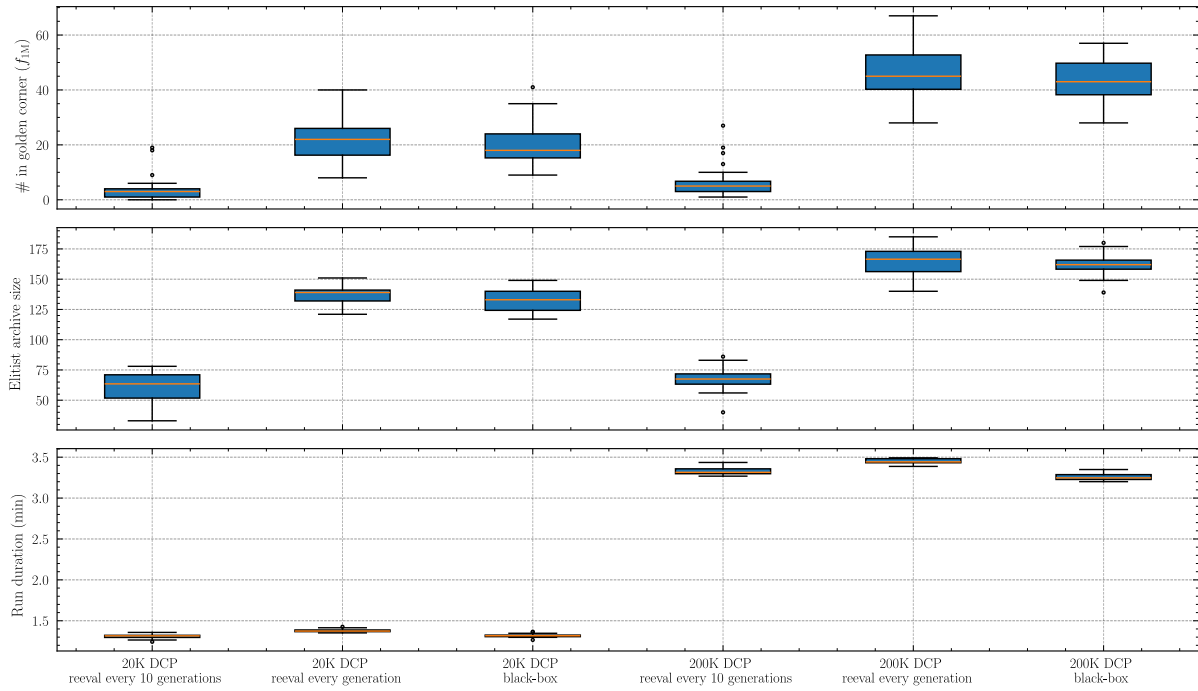


Figure 4.4: Number of plans in the golden corner after reevaluation at 1 000 000 DCPs, elitist archive size, and wall-clock duration of a run of 300 generations, per configuration, for patient A. Run durations are excluding initialisation time and reevaluation time.

Figure 4.5 reports for both LCI and LSI which DVI decides it, as a fraction of all individuals in all 30 runs. Little difference is observed between the studied configurations, other than a sawtooth pattern in the configurations which reevaluate only once every 10 generations. LCI is decided by the $D_{90\%}$ coverage aim on CTV_{HR} in essentially every evaluation of every configuration. LSI is decided by the $D_{2\text{cm}^3}$ aim on the bladder in roughly 85% of evaluations after the first fifty generations, and by the $D_{90\%}$ sparing aim on CTV_{HR} in nearly all of the remainder. Two DVIs therefore decide the objectives almost all of the time and suggest that, for this patient and these runs, the DCP resolution can maybe be lowered for those ROIs that never (or infrequently) decide LCI and LSI (and contribute little to LCI_w and LSI_w).

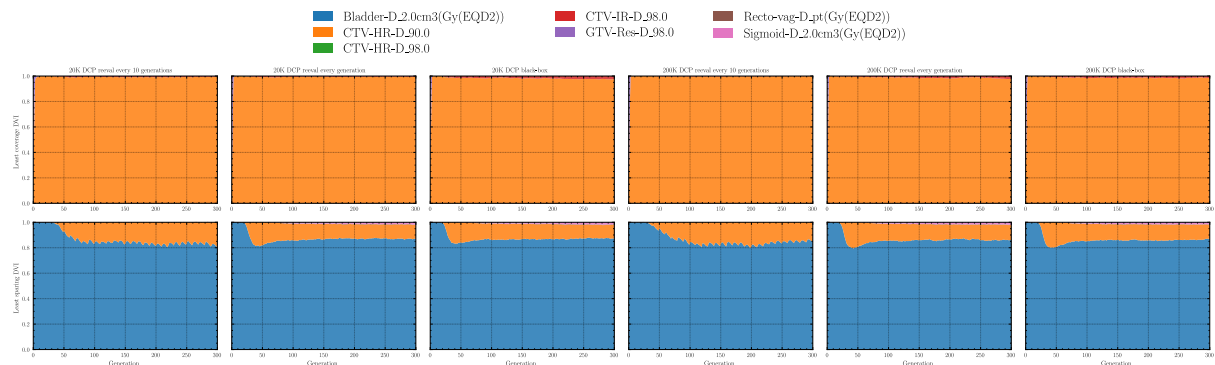


Figure 4.5: Fraction of the 30 populations whose LCI, respectively LSI, is decided by each aim of EMBRACE-II, against the generation, for patient A.

Although we only ran experiments for one patient, we have gathered insights concerning partial evaluations and overfitting behaviour at low DCP counts.

Judged by Kendall’s τ , the number of DCPs makes little difference, except for briefly early in the run in the first 100 out of 300 generations. The strong positive bias on $f_n - f_{1M}$ however is proof of overfitting: the difference is not just large, but there is a structural tendency for MO-RV-GOMEA to find solutions which seem better on f_n than they are on f_{1M} . This overfitting, as seen in Figure 4.1 already occurs in early generations. Overfitting occurring early in the run, at least for patient A, is an argument against a multi-resolution scheme where fewer DCPs are used early in the run.

The accumulated rounding error over multiple partial evaluations seems to introduce great uncertainty and be a large problem for optimisation. The errors can be avoided by simply recomputing the fitnesses more frequently, or by avoiding partial evaluations altogether. Surprisingly, not utilising partial evaluations is shown to not be disadvantageous to performance, likely due to the ability of the GPU to perform the matrix multiplication nearly as fast for the full dose-rate computation as for a partial one. It shows that partial evaluations might contribute little performance improvements when performing them on a GPU, as one would simply not fully saturate the GPU’s capacity, compared to CPUs and points to future research directions which are discussed in Section 7.1. We later discuss directions that may be of interest to future work regarding the surprisingly fast black-box setting, but provide a solution that we carry forward for the work of this thesis. It is inexpensive to simply recompute the dose at each DCP for each individual, without recomputing the DVIs. We thus keep using partial evaluations but fully recompute the dose at each DCP once every generation for all individuals. We use this strategy moving forward.

Three limitations of the experiment are identified. The first is that Kendall’s τ is computed over the population once per generation, whereas MO-GOM compares a solution against its own previous state and the archive at every FOS element, so the orderings which decide acceptance are finer-grained than those measured here. The second is the protocol. We optimised for the EMBRACE-II protocol, which only has the two LCI and LSI objectives, rather than the full EMBRACE-II+SHAPE version which includes the SHAPE objective. Findings concerning partial evaluations and the recomputation of the fitnesses hold, as the rounding error accumulates just the same for the different protocols and the computational cost of calculating the dose received at DCPs is also the same. This section’s comparison between the exact numbers 20 000 and 200 000 DCPs is however protocol-dependent. The third is that a single patient is used, so the observations of Figure 4.5 do not necessarily carry over to the other patients, nor to a protocol under which SHAPE might change which solutions are in the population.

4.3 Reduced Dose Calculation Resolution

Section 4.2 showed that overfitting occurs at a lower dose calculation resolution, at least on one patient. It examined runs with an evaluation budget, rather than a time budget. In reality we are interested in how fast good results can be obtained in terms of time rather than number of evaluations. We saw that the same number of generations can be covered in less than half the optimisation time if the number of DCPs is reduced. Possibly, this additional available time allows for finding similar or better plans given the same time budget.

In this experiment we perform runs using 10 000 DCPs, 50 000 DCPs and 250 000 DCPs. Five patients, A–E, are optimised under the EMBRACE-II+SHAPE protocol, with ten independent runs per setting. Approximation sets are recorded at ten wall-clock time limits, as if the algorithm was terminated at that moment: {0.5, 1, 2, 5, 10, 30, 60, 120, 240, 480} seconds. The approximation sets are always reevaluated at 500 000 DCPs. Graphs over time read as the quality a run would have delivered had it been stopped there. Time limits always exclude initialisation time and the reevaluation time.

The longest time limit is 480 seconds as this time limit was found to be sufficient to achieve the maximum treatment plan quality for all configurations, although the density of the fronts does seem to still improve after 480 seconds. The objectives are three-dimensional, but fronts are shown in LCI and LSI (or SHAPE and LCI) by plotting only the solutions non-dominated in the two plotted objectives. Patient E is reported against a relaxed corner for some of its performance metrics, $LCI, LSI > -0.1$, because not all of its runs reach the golden corner within 480 seconds.

Figure 4.6 shows the fronts found after 480 seconds in LSI over LCI and in SHAPE over LCI. The 10 000 DCP fronts perform, at least for some runs, clearly worse than the two higher resolution configurations. The 50 000 DCP setting performs worse than 250 000 DCP on at least patient B (SHAPE), patient C (LSI/LCI) and patient D (LSI/LCI).

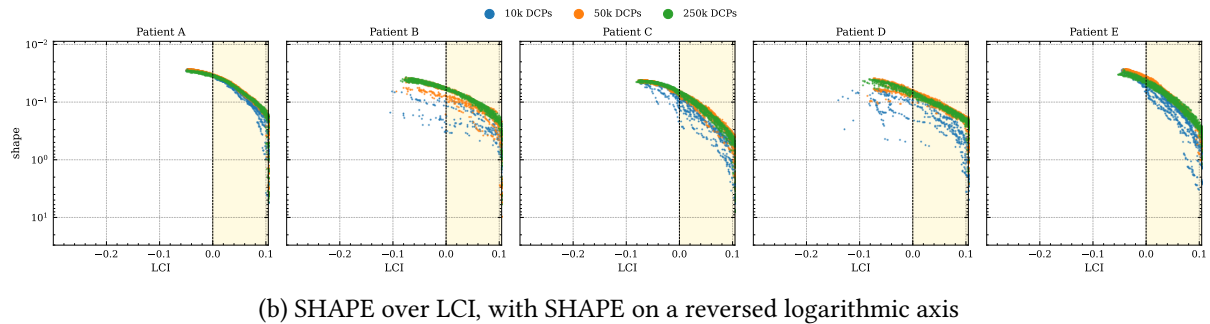
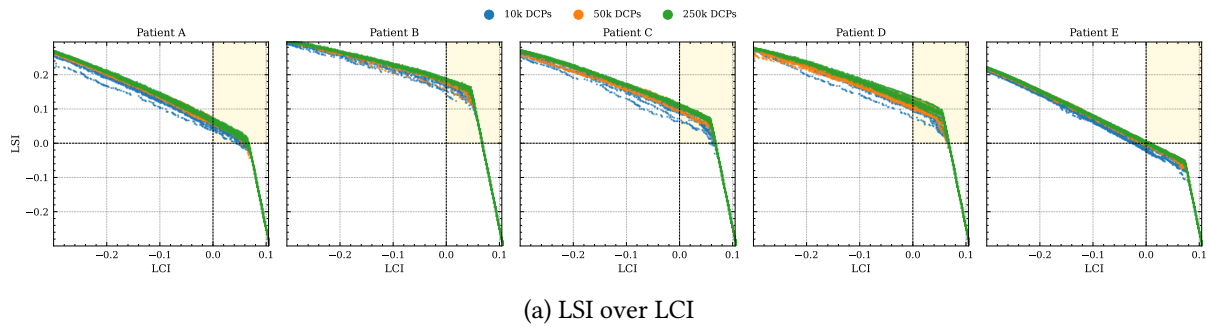


Figure 4.6: Approximation fronts at the 480 s time limit, for DCP sets of 10 000, 50 000 or 250 000. 10 runs of a configuration are overlaid and only the points which are non-dominated in their respective runs in the two plotted axes are shown. All values shown are those of a reevaluation at 500 000 DCPs. Figure A.3 and Figure A.4 show the fronts found after all recorded time limits.

Figure 4.7 shows four performance indicators against the time limit. HV and L-score show the same pattern for all five patients: the 10 000 and 50 000 DCP settings perform better than 250 000 DCPs early on. The three settings finish at 480 seconds on nearly the same HV and L-score for all patients. The other two indicators, best SHAPE in the golden corner and number of points in golden corner, do show clearer differences between the configurations at 480 seconds, with more DCPs performing better. Three out of five of the median 250 000 DCP runs finish with significantly more points in the golden corner than the median 50 000 DCP runs. The median best SHAPE in the golden corner at 480 seconds is similar between the two higher resolution settings.

