UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

Conrado Silva Miranda

Multi-Objective Optimization Involving Function Approximation via Gaussian Processes and Hybrid Algorithms that Employ Direct Optimization of the Hypervolume

Otimização Multi-Objetivo Envolvendo Aproximadores de Função via Processos Gaussianos e Algoritmos Híbridos que Empregam Otimização Direta do Hipervolume

CAMPINAS
2018

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

Conrado Silva Miranda

Multi-Objective Optimization Involving Function Approximation via Gaussian Processes and Hybrid Algorithms that Employ Direct Optimization of the Hypervolume

Otimização Multi-Objetivo Envolvendo Aproximadores de Função via Processos Gaussianos e Algoritmos Híbridos que Empregam Otimização Direta do Hipervolume

Thesis presented to the School of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering, in the area of Computer Engineering.

Tese apresentada a Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica, na Área de Engenharia de Computação.

Orientador: Prof. Dr. Fernando José Von Zuben

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA TESE DEFENDIDA PELO ALUNO Conrado Silva Miranda, E ORIENTADO PELO PROF. DR. Fernando José Von Zuben.

.....
ASSINATURA DO ORIENTADOR

CAMPINAS
2018

Agência(s) de fomento e nº(s) de processo(s): CAPES; FAPESP, 2015/09199-0

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

M672m

Miranda, Conrado Silva, 1989-
Multi-objective optimization involving function approximation via gaussian processes and hybrid algorithms that employ direct optimization of the hypervolume / Conrado Silva Miranda. – Campinas, SP : [s.n.], 2018.

Orientador: Fernando José Von Zuben.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Algoritmos de aproximação. 2. Híbridos. 3. Otimização multiobjetivo. 4. Processos gaussianos. I. Von Zuben, Fernando José, 1968-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Otimização multi-objetivo envolvendo aproximadores de função via processos gaussianos e algoritmos híbridos que empregam otimização direta do hipervolume
Palavras-chave em inglês:
Approximation algorithms
Hybrid
Multi-objective optimization
Gaussian process
Área de concentração: Engenharia de Computação
Titulação: Doutor em Engenharia Elétrica
Banca examinadora:
Fernando José Von Zuben [Orientador]
Felipe Campelo Franca Pinto
Myriam Regattieri de Biase da Silva Delgado
Romis Ribeiro de Faissol Attux
Guilherme Palermo Coelho
Data de defesa: 27-04-2018
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: Conrado Silva Miranda RA: 070498
Data da Defesa: 27 de abril de 2018

Título da tese em inglês: "Multi-Objective Optimization Involving Function Approximation via Gaussian Processes and Hybrid Algorithms that Employ Direct Optimization of the Hypervolume"
Título da tese: "Otimização Multi-Objetivo Envolvendo Aproximadores de Função via Processos Gaussianos e Algoritmos Híbridos que Empregam Otimização Direta do Hipervolume"

Prof. Dr. Fernando José Von Zuben(Presidente, FEEC/UNICAMP)
Prof. Dr. Felipe Campelo Franca Pinto (UFMG)
Profa. Dra. Myriam Regattieri de Biase da Silva Delgado (UTFPR)
Prof. Dr. Romis Ribeiro de Faissol Attux (FEEC/UNICAMP)
Prof. Dr. Guilherme Palermo Coelho (FT/UNICAMP)

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no processo de vida acadêmica do aluno.

Acknowledgements

I owe huge thanks to my advisor, Fernando, who has always been patient with my mistakes, supportive when I failed and helped me guide through the academic maze. His wisdom at the right times kept the ball rolling and the target achievable.

Another huge base of support is my friends, either from long ago or my new research family. I've learned a lot from them, even when they didn't think so. They have enlightened my dark days and made life much easier.

I also thank FAPESP (São Paulo Research Foundation) for their grant 2015/09199-0 and CAPES, whose financial support helped me achieve the results here described.

Abstract

The main purpose of this thesis is to bridge the gap between single-objective and multi-objective optimization and to show that connecting techniques from both ends can lead to improved results. To reach this goal, we provide contributions in three directions.

First, we show the connection between optimality of a mean loss and the hypervolume when evaluating a single solution, proving optimality bounds when the solution from one is applied to the other. Furthermore, an evaluation of the gradient of the hypervolume shows that it can be interpreted as a particular case of the weighted mean loss, where the weights increase as their associated losses increases. We hypothesize that this can help to train a machine learning model, since samples with high error will also have high weight. An experiment with a neural network validates the hypothesis, showing improved performance.

Second, we evaluate previous attempts at using gradient-based hypervolume optimization to solve multi-objective problems and why they have failed. Based on the analysis, we propose a hybrid algorithm that combines gradient-based and evolutionary optimization. Experiments on the benchmark functions ZDT show improved performance and faster convergence compared with reference evolutionary algorithms.

Finally, we prove necessary and sufficient conditions for a function to describe a valid Pareto frontier. Based on this result, we adapt a Gaussian process to penalize violation of the conditions and show that it provides better estimates than other approximation algorithms. In particular, it creates a curve that does not violate the constraints as much as done by algorithms that do not consider the restrictions, being a more reliable performance indicator. We also show that a common optimization metric when approximating functions with Gaussian processes is a good indicator of the regions an algorithm should explore to find the Pareto frontier.

Keywords: Function approximation, Gaussian process, Hybrid algorithm, Hypervolume, Multi-objective optimization.

Resumo

O principal propósito desta tese é reduzir a lacuna entre otimização mono-objetivo e multi-objetivo e mostrar que conectar técnicas de lados opostos pode gerar melhores resultados. Para atingir esta meta, nós fornecemos contribuições em três direções.

Primeiro, mostra-se a conexão entre otimalidade da perda média e do hipervolume quando avaliando uma única solução, provando limites de otimalidade quando a solução de um é aplicada ao outro. Ademais, uma avaliação do gradiente do hipervolume mostra que ele pode ser interpretado como um caso particular da perda média ponderada, onde os pesos aumentam conforme as perdas associadas aumentam. Levantou-se a hipótese de que isto pode ajudar a treinar modelos de aprendizado de máquina, uma vez que amostras com erro alto também terão peso alto. Um experimento com uma rede neural valida a hipótese, mostrando melhor desempenho.

Segundo, avaliaram-se tentativas anteriores de usar otimização do hipervolume baseada em gradiente para resolver problemas multi-objetivo e por que elas falharam. Baseado na análise, foi proposto um algoritmo híbrido que combina otimização evolutiva e baseada em gradiente. Experimentos nas funções de benchmark ZDT mostram melhor desempenho e convergência mais rápida comparado a algoritmos evolutivos de referência.

Finalmente, foram apresentadas condições necessárias e suficientes para que uma função descreva uma fronteira de Pareto válida. Com base nestes resultados, adaptou-se um processo Gaussiano para penalizar violações das condições e mostrou-se que ele fornece melhores estimativas do que outros algoritmos de aproximação. Em particular, ele cria uma curva que não viola as restrições tanto quanto algoritmos que não consideram as condições, sendo mais confiável como um indicador de performance. Foi também demonstrado que uma métrica de otimização comum, quando aproximando funções com processos Gaussianos, é uma boa indicadora das regiões que um algoritmo deveria explorar para encontrar a fronteira de Pareto.

Palavras-chave: Algoritmos híbridos, Aproximação de função, Hipervolume, Otimização multi-objetivo, Processos Gaussianos.

Nomenclature

Functions

$\mu(x)$	Mean function of a Gaussian process
$f(y)$	Score function
$g_i(x)$	Objective function
$K(x)$	Covariance function of a Gaussian process
$k(x, x')$	Kernel for covariance in a Gaussian process

Hyperparameters

α	Shape for the inverse-gamma distribution
β	Scale for the inverse-gamma distribution
η	Scaling of the covariance in a squared exponential kernel
ρ_i	Scaling of the i-th dimension in a squared exponential kernel
Ξ	Scheduling sequence for ξ
ξ	Scale for the reference point distance from the maximum loss
ζ	Step hardness for monotonic distribution

Operators

$\Delta(h)$	Generalized gradient
$\prec$	Strong dominance operator
$\preceq$	Dominance operator
$\sim$	Distribution operator, such that $a \sim b$ means a follows distribution b

Optimality symbols

$\alpha_i(z, x)$	Relative importance of the i-th objective
$\beta_i(z, x)$	Importance of the i-th objective
Δ	Mean absolute difference between objectives
δ	Neighborhood of an optimal point
γ	Minimum value for the reference point
$\mathcal{X}_\delta$	Subspace of $\mathcal{X}$ in the neighborhood of an optimal point
ν	Maximum absolute difference of relative importances to the balanced case

Quantities and known values

μ	Mean of a Gaussian distribution
Σ	Covariance of a Gaussian distribution
σ^2	Noise variance between the latent and observed spaces in a Gaussian process
M	Number of objectives
P	Number of solutions in a set X
x	Value in the decision space
y	Value in the objective space
z	Reference point in the objective space
z^*	Nadir point

Spaces

$\mathcal{F}$	Pareto frontier
$\mathcal{P}$	Pareto set
$\mathcal{Y}$	Objective space
$\mathcal{Y}_i$	Objective or codomain space for the i-th objective
$\mathcal{X}$	Decision or domain space
$\overline{D}$	Non-dominated set
D	Dominated set

F	Estimated frontier
F_s	Estimated strict frontier
X	Set of values in the decision space

Contents

Chapter 1

Introduction

Multi-objective optimization (MOO), also called multiple criteria optimization [1], is an extension of the standard single-objective optimization (SOO), where the objectives may be conflicting with each other [2, 3]. When a conflict exists, we are no more looking for a single optimal solution but for a set of solutions, each one providing a trade-off on the objectives and none being better than the others. This solution set is called the Pareto set and its counterpart in the objective space is denoted the Pareto frontier.

Although interpreting the problem from this perspective might have benefits, such as evaluating the trade-offs after some solutions have been found [4], which allows clearer comparison, solving the problem is considerably harder than a single-objective one. Therefore, these two areas of research, although clearly connected, have been investigated separately for the most part.

In this work, we try to bridge the gap between them, providing connections that might be useful to extend results from one to the other, with the goal of bringing them closer so that both sides can benefit from contributions. This thesis is divided into 4 chapters besides introduction and conclusion. Chapter 2 provides a general background on the theory and knowledge base required to understand the main contributions, and the other three main contributions divided as follows.

1.1 Connecting single-objective and multi-objective approaches

Chapter 3 introduces the hypervolume maximization with a single solution as an alternative to the mean loss minimization. The relationship between the two problems is proved

through bounds on the cost function when an optimal solution to one of the problems is evaluated on the other, with a hyperparameter, given by the reference point, to control the similarity between the two problems. This same hyperparameter allows higher weight to be placed on samples with higher loss when computing the hypervolume's gradient, whose normalized version can range from the mean loss to the max loss. An experiment on MNIST with a neural network is used to validate the theory developed, showing that the hypervolume maximization can behave similarly to the mean loss minimization and can also provide better performance, resulting on a 20% reduction of the classification error on the test set.