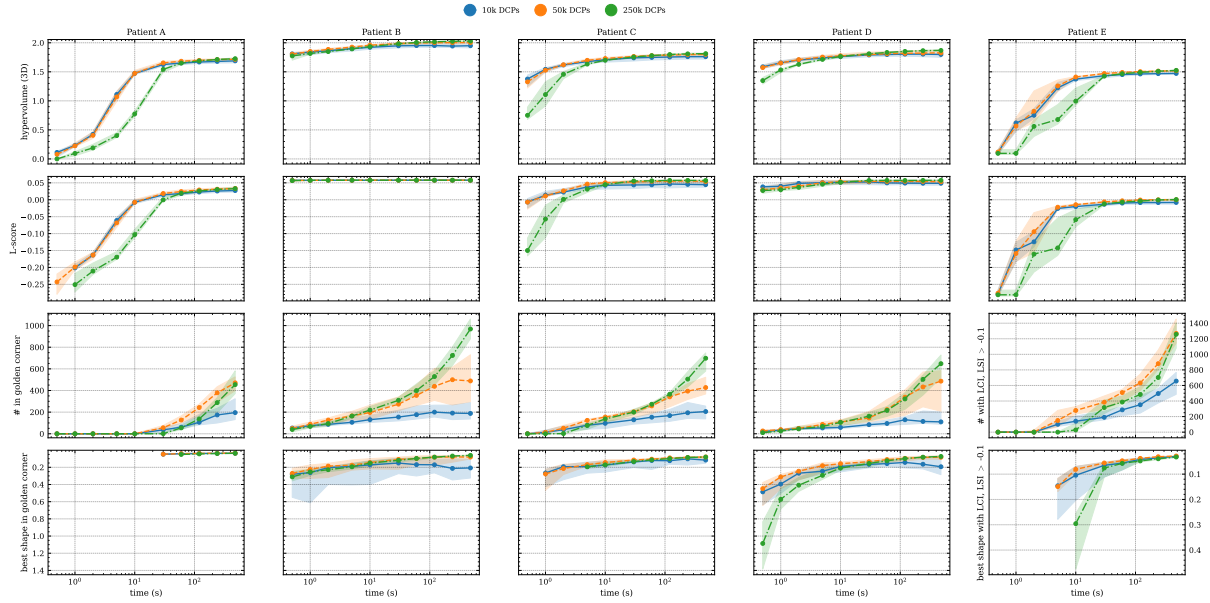


Figure 4.7: HV, L-score, number of plans in the golden corner, and best SHAPE among those plans, against the wall-clock time limit on a log scale for each patient. Lines show the median over 10 runs and shaded regions the interdecile range. All values shown are computed using fitnesses of reevaluations at 500 000 DCPs. Patient E is reported against a relaxed corner version of the golden corner: LCI, LSI > -0.1.

Figure 4.8 reports the time spent on initialisation. Initialisation takes between 10.5 and 13 seconds and differs by less than a second for different DCP set sizes on the same patient.

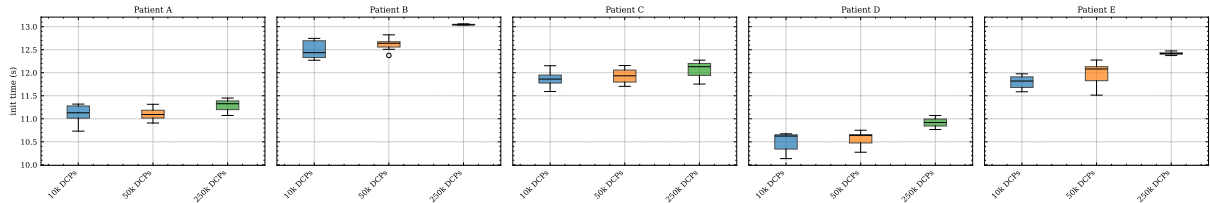


Figure 4.8: Time spent on initialisation, for runs using a single set of 10 000 DCPs, 50 000 DCPs and 250 000 DCPs.

We can conclude that using fewer DCPs for fitness evaluations throughout an optimisation run speeds up early progress but does not improve the final result and sometimes worsens it. At 50 000 DCPs, the HV and L-score found by runs using 250 000 DCPs are reached earlier, at wall-clock time limits of 10 seconds and shorter, but by the longest time limit that advantage is gone.

Using 10 000 DCPs clearly worse treatment plans are found: approximation fronts are seen to lie inside those of the other settings on most patients in Figure 4.6. The difference in front quality at 480 seconds is clearly visible in the number of solutions in the golden corner but not in the HV and L-score metrics. Approximation sets found using 50 000 DCPs are not worse than those found by 250 000 DCPs in terms of HV and L-score at every time limit, but contain fewer plans in the golden corner for three of the five patients. Possibly, the cheaper settings generally deliver a front equally good in plan quality, but with fewer plans to choose from. The three configurations evaluated here are carried forward as baselines to Chapter 5.

4.4 Conclusion

This chapter asked [SRQ3](#). Results show that runs using reduced dose calculation resolutions are able to find good solutions faster, yet, the final approximation fronts found are worse. Initialisation time is nearly unaffected by the DCP count at 10.5 – 13 seconds. Besides differences stemming from changes in dose-calculation resolutions the experiments of this chapter also revealed that a rounding error accumulates from the use of partial evaluations that is cleared, by recomputation of the fitnesses, too infrequently. Instead of the default recomputation of the fitness after every 10 generations, we continued with a recomputation every generation.

We have seen that using fewer DCPs can accelerate optimisation early on, yet overfitting also can start early. We therefore conclude that one should be cautious with using multi-resolution schemes where a lower dose-calculation resolution is used during the early stages of optimisation, because overfitting can occur early. Lastly, we have seen that specific DVIs, and thus specific ROIs, can dominate the LSI and LCI, which means that the majority of optimisation would focus on a subset of ROIs. Chapter 5 proposes both a scheme to dynamically use fewer DCPs in ROIs that are assigned a small weight in the objective values, and proposes a scheme which avoids overfitting to a single DCP set by performing fitness evaluations on different DCP sets throughout a run.

Mitigating Overfitting

Chapter 4 showed that a smaller set of DCPs makes every evaluation cheaper. Using fewer DCPs, BRIGHT can find better plans in the first 10 seconds, but worse plans are found later when compared to using more DCPs. In this chapter we propose two modifications to BRIGHT, which aim to mitigate overfitting while using fewer DCPs. In doing so, we answer SRQ4.

SRQ4 What strategies can be used to speed up optimisation using low-resolution dose calculations in BRIGHT, without loss of plan quality?

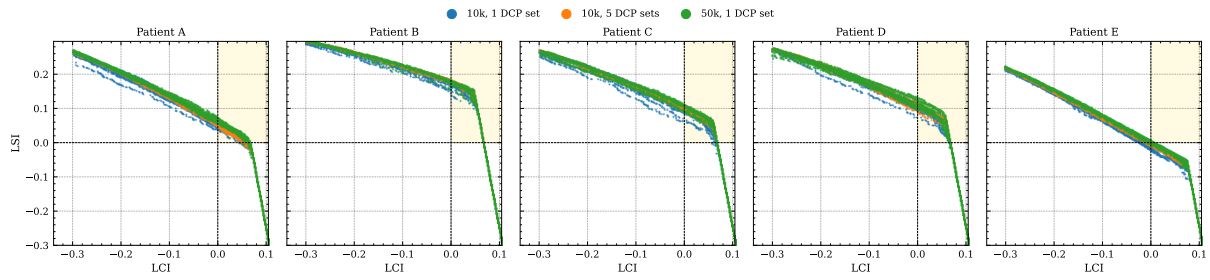
The proposed schemes are measured against the baseline set by Section 4.3, where the dose is recomputed after every generation to remove rounding errors accumulated by the use of partial evaluations. Section 5.3 closes this chapter by asking whether the plans found by two configurations of the proposed variants of BRIGHT, which showed good performance on Patients A through E, generate the same plans as the baseline with 250 000 DCPs on 8 unseen patients by using a γ analysis between the L-plans. The experimental setup used throughout this chapter is that described in Section 4.1.

5.1 Rotating Multiple Sets of Dose Calculation Points

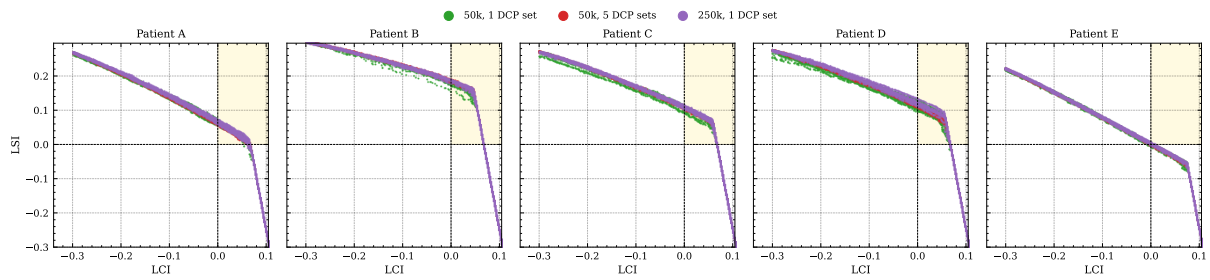
Section 4.3 shows what happens when a single small set of DCPs is used. Using a single set of DCPs aligns with the use of a single draw of the approximation error $e_i(\mathbf{x})$ in Subsection 2.3.2. In BRIGHT it is infeasible to use many different low-fidelity functions due to the cost of sampling a new DCP set. It is however feasible to rotate through a small number of independent DCP sets.

We modify BRIGHT as follows: N independent DCP sets are sampled at initialisation. In each generation g the DCP set used to evaluate fitness is indexed by $g \bmod N$. The whole population is evaluated against a single set within a generation and against a different one in the next. Each DCP set is given its own elitist archive: by doing so, the acceptance test inside MO-GOM (Algorithm 2.6, line 4) consults only the archive containing solutions evaluated on the same DCP set. This avoids testing the fitness of solutions against elitist solutions which were, supposedly, overfitted to another DCP set. At the end of a run every archive is reevaluated at 500 000 DCPs and the results are merged into one mutually non-dominated set, which is the approximation set reported. The per-evaluation cost does not depend on N : one set is used per evaluation regardless of the value of N . What a larger N does cost is memory, initialisation and reevaluation time, as N sets of dose rates are sampled, stored, and at the end all N elitist archives are reevaluated on the reference DCP set of size 500 000 before combining them into one final approximation set. Every configuration in this experiment uses either $N = 5$ or $N = 1$ with the latter simply being the baseline. We only use $N = 5$ for 10 000 and 50 000 as 5 sets of 250 000 DCPs exceeds the GPU's memory capacity.

Figure 5.1 shows the approximation fronts found after 480 seconds, reevaluated on 500 000 DCPs per patient. One set of 10 000 DCPs is seen to perform worse than rotating through 5 different DCP sets of 10 000 DCPs. Similarly, one set of 50 000 DCPs is seen to perform worse than rotating through 5 different DCP sets of 50 000 DCPs. The final fronts do not show a significant difference between 5 sets of 10 000 DCPs and 1 set of 50 000 DCPs, and between 5 sets of 50 000 DCPs and 1 set of 250 000 DCPs.



(a) 1 set of 10 000 DCPs, 5 rotating sets of 10 000 DCPs, and 1 set of 50 000 DCPs

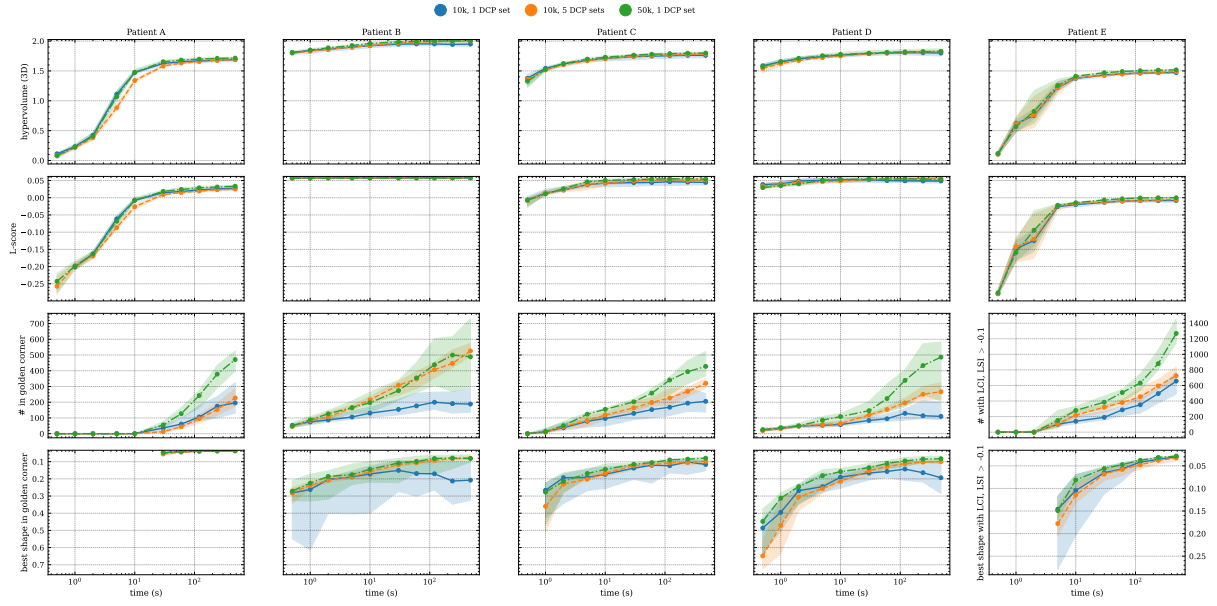


(b) 1 set of 50 000 DCPs, 5 rotating sets of 50 000 DCPs, and 1 set of 250 000 DCPs