1.2 Hybrid gradient-based multi-objective optimization

Once we determine that optimizing the hypervolume with gradient can give good results, we investigate the literature on applying the same idea for multi-objective. Based on current results, we identify current attempts haven't succeeded and the main culprit is the fact that dominated points don't contribute to the hypervolume, becoming ignored. With this observation, we develop an algorithm to try to perform the optimization but without the same issues as existing methods, by optimizing one point at a time.

Chapter 4 introduces a high-performance hybrid algorithm, called Hybrid Hypervolume Maximization Algorithm (H2MA), for multi-objective optimization that alternates between exploring the decision space and exploiting the already obtained non-dominated solutions. The proposal is centered on maximizing the hypervolume indicator, thus converting the multi-objective problem into a single-objective one. The exploitation employs gradient-based methods, but considering a single candidate efficient solution at a time, to overcome limitations associated with population-based approaches and also to allow easy control of the number of solutions provided. There is an interchange between two steps. The first step is a deterministic local exploration, endowed with an automatic procedure to detect stagnation. When stagnation is detected, the search is switched to a second step characterized by a stochastic global exploration using an evolutionary algorithm. Using five ZDT benchmarks with 30 variables, the performance of the new algorithm is compared to reference algorithms for multi-objective optimization, more specifically NSGA-II, SPEA2, and SMS-EMOA. The solutions found by the H2MA guide to higher hypervolume and smaller distance to the true Pareto frontier with significantly less function evaluations, even when the gradient is estimated numerically. Furthermore, although only continuous decision spaces have been considered here, discrete decision spaces could also have been treated, replacing gradient-based search by hill-climbing. Finally, a thorough explanation is provided to support the expressive gain in performance that was achieved.

1.3 Approximating the optimal objective space

Although H2MA is able to reach good performance, it can be expensive to compute in real problems due to the numeric gradient and the computation of the hypervolume itself, which is exponential in the number of objectives for the general case. In order to reduce this cost, we pursued the field of approximation through surrogates so that we could define a less expensive metric and that could be integrated more easily with other surrogate methods for the objectives themselves. Since the Pareto frontier is the main target of the multi-objective optimization, we focus on understanding its behavior and drive metrics from its shape.

Chapter 5 introduces necessary and sufficient conditions that surrogate functions must satisfy to properly define frontiers of non-dominated solutions in multi-objective optimization problems. These new conditions work directly on the objective space, thus being agnostic about how the solutions are evaluated. Therefore, real objectives or user-designed objectives' surrogates are allowed, opening the possibility of linking independent objective surrogates. To illustrate the practical consequences of adopting the proposed conditions, we use Gaussian processes as surrogates endowed with monotonicity soft constraints and with an adjustable degree of flexibility, and compare them to regular Gaussian processes and to a frontier surrogate method in the literature that is the closest to the method proposed in this paper. Results show that the necessary and sufficient conditions proposed here are finely managed by the constrained Gaussian process, guiding to high-quality surrogates capable of suitably synthesizing an approximation to the Pareto frontier in challenging instances of multi-objective optimization, while an existing approach that does not take the theory proposed in consideration defines surrogates which greatly violate the conditions to describe a valid frontier.

Chapter 2

Background

2.1 Multi-objective optimization

Multi-objective optimization (MOO) is a generalization of the standard single-objective optimization to problems where multiple criteria are defined and they conflict with each other [2]. In this case, there can be multiple optimal solutions with different trade-offs between the objectives. Since the optimal set can be continuous, an MOO problem is given by finding samples from the optimal set, called Pareto set. However, we also wish that the mapping of the obtained samples of the Pareto set into the objective space may be well-distributed along the Pareto frontier, which is the counterpart for the Pareto set, so that the solutions present more diverse trade-offs.

A multi-objective optimization problem is described by its decision space $\mathcal{X}$ and a set of objective functions $g_i(x)\colon \mathcal{X} \to \mathcal{Y}_i, i \in \{1, \ldots, M\}$, where $\mathcal{Y}_i \subseteq \mathbb{R}$ is the associated objective space for each objective function [5]. Due to the symmetry between maximization and minimization, only the minimization problem is considered here. Each point x in the decision space has a counterpart in the objective space $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_M$ given by $y = g(x) = (g_1(x), \ldots, g_M(x))$.

Since there are multiple objectives, a new operator for comparing solutions must be used, since the conventional "less than" ($<$) and "less than or equal" ($\leq$) operators can only compare two numbers. This operator is denoted the dominance operator and is defined as follows.

Definition 1 (Dominance)**.** Let y and y' be points in $\mathbb{R}^M$, the objective space. Then y dominates y', denoted $y \preceq y'$, if $y_i \leq y'_i$ for all i.

The definition of dominance used in this thesis is the same provided in [6], which allows a point to dominate itself. This relation is usually called weak dominance, but we call it "dominance" for simplicity, since it is the main dominance relation used in this thesis. Another common definition is to require that $y_i < y'_i$ for at least one i, and both definitions are consistent with the theory developed in throughout the thesis. Another operator can be defined to enforce strict inequalities between the values on each dimension, denoted strong dominance.

Definition 2 (Strong Dominance)**.** Let y and y' be points in $\mathbb{R}^M$, the objective space. Then y strongly dominates y', denoted $y \prec y'$, if $y_i < y'_i$ for all i.

Using the dominance, we can define the set of points characterized by the fact that no other point can have better performance in all objectives.

Definition 3 (Pareto Set and Frontier)**.** The Pareto set is defined by the set of all points in the decision space that are not strongly dominated by any other point in the decision space. That is, the Pareto set is given by $\mathcal{P} = \{x \in \mathcal{X} \mid \nexists x' \in \mathcal{X} \colon f(x') \prec f(x)\}$. The Pareto frontier is the associated set in the objective space, given by $\mathcal{F} = \{f(x) \mid x \in \mathcal{P}\}$.

Therefore, the Pareto frontier can be seen as a generalization of the concept of "global minima" from single objective optimization, describing all the possible trade-offs between solutions, and the Pareto set is a generalization of the set of optimal solutions. This approach can be easily contrasted to the most commonly used by single-objective optimization to deal with the conflicting objectives, where a weight is placed on each to specify the importance and a new objective is constructed as

$$G(x) = \sum_{i=1}^{m} w_i g_i(x), \quad w_i \geq 0. \tag{2.1}$$

While this approach is frequently used, it has two major drawbacks. The first one is that it can only find all the possible optima if each objective g_i is convex [7]. That is, if at least one of them is not, then there exists at least one point in the Pareto set x^* which is not in the optimal space for $G(x)$ for any choice of w_i. Therefore, using this approach for transforming the multi-objective problem into a single objective will, in many cases, fail to find a solution whose trade-offs would have been preferred by the user if they had an option.

The second drawback is that it forces the user to establish a preference between the objectives (through the weights) before they ever see any solution [4]. This prior preference thus forces the optimization algorithm to solve that particular problem instead of presenting a set of "interesting" solutions so that the user can choose from the options provided.

To counter-balance this, MOO algorithms usually focus on providing multiple possible solutions, with different trade-offs. Therefore, a good MOO solver must put pressure both to optimize all objectives and to create a diverse set of candidates, presenting all the possibilities in the Pareto frontier.

2.2 Hypervolume indicator

The current state-of-the-art for MOO relies on the use of evolutionary algorithms for finding the desired optimal set [3]. One of these algorithms is the NSGA-II [8], which performs non-dominance sorting, thus ordering the proposed solutions according to their relative dominance degree, and dividing the solution set in subsequent frontiers of non-dominated solutions. NSGA-II also uses crowding distance, which measures how close the nearby solutions are, to maintain diversity in the objective space. Another well-known algorithm is the SPEA2 [9], where the solutions have a selective pressure to move towards the Pareto frontier and also to stay away from each other.

These algorithms are based on heuristics to define what characterizes a good set of solutions. However, the hypervolume indicator [10] defines a metric of performance for a set of solutions, thus allowing a direct comparison of multiple distinct sets of solutions [6], with higher values indicating possible better quality. The hypervolume is maximal at the Pareto frontier and increases if the samples are better distributed along the frontier [11]. Moreover, the hypervolume is the only known unary performance indicator that satisfies all theoretical guarantees for such type of indicators, such as providing valid quality and diversity comparisons [6]. There are other indicators that can achieve better guarantees, but they require multiple sets of candidates to measure the relative performance of the whole set, thus being able to perform only relative comparisons. Due to these properties, it represents a good candidate to be maximized in MOO, being explicitly explored in the SMS-EMOA [12], where solutions that contribute the least to the hypervolume are discarded.

In order to define the hypervolume indicator [10], we must first define the reference point, which is a point in the objective space that is dominated by every point in a set.

Definition 4 (Reference Point)**.** Let $X = \{x_1, \ldots, x_P\} \subseteq \mathcal{X}$ be a set of points in the decision space. Let $z \in \mathbb{R}^M$ be a point in the objective space. Then z is a valid reference point if, for all $x \in X$ and $i \in \{1, \ldots, M\}$, we have that $g_i(x) < z_i$. Using Definition 1, this can be written as $g(x) \prec z$.

There is one reference point of particular importance for MOO, which can be used as standardized metric of performance, called the Nadir point. It can be defined as:

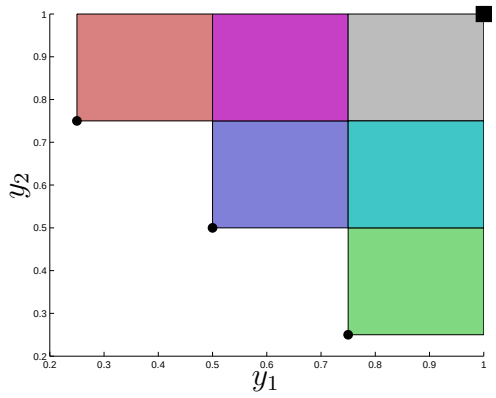


Figure 2.1: Example of hypervolume. The non-dominated solutions in the objective space are shown in black circles, and the reference point is shown in the black square. For each non-dominated solution, the region between it and the reference point is filled, with colors combining when there is overlap, and the total hypervolume is given by the area of the shaded regions. Best viewed in color.

Definition 5 (Nadir Point)**.** A reference point $z^* \in \mathbb{R}^M$ is a Nadir point if and only if it is dominated by all other reference points. That is, let $\mathcal{R}$ be the set of all possible reference points, then $\nexists z \in \mathcal{R}, : z \prec z^*$.

Again, it is possible to allow equality in the definition of the reference point, just like in the definition of dominance. However, when equality is allowed, it is possible for some point to have a null hypervolume, which can guide to undesired decisions when using the hypervolume as a performance metric, since such points would not contribute to the hypervolume and would be replaced by other points. Using the definition of a reference point, we can define the hypervolume for a set of points.