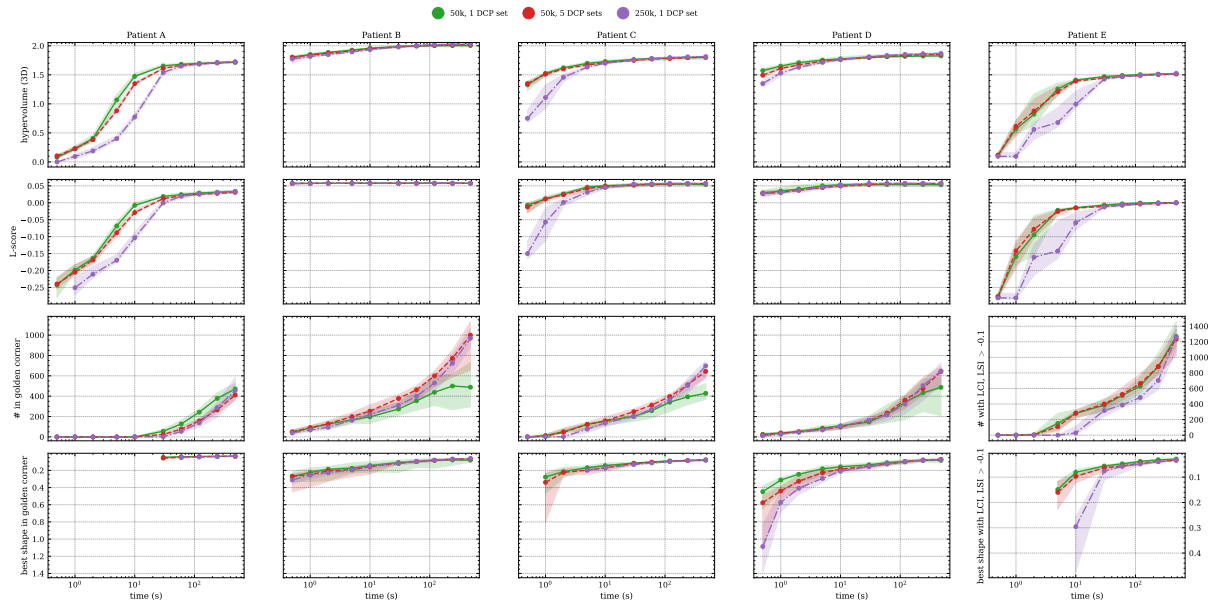
Figure 5.1: Approximation fronts at the 480 s time limit, for DCP sets of sizes 10 000, 50 000 and 250 000 at $N = 1$ or $N = 5$ rotating DCP sets. 10 runs of a configuration are overlaid and only the solutions which are non-dominated in their respective runs in the two plotted axes are shown. All values shown are those of a reevaluation using 500 000 DCPs.

Figure A.5 and Figure A.6 show the approximation fronts over time, all reevaluated using 500 000 DCPs, for LSI versus LCI, and SHAPE versus LCI respectively. We broadly observe two things: Better approximation sets are found when using 5 sets of k DCPs compared to using 1 set of k DCPs, and, notably, using 5 sets of 50 000 DCPs finds final approximation fronts as good as those found by 1 set of 250 000 DCPs, except for small differences in the SHAPE values of (some of the) approximation sets found on Patient E, as can be seen in the bottom right subplot of Figure A.6.

Figure 5.2 shows the HV, L-score, number of plans in the golden corner, and best SHAPE found in the golden corner over time. One set of 50 000 DCPs performs better in terms of the number of solutions found in the golden corner on 4 out of 5 patients compared to 5 rotating DCP sets of size 10 000. 5 sets of 50 000 DCPs show a similar number of solutions found in the golden corner as one set of 250 000 DCPs. 5 sets of 50 000 DCPs do not perform significantly worse than 250 000 at any performance metric at any time limit, and perform as good in HV and L-score as one set of 50 000 DCPs early on.



(a) $N = 5$ DCP sets of size 10 000, against $N = 1$ DCP set of size 10 000 and $N = 1$ DCP set of size 50 000



(b) $N = 5$ DCP sets of size 50 000, against $N = 1$ DCP set of size 50 000 and $N = 1$ DCP set of size 250 000

Figure 5.2: Five rotating DCP sets against a single set: showing HV, L-score, number of plans in the golden corner, and best SHAPE found in the golden corner, per patient.

Figure 5.3 shows the runtime breakdown of the performed runs. Using multiple DCP sets brings a large overhead, because N archives need to be reevaluated at the reference set of 500 000 DCPs. Reevaluation times are 2 to 3 times larger for $N = 5$ compared to the baseline. This difference is greater than any performance gain made by using multiple rotating DCP sets.

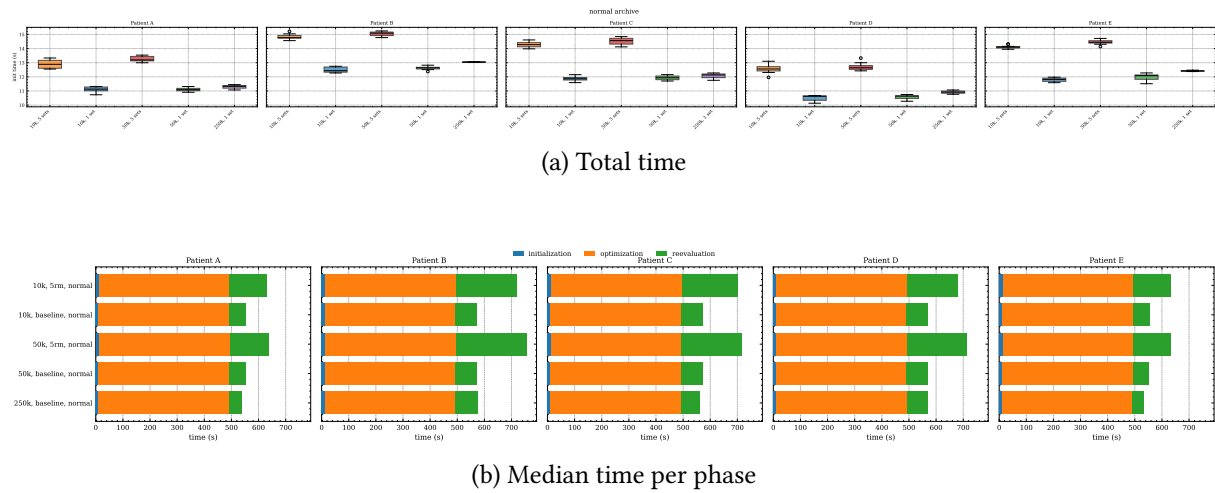


Figure 5.3: Five rotating DCP sets against a single set: Total run time and time per run phase.

The use of multiple smaller DCP sets, each with their own archive, is shown to find good treatment plans. Specifically, runs with $N = 5$ DCP sets of size 50 000 find fronts that are at least as good as $N = 1$ DCP set of size 250 000 at each recorded wall-clock time. This however excludes reevaluation time, which is significant and when included removes any benefits that $N > 1$ sets provide. We therefore next examine a different use of the elitist archive, and whether it can be removed altogether.

The elitist archive is used in two ways. Firstly, at the start of each generation, archive members are put back into the population (line 7 of Algorithm 2.5). Secondly, the elitist archive is used as an acceptance test for a change to an individual in GOM (line 4 of Algorithm 2.6). When rotating through multiple DCP sets, we now use multiple elitist archives such that in each archive we do not compare fitness values of evaluations on different DCP sets. Solutions that seem better on a given DCP set than they are in reality would still be stored in their respective elitist archives and be returned to the population. We experiment with removing either the first mechanism, or both mechanisms in which the elitist archive is used. We refer to the configuration where neither mechanism is active as `no archive` and only not copying the elites to the population at the start of each generation as `no elitism`. In `no elitism`, the archives of size 7 500 still exist, resulting in similar reevaluation times, while in the `no archive` setting the archives do not exist and thus also do not contribute solutions to the final approximation sets.

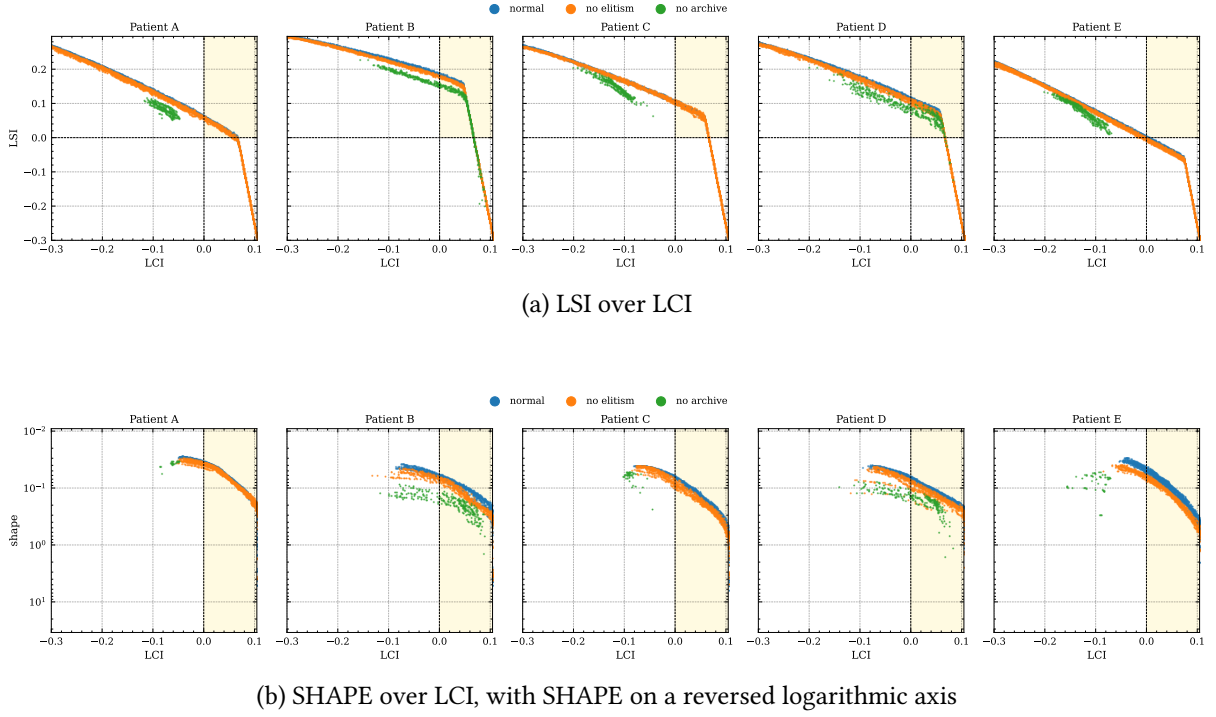


Figure 5.4: Approximation fronts at the 480 s time limit for $N = 5$ rotating DCP sets of size 50 000, with the normal elitist archive, without copying elites into the population (no elitism), and without an elitist archive (no archive). Treatment plans found by ten runs of each configuration are overlaid and only the plans which are non-dominated in the two plotted axes in their respective runs are shown. All values shown are those of a reevaluation using 500 000 DCPs.

Figure 5.4 shows the approximation fronts found after 480 seconds for the normal archive compared against no archive and no elitism. no archive performs much worse than the version with the default archive. no elitism performs better than no archive, but does not outperform the default archive, and most importantly, does not improve reevaluation time, as all $N = 5$ archives are still reevaluated.

5.2 Different Dose Calculation Resolution Per Region

Every experiment so far divides DCPs across ROIs equally. The objectives LCI_w and LSI_w are exponentially weighted sums over the aims (Eq. (2.18)), so the DVI furthest away from satisfying its aim carries nearly all of the weight, while others are assigned small weights. Figure 4.5 showed for one patient how monotone this can be throughout a run: LCI was decided by the $D_{90\%}$ aim on CTV_{HR} in nearly every evaluation and LSI by the D_{2cm^3} aim on the bladder in a vast majority. We thus hypothesise that fewer DCPs suffice for computing the DVIs of the ROIs which carry less weight in the objective function, and propose the *DVI subsampling* variant of BRIGHT.

The number of DCPs used to compute the DVIs of each ROI is determined anew every generation. The dose is still computed at all DCPs because otherwise the dose-rate matrix R would need to change every generation, creating additional overhead. We call the fraction of an ROI's DCPs which are used to compute its DVIs its *fidelity* φ . An ROI whose DVIs are never assigned the greatest weight in LCI_w nor LSI_w in any evaluation of the previous generation is assigned fidelity $\varphi = \varphi_{idle}$. All other ROIs

are assigned fidelity $\varphi_{\min} = 1$. We use $\varphi_{\text{idle}} = 0.25$. Algorithm 5.8 describes this mechanism using pseudocode.

Algorithm 5.8: Determine what fraction of the DCPs of each ROI to use for the computation of its DVIs, based on whether that ROI was assigned the greatest weight in LCI_w or LSI_w in the previous generation.

```

1: for ROI  $\in \mathcal{R}$  do
2:    $\varphi_{\text{ROI}} \leftarrow \varphi_{\text{idle}}$ 
3: end
4: for  $x$  evaluated during the previous generation do
5:   for  $i \in \{\text{LCI}_w, \text{LSI}_w\}$  do
6:      $a^* \leftarrow \arg \max_{a \in A_i} \delta_a(x)$  ▷ aim with the largest relative violation
7:      $\varphi_{\text{ROI}(a^*)} \leftarrow \varphi_{\min}$ 
8:   end
9: end

```

We compare DVI subsampling against the default configuration, all DCPs, using 50 000 and 250 000 DCPs. Figure 5.5 shows the approximation fronts found after 480 seconds and Figure 5.6 the mean fidelity per ROI over time. The fronts show very little difference between all DCPs and $\varphi_{\text{idle}} = 0.25$. The mean fidelity φ per ROI throughout a run is different between patients. For instance, the mean fidelity φ of the ROI bowel differs greatly between runs for the different patients.

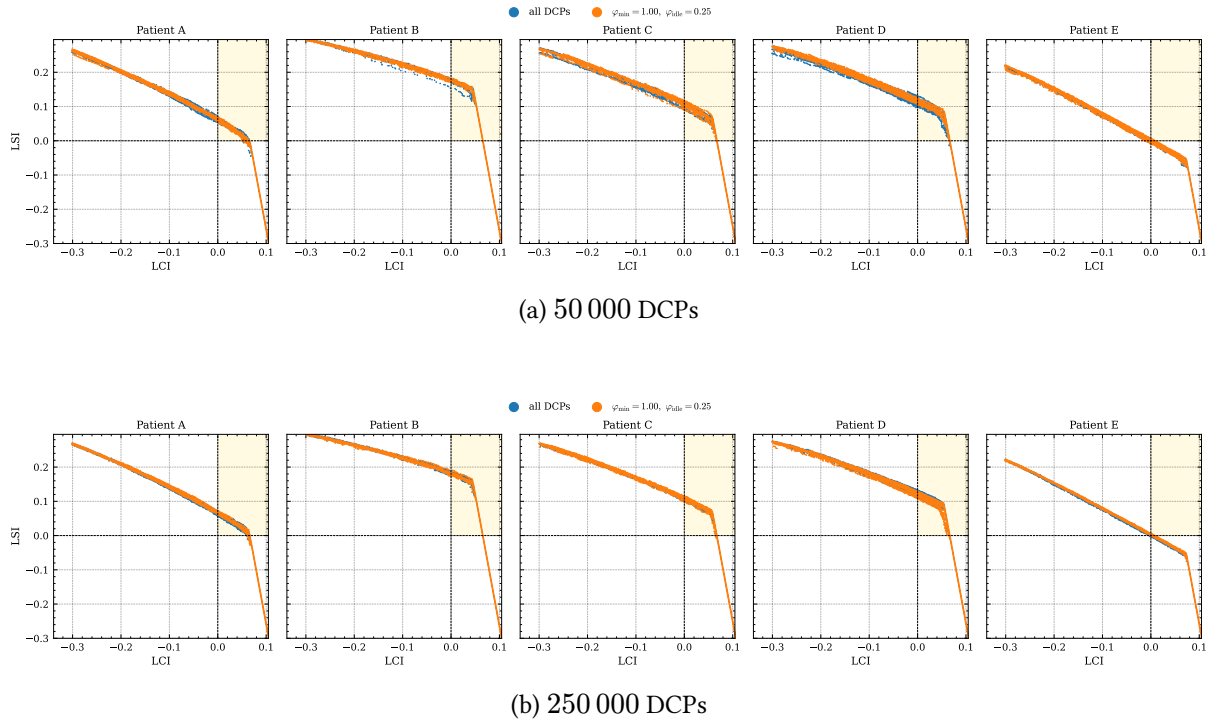


Figure 5.5: Approximation fronts found at the 480 s time limit by DVI subsampling with $\varphi_{\text{idle}} = 0.25$ (Algorithm 5.8), compared against the default configuration in which all DCPs are used for every ROI. Treatment plans found by ten runs of each configuration are overlaid and only the plans which are non-dominated in the two plotted axes in their respective runs are shown. All values shown are those of a reevaluation at 500 000 DCPs. The shaded region is the golden corner, in which all coverage and sparing aims are met.

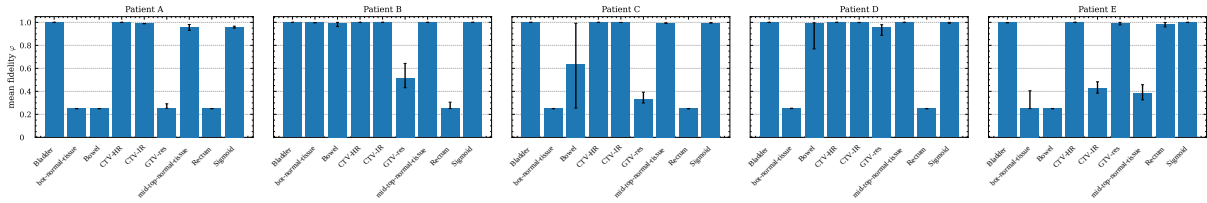


Figure 5.6: Mean fidelity φ per ROI over the generations, per patient, for runs using 50 000 DCPs with $\varphi_{\text{idle}} = 0.25$.

Using fewer DCPs in the ROIs of the DVIs which are assigned little weight in LCI_w and LSI_w neither significantly degrades nor significantly improves plan quality. No speed-up is observed either in terms of HV, L-score, number of plans in the golden corner and best SHAPE value, as can be seen in Figure A.7. DVI subsampling might however provide benefits in a less parallelised setting where no GPU is used, since the dose could then also be left uncomputed at the DCPs which are not used.

5.3 Comparing Dose Distributions

Every comparison of approximation sets made so far was in the objective space of BRIGHT: the DVIs and shape aims, summarised in the LCI_w , LSI_w and SHAPE objectives. Two plans which agree on these aims and objectives can still distribute dose differently in space. Clinicians also consider how dose is distributed when choosing between treatment plans. This section therefore evaluates whether the dose distributions of the treatment plans found by two variants of BRIGHT which were proposed in Section 5.1 and Section 5.2 are the same as those of plans found by the baseline. The γ tool, or γ analysis, is used to compare the planned dose distribution of two treatment plans [37]. A γ analysis takes as input the planned dose at points in a 3D grid for both the *reference plan* and the *evaluated plan*. Each grid point of the reference plan is compared with every grid point of the evaluated plan. Each comparison outputs a value determined by the dose difference, distance between points, and predefined tolerances for both. The γ value of one point in the reference plan is the smallest of all its comparisons with points in the evaluated plan. The point is said to pass if $\gamma \leq 1$. The γ pass-rate is the fraction of points that pass and is a measure of how similar two treatment plans are at the predefined tolerances.

Comparing full approximation sets using γ analyses would require many γ analyses, which is infeasible for this work for many runs. Instead we use a γ analysis between L-plans as a measure of similarity of two approximation sets. BRIGHT is stochastic, so the plans found by one run will always differ from those of another run, even if the same algorithm and parameters are used. We are therefore not solely interested in how much L-plans differ between runs of a baseline and a different variant of BRIGHT. What is instead of interest is whether the difference between L-plans of two runs of the baseline is smaller than the difference between L-plans of one found by the baseline and one found by another variant of BRIGHT. Finding that the γ pass-rates of baseline-baseline comparisons are similar to those of baseline-variant would suggest that the proposed variant of BRIGHT does find treatment plans with similar dose distributions as the baseline does.

We evaluate the baseline version of BRIGHT with 250 000 DCPs against the rotating DCP sets variant with $N = 5$ sets of 50 000 DCPs each, and against DVI subsampling with 50 000 DCPs and $\varphi_{\text{idle}} = 0.25$. We compare pairs of two baseline runs with pairs of one baseline run and one run of the variant being evaluated. A γ analysis is performed for the L-plans of each pair. Per patient we run 45

baseline runs and 15 runs of each of the two other BRIGHT versions, giving 15 baseline-baseline pairs and 15 pairs per variant.

The baseline run is always the reference plan in the γ analysis. The tolerances are set to 2 mm and 3% of the prescribed dose and the grid spacing is 2 mm along all three axes. Areas with an extremely high planned dose are those that lie close to the dwell positions; such points are always present and not considered by clinicians when selecting or comparing treatment plans. Similarly, areas with a very low planned dose, which often lie far away from the ROIs, are also rarely considered. For instance, the BRIGHT user interface allows only isodose surfaces to be shown between 25% and 400% of the prescribed dose. The γ analysis used here considers only points that receive between 20% and 500% of the prescribed dose in the reference plan.

Figure 5.7 shows γ pass-rates of the baseline-baseline compared with that of the baseline vs. DVI subsampling using 50 000 DCPs with $\varphi_{\text{idle}} = 0.25$. Similarly, Figure 5.8 shows the γ pass-rates of the baseline-baseline compared with that of the baseline vs. the rotating DCP sets variant with $N = 5$ rotating sets of 50 000 DCPs each. These figures give a visual impression of the differences but do not support any claim about whether the distributions are actually similar. Instead, statistical tests are needed.

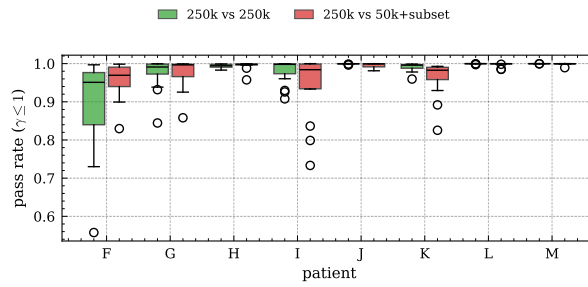


Figure 5.7: γ pass-rates per patient of the L-plans, for pairs of two baseline runs using 250 000 DCPs and for pairs of a baseline run of DVI subsampling using 50 000 DCPs and $\varphi_{\text{idle}} = 0.25$. The baseline run is the reference plan in each pair, and each group contains 15 pairs per patient. The tolerances are 2 mm and 3% of the prescribed dose.

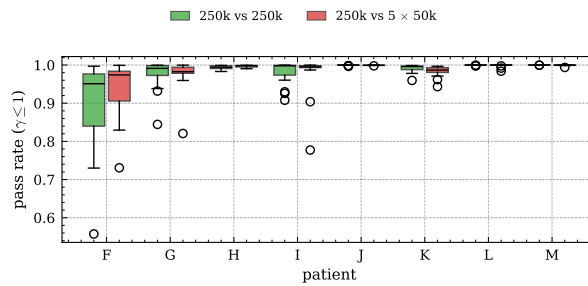


Figure 5.8: γ pass-rates per patient of the L-plans, for pairs of two baseline runs using 250 000 DCPs and for pairs of a baseline run with a run using $N = 5$ rotating DCP sets of 50 000 DCPs each. The baseline run is the reference plan in each pair, and each group contains 15 pairs per patient. The tolerances are 2 mm and 3% of the prescribed dose.

We have few samples for each group, do not assume normality, and assume the γ pass-rates to be independent because each run is used in only one pair. We therefore use a Mann-Whitney U test per patient to detect whether the baseline-baseline and the baseline-variant pairs differ [38]:

$$H_0 : \text{the two distributions are the same, } H_1 : \text{the two distributions differ.} \quad (5.1)$$

The test serves to answer whether for at least one patient the two algorithms find different L-plans. But the more patients we test, the more likely it becomes that we find such a difference by chance. We therefore apply a Bonferroni correction over the eight patients. Not rejecting H_0 of this test for difference does not imply the two distributions are the same, but rather that we have not found a difference. An equivalence test is used for testing whether two distributions are equivalent [39]. In our case only one direction matters: If the baseline-variant pairs γ pass-rates are higher than the baseline-baseline γ pass-rates then they simply fall within the run-to-run spread that the baseline already has. We therefore use a non-inferiority test, which is a one directional equivalence test, with a margin m :

$$H_0 : \Delta \geq m, \quad H_1 : \Delta < m. \quad (5.2)$$

Here Δ is the amount by which the γ pass-rates are lower for the baseline-variant pairs than the baseline-baseline pairs. The Hodges-Lehmann (HL) shift is the median of all pairwise differences between two sets of γ pass-rates. Rejecting H_0 of Eq. (5.2) would show that the proposed version loses less than m γ pass-rate with respect to the baseline. We are not interested in whether H_0 can be rejected for any patient, but whether it can be rejected for all patients. We therefore do not apply a correction such as a Bonferroni correction. What is left is the choice of m . There is no convention for how much two γ pass-rates may differ. TG-218 [40] does describe how to derive what acceptable γ pass-rates may be when comparing plans for intensity-modulated radiation therapy, which is a type of EBRT. Yet, we are considering a difference of γ pass-rates, instead of the γ pass-rate itself. It is out of the scope of this work to identify what the allowed clinical tolerance would be, but the test needs a margin, so we fix the margin at $m = 1\%$ γ pass-rate. This specific value is something we cannot defend and next to it we report $m_{\min}$ per patient, which is the smallest margin at which $\Delta < m$ would still hold for that patient, at 95% confidence. $m_{\min}$ lets the reader apply their own margin m .

Table 5.1 and Table 5.2 show the results for the DVI subsampling with a total initial DCP set of 50 000 DCPs, and Table 5.3 and Table 5.4 those for the five rotating DCP sets at DCP set size of 50 000. For the five rotating DCP sets we detect no difference on any of the eight patients. For the DVI subsampling we detect one, on Patient K, with a median difference in γ pass-rate of 1.21%. On the other seven patients no difference is detected.

On the test for non-inferiority the five rotating DCP sets have a largest $m_{\min}$ of 1.26%: they have an $m_{\min}$ of less than 1.3% on every tested patient. The DVI subsampling is worse on two patients: $m_{\min}$ is 5.44% on Patient I and 2.68% on Patient K.

Patient	250k vs 250k	250k vs 50k+subset	Shift (HL)	p	p_{Bonf}	Verdict based on p_{Bonf}
F	95.10	96.95	−2.14	0.089	0.712	none detected
G	99.12	99.66	−0.02	0.967	1.000	none detected
H	99.43	99.77	−0.20	0.056	0.451	none detected
I	99.80	98.41	0.41	0.340	1.000	none detected
J	99.98	99.92	0.04	0.020	0.162	none detected
K	99.60	98.27	1.21	0.0019	0.015	difference detected
L	100.00	99.97	0.02	0.168	1.000	none detected
M	99.99	99.98	0.01	0.097	0.776	none detected

Table 5.1: Mann-Whitney U test per patient for a difference between the two sets of γ pass-rates of Figure 5.7. The second and third column give the median γ pass-rate of each group in percent. p_{Bonf} is p after a Bonferroni correction over the eight patients, and the verdict is taken at a significance level of 0.05.