Definition 6 (Hypervolume)**.** Let $X = \{x_1, \ldots, x_P\} \subseteq \mathcal{X}$ be a set of points in the decision space. Let $z \in \mathbb{R}^M$ be a valid reference point in the objective space. Then the hypervolume can be defined as:

$$H(X; z) = \int_{\mathbb{R}^M} 1[\exists x \in X : f(x) \prec y \prec z]\mathrm{d}y, \tag{2.2}$$

where $1[\cdot]$ is the indicator function.

The hypervolume measures how much of the objective space is dominated by a current set X and dominates the reference point z. Fig. 2.1 shows an example of the hypervolume for a set of three non-dominated points. For each point, the shaded region represents the area dominated by the given point, with colors combining when there is overlap.

Note that the definition of hypervolume given here is relative to an arbitrary reference point, while other definitions require the Nadir point. The difference usually occurs due to the

moment where the hypervolume is computed, since the Nadir point cannot be know a priori. Therefore, for optimization algorithms that incrementally maximize the hypervolume, a reference point must be chosen, like it is done in SMS-EMOA [12], while the real Nadir can be used to compute the effective hypervolume to compare solutions.

2.3 Bayesian statistics

Handling uncertainty is key for many real world optimization problems [4], since real evaluations are subject to noise both from the process itself, like a sudden spike in access to a viral video, and from measurement, like sensors responding to environment vibration. Therefore, developing methods that can deal with uncertainty from its construction are of high value.

In machine learning, a common approach is to define a probabilistic parametric model that, given some input, computes a prediction on its output and whose objective is to minimize some statistical measure of error. For instance, a logistic regression defines a parametric model

$$\hat{y} = \sigma\left(\sum_{i=1}^{N} w_i x_i\right), \tag{2.3}$$

where σ is the sigmoid function, w_i are the parameters of the model and x_i is the data. Its objective is to maximize the likelihood of the labels in the dataset, defining a cost

$$J(w) = -\sum_{s\in\mathcal{S}} \log p(y_s|\hat{y}_s), \tag{2.4}$$

where this cost function is known as negative log likelihood and this approach is known as maximum likelihood [13] and $\mathcal{S} = \{(x, y)\}$ are the samples.

However, direct optimization of this cost can lead to issues in ill-conditioned problems [13]. A common approach to counter this is to enforce a prior on the possible parameters and optimize the posterior probability of the data given this prior of the model. Therefore, for a given prior $p(w)$, the problem becomes

$$J(w) = -\log p(w) - \sum_{s\in\mathcal{S}} \log p(y_s|\hat{y}_s), \tag{2.5}$$

which can be seen as placing a regularization on the optimization.

Alternatively to the maximum likelihood and maximum posterior methods, we can also find the full distribution of potential values for w based on the data, instead of picking only ones that maximize some metric. Using Bayes' theorem, we end up not with one set of optimal

solutions but a distribution over the possible solutions:

$$p(w|\mathcal{S}) = \frac{p(\mathcal{S}|w)p(w)}{p(\mathcal{S})} = \frac{p(\mathcal{S}|w)p(w)}{\int p(\mathcal{S}|w)p(w)dw}, \tag{2.6}$$

where $p(\mathcal{S}|w)$ is called the likelihood, describing how likely the data is given a set of parameters, and $p(\mathcal{S})$ is called the marginal, describing the likelihood of the data itself and usually is considered just to ensure the distribution is normalized.

Given a learn probability of the parameters, we can make an statistical prediction over a new sample x^* by computing its posterior, that is,

$$p(y^*|x^*, \mathcal{S}) = \int p(y^*|w, x^*)p(w|\mathcal{S})dw. \tag{2.7}$$

As one can imagine, computing these functions in the general case is tricky and many approximations have been studied [13]. We will focus on one particular method, namely Gaussian process, that uses the normal distribution as basis and has well defined solutions to these problems.

2.4 Gaussian process

A Gaussian process (GP) is a generalization of the multivariate normal distribution to infinite dimensions and can be used to solve a regression problem. A GP defines a probability distribution over functions, such that the outputs are jointly normally distributed. Due to the particular properties of the Gaussian distribution, this infinity does not cause issues, as we will show.

2.4.1 Gaussian distribution

The Gaussian or normal distribution can be described as a probability distribution over a space $\mathbb{R}^N$ with the following probability density function:

$$p(x|\mu, \Sigma) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right), \quad X \sim \mathcal{N}(\mu, \Sigma), \tag{2.8}$$

where $\mu \in \mathbb{R}^N$ is the mean vector and $\Sigma \in \mathbb{R}^N \times \mathbb{R}^N$ is the positive definite covariance matrix.

If we divide the vector x in two non-empty parts x_1 and x_2 and make appropriate divisions

of μ and Σ such that

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}, \tag{2.9}$$

then the following two properties hold:

1. both x_1 and x_2 are distributed normally, with distributions $X_1 \sim \mathcal{N}(\mu_1, \Sigma_{11})$ and $X_2 \sim \mathcal{N}(\mu_2, \Sigma_{22})$; and

2. the posterior distribution for x_1 given certain values for x_2 is normally distributed according to $X_1|X_2 \sim \mathcal{N}(\hat{\mu}_1, \hat{\Sigma}_{11})$, where

$$\hat{\mu}_1 = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) \tag{2.10a}$$

$$\hat{\Sigma}_{11} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}. \tag{2.10b}$$

Therefore, a Gaussian distribution can have any number of dimensions but, if we restrict ourselves to evaluate only a few of them, those few are also normally distributed and their joint distribution can be easily calculated by just ignoring the others. This also means that marginalizing a Gaussian distribution over some of its variables defines another easy to compute Gaussian distribution. Therefore, most of the issues in computation for infinite dimensions are easily handled.

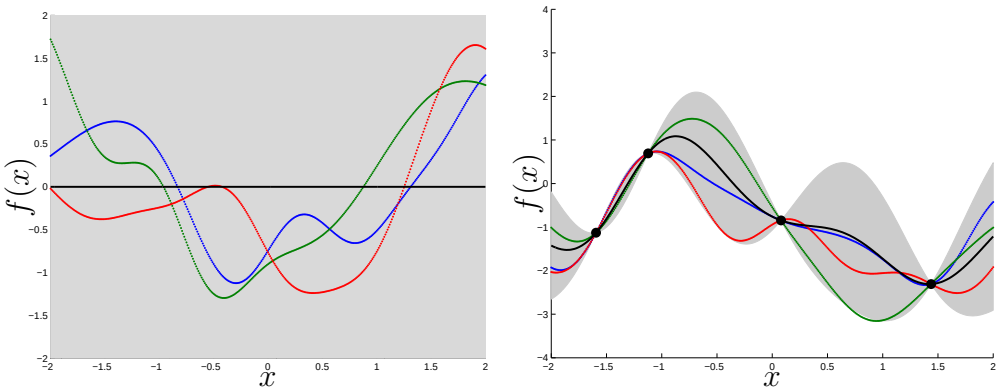
2.4.2 Gaussian process as function approximation

In general, a function approximation problem is defined by finding a function that (approximately) fits a given set of data points. Since the model should have enough flexibility to fit the given samples, an appropriate choice for a surrogate function is a Gaussian process, which always has enough capacity to fit the data [14].

To better understand this concept, consider an infinite column vector $y \in \mathbb{R}^\infty$ and an infinite matrix $x \in \mathbb{R}^{\infty \times D}$. Then a function $f\colon \mathbb{R}^D \to \mathbb{R}$ can be described by associating the row indexes, such that $f(x_{i,:}) = y_i$. The GP relies on the fact that the relationship between x and y can be written as:

$$y \sim \mathcal{N}(\mu(x), K(x)), \tag{2.11}$$

which states that all dimensions of y are distributed according to a multivariate normal distribution with mean $\mu(x)$ and covariance $K(x)$. Moreover, the mean for a given dimension is given by $\mathbb{E}[y_i] = \mu(x_{i,:})$ and the covariance is given by $\mathrm{Cov}(y_i, y_j) = k(x_{i,:}, x_{j,:})$, where $k(\cdot, \cdot)$ is a positive definite kernel function.



(a) Before observations **(b)** After observations

Figure 2.2: Function distribution using a Gaussian process. Before the observations, the distribution is the same over all the space. After the observations, the distribution adapts to constrain the possible functions. The distribution mean is given by the black line and the 95% confidence interval is given by the shadowed region. Three function samples are also provided for each case.

Although continuous functions, and thus Gaussian processes, are defined for an infinite number of points, which caused the vectors x and y to have infinite dimensions, only a finite number of observations are actually made in practice. Let N be such number of observations. Then, by the marginalization property of the multivariate normal distribution, we only have to consider N observed dimensions of x and y. Furthermore, the finite-dimension version of y is still normally distributed according to Eq. (2.11) when considering only the observed dimensions.

Usual choices for the mean and covariance functions are the null mean [14], such that $\mu(x) = 0$, and the squared exponential kernel, defined by:

$$k(x, x') = \eta^2 \exp\left(-\frac{1}{2}\sum_{i=1}^{D}\frac{(x_i - x_i')^2}{\rho_i^2}\right), \tag{2.12}$$

where $\eta, \rho_i > 0$ and ρ_i are the scale parameters, which define a representative scale for the smoothness of the function.

The choice of the kernel function establishes the shape and smoothness of the functions defined by the GP, with the squared exponential kernel defining infinitely differentiable functions. Other choices of kernel are possible and provide different compromises regarding the shape of the function being approximated, such as faster changes and periodicity of values.

Figure 2.2a shows the prior distribution over functions using the squared exponential kernel with $\eta = 1$, $\rho = 0.5$, $D = 1$, and the zero mean. This highlights the fact that the GP defines a distribution over functions, not a unique function. Three sample functions from this GP are

also shown in the same figure. Note that the functions are not shown as continuous, which would require an infinite number of points, but as finite approximations.

To use the GP to make predictions, the observed values of x are split into a training set X, whose output Y is known, and a test set X_*, whose output Y_* we want to predict. Since all observations are jointly normally distributed, we have that the posterior distribution is given by:

$$Y_*|X_*, X, Y \sim \mathcal{N}(\mu_*, \Sigma_*) \tag{2.13a}$$
$$\mu_* = K(X_*, X)K(X, X)^{-1}Y \tag{2.13b}$$
$$\Sigma_* = K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*), \tag{2.13c}$$

where $K(\cdot, \cdot)$ are matrices built by computing the kernel function for each combination of the arguments values. For details of the derivation, please consult [13].

The posterior distribution for the previous GP, after four observations marked as black dots, is shown in Fig. 2.2b. Note that the uncertainty around the observed points is reduced due to the observation themselves, and the mean function passes over the points, as expected. Again, three functions are sampled from the posterior, and all agree on the value the function must assume over the observations.