Patient	Shift (HL)	$m_{\min}$	p	Verdict at $m = 1$
F	-2.14	0.00	0.0113	$\Delta < 1$
G	-0.02	0.29	0.0190	$\Delta < 1$
H	-0.20	0.00	$4.8 \cdot 10^{-5}$	$\Delta < 1$
I	0.41	5.44	0.355	inconclusive
J	0.04	0.62	0.0018	$\Delta < 1$
K	1.21	2.68	0.630	inconclusive
L	0.02	0.04	$3.3 \cdot 10^{-4}$	$\Delta < 1$
M	0.01	0.03	$2.9 \cdot 10^{-5}$	$\Delta < 1$

Table 5.2: Non-inferiority test per patient (Eq. (5.2)) on the γ pass-rates of Figure 5.7. Δ is the amount by which the pass-rates of the pairs involving DVI subsampling are lower than those of the baseline-baseline pairs. $m_{\min}$ is the smallest margin at which $\Delta < m$ still holds at 95% confidence, and the verdict is taken at $m = 1\%$.

Patient	250k vs 250k	250k vs $5 \times 50k$	Shift (HL)	p	p_{Bonf}	Verdict based on p_{Bonf}
F	95.10	97.41	-1.79	0.263	1.000	none detected
G	99.12	98.24	0.27	0.590	1.000	none detected
H	99.43	99.82	-0.22	0.081	0.652	none detected
I	99.80	99.51	0.06	0.648	1.000	none detected
J	99.98	99.92	0.05	0.062	0.496	none detected
K	99.60	98.67	0.59	0.028	0.223	none detected
L	100.00	100.00	0.00	0.668	1.000	none detected
M	99.99	99.98	0.01	0.115	0.917	none detected

Table 5.3: Mann-Whitney U test per patient for a difference between the two sets of γ pass-rates of Figure 5.8. The second and third column give the median γ pass-rate of each group in percent. p_{Bonf} is p after a Bonferroni correction over the eight patients, and the verdict is taken at a significance level of 0.05.

Patient	Shift (HL)	$m_{\min}$	p	Verdict at $m = 1$
F	-1.79	0.46	0.0256	$\Delta < 1$
G	0.27	1.21	0.107	inconclusive
H	-0.22	0.00	$1.7 \cdot 10^{-6}$	$\Delta < 1$
I	0.06	0.39	$9.3 \cdot 10^{-4}$	$\Delta < 1$
J	0.05	0.07	$1.7 \cdot 10^{-6}$	$\Delta < 1$
K	0.59	1.26	0.181	inconclusive
L	0.00	0.00	$2.4 \cdot 10^{-5}$	$\Delta < 1$
M	0.01	0.06	$1.7 \cdot 10^{-6}$	$\Delta < 1$

Table 5.4: Non-inferiority test per patient (Eq. (5.2)) on the γ pass-rates of Figure 5.8. Δ is the amount by which the pass-rates of the pairs involving the rotating DCP sets are lower than those of the baseline-baseline pairs. $m_{\min}$ is the smallest margin at which $\Delta < m$ still holds at 95% confidence, and the verdict is taken at $m = 1\%$.

DVI subsampling reproduces the baseline less well than the rotating DCP sets do. It is impossible to determine whether either scheme reproduces the L-plans of the baseline without knowing what a clinically acceptable margin m of γ pass-rate would be. Additionally, any statements would only concern the L-plans, rather than the whole front, and these runs are all of the same wall-clock optimisation time. The results of Section 5.1 and Section 5.2 suggested that the benefit of the proposed strategies would be a speed-up during early optimisation. What we tested here is whether these strategies converge, when run until plan quality does not improve anymore, to the same plans. Further experiments would be required to see how well the proposed settings, given a smaller time budget, reproduce the plans found by the baseline.

5.4 Conclusion

This chapter concerned itself with answering SRQ4. It broadly proposed two methods of mitigating overfitting in BRIGHT when using low-resolution dose calculations. Rotating through small DCP sets, where one is used to determine the fitness for each generation, is shown to improve performance in terms of optimisation time compared to using a single small DCP set of the same size when keeping an elitist archive for each DCP set. Initialisation and especially reevaluation on the large reference DCP set is expensive when the number of small DCP sets N is increased. This makes for a total increase of runtime when using rotating DCP sets. Removing the archives altogether worsens results, and no improvements were observed when only not reintroducing solutions from the elitist archive in the population. A variant of BRIGHT where only a subset of DCPs is used for the computation of DVIs, based on which DVIs were the least sparing and least covering DVIs in the previous generation, was also evaluated. No significant speed-ups or improvements in plan quality were observed in our highly GPU-parallelised setting. No worsening of plan quality in terms of LCI, LSI and SHAPE was observed either. Using only a subset of DCPs might be of significant benefit in non-parallelised settings, where dose calculations are performed on a CPU.

In Section 5.3 we performed γ analyses in order to evaluate to what extent the approximation fronts found by the baseline could be reproduced using two of the proposed modifications. The L-plans produced by runs using $N = 5$ rotating DCP sets each of size 50 000, did not show statistically significant different dose distributions compared to the L-plans of the baseline, but determining whether these L-plans are at least as good is not possible without knowing what a clinically acceptable difference in γ pass-rate would be. Additionally, the experiment only considered the L-plans and a low number of patients, and can thus only provide suggestions for future research directions. Any more conclusive results would require evaluation of the found treatment plans by clinicians.

Discussion

The experiments of this work can be divided into experiments with benchmark problems (Chapter 3) and experiments in the HDR brachytherapy treatment planning setting of BRIGHT (Chapter 4 and Chapter 5). It is worth asking whether and how the results of Chapter 3 influenced the following chapters. Benchmark problems were used because experiments were easier to set up, faster to run, and any resulting conclusions may generalise to real-world problems other than HDR brachytherapy treatment planning. Noisy fitness functions were a proxy for the case where a new DCP set is used each evaluation, but such a setting was already not deemed likely to be effective in BRIGHT, as the sampling of a new DCP set for each evaluation is very costly. The experiments of Section 3.2 did model the setting of BRIGHT more closely, where a function $e(\mathbf{x})$, being the approximation error, could be re-used and only be re-sampled on demand much like a set of DCPs. Although noisy fitness functions and the use of surrogate fitness functions are each by itself common in real-world settings besides BRIGHT, a setting where defining the low-fidelity fitness function itself is a stochastic process, as modelled in Section 3.2, is not known to us in any other real-world problem. At the same time the used $e(\mathbf{x})$ was not shown to match the approximation error of BRIGHT, so insights of Section 3.2 do not necessarily carry over to BRIGHT. Greater value would possibly have lied in experimenting more in the treatment planning setting than general benchmark settings. This weakens the argument for using benchmark problems rather than the treatment planning optimisation problem itself in Section 3.2.

The comparison of approximation fronts of Section 5.3 does not allow for any strong conclusions: L-plans do not represent an entire front and it is unknown what a clinically acceptable margin for the difference of γ pass-rates is. The experiment only tested whether the same (given a certain margin) treatment plans as those found by the baseline are found using the two novel variants of BRIGHT after 480 seconds. To answer whether the same treatment plans as those found by the baseline can be found by the two novel variants of BRIGHT in a shorter period of time, the same experiment would need to be performed but with a shorter wall-clock budget for the novel variants of BRIGHT. Additionally, we never studied the effects of using a subset of the DCPs for calculating DVIs that contribute little to the objectives on its own, without changing the total number of available DCPs. Possibly, using the DVI subsampling approach with the same total number of DCPs does find the same treatment plans, and possibly, better results can be found using the DVI subsampling approach with a larger total number of DCPs but (approximately) the same effective number of DCPs. These questions now remain unanswered and more experiments containing γ analyses would be needed.

All HDR brachytherapy treatment planning experiments in this work were run on the GPU-parallelised version of BRIGHT. Different approaches, such as using different DCP sets for evaluations of individuals in the same generation, may be viable in less parallelised settings. Many findings, especially those of results expressed in terms of wall-clock time, are not transferable to a CPU-only setting of BRIGHT. However, the use of GPU parallelisation has been shown to significantly speed up BRIGHT [41] and GPUs are cheap compared to the cost of all equipment necessary for HDR brachytherapy, so it can be reasonably expected that most if not all applications of BRIGHT will use a GPU.

In practice BRIGHT is only run once for each HDR brachytherapy treatment. Convergence plots which show the median and interdecile range, as is common in the EA research community, may not be what we are interested in for clinical applications, as the worst case is of specific interest if only one

run is used per brachytherapy treatment. Before bringing any changes to BRIGHT to clinical practice, evaluation of treatment plan quality should consider the worst-case runs.

Conclusion

This thesis asked [RQ](#), with its four sub-questions:

RQ How can the use of low-resolution dose calculations speed up the optimisation of brachytherapy treatment planning using BRIGHT without loss of plan quality?

SRQ1 How does the use of noisy fitness functions affect MO-RV-GOMEA's convergence, computational cost and final solution quality?

SRQ2 How does the number of distinct low-fidelity fitness functions used during a run affect MO-RV-GOMEA's convergence, computational cost and final solution quality?

SRQ3 How does reducing the number of dose-calculation points affect BRIGHT's convergence, computation time and final plan quality?

SRQ4 What strategies can be used to speed up optimisation using low-resolution dose calculations in BRIGHT, without loss of plan quality?

Section [3.1](#) concerned itself with [SRQ1](#), where Gaussian noise was added to benchmark problems. Progress halted at some distance from the optimum which increases with the noise level. Progress also stopped at some distance from the optimum when non-elitist algorithms were used. In an SO setting the rate of progress over the number of evaluations was the same for different noise levels, until a halt of progress at a distance from the optimum dependent on the noise level. In the MO setting this rate of progress, until the halt at a distance from the optimum dependent on the noise level, was not the same for different levels of noise. The final solution quality of settings with more noise on the fitness evaluations were never able to reach a final solution quality that was as good as settings with less noise on the fitness evaluations. No statements about computational cost can be made, unless it is known how the cost of evaluations changes with the noise level, which is problem dependent and not defined in the benchmark setting.

Section [3.2](#) concerned itself with [SRQ2](#), where an approximation error $e(x)$ is added to the true fitness function of benchmark problems. $e(x)$ is redrawn at different frequencies throughout the optimisation runs. Convergence is faster in terms of number of evaluations when $e(x)$ is redrawn less frequently. Bigger variances in solution quality between runs were found when $e(x)$ was redrawn less frequently. For the benchmark problem ZDT4, where many local Pareto fronts exist, optimisers, when $e(x)$ is redrawn throughout the run, never found solutions that were as good as when $e(x)$ is never redrawn. No statements about computational cost can be made, unless it is known how the cost of evaluations changes with the noise level, which is problem dependent and not defined in the benchmark setting.

Chapter [4](#) studied what happened during the optimisation of HDR brachytherapy treatment plans when fewer DCPs were used and, in doing so, answered [SRQ3](#). Better solutions are found in the first 10 seconds when fewer DCPs are used, yet the final approximation fronts found are sometimes worse in terms of LCI, LSI and SHAPE when a total of 10 000 or 50 000 DCPs are used compared to when a total of 250 000 DCPs are used. The initialisation time is only affected minimally by the DCP count. Overfitting was shown to occur early in a run, suggesting that a high resolution of dose calculations is also necessary during the first generations.

Regardless of the number of DCPs used, rounding errors stemming from partial evaluations accumulate in the fitness values, and were shown to worsen the quality of the final approximation front. Additionally, never using partial evaluations was shown to be as fast as using partial evaluations. The implications of these finding, which are independent of the dose calculation resolution, are discussed below in Section 7.1.

Chapter 5 addresses SRQ4 and proposed variants of BRIGHT that aim to avoid overfitting at fewer DCPs. Rotating through five sets of in total 50 000 DCPs, each with their own elitist archive, seems to combine the benefits of fewer DCPs compared to 250 000 DCPs, while reducing or removing any (additional) overfitting. However, the benefit is not greater than the additional cost of reevaluating all individuals in all archives at the end of a 480 second run. Never reintroducing elitist individuals into the population provided no improvements in plan quality or computation time, and not using an elitist archive degraded plan quality significantly. Another proposed strategy is one where the GPU still computes the dose on each DCP, but only a subset of points is used to compute DVIs, specifically for the DVIs which were assigned a small weight in the objective function in the previous generation. This approach provided no significant performance gain, but also did not decrease plan quality in terms of LCI, LSI and SHAPE. Finally, Section 5.3 investigated whether two configurations (5 sets of 50 000 DCPs, and the DVI subsampling approach with a total of 50 000 DCPs) found similar treatment plans as the baseline at 250 000 DCPs in the same optimisation time of 480 seconds. The used γ analyses compare two plans and not entire approximation sets. The L-plans were used as representative plans for each front, and thus results do not concern entire approximation sets. 5 sets of 50 000 DCPs reproduced the baseline more closely than the DVI subsampling approach with a total of 50 000 DCPs. Without knowing what a clinically acceptable difference in γ pass-rate is, it is impossible to know whether the novel approaches can find the same plans as the baseline. We did not evaluate whether the novel approaches can find the same plans as the baseline in a shorter period of time, but such an analysis would also require knowing what difference in γ pass-rate is clinically acceptable.

Although using fewer DCPs allows for performing more evaluations in the same time, the quality of plans found within the same wall-clock time is not better after about 10 seconds of optimisation. Reducing the number of DCPs provides no significant speed-up after 10 seconds of optimisation time, likely due to the GPU's parallelisation capabilities, which allows for more DCPs at little additional cost. Notably for BRIGHT, we have seen that overfitting can occur early when a lower dose calculation resolution is used. These results argue against a multi-resolution scheme, which is currently used by default, where fewer DCPs are used during early optimisation.

Rotating through multiple DCP sets is costly in reevaluation, but adaptively reducing DCPs in ROIs that contribute little to the objectives comes at no additional cost and provides no decrease in quality of the final approximation front. Little benefit is found in using fewer DCPs, but using a subset of DCPs in some ROIs could instead be used to increase the effective dose calculation resolution.

The used γ analyses in Section 5.3 provide little insight into whether the approximation fronts of novel approaches consist of plans which are of the same clinical quality as the baseline provides, but the γ analyses do provide an additional tool to compare plan quality.

7.1 Future Work

Using a lower dose calculation resolution for HDR brachytherapy treatment planning using BRIGHT provides little benefit at the cost of possibly finding worse treatment plans. In a less parallelised CPU setting the benefits of using fewer DCPs may be greater, but research in such settings would need

to be motivated with a need to use BRIGHT without a GPU. Using only a subset of DCPs for some ROIs is shown to provide no or insignificant disadvantages in terms of LCI, LSI and SHAPE. Future work could explore whether DVI subsampling (Algorithm 5.8) could be used to increase the effective number of DCPs, reducing overfitting over the baseline. The dose-rate matrix $\mathbf{R}$ (Eq. (2.16)) grows with the number of DCPs and the population size. This work studied the number of DCPs. It may be of interest to also see how the population size changes optimisation performance. If, as we have seen, the parallelisation capabilities of the GPU are such that more DCPs come at little cost, then an increase in population size would, by this logic, also come at little cost. Compared to early work such as that of Luong et al. [42], BRIGHT has become very fast in dose calculations due to large-scale parallelisation. Overhead of the initialisation process has as a result increased in relative terms.

Observations were made throughout this thesis that do not concern the dose calculation resolution which hint at various new possible directions for future work. Firstly, the observation that, at least on one patient, partial evaluations provide no performance benefit is surprising. MO-RV-GOMEA is especially well suited to problems that lend themselves to partial evaluations, which HDR brachytherapy treatment planning does, but it seems that partial evaluations do not fully saturate the GPU making full evaluations practically as fast as partial ones. It may be worthwhile to perform comparisons of different algorithms in the current large-scale GPU-parallelised version of BRIGHT, similar to the comparison by Luong et al. [42].

Secondly, figures such as Figure A.3 show that many runs are already able to find plans in the golden corner within 1 second of optimisation. Patients for which this is the case show better final front quality after 480 seconds of optimisation. It seems that there is a correlation between the quality of the plans found after a few seconds and the quality of the plans found after a full run of, for instance, 480 seconds. Future work could explore how well the quality of plans found after short runs predicts the quality of the found treatment plans after longer runs. If successful, such predictions would enable performing short runs for many different hypothetical dwell positions, informing where the applicator/needles could best be placed.

References

- [1] S. Dauzère-Pérès, J. Ding, L. Shen, and K. Tamssaouet, ‘The Flexible Job Shop Scheduling Problem: A Review’, *European Journal of Operational Research*, vol. 314, no. 2, pp. 409–432, Apr. 2024, doi: 10.1016/j.ejor.2023.05.017.
- [2] R. Baños, F. Manzano-Agugliaro, F. G. Montoya, C. Gil, A. Alcayde, and J. Gómez, ‘Optimization Methods Applied to Renewable and Sustainable Energy: A Review’, *Renewable and Sustainable Energy Reviews*, vol. 15, no. 4, pp. 1753–1766, May 2011, doi: 10.1016/j.rser.2010.12.008.
- [3] A. Salo, M. Doumpos, J. Liesiö, and C. Zopounidis, ‘Fifty Years of Portfolio Optimization’, *European Journal of Operational Research*, vol. 318, no. 1, pp. 1–18, Oct. 2024, doi: 10.1016/j.ejor.2023.12.031.
- [4] D. H. Wolpert and W. G. Macready, ‘No Free Lunch Theorems for Optimization’, *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, Apr. 1997, doi: 10.1109/4235.585893.
- [5] T. Koch, T. Berthold, J. Pedersen, and C. Vanaret, ‘Progress in Mathematical Programming Solvers from 2001 to 2020’, *EURO Journal on Computational Optimization*, vol. 10, p. 100031, Jan. 2022, doi: 10.1016/j.ejco.2022.100031.
- [6] Y. Sherry and N. C. Thompson, ‘How Fast Do Algorithms Improve? [Point of View]’, *Proceedings of the IEEE*, vol. 109, no. 11, pp. 1768–1777, Nov. 2021, doi: 10.1109/JPROC.2021.3107219.
- [7] T. Back, U. Hammel, and H.-P. Schwefel, ‘Evolutionary Computation: Comments on the History and Current State’, *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 3–17, Apr. 1997, doi: 10.1109/4235.585888.
- [8] A. Mukherji, ‘An Introduction to Brachytherapy’, *Basics of Planning and Management of Patients during Radiation Therapy: A Guide for Students and Practitioners*. Springer, pp. 171–182, Mar. 2018. doi: 10.1007/978-981-10-6659-7_13.
- [9] L. R. M. Dickhoff *et al.*, ‘End-User Validation of BRIGHT with Custom-Developed Graphical User Interface Applied to Cervical Cancer Brachytherapy’, 2026, doi: 10.48550/ARXIV.2602.16321.
- [10] D. L. J. Barten *et al.*, ‘Towards Artificial Intelligence-Based Automated Treatment Planning in Clinical Practice: A Prospective Study of the First Clinical Experiences in High-Dose-Rate Prostate Brachytherapy’, *Brachytherapy*, vol. 22, no. 2, pp. 279–289, Mar. 2023, doi: 10.1016/j.brachy.2022.11.013.
- [11] A. Bouter, N. H. Luong, C. Witteveen, T. Alderliesten, and P. A. N. Bosman, ‘The Multi-Objective Real-Valued Gene-Pool Optimal Mixing Evolutionary Algorithm’, in *Proceedings of the Genetic and Evolutionary Computation Conference*, in GECCO ’17. Association for Computing Machinery, July 2017, pp. 537–544. doi: 10.1145/3071178.3071274.
- [12] A. Bouter, T. Alderliesten, B. R. Pieters, A. Bel, Y. Niatsetski, and P. A. N. Bosman, ‘GPU-accelerated Bi-Objective Treatment Planning for Prostate High-Dose-Rate Brachytherapy’, *Medical Physics*, vol. 46, no. 9, pp. 3776–3787, June 2019, doi: 10.1002/mp.13681.
- [13] R. Bellman, ‘Dynamic Programming’, *Science*, vol. 153, no. 3731, pp. 34–37, July 1966, doi: 10.1126/science.153.3731.34.
- [14] G. B. Dantzig, *Linear Programming and Extensions*. Princeton University Press, 2016. Accessed: May 20, 2026. [Online]. Available: <https://www.torrossa.com/en/resources/an/5793009>
- [15] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, ‘Optimization by Simulated Annealing’, *Science*, vol. 220, pp. 671–680, May 1983, doi: 10.1126/science.220.4598.671.
- [16] P. Ngatchou, A. Zarei, and A. El-Sharkawi, ‘Pareto Multi Objective Optimization’, in *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*, Nov. 2005, pp. 84–91. doi: 10.1109/ISAP.2005.1599245.

- [17] N. Riquelme, C. Von Lücken, and B. Barán, ‘Performance Metrics in Multi-Objective Optimization’, in *2015 Latin American Computing Conference (CLEI)*, Oct. 2015, pp. 1–11. doi: 10.1109/CLEI.2015.7360024.
- [18] M. Črepinšek, S.-H. Liu, and M. Mernik, ‘Exploration and Exploitation in Evolutionary Algorithms’, *ACM Computing Surveys*, vol. 45, no. 3, pp. 1–33, July 2013, doi: 10.1145/2480741.2480752.
- [19] P. A. N. Bosman, J. Grahl, and D. Thierens, ‘Benchmarking Parameter-Free AMaLGaM on Functions With and Without Noise’, *Evolutionary Computation*, vol. 21, no. 3, pp. 445–469, Sept. 2013, doi: 10.1162/EVCO_a_00094.
- [20] D. Thierens and P. A. N. Bosman, ‘Optimal Mixing Evolutionary Algorithms’, in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, in GECCO ’11. Association for Computing Machinery, July 2011, pp. 617–624. doi: 10.1145/2001576.2001661.
- [21] A. Bouter, T. Alderliesten, C. Witteveen, and P. A. N. Bosman, ‘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*, in GECCO ’17. Association for Computing Machinery, July 2017, pp. 705–712. doi: 10.1145/3071178.3071272.
- [22] A. Bouter, ‘Optimal Mixing Evolutionary Algorithms for Large-Scale Real-Valued Optimization: Including Real-World Medical Applications’, Delft, The Netherlands, 2023. doi: 10.4233/uuid:0e03913c-898e-4392-8de5-072a7ead7fd6.
- [23] Y. Jin and J. Branke, ‘Evolutionary Optimization in Uncertain Environments-A Survey’, *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, June 2005, doi: 10.1109/TEVC.2005.846356.
- [24] P. Rakshit, A. Konar, and S. Das, ‘Noisy Evolutionary Optimization Algorithms – A Comprehensive Survey’, *Swarm and Evolutionary Computation*, vol. 33, pp. 18–45, Apr. 2017, doi: 10.1016/j.swevo.2016.09.002.
- [25] A. Syberfeldt, A. Ng, R. I. John, and P. Moore, ‘Evolutionary Optimisation of Noisy Multi-Objective Problems Using Confidence-Based Dynamic Resampling’, *European Journal of Operational Research*, vol. 204, no. 3, pp. 533–544, Aug. 2010, doi: 10.1016/j.ejor.2009.11.003.
- [26] P. Limbourg, ‘Multi-Objective Optimization of Problems with Epistemic Uncertainty’, in *Evolutionary Multi-Criterion Optimization*, C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, Eds, Springer, 2005, pp. 413–427. doi: 10.1007/978-3-540-31880-4_29.
- [27] J. Teich, ‘Pareto-Front Exploration with Uncertain Objectives’, in *Evolutionary Multi-Criterion Optimization*, E. Zitzler, L. Thiele, K. Deb, C. A. Coello Coello, and D. Corne, Eds, Springer, 2001, pp. 314–328. doi: 10.1007/3-540-44719-9_22.
- [28] Y. Jin, ‘A Comprehensive Survey of Fitness Approximation in Evolutionary Computation’, *Soft Computing*, vol. 9, no. 1, pp. 3–12, Jan. 2005, doi: 10.1007/s00500-003-0328-5.
- [29] Y. Jin, ‘Surrogate-Assisted Evolutionary Computation: Recent Advances and Future Challenges’, *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, June 2011, doi: 10.1016/j.swevo.2011.05.001.
- [30] H. Wang, Y. Jin, and J. Doherty, ‘A Generic Test Suite for Evolutionary Multifidelity Optimization’, *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 6, pp. 836–850, Dec. 2018, doi: 10.1109/TEVC.2017.2758360.
- [31] L. Mainini *et al.*, ‘Analytical Benchmark Problems and Methodological Framework for the Assessment and Comparison of Multifidelity Optimization Methods’, *Archives of Computational Methods in Engineering*, Nov. 2025, doi: 10.1007/s11831-025-10392-8.

- [32] M. J. Rivard *et al.*, ‘Update of AAPM Task Group No. 43 Report: A Revised AAPM Protocol for Brachytherapy Dose Calculations’, *Medical Physics*, vol. 31, no. 3, pp. 633–674, Feb. 2004, doi: 10.1118/1.1646040.
- [33] L. R. M. Dickhoff *et al.*, ‘Keeping Your Best Options Open with AI-based Treatment Planning in Prostate and Cervix Brachytherapy’, *Brachytherapy*, vol. 23, no. 2, pp. 188–198, Mar. 2024, doi: 10.1016/j.brachy.2023.10.005.
- [34] M. C. van der Meer *et al.*, ‘Sensitivity of Dose-Volume Indices to Computation Settings in High-Dose-Rate Prostate Brachytherapy Treatment Plan Evaluation’, *Journal of Applied Clinical Medical Physics*, vol. 20, no. 4, pp. 66–74, Mar. 2019, doi: 10.1002/acm2.12563.
- [35] P. A. N. Bosman, ‘Source Code’. Accessed: Aug. 09, 2026. [Online]. Available: https://homepages.cwi.nl/~bosman/source_code.php
- [36] E. Zitzler, K. Deb, and L. Thiele, ‘Comparison of Multiobjective Evolutionary Algorithms: Empirical Results’, *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, June 2000, doi: 10.1162/106365600568202.
- [37] D. A. Low and J. F. Dempsey, ‘Evaluation of the Gamma Dose Distribution Comparison Method’, *Medical Physics*, vol. 30, no. 9, pp. 2455–2464, Aug. 2003, doi: 10.1118/1.1598711.
- [38] A. Petrie and C. Sabin, *Medical Statistics at a Glance*. John Wiley & Sons, 2019.
- [39] S. Wellek, *Testing Statistical Hypotheses of Equivalence and Noninferiority*. CRC Press, 2010.
- [40] M. Miften *et al.*, ‘Tolerance Limits and Methodologies for IMRT Measurement-Based Verification QA: Recommendations of AAPM Task Group No. 218’, *Medical Physics*, vol. 45, no. 4, pp. e53–e83, Feb. 2018, doi: 10.1002/mp.12810.
- [41] A. Bouter, T. Alderliesten, A. Bel, C. Witteveen, and P. A. N. Bosman, ‘Large-Scale Parallelization of Partial Evaluations in Evolutionary Algorithms for Real-World Problems’, *Proceedings of the Genetic and Evolutionary Computation Conference*. in ACM Conferences. pp. 1199–1206, July 02, 2018. doi: 10.1145/3205455.3205610.
- [42] N. H. Luong, T. Alderliesten, A. Bel, Y. Niatsetski, and P. A. N. Bosman, ‘Application and Benchmarking of Multi-Objective Evolutionary Algorithms on High-Dose-Rate Brachytherapy Planning for Prostate Cancer Treatment’, *Swarm and Evolutionary Computation*, vol. 40, pp. 37–52, June 2018, doi: 10.1016/j.swevo.2017.12.003.