In order to avoid some numerical issues and to consider noisy observations, we can assume that the covariance has a noise term. Assuming that $y_i = f(x_i) + \epsilon_i$, where ϵ_i is normally distributed with zero mean and variance σ^2, then the covariance of the observations is given by $\mathrm{Cov}(y_i, y_j) = k(x_i, x_j) + \sigma^2\delta_{ij}$, where δ_{ij} is the Kronecker delta. The noiseless value $l_i = f(x_i)$ can then be estimated by:

$$L_*|X_*, X, Y \sim \mathcal{N}(\mu_*, \Sigma_*) \tag{2.14a}$$
$$\mu_* = K(X_*, X)\Omega Y \tag{2.14b}$$
$$\Sigma_* = K(X_*, X_*) - K(X_*, X)\Omega K(X, X_*) \tag{2.14c}$$
$$\Omega = \left[K(X, X) + \sigma^2 I\right]^{-1}, \tag{2.14d}$$

which is similar to Eq. (2.13), except for the added term in Ω corresponding to the noise.

2.4.3 Expectation propagation

As discussed in Sec. 2.4.1, if the prior and likelihood are normal distributions, so is the posterior. However, if one of them is not normal, then the posterior computation can be intractable, due to the integral, and not have closed form. This is the case of the monotonicity

constraints introduced in Sec. 5.4.1. The expectation propagation algorithm [15] solves this problem by approximating the non-normal term by a normal distribution.

Let $q^{\setminus i}(x)$ be the normal prior[1] before incorporating the i-th evidence and $t_i(x)$ the i-th non-normal likelihood based on data. With this, we can write the posterior as

$$\hat{p}(x) = \frac{t_i(x)q^{\setminus i}(x)}{\int t_i(x)q^{\setminus i}(x)dx}, \tag{2.15}$$

which is clearly non-normal due to $t_i(x)$. However, we can find an approximation $q(x)$ to $\hat{p}(x)$ which is normal and minimizes the KL-divergence $D(\hat{p}(x)||q(x))$, and use that as the posterior when incorporating the next piece of evidence.

An alternative interpretation to the posterior approximation can be defined from an approximation of the likelihood $t_i(x)$ itself by some other function $\tilde{t}_i(x)$ and use that approximation to compute the exact posterior $q(x)$. This interpretation is possible because we can define

$$\tilde{t}_i(x) = Z_i \frac{q(x)}{q^{\setminus i}(x)}, \tag{2.16}$$

where Z_i is a normalizing constant. Applying this to Eq. (2.15) makes it clear that the result is indeed the desired posterior $q(x)$. Moreover, since the normal distribution is part of the exponential family, $\tilde{t}_i(x)$ is also a normal distribution.

Therefore, the expectation propagation algorithm, in this context, approximates each non-normal piece of evidence by a normal distribution such that the posterior becomes close to the true posterior. Moreover, the order in which the approximated evidence is applied does not matter as they are all multiplied together.

Given initial approximations for all $\tilde{t}_i(x)$ and using the algorithm, one can iterate over i between computing $q^{\setminus i}(x)$ from Eq. (2.16), optimizing $q(x)$ to fit the posterior in Eq. (2.15) and re-computing approximations for the likelihood through Eq. (2.16). Each step of the algorithm provides better approximations $\tilde{t}_i(x)$ for the likelihood and $q(x)$ for the posterior. Please refer to [15] for more details and examples.

[1] As we will see, the formulation describes both the prior and the likelihood as only depending on x, so it does not matter which one is normal and the results work on either case. However, assuming it is the prior allows us to chain the results to incorporate more evidence.

2.4.4 Expected improvement

Using a distribution over functions given by a Gaussian process, a reasonable task for the optimizer is to find good regions of the space to try before evaluating the real function. A method for finding these regions must take into account the actual predictions and the uncertainty over the possible functions being optimized.

The efficient global optimization (EGO) algorithm [16] proposes to solve this problem by defining a performance metric called the expected improvement (EI). By choosing the points with highest value of this metric to be evaluated in the real objective, it leverages the information collected about the objective to avoid regions that probably will not provide improvements of the objective's value.

Let $f(x)$ be the function being approximated by the GP, then the EI is given by

$$EI(x) = p(1[f(x) < y_{min}]), \tag{2.17}$$

where the probability is based on the current predictions from the GP and y_{min} is the minimum value of the function observed so far.

Therefore, the EI measures the probability that a given point x produces a lower value then the current minimum according to the distribution over functions described by the GP. This simple metric can automatically handle both high uncertainty, which makes the probability high due to the variance, and low, accurate predictions, which would also make the probability high due to the concentrated density. It is also an efficient metric, since the EI can be computed in closed form directly from the GP model, which in turn can be computed in polynomial time.

Let $f(x)$ be distributed from $\mathcal{N}(\mu(x), \sigma^2(x))$, where $\mu(x)$ and $\sigma(x)$ can be found from the predictive posterior of the Gaussian process. Then the Eq. (2.17) can be written in closed form as

$$EI(x) = \sigma(x)\frac{1}{\sqrt{2\pi}}\exp\left(\frac{-\xi(x)^2}{2}\right) - \mu(x)\frac{1+\operatorname{erf}\left(\frac{\xi(x)}{\sqrt{2}}\right)}{2}, \quad \xi(x) = -\frac{\mu(x)}{\sigma(x)}, \tag{2.18}$$

where $\exp(\cdot)$ is the exponential function and $\operatorname{erf}(\cdot)$ is the error function [13], given by:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp(-t^2)dt. \tag{2.19}$$

Chapter 3

Single-solution hypervolume optimization

When comparing the fields of single-objective optimization (SOO) and multi-objective optimization (MOO), one striking difference between them is that SOO is usually concerned with the optimality of a single solution [7]. In the case of multi-modal or noisy problems, we might be interested in finding multiple solutions with some diversity among them to find all the optimal modes, but the optimality of each solution on the original problem can usually be isolated. On the other hand, MOO problems inherently focus on multiple solutions [5] since there is a space of possible solutions based on the trade-offs between objectives, even if they are convex. Hence, not many connections between the two areas have been done as they seem to focus on different styles of solutions.

In this chapter, we are interested in exploring the relationship between the two areas as a two-way link and focusing on single-solution problems. Since SOO has been more extensively studied, both theoretically and pragmatically, these connections might allow us to bring existing knowledge to MOO helping to better understand the problem. On the other hand, MOO is a generalization of SOO, so we are also interested in how using this higher level view can help define the SOO problems to be solved with classical tools better.

With this goal in mind, Section 3.1 will provide theoretical bounds between optimal solutions to the mean loss cost and to the hypervolume. These bounds show, based on the transfer of the problem, how far away from the optimal in problem A can an optimal solution to problem B be. This is a common technique in SOO to evaluate optimality of a problem which is hard to evaluate directly but easy to define a bound based on another simpler problem [17].

In the other direction, Section 3.2 shows how the choice of weights in the weighted mean loss can be made such that, for a convex problem, it has the same optimal solution as the

hypervolume. Using this weight definition as inspiration, we evaluate it throughout a non-convex optimization with mini-batches of samples, which is a more complex problem. We show that this choice can lead to better results than the naive mean loss.

3.1 Optimality relationship between mean loss and hypervolume

In order to define what is the behavior of an optimal solution to one problem in the other, we first have to define mathematically what an optimal solution means and how to write the problems from the same bases.

From the intuitive notion that a locally optimal solution is one where no nearby solutions are better, the first definition is simple and based on a space $\mathcal{X}$ of possible solutions:

Definition 7 (Local Minimum)**.** Let $(\mathcal{X}, d)$ be a metric space. Let $x^* \in \mathcal{X}$ and $\mathbb{R} \ni \delta > 0$. Let $f\colon \mathcal{X} \to \mathbb{R}$ be the function evaluated. Then x^* is said to be a local minimum of f with neighborhood $\mathbb{R} \ni \delta > 0$ if and only if

$$\forall x \in \mathcal{X}\colon d(x, x^*) < \delta \Rightarrow f(x^*) \leq f(x). \tag{3.1}$$

A similar definition can be done for a maximum. Note that this definition does not assume that the function f is differentiable or even continuous. Similarly, the space $\mathcal{X}$ can be any set to which we can associate a metric d. If we could not associate a metric with such space, then we would not have the notion of how close two solutions are in order to compare their optimality.

Using this same space and a set of functions representing the objectives, we can define the mean loss and hypervolume[2] as follows.

Definition 8 (Mean Loss)**.** Let $\mathcal{X}$ be a set and let $\mathcal{G} = \{g \mid g\colon \mathcal{X} \to \mathbb{R}\}$ be a set of functions. Then the mean loss is a function $J_m\colon \mathcal{X} \to \mathbb{R}$ defined by

$$J_m(x) \coloneqq \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} G(x). \tag{3.2}$$

Definition 9 (Hypervolume)**.** Let $\mathcal{X}$ be a set and let $\mathcal{G} = \{g \mid g\colon \mathcal{X} \to \mathbb{R}\}$ be a set of functions.

[2]The function being defined is actually the logarithm of the hypervolume when a single solution is present. This name overlapping will happen in this chapter for simplicity.

Then the hypervolume is a function $H\colon \mathbb{R} \times \mathcal{X} \to \mathbb{R}$ defined by

$$H(z, x) \coloneqq \begin{cases} \sum_{g \in \mathcal{G}} \log(z_g - g(x)), & \text{if } z_g > g(x)\ \forall g \in \mathcal{G}, \\ \text{undefined}, & \text{otherwise} \end{cases} \tag{3.3}$$

Note that the hypervolume has an extra parameter z representing the reference point used for its measure, such as the Nadir point [11]. Since the metric is meaningless if any of the losses is higher than the reference, we say that the hypervolume is undefined in such cases and this automatically reduces the space of feasible solutions $\mathcal{X}$. Also, note that we use the same reference for all objectives without loss of generalization since the pairs $(z_i, g_i(x))$ and $(z_i - c_i, g_i(x) + c_i)$ would give the same hypervolume and adding constants to the mean loss changes its value but not the optimality of solutions. Hence we'll consider all z_i to have the same value z.

With these definitions, Section 3.1.1 will describe a mathematical link between the hypervolume and the mean loss for a set of proposed solutions. Based on this relationship, Sections 3.1.2 and 3.1.3 will define bounds on the optimality in problem B for an optimal solution for problem A.

3.1.1 Linking the objectives

Now that we have a precise definition for both objectives being analyzed, we have to find a link between the two problems that can be used to analyze the bounds for optimal solutions. The following lemma provides such link for two candidate solutions to the two problems:

Lemma 1 (Link between mean loss and hypervolume)**.** *Let $\mathcal{X}$ be a set and let $\mathcal{G} = \{g_1, \ldots, g_N \mid g_i\colon \mathcal{X} \to \mathbb{R}\}$ be a set of functions. Let $x_1, x_2 \in \mathcal{X}$. Let J_m and H be as in Definitions 8 and 9, respectively. Let $\lceil g(x_1) \rceil, \lfloor g(x_2) \rfloor, \lceil g(x_2) \rceil \in \mathbb{R}$ satisfy $g(x_1) \leq \lceil g(x_1) \rceil$ and $\lfloor g(x_2) \rfloor \leq g(x_2) \leq \lceil g(x_2) \rceil$ for all $l \in \mathcal{G}$. Let $\Delta \in \mathbb{R}$ such that $\sum_{i=1}^{N} |g_i(x_1) - g_i(x_2)| \leq N\Delta$. Let $\beta_i(z, x) \coloneqq \frac{1}{z - g_i(x)}$ and $\alpha_i(z, x) \coloneqq \frac{\beta_i(z,x)}{\sum_{j=1}^{N} \beta_j(z,x)}$. Let there be $\mathbb{R} \ni \nu > 0$ and $\mathbb{R} \ni \gamma \geq \max\{\lceil g(x_1) \rceil, \lceil g(x_2) \rceil\}$ such that*

$$|N\alpha_i(z, x_2) - 1| \leq \nu \tag{3.4}$$

for all $i \in [N]$ and $z > \gamma$. Then

$$H(z, x_1) - H(z, x_2) \leq N \log\left(1 + \max\left\{\frac{\nu\Delta + J_m(x_2) - J_m(x_1)}{z - \lceil g(x_2) \rceil}, \frac{\nu\Delta + J_m(x_2) - J_m(x_1)}{z - \lfloor g(x_2) \rfloor}\right\}\right) \tag{3.5}$$

and

$$H(z, x_1) - H(z, x_2) \leq N \log\left(1 + \frac{\nu\Delta + |J_m(x_2) - J_m(x_1)|}{z - \lceil g(x_2) \rceil}\right) \tag{3.6}$$

for all $z > \gamma$.

Proof. From the definition of hypervolume and Jensen's inequality [17], we have that

$$\begin{aligned}
&H(z, x_1) - H(z, x_2) && \text{(3.7a)}\\
&= \sum_{i=1}^{N} (\log(z - g_i(x_1)) - \log(z - g_i(x_2))) && \text{(3.7b)}\\
&= \sum_{i=1}^{N} \log\left(\frac{z - g_i(x_1)}{z - g_i(x_2)}\right) && \text{(3.7c)}\\
&= N \sum_{i=1}^{N} \frac{1}{N} \log\left(\frac{z - g_i(x_1)}{z - g_i(x_2)}\right) && \text{(3.7d)}\\
&= N E_i \left[\log\left(\frac{z - g_i(x_1)}{z - g_i(x_2)}\right)\right] && \text{(3.7e)}\\
&\leq N \log E_i \left[\frac{z - g_i(x_1)}{z - g_i(x_2)}\right] && \text{(3.7f)}\\
&\leq N \log\left(\frac{1}{N} \sum_{i=1}^{N} \frac{z - g_i(x_1)}{z - g_i(x_2)}\right). && \text{(3.7g)}
\end{aligned}$$

Simplifying the expression, we can write it as:

$$\begin{aligned}
&H(z, x_1) - H(z, x_2) && \text{(3.8a)}\\
&\leq N \log\left(\frac{1}{N} \sum_{i=1}^{N} \frac{z - g_i(x_1)}{z - g_i(x_2)}\right) && \text{(3.8b)}\\
&= N \log\left(\frac{1}{N} \sum_{i=1}^{N} \frac{z - g_i(x_2) + g_i(x_2) - g_i(x_1)}{z - g_i(x_2)}\right) && \text{(3.8c)}\\
&= N \log\left(\frac{1}{N} \sum_{i=1}^{N} \left(1 + \frac{g_i(x_2) - g_i(x_1)}{z - g_i(x_2)}\right)\right) && \text{(3.8d)}\\
&= N \log\left(1 + \frac{1}{N} \sum_{i=1}^{N} \frac{g_i(x_2) - g_i(x_1)}{z - g_i(x_2)}\right). && \text{(3.8e)}
\end{aligned}$$

Using the definition of α and β given and the bounds provided, the bound can be found

as:

$$
\begin{aligned}
& H(z, x_1) - H(z, x_2) && \text{(3.9a)}\\
& \leq N \log\left(1 + \frac{1}{N}\sum_{i=1}^{N} \beta_i(z, x_2)(g_i(x_2) - g_i(x_1))\right) && \text{(3.9b)}\\
& = N \log\left(1 + \frac{\sum_{j=1}^{N} \beta_j(z, x_2)}{N}\sum_{i=1}^{N} \frac{\beta_i(z, x_2)}{\sum_{j=1}^{N} \beta_j(z, x_2)}(g_i(x_2) - g_i(x_1))\right) && \text{(3.9c)}\\
& = N \log\left(1 + \frac{\sum_{j=1}^{N} \beta_j(z, x_2)}{N}\sum_{i=1}^{N} \alpha_i(z, x_2)(g_i(x_2) - g_i(x_1))\right) && \text{(3.9d)}\\
& = N \log\left(1 + \frac{\sum_{j=1}^{N} \beta_j(z, x_2)}{N^2}\sum_{i=1}^{N} (N\alpha_i(z, x_2) - 1 + 1)(g_i(x_2) - g_i(x_1))\right) && \text{(3.9e)}\\
& = N \log\left(1 + \frac{\sum_{j=1}^{N} \beta_j(z, x_2)}{N^2}\sum_{i=1}^{N} (N\alpha_i(z, x_2) - 1)(g_i(x_2) - g_i(x_1)) + g_i(x_2) - g_i(x_1)\right) && \text{(3.9f)}\\
& \leq N \log\left(1 + \frac{\sum_{j=1}^{N} \beta_j(z, x_2)}{N^2}\sum_{i=1}^{N} \underbrace{|N\alpha_i(z, x_2) - 1|}_{\leq \nu} |g_i(x_2) - g_i(x_1)| + g_i(x_2) - g_i(x_1)\right) && \text{(3.9g)}\\
& \leq N \log\left(1 + \frac{\sum_{j=1}^{N} \beta_j(z, x_2)}{N^2}\left(\nu \underbrace{\sum_{i=1}^{N} |g_i(x_2) - g_i(x_1)|}_{\leq N\Delta} + \sum_{i=1}^{N} g_i(x_2) - g_i(x_1)\right)\right) && \text{(3.9h)}\\
& \leq N \log\left(1 + \frac{\sum_{j=1}^{N} \beta_j(z, x_2)}{N^2}\left(N\nu\Delta + \sum_{i=1}^{N}(g_i(x_2) - g_i(x_1))\right)\right) && \text{(3.9i)}\\
& = N \log\left(1 + \frac{\sum_{j=1}^{N} \beta_j(z, x_2)}{N}\left(\nu\Delta + \frac{1}{N}\sum_{i=1}^{N}(g_i(x_2) - g_i(x_1))\right)\right) && \text{(3.9j)}\\
& = N \log\left(1 + \frac{\sum_{j=1}^{N} \beta_j(z, x_2)}{N}\left(\nu\Delta + J_m(x_2) - J_m(x_1)\right)\right) && \text{(3.9k)}\\
& && \text{(3.9l)}
\end{aligned}
$$

The bound on the sum of β_j depends on the signal of $\nu\Delta + J_m(x_2) - J_m(x_1)$, as it decides whether β_j should be made large or small. This can be written as:

$$
H(z, x_1) - H(z, x_2) \leq N \log\left(1 + \max\left\{\frac{\nu\Delta + J_m(x_2) - J_m(x_1)}{z - \lceil g(x_2)\rceil}, \frac{\nu\Delta + J_m(x_2) - J_m(x_1)}{z - \lfloor g(x_2)\rfloor}\right\}\right) \tag{3.10}
$$

or, alternatively,

$$H(z, x_1) - H(z, x_2) \leq N \log\left(1 + \frac{\nu\Delta + |J_m(x_2) - J_m(x_1)|}{z - \lceil g(x_2) \rceil}\right). \tag{3.11}$$

■

3.1.2 Hypervolume bound for optimal mean loss

To use the bound proven in Lemma 1, we first have to prove the inequality in Eq. (3.4) holds for the space considered.

Lemma 2. *Let $\mathcal{X}$ be a set and let $\mathcal{G} = \{g_1, \ldots, g_N \mid g_i \colon \mathcal{X} \to \mathbb{R}\}$ be a set of functions. Let $x \in \mathcal{X}$ and $\mathbb{R} \ni \nu > 0$. Let $\beta_i(z, x) \coloneqq \frac{1}{z - g_i(x)}$ and $\alpha_i(z, x) \coloneqq \frac{\beta_i(z,x)}{\sum_{j=1}^{N} \beta_j(z,x)}$. Let $\lfloor g(x) \rfloor \leq g(x) \leq \lceil g(x) \rceil$ for all $l \in \mathcal{G}$. Then*

$$|N\alpha_i(z, x) - 1| \leq \nu \tag{3.12}$$

for all $i \in [N]$ and $z > \gamma$, where

$$\gamma = \max\left\{\lceil g(x) \rceil, \frac{(1+\nu)\lceil g(x) \rceil - \lfloor g(x) \rfloor}{\nu}, \frac{\lceil g(x) \rceil - (1-\nu)\lfloor g(x) \rfloor}{\nu}\right\}. \tag{3.13}$$

Proof. For Eq. (3.12) to hold, we must have

$$N\alpha_i(z, x) - 1 \leq N \max_{i \in [N]} \alpha_i(z, x) - 1 \leq \nu \tag{3.14a}$$

$$N\alpha_i(z, x) - 1 \geq N \min_{i \in [N]} \alpha_i(z, x) - 1 \geq -\nu. \tag{3.14b}$$

Using the bounds $\lfloor g(x) \rfloor$ and $\lceil g(x) \rceil$, we can bound $\beta_i(\delta)$ as:

$$\max_{i \in [N]} \beta_i(z, x) \leq \frac{1}{z - \lceil g(x) \rceil}, \quad \min_{i \in [N]} \beta_i(z, x) \geq \frac{1}{z - \lfloor g(x) \rfloor}, \tag{3.15}$$

and $\alpha_i(\delta)$ as

$$\max_{i \in [N]} \alpha_i(z, x) \leq \frac{\max_{i \in [N]} \beta_i(z, x)}{N \min_{i \in [N]} \beta_i(z, x)} \leq \frac{\frac{1}{z - \lceil g(x) \rceil}}{N \frac{1}{z - \lfloor g(x) \rfloor}} \tag{3.16a}$$

$$\min_{i \in [N]} \alpha_i(z, x) \geq \frac{\min_{i \in [N]} \beta_i(z, x)}{N \max_{i \in [N]} \beta_i(z, x)} \geq \frac{\frac{1}{z - \lfloor g(x) \rfloor}}{N \frac{1}{z - \lceil g(x) \rceil}} \tag{3.16b}$$

Hence, we have that Eq. (3.14a) can be satisfied by:

$$\frac{\frac{1}{z-\lceil g(x)\rceil}}{N\frac{1}{z-\lfloor g(x)\rfloor}} \leq \frac{1+\nu}{N} \Rightarrow \frac{(1+\nu)\lceil g(x)\rceil - \lfloor g(x)\rfloor}{\nu} \leq z \tag{3.17}$$

and Eq. (3.14b) can be satisfied by:

$$\frac{\frac{1}{z-\lfloor g(x)\rfloor}}{N\frac{1}{z-\lceil g(x)\rceil}} \geq \frac{1-\nu}{N} \Rightarrow \frac{\lceil g(x)\rceil - (1-\nu)\lfloor g(x)\rfloor}{\nu} \leq z. \tag{3.18}$$

The additional value in the definition of γ guarantees that z does not become invalid. ■

Using this bound and Lemma 1, we can bound the hypervolume improvement possible over the value achieved by an optimal solution to the mean loss in its neighborhood:

Theorem 1 (Hypervolume bound from mean loss optimality)**.** *Let $\mathcal{X}$ be a set and let $\mathcal{G} = \{g_1, \ldots, g_N \mid g_i \colon \mathcal{X} \to \mathbb{R}\}$ be a set of functions. Let J_m and H be as in Definitions 8 and 9, respectively. Let $x^* \in \mathcal{X}$ be a local minimum of J_m with neighborhood ϵ. Let $\delta \in (0, \epsilon]$ and $\mathbb{R} \ni \nu > 0$. Let $\mathcal{X}_\delta = \{x \in \mathcal{X} \mid d(x, x^*) < \delta\}$. Let $g(x) \leq \lceil g\rceil$ and $\sum_{j=1}^{N} |g_j(x) - g_j(x^*)| \leq N\Delta$ for all $l \in \mathcal{G}$ and $x \in \mathcal{X}_\delta$. Let $\lfloor g(x^*)\rfloor \leq g(x^*) \leq \lceil g(x^*)\rceil$ for all $l \in \mathcal{G}$. Then*

$$H(z, x) \leq H(z, x^*) + N \log\left(1 + \frac{\nu\Delta + J_m(x) - J_m(x^*)}{z - \lceil g(x^*)\rceil}\right) \tag{3.19}$$

and

$$H(z, x) \leq H(z, x^*) + N \log\left(1 + \frac{(1+\nu)\Delta}{z - \lceil g(x^*)\rceil}\right) \tag{3.20}$$

for all $x \in \mathcal{X}_\delta$ and $z > \gamma$, where

$$\gamma = \max\left\{\lceil g\rceil, \frac{(1+\nu)\lceil g(x^*)\rceil - \lfloor g(x^*)\rfloor}{\nu}, \frac{\lceil g(x^*)\rceil - (1-\nu)\lfloor g(x^*)\rfloor}{\nu}\right\}. \tag{3.21}$$

Proof. Let x^* and ν be used in Lemma 2. Then we know that

$$|N\alpha_i(z, x^*) - 1| \leq \nu \tag{3.22}$$

for all $i \in [N]$ and $z > \gamma'$, where

$$\gamma' = \max\left\{\lceil g(x^*)\rceil, \frac{(1+\nu)\lceil g(x^*)\rceil - \lfloor g(x^*)\rfloor}{\nu}, \frac{\lceil g(x^*)\rceil - (1-\nu)\lfloor g(x^*)\rfloor}{\nu}\right\}. \tag{3.23}$$

Let $\gamma = \max\{\gamma', \lceil g\rceil\}$ so that $z > \gamma$ is always valid for the ν bound and for hypervolume computation. Since $\lceil g\rceil \geq \lceil g(x^*)\rceil$, the value $\lceil g(x^*)\rceil$ from γ' can be dropped when computing

γ.

Let $x_2 = x^*$ and $x_1 = x \in \mathcal{X}_\delta$ in Lemma 1. Then we have that

$$H(z,x) - H(z,x^*) \leq N \log\left(1 + \frac{\nu\Delta + |J_m(x^*) - J_m(x)|}{z - \lceil g(x^*) \rceil}\right). \tag{3.24}$$

Since $J_m(x^*) \leq J_m(x)$ due to the optimality, the bound can be written as

$$H(z,x) \leq H(z,x^*) + N \log\left(1 + \frac{\nu\Delta + J_m(x) - J_m(x^*)}{z - \lceil g(x^*) \rceil}\right). \tag{3.25}$$

Since $|J_m(x^*) - J_m(x)| \leq \frac{1}{N}\sum_{i=1}^{N} |g_i(x^*) - g_i(x)| \leq \Delta$, the bound can be rewritten as

$$H(z,x) \leq H(z,x^*) + N \log\left(1 + \frac{(1+\nu)\Delta}{z - \lceil g(x^*) \rceil}\right). \tag{3.26}$$

■

3.1.3 Mean loss bound for optimal hypervolume

Similarly to Lemma 2, we have to find another bound to Eq. (3.4) in order to use Lemma 1:

Lemma 3. *Let $\mathcal{X}$ be a set and let $\mathcal{G} = \{g_1, \ldots, g_N \mid g_i \colon \mathcal{X} \to \mathbb{R}\}$ be a set of functions. Let $\beta_i(z,x) \coloneqq \frac{1}{z - g_i(x)}$ and $\alpha_i(z,x) \coloneqq \frac{\beta_i(z,x)}{\sum_{j=1}^{N} \beta_j(z,x)}$. Let $\lfloor g \rfloor \leq g(x) \leq \lceil g \rceil$ for all $l \in \mathcal{G}$ and $x \in \mathcal{X}$. Then*

$$|N\alpha_i(z,x) - 1| \leq \nu \tag{3.27}$$

for all $i \in [N]$, $x \in \mathcal{X}$ and $z > \lceil g \rceil$, where

$$\nu = \max\left\{\frac{z - \lfloor g \rfloor}{z - \lceil g \rceil} - 1, 1 - \frac{z - \lceil g \rceil}{z - \lfloor g \rfloor}\right\}. \tag{3.28}$$

Proof. For Eq. (3.27) to hold, we must satisfy the conditions in Eq. (3.14). Using the bounds $\lfloor g \rfloor$ and $\lceil g \rceil$, we can bound $\beta_i(z,x)$ as in Eq. (3.15). Hence, we have that Eq. (3.17) and Eq. (3.18) can be satisfied by:

$$\frac{z - \lfloor g \rfloor}{z - \lceil g \rceil} - 1 \leq \nu, \quad 1 - \frac{z - \lceil g \rceil}{z - \lfloor g \rfloor} \leq \nu. \tag{3.29}$$

■

Using this lemma, we can bound how much lower the mean loss can get around an optimal solution to the hypervolume:

Theorem 2 (Mean loss bound from hypervolume optimality)**.** *Let $\mathcal{X}$ be a set and let $\mathcal{G} = \{g_1, \ldots, g_N \mid g_i \colon \mathcal{X} \to \mathbb{R}\}$ be a set of functions. Let J_m and H be as in Definitions 8 and 9, respectively. Let $x^* \in \mathcal{X}$ be a local maximum of H with neighborhood ϵ for some $z \in \mathbb{R}$. Let $\delta \in (0, \epsilon]$ and $\mathcal{X}_\delta = \{x \in \mathcal{X} \mid d(x, x^*) < \delta\}$ such that $H(z, x)$ is defined for all $x \in \mathcal{X}_\delta$. Let $\lfloor g \rfloor \leq g_i(x) \leq \lceil g \rceil$ and $\sum_{j=1}^{N} |g_j(x) - g_j(x^*)| \leq N\Delta$ for all $i \in [N]$ and $x \in \mathcal{X}_\delta$. Then*

$$J_m(x^*) - \nu\Delta \leq J_m(x) \tag{3.30}$$

for all $x \in \mathcal{X}_\delta$, where

$$\nu = \max\left\{\frac{z - \lfloor g \rfloor}{z - \lceil g \rceil} - 1, 1 - \frac{z - \lceil g \rceil}{z - \lfloor g \rfloor}\right\}. \tag{3.31}$$

Proof. Let $\mathcal{X}_\delta$ be the set considered in Lemma 3. Since $\nu(z') < \nu(z)$ for all $z' > z$, the bound is valid for all $z' > z$.

Let $x_1 = x^*$ and $x_2 = x \in \mathcal{X}_\delta$ in Lemma 1. Then we have that

$$H(z, x^*) - H(z, x) \leq N \log\left(1 + \max\left\{\frac{\nu\Delta + J_m(x) - J_m(x^*)}{z - \lceil g \rceil}, \frac{\nu\Delta + J_m(x) - J_m(x^*)}{z - \lfloor g \rfloor}\right\}\right). \tag{3.32}$$

From the optimality of x^*, the bound can be expanded as

$$0 \leq \log\left(1 + \max\left\{\frac{\nu\Delta + J_m(x) - J_m(x^*)}{z - \lceil g \rceil}, \frac{\nu\Delta + J_m(x) - J_m(x^*)}{z - \lfloor g \rfloor}\right\}\right) \tag{3.33a}$$

$$0 \leq \max\left\{\frac{\nu\Delta + J_m(x) - J_m(x^*)}{z - \lceil g \rceil}, \frac{\nu\Delta + J_m(x) - J_m(x^*)}{z - \lfloor g \rfloor}\right\} \tag{3.33b}$$

Assuming that either term is the maximum provides the same bound:

$$J_m(x^*) - \nu\Delta \leq J_m(x). \tag{3.34}$$

■

3.2 Adjustable weights in mean loss

One of the standard ways to tackle a multi-objective optimization problem is to transform it into a single-objective problem by considering each objective as a loss function to be minimized and taking a weighted mean loss [7], with the weights representing the importance of a given loss compared to the others.

We know that, if the losses are convex, any point in the Pareto frontier can be achieved by an appropriate choice of weights [7] and Section 3.2.1 will show how these weights must be chosen to coincide with the optimal solution to the hypervolume for a given reference z. Since the hypervolume is known to provide solutions with reasonable trade-offs between the objectives [11], Section 3.2.2 will go one step further and apply those weights during the optimization in an image classification problem using a neural network, evaluating the potential efficacy for non-convex losses.

3.2.1 Hypervolume and weighted mean on convex losses

For convex problems, one can define optimality of a solution based on the subgradient:

Definition 10 (Subgradient)**.** Let $f\colon \mathcal{X} \to \mathbb{R}$ be a real-valued convex function defined on a convex open set in the Euclidean space $\mathcal{X} \subseteq \mathbb{R}^n$. Then a vector $v \in \mathcal{X}$ in that space is called a subgradient at a point $x_0 \in \mathcal{X}$ if for any $x \in \mathcal{X}$ one has

$$f(x) - f(x_0) \geq v^T(x - x_0). \tag{3.35}$$

The set of all subgradients at x_0 is called the subdifferential at x_0 and is denoted as $\partial f(x_0)$. The subdifferential is always a nonempty convex compact set. Moreover, x^* is a global minimum of f if and only if zero is contained in the subdifferential, that is, $0 \in \partial f(x^*)$.

We can also extend Definition 8 to include weights in the mean loss:

Definition 11 (Weighted Mean Loss)**.** Let $\mathcal{X}$ be a set and let $\mathcal{G} = \{g_1, \ldots, g_N \mid g_i\colon \mathcal{X} \to \mathbb{R}\}$ be a set of functions. Let $\mathbb{R}^+ = \{v \in \mathbb{R} \mid v \geq 0\}$. Then the weighted mean loss is a function $J_w\colon \mathcal{X} \times \mathbb{R}^{+N} \to \mathbb{R}$ defined by

$$J_w(x, w) \coloneqq \sum_{i=1}^{N} w_i g_i(x). \tag{3.36}$$

Using these two definitions, we can establish which weights in the weighted mean loss would provide the same solution that the hypervolume provides:

Theorem 3. *Let $\mathcal{X}$ be a convex open set in the Euclidean space $\mathbb{R}^n$. Let $\mathcal{G} = \{g_1, \ldots, g_N \mid g_i\colon \mathcal{X} \to \mathbb{R}\}$, be a set of real-valued convex functions. Let J_w and H be as in Definitions 11 and 9, respectively. Then $x^* \in \mathcal{X}$ is a local maximum of H for some z if and only if it is also a local minimum of J_w with $w_i = \frac{1}{z - g_i(x^*)}$.*

Proof. To prove this theorem, we will show that the subdifferentials of the weighted mean loss

and the negative of the hypervolume are the same at x^*. Then, from Definition 10 and from the convexity, if the null vector is one of the subdifferentials it must also be in the other and the point is optimal for both functions.

From the linearity of the subgradient, we find that the subdifferential for the weighted mean is given by

$$\partial_x J_w(x^*, w) = \left\{ \sum_{i=1}^{N} w_i v_i \mid v_i \in \partial g_i(x^*) \right\}. \tag{3.37}$$

Since the logarithm is a differentiable function, we can use the chain rule on the subgradient to find that

$$\partial \log(z - g_i(x^*)) = \left\{ -\frac{1}{z - g_i(x^*)} v_i \mid v_i \in \partial g_i(x^*) \right\} \tag{3.38a}$$

$$= \{ -w_i v_i \mid v_i \in \partial g_i(x^*) \}. \tag{3.38b}$$

Using this subdifferential to compute the hypervolume's subdifferential, we get

$$\partial_x H(z, x^*) = \left\{ \sum_{i=1}^{N} u_i \mid u_i \in \partial \log(z - g_i(x^*)) \right\} \tag{3.39a}$$

$$= \left\{ -\sum_{i=1}^{N} w_i v_i \mid v_i \in \partial g_i(x^*) \right\} \tag{3.39b}$$

$$= \{ -d \mid d \in \partial_x J_w(x^*, w) \}, \tag{3.39c}$$

which shows that the hypervolume and weighted mean loss have opposite subdifferentials, proving the result. ■

3.2.2 Case study: optimizing neural networks for digit recognition

Based on the previous result, we see that the weight of a loss is directly linked to how well it performs. If we define the objectives as $g_i(x) = l(x, s_i)$, where $l \colon \mathcal{X} \times \mathcal{S} \to \mathbb{R}$ represents a base loss function and $s_i \in \mathcal{S}$ represents a sample, we can apply the MOO results to areas such as machine learning. In particular, one objective is linked to one sample, which would make it impractical to apply standard MOO techniques since they cannot handle large number of objectives efficiently [18]. However, interpreting the problem as SOO with focus on a single solution makes it feasible, but loses the information about the different objectives. A middle ground is to use a MOO perspective but optimize for a single solution, taking advantage if the efficient methods defined for SOO.

With the weight connection defined in the last section, we can bring some information from the MOO perspective because, since defining the weights as

$$w_i = \frac{1}{z - l(x, s_i)}, \tag{3.40}$$

causes a sample to have higher weight if the current model x has a higher loss to it. So instead of all samples having pre-defined weights, either based on previous knowledge or the naive interpretation that all contribute equally, we can use this definition of weights to automatically focus more on samples with higher losses. We conjecture that this will guide the model towards a more balanced solution, providing better generalization instead of focusing too much on the samples that the model can fit easily.

In order to make the optimization comparable to minimizing the mean loss and avoid faster convergence just due to bigger steps on the same losses, we also normalize the weights such that

$$w'_i = \frac{w_i}{\sum_{j=1}^{N} w_j} \tag{3.41}$$

and $\sum_{i=1}^{N} w'_i = 1$, which is the same normalization described by the mean loss, since $w_i = \frac{1}{N}$ when comparing Definitions 8 and 11.

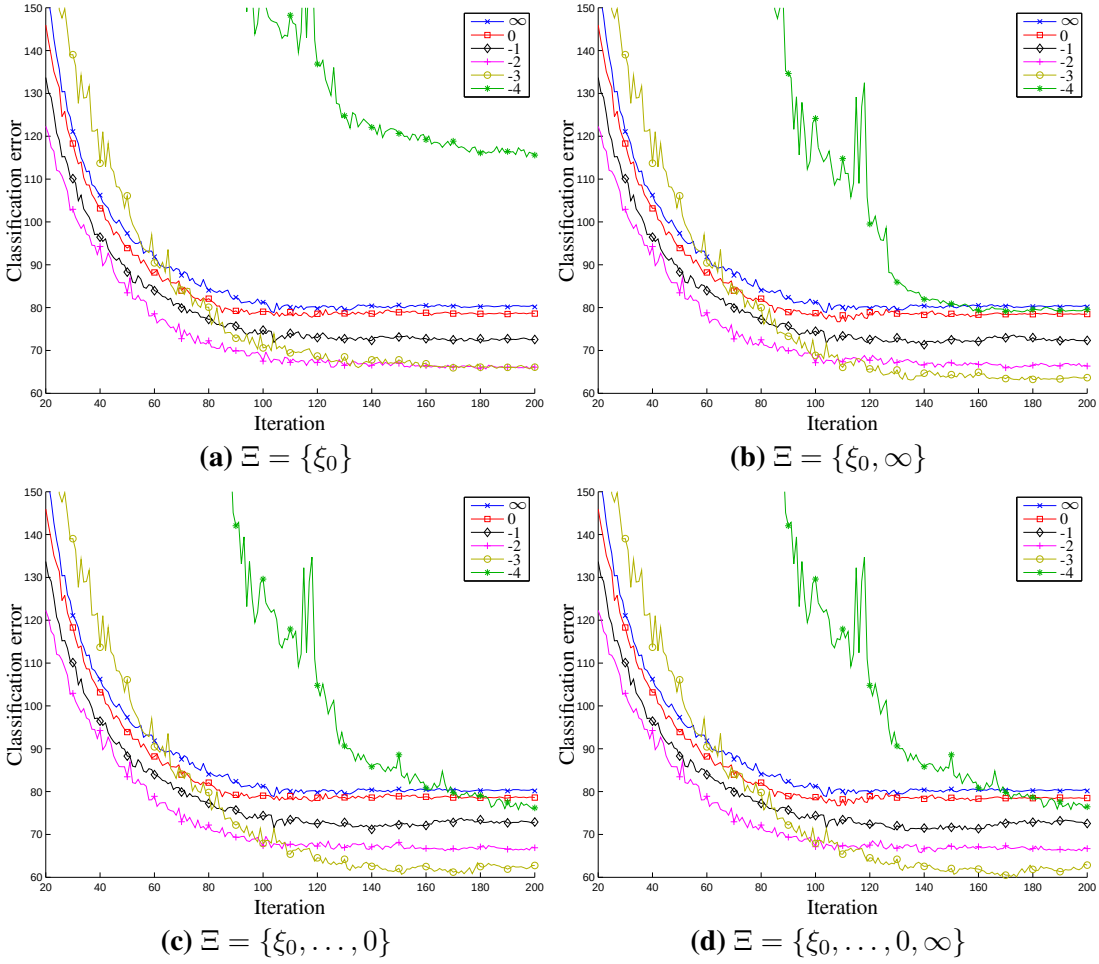
To experiment with using the single-solution hypervolume instead of the mean loss for training neural networks, we used a LeNet-like network on the MNIST dataset, known to be noisy but tractable. This network is composed of three layers, all with ReLU activation, where the first two layers are convolutions with 20 and 50 filters, respectively, of size 5x5, both followed by max-pooling of size 2x2, while the last layer is fully connected and composed of 500 hidden units with dropout probability of 0.5.

The learning was performed by gradient descent with base learning rate of 0.1 and momentum of 0.9, which were selected using the validation set to provide the best performance for the mean loss optimization, the baseline for the experiment, and mini-batches of 500 samples. After 20 iterations without improvement on the validation error, the learning rate is reduced by a factor of 0.1 until it reaches the value of 0.001, after which it is kept constant until 200 iterations occurred.

For the hypervolume, instead of fixing a single value for z, which would require it to be large as the neural network has high loss when initialized, we allowed z to change on each iteration, being defined by

$$z = (1 + 10^{\xi}) \max_i g_i(x) \tag{3.42}$$

so that it can follow the improvement on the loss functions, where i are the samples in the



(a) $\Xi = \{\xi_0\}$ **(b)** $\Xi = \{\xi_0, \infty\}$

(c) $\Xi = \{\xi_0, \ldots, 0\}$ **(d)** $\Xi = \{\xi_0, \ldots, 0, \infty\}$

Figure 3.1: Mean number of misclassified samples in the test set over 20 runs for different initial values ξ_0 and training strategies, with $\xi = \infty$ representing the mean loss and Ξ representing the schedule of values of ξ used when no improvement is observed.

mini-batch and the parameters' gradients are not backpropagated through z. Any value $\xi \in \Xi \subseteq \mathbb{R} \cup \{\infty\}$ provides a valid reference point and larger values make the problem closer to using the mean loss. We tested $\Xi = \{-4, -3, -2, -1, 0, \infty\}$, where $\xi = \infty$ represents the mean loss, and allowed for scheduling of ξ. In this case, before decreasing the learning rate when the learning stalled, ξ is incremented to the next value in Ξ. We have also considered the possibility of $\infty \notin \Xi$, to evaluate the effect of not using the mean together with the schedule. Note again that we use the same reference z for all losses, as discussed in Section 3.1, and that this assumption makes sense in this case as we want to compare all samples to the same standard.

Figure 3.1 shows the results for each scenario considered. First, note that using $\xi_0 = 0$ provided results close to the mean loss throughout the iterations on all scenarios, which empirically validates the theory that large values of z makes maximization of the hypervolume similar to minimization of the mean loss and provides evidence that z does not have to be so large in comparison to the loss functions for this to happen. Moreover, Figs. 3.1c and 3.1d are similar

for all values of ξ_0, which provides further evidence that $\xi = 0$ is large enough to approximate the mean loss well, as including $\xi = \infty$ in the schedule does not change the performance.

On the other hand, $\xi_0 = -4$ was not able to provide good classification by itself, requiring the use of other values of ξ to achieve an error rate similar to the mean loss. Although it is able to get better results with the help of schedule, as shown in Figs. 3.1c and 3.1d, this is due to the other values of ξ themselves instead of $\xi_0 = -4$ providing direct benefit, as it achieved an error similar to the mean loss when no schedule except for $\xi = \infty$ was used, as shown in Fig. 3.1b. This indicates that too much pressure on the samples with high loss is not beneficial, which is explained by the optimization becoming close to minimizing the maximum loss as only the one with highest weight will effectively impact the loss, thus ignoring most of the samples and trying too hard to fit noisy ones, focusing on the worst case.

When optimizing the hypervolume starting from $\xi_0 \in \{-1, -2, -3\}$, all scenarios showed improvements on the classification error, with all the differences after convergence between optimizing the hypervolume and the mean loss being statistically significant. Moreover, better results were obtained when the schedule started from a smaller value of ξ_0. This provides evidence to the conjecture that placing higher pressure on samples with high loss, which is represented by higher values of w_i in Eq. (3.36), is beneficial and might help the network to achieve higher generalization, but also warns that too much pressure can be prejudicial as the results for $\xi_0 = -4$ show.

Furthermore, Fig. 3.1 indicates that, even if this pressure is kept throughout the training, it might improve the results compared with using only the mean loss, but that reducing the pressure as the training progresses improves the results. We suspect that reducing the pressure allows rare samples that cannot be well learned by the model to be less relevant in favor of more common samples, which improves the generalization overall, and that the initial pressure allowed the model to learn better representations for the data from the start, as it was forced to take more into account the bad samples, instead of exploiting easy gains. The presence of these rare and bad samples also explain why $\xi_0 = -4$ provided bad results, as the learning algorithm focused mainly on samples that cannot be appropriately learnt by the model instead of focusing on the more representative ones. Thus using a schedule mechanism such as the one described here, where ξ is incremented by one every time the optimization reaches a local minima, is essential for real applications.

Table 3.1 provides the errors for the mean loss, represented by $\xi_0 = \infty$, and for hypervolume with $\xi_0 = -3$, which presented the best improvements. We used the classification error on the validation set to select the parameters used for computing the classification error on the test set. If not used alone, with either scheduling or mean or both, maximizing the hypervolume

Table 3.1: Mean number of misclassified samples in the test set over 20 runs. The differences between $\xi_0 = \infty$ and $\xi_0 = -3$ are statistically significant ($p \ll 0.001$).

ξ_0	Schedule	Mean	Errors	Reduction
∞			80.8	
-3	X	X	67.5	16.5%
-3	X	✓	64.2	20.5%
-3	✓	X	62.9	22.2%
-3	✓	✓	63.4	21.6%

leads to a reduction of at least 20% in the classification error without changing the convergence time significantly, as observed in Fig. 3.1, which motivates its use in real problems.

3.3 Conclusion

In this chapter, we introduced the idea of using the hypervolume with a single solution as an optimization objective and presented a theory for its use. We showed how an optimal solution for the hypervolume relates to the mean loss problem and vice-versa, providing bounds on the neighborhood of the optimal point. Using these bounds, we can transfer information and results from one problem to the other, opening an important door to more fundamental studies on multi-objective optimization, which has mostly experimental results and whose theory has not been so thoroughly explored.

We also showed a link between the weights in a weighted mean loss and the reference point and losses when optimizing the hypervolume. This analysis raised the conjecture that using the hypervolume in machine learning might result in better models, as the hypervolume's gradient is composed of an automatically weighted average of the gradient for each sample, with higher weights representing higher losses. This weighting makes the learning algorithm focus more on samples that are not well represented by the current set of parameters, indicated by a higher loss for the sample, even if it means a slower reduction of the mean loss. Hence, it forces the learning algorithm to search for regions where all samples can be well represented, avoiding early commitment to less promising regions.

The conjecture was validated in an experiment with MNIST, where using the hypervolume maximization led to a reduction of 20% in the classification errors in comparison to the mean loss minimization. We also showed how the gradient of the hypervolume behaves when changing the reference point and how to stabilize it for practical applications.

Future research should focus on studying more theoretical and empirical properties of the single-solution hypervolume maximization, to provide a solid explanation for its improvement

over the mean loss and in which scenarios this could be expected. The robustness of the method should also be investigated, as too much noise or the presence of outliers might cause large losses, which opens the possibility of inducing the learner to place high importance on these losses in detriment of more common cases.

Another direction is defining bounds for the hypervolume as more points in the Pareto frontier are considered. This poses the problem of defining the bounds themselves as more points are considered, which could leverage the theoretical research on the hypervolume as a metric. It also raises the question of how to associate an optimal solution for a single-objective problem to the multiple elements of an optimal set for a multi-objective problem.

Chapter 4

Multi-solution hypervolume optimization

Local search methods have been successful in single-objective optimization (SOO) due to their efficiency in finding a local optimum for some problems [19, 20], so that research has been performed to try to adapt these methods for MOO problems. For instance, [21] defined a method for finding all minimizing directions in a MOO problem, but the proposed algorithm achieved low performance on usual benchmark functions.

Alternatively, instead of adapting the single-objective methods to work on MOO problems, we can create a SOO problem associated with the MOO one, such that a good solution for the single-objective case is a good solution for the multi-objective case. Since the hypervolume is able to describe how good a population is, based on a single indicator, the MOO problem can be converted into the maximization of the population's hypervolume.

Based on this idea, [22] proposed a method to compute the hypervolume's gradient for a given population, so that the optimal search direction for each individual could be established. However, [23] showed that adjusting the population through integration of the hypervolume's gradient not always work, with some initially non-dominated points becoming dominated and others changing very little over the integration.

In this chapter, we introduce an algorithm for maximizing the hypervolume by optimizing one point at a time, instead of adjusting a whole population at once. The algorithm alternates between exploring the space for non-dominated solutions and, when they are found, exploiting them using local search methods to maximize the populations' hypervolume when only this active point can be moved. Therefore, once the hypervolume has converged, which is guaranteed to happen because the problem is bounded, the point is fixed in all further iterations. We found that this restriction is enough to overcome the issues presented in [23] when using

the hypervolume's gradient. The proposed algorithm, called Hybrid Hypervolume Maximization Algorithm (H2MA), is a hybrid one, since it is composed of global exploration and local exploitation procedures, properly managed to be executed alternately.

Results over the ZDT benchmark [24] show that the new algorithm performs better than the reference evolutionary algorithms, both in terms of total hypervolume and distance to the Pareto frontier. Moreover, the algorithm was able to work deterministically in most of the benchmark problems, which makes it less susceptible to variations due to random number generation. Due to the high quality of the solutions found in less function evaluations than what is achieved by the current reference, we consider that the new algorithm might be a viable choice for solving MOO problems. Moreover, since a single solution is introduced at a time, the user is able to stop the algorithm when the desired number of solutions is found, while evolutionary algorithms must evolve the whole population at the same time.

This chapter is organized as follows. Section 4.1 discusses the problems with the gradient-based approach for hypervolume maximization introduced in [23]. Section 4.2 provides the details of the new H2MA algorithm, and Section 4.3 shows the comparison with the reference algorithms. Finally, Section 4.4 summarizes the results and discusses future research direction.

4.1 Gradient of the Hypervolume

As stated earlier, since the hypervolume provides such a good indicator of performance in multi-objective problems, it can be used to transform the multi-objective problem into a single-objective one, characterized by the maximization of the hypervolume.

Although such approach proved to be successful when using evolutionary algorithms as the optimization method [12], the same did not happen when using the hypervolume's gradient to perform the optimization [23]. However, it is well-known that gradient methods have been successful in single-objective optimization [19, 20], thus suggesting that they should be a reasonable choice for multi-objective optimization devoted to maximizing the hypervolume, since the hypervolume operator is well-defined almost everywhere in the objective space.

The hypervolume's gradient for a set of points was introduced in [22], and it can be used to compute the optimal direction in which a given point should move to increase the hypervolume associated with the current set of non-dominated solutions. Although the hypervolume is not a continuously differentiable function of its arguments, the gradient can be computed whenever any two points have different values for all objectives.

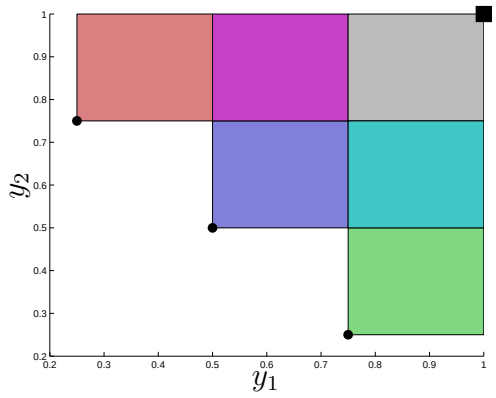


Figure 2.1: Example of hypervolume. The non-dominated solutions in the objective space are shown in black circles, and the reference point is shown in the black square. For each non-dominated solution, the region between it and the reference point is filled, with colors combining when there is overlap, and the total hypervolume is given by the area of the shaded regions. Best viewed in color. (repeated from page 18)

Based on this motivation, [23] used the hypervolume's gradient as a guide for adjusting a set of points by numerical integration, that is, performing a small step in the direction pointed by the gradient. Even though the algorithm was able to achieve the Pareto set in some cases, it failed to converge to efficient solutions when some points got stuck along the iterative process, either because their gradients became very small or because they became dominated by other candidates. Once dominated, these points do not contribute to the hypervolume and remain fixed. This causes a major issue to using the hypervolume gradient in practice since dominated points can be discarded, as there is no possibility to revert them to non-dominated points anymore, and the points with small gradients remain almost stagnant.

If we analyze Eq. (2.2), we can see that points at the border in the objective space are the only ones that can fill some portions of the objective space. On the other hand, points that are not at the border have less influence in the hypervolume, since part of the area dominated by them is also dominated by some other points. In the analysis presented in [23], it is clear that the cases where some points got stuck had higher gradients for the border points in the objective space, which led to the dominance or decrease of contribution of some or all central points.

To make this idea clearer, consider the example in Fig. 2.1. If the point located at $(0.75, 0.25)$ decreases its value on the second objective, it can increase the population's hypervolume. Moreover, it is the only point that can do so without competition for that portion