Appendix

A.1 Normalisation of Dose-Volume Indices Stated in EQD2

DVI aims can be stated either as a physical dose or as EQD2. The aims which are used in BRIGHT for prostate HDR brachytherapy planning are expressed in physical dose [12], [42]. δ_a is defined such that a plan which delivers no dose at all yields $\delta_a = 1$ for every sparing aim, for aims expressed in physical dose (Eq. (2.17)). EQD2 is defined such that it always includes the EBRT contribution. Figure A.1 shows the conversion from physical dose to EQD2 for a sparing aim, which assumes an EBRT contribution of 43.2 Gy and four brachytherapy fractions.

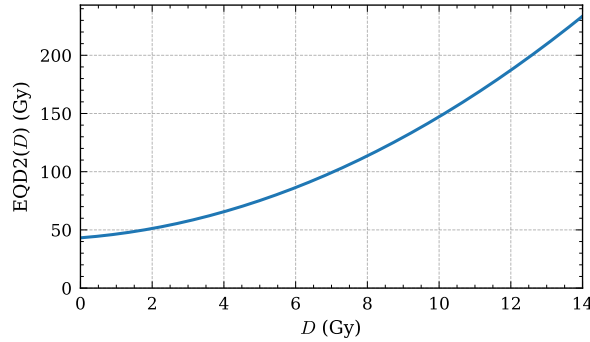


Figure A.1: The EQD2 of a DVI against its physical dose in Gy per fraction for a sparing aim, which assumes an EBRT contribution of 43.2 Gy and four brachytherapy fractions.

The conversion between physical dose and EQD2, if the computation of δ_a is left unchanged, changes what the extremes of δ_a mean. A plan which has no planned dose at all still has an EQD2 of 43.2 Gy, so δ_a no longer reaches -1 but is -0.55 for the aim of 95 Gy(EQD2) on CTV_{HR} and -0.34 for the aim of 65 Gy(EQD2) on the rectum. Each δ_a thus has its own lower bound, even though LCI_w and LSI_w compare those values across aims.

The conversion can also rank two aims one way when they are normalised on the physical dose and the other way when they are normalised on the EQD2, as Figure A.2 shows. Since the aims are weighted exponentially by the rank of their violation, such an inversion changes how LCI_w and LSI_w are composed, and thus what the optimiser optimises.

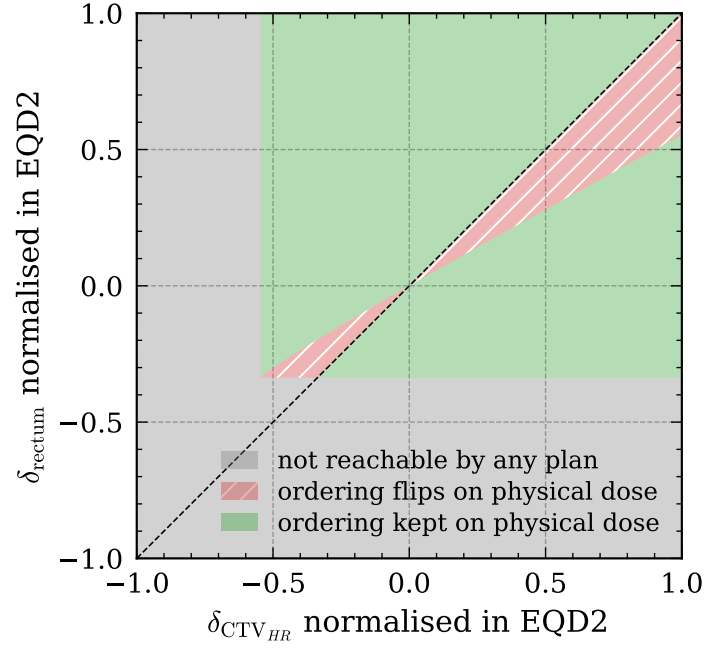


Figure A.2: Every pair of values of δ_a (Eq. (2.17)) for two sparing aims of Table 2.1, $D_{90\%} < 95$ Gy(EQD2) on CTV_{HR} and $D_{2\text{cm}^3} < 65$ Gy(EQD2) on the rectum, as BRIGHT computes them, normalised in EQD2. The grey region cannot be reached by any plan, since the EBRT contribution alone already puts each aim at -0.55 and -0.34 respectively. Pairs in the hatched region would be ranked the other way round if the aims were normalised on the physical dose, which is what decides the weights of Eq. (2.18). The dashed diagonal is where the two violations are equal.

This thesis uses the normalisation of Eq. (2.17) throughout, as earlier work with BRIGHT does, but it may be important to note that LCI_w , LSI_w , LCI and LSI carry a slightly different meaning than they would for aims stated in physical dose expressed in Gy.

A.2 Approximation Fronts over the Full Time-Limit Sweep

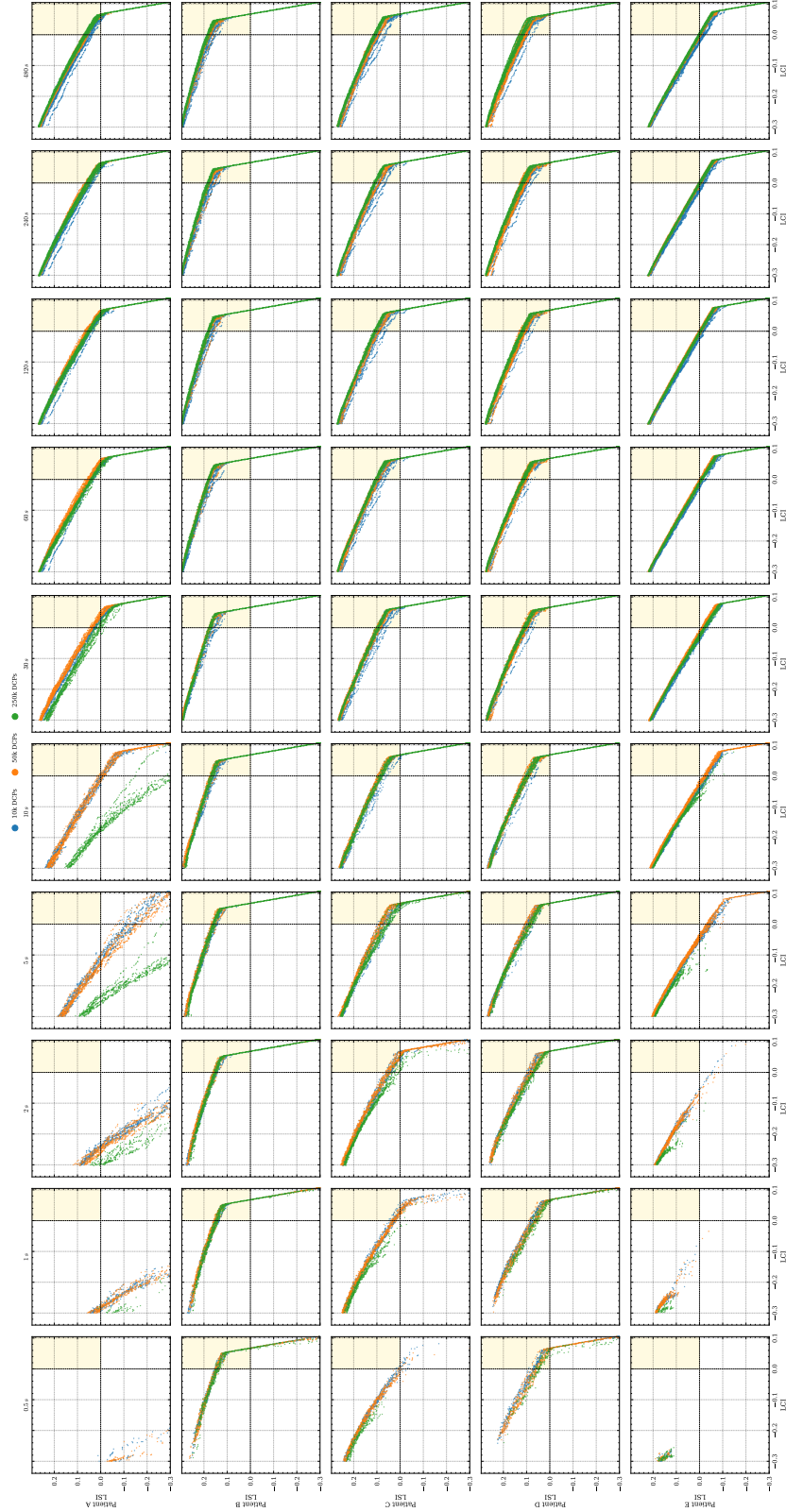


Figure A.3: Approximation fronts of LSI over LCI over the full time-limit sweep, for 10 000 DCPs, 50 000 DCPs and 250 000 DCPs, with patients as rows and time limits as columns. Treatment plans found by ten runs of each configuration are overlaid. Every plan is reevaluated at 500 000 DCPs. The shaded region is the golden corner, in which all coverage and sparing aims are met. Only plans are shown which are non-dominated in LSI and LCI in their respective runs. The 480 s column is also shown in Figure 4.6(a).

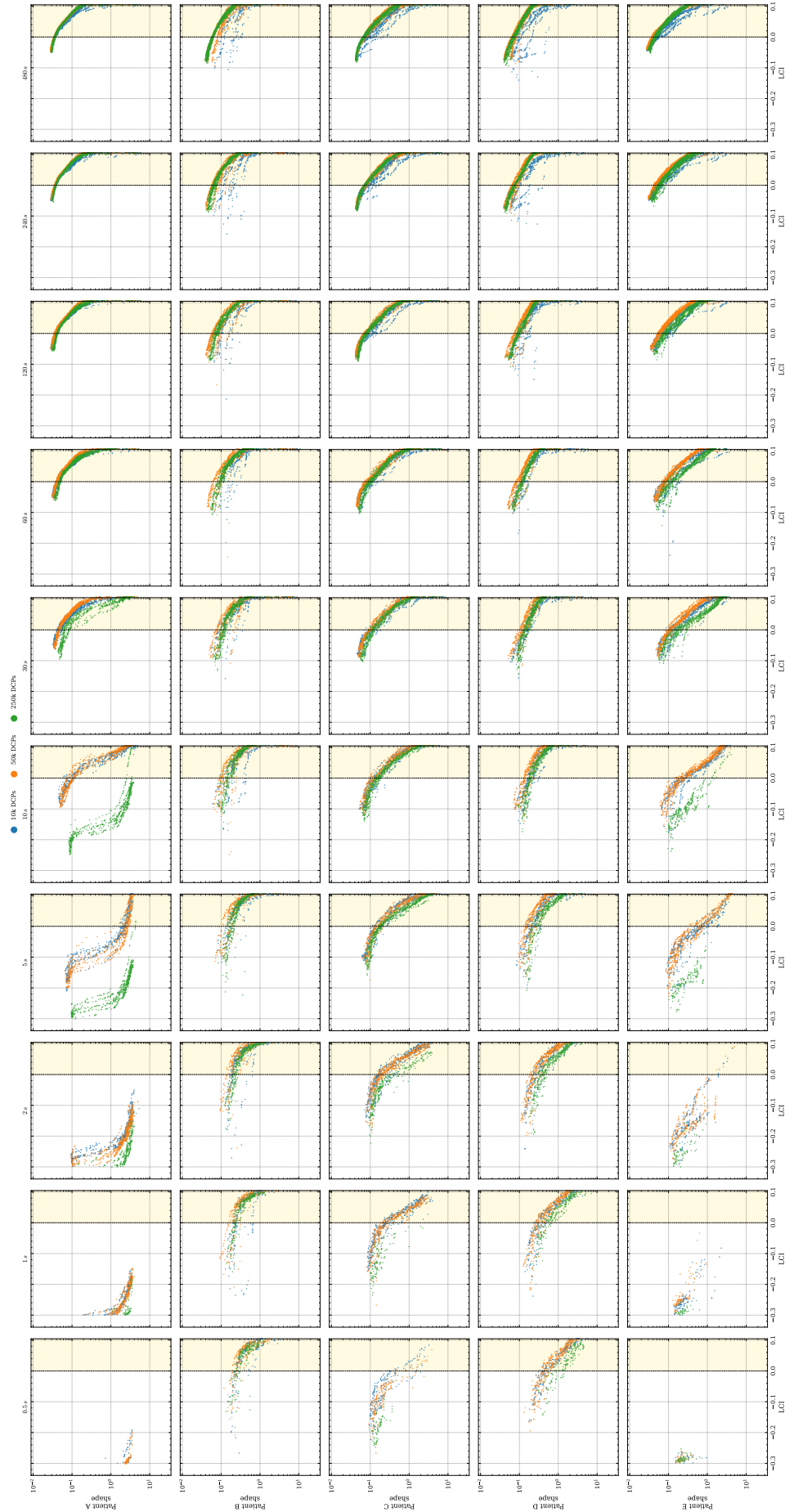


Figure A.4: Approximation fronts of SHAP over LCI, with SHAP on a reversed logarithmic axis, over the full time-limit sweep, for 10 000 DCPs, 50 000 DCPs and 250 000 DCPs, with patients as rows and time limits as columns. Treatment plans found by ten runs of each configuration are overlaid. Every plan is reevaluated at 500 000 DCPs. The shaded region is where all coverage aims are met. Only plans are shown which are non-dominated in SHAP and LCI in their respective runs. The 480 s column is also shown in Figure 4.6(b).

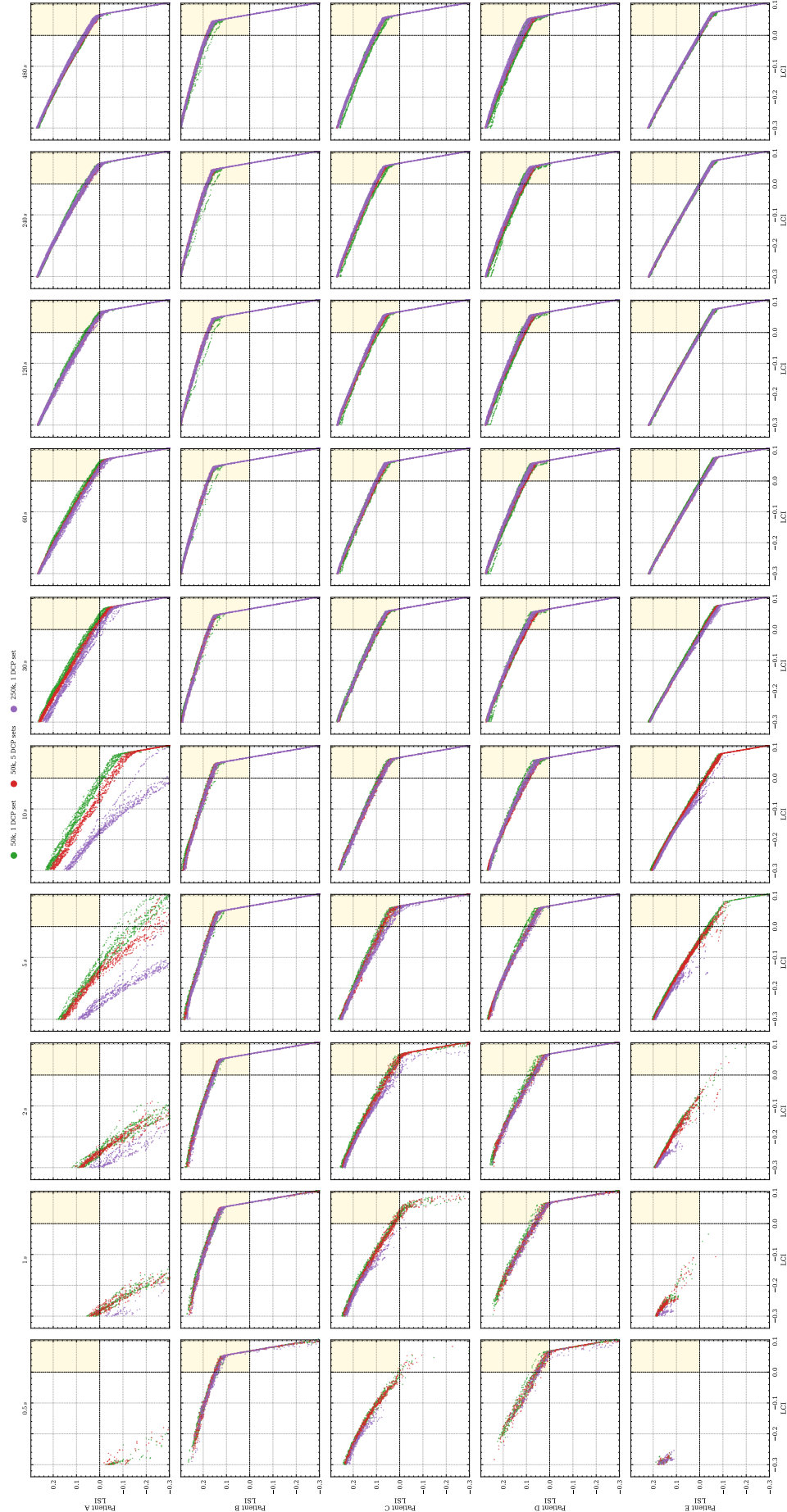


Figure A.5: Approximation fronts of LSI over LCI, over the full time-limit sweep, for 1 set of 50 000 DCPs, 5 rotating sets of 50 000 DCPs and 1 set of 250 000 DCPs, with patients as rows and time limits as columns. Treatment plans found by ten runs of each configuration are overlaid. Every plan is reevaluated at 500 000 DCPs. The shaded region is the golden corner, in which all coverage and sparing aims are met. Only plans are shown which are non-dominated in LSI and LCI in their respective runs. The 480 s column is also shown in Figure 5.1(b).

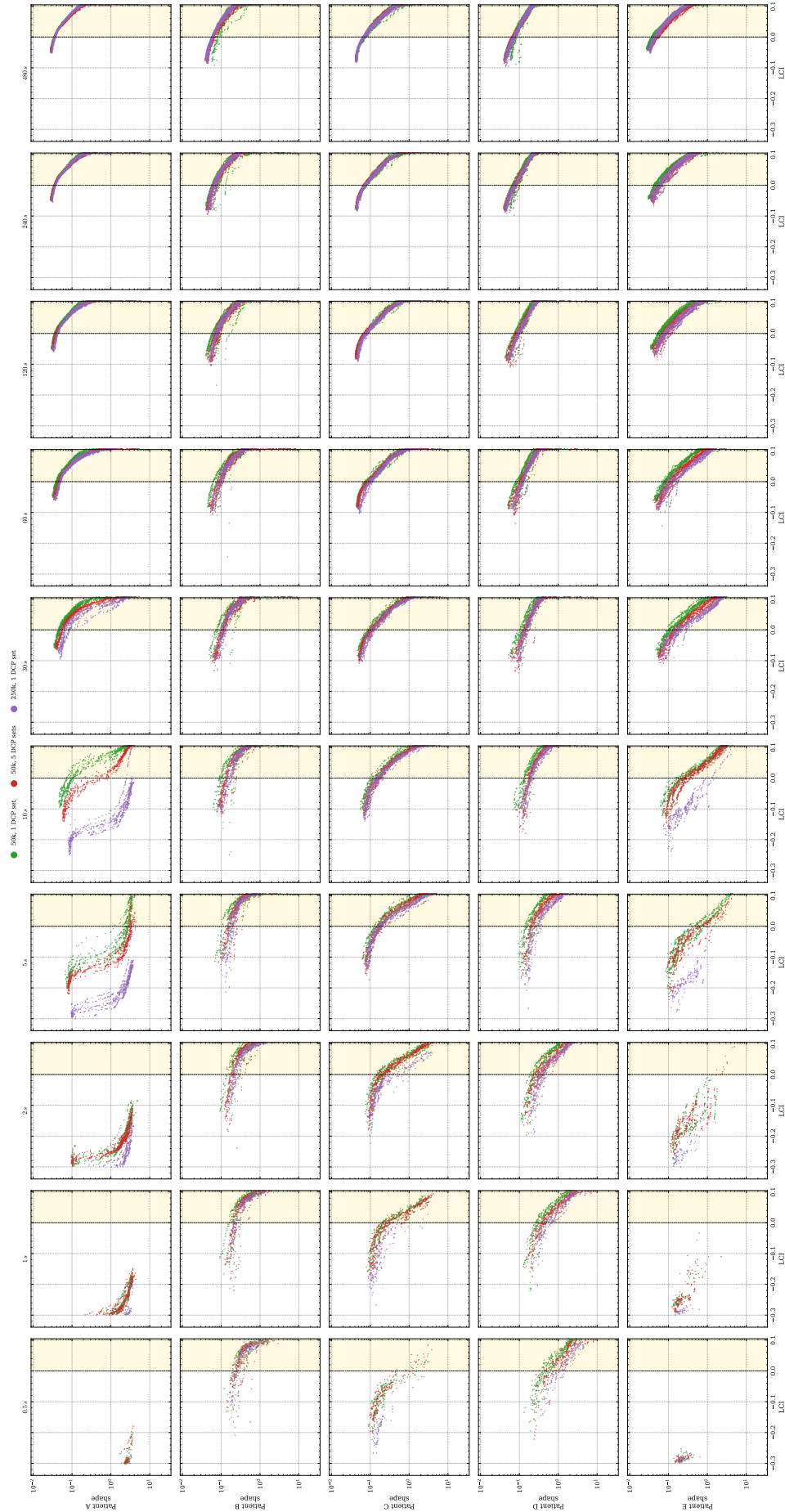
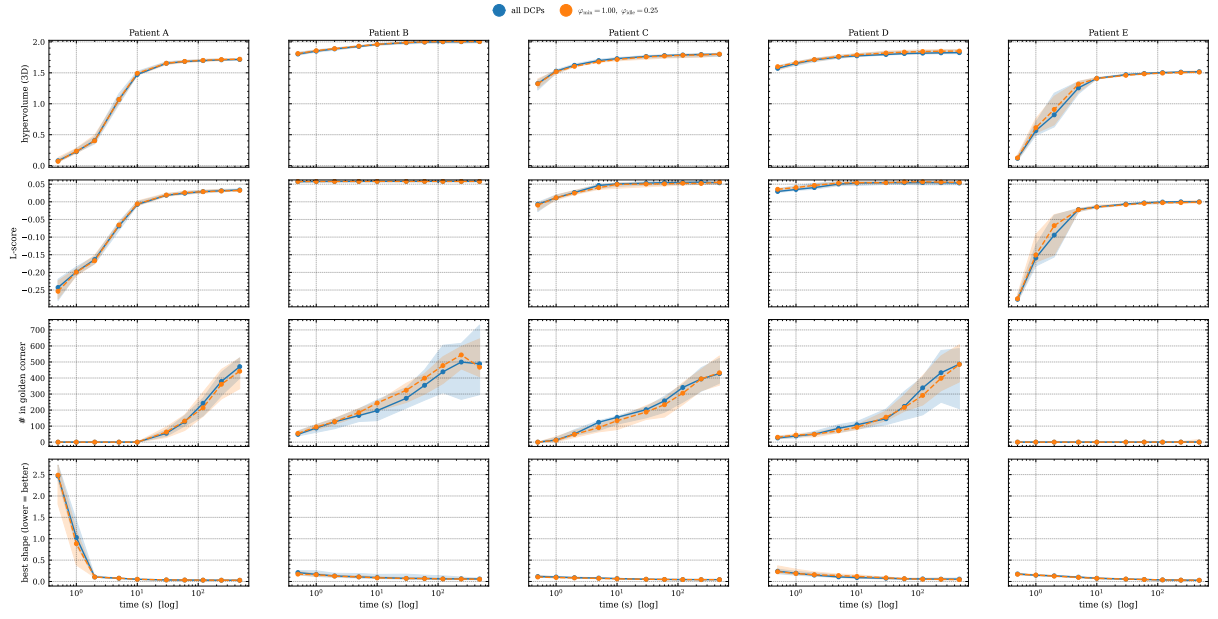
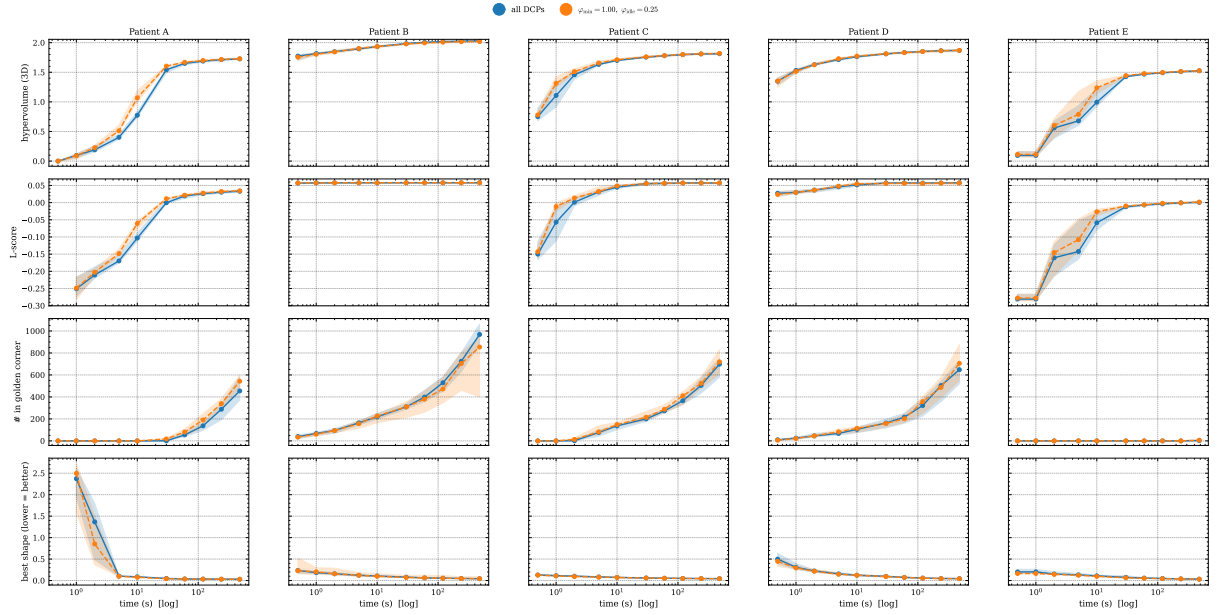


Figure A.6: Approximation fronts of SHAP over LCI, with SHAP on a reversed logarithmic axis, over the full time-limit sweep, for 1 set of 50 000 DCPs, 5 rotating sets of 50 000 DCPs and 1 set of 250 000 DCPs, with patients as rows and time limits as columns. Treatment plans found by ten runs of each configuration are overlaid. Every plan is reevaluated at 500 000 DCPs. The shaded region is where all coverage aims are met. Only plans are shown which are non-dominated in SHAP and LCI in their respective runs.



(a) 50 000 DCPs



(b) 250 000 DCPs

Figure A.7: HV, L-score, number of plans in the golden corner, and best SHAPE found in the golden corner against the wall-clock time limit, for DVI subsampling with $\varphi_{idle} = 0.25$ against the default configuration in which all DCPs are used for every DVI. Lines show the median over ten runs and the shaded bands the interdecile range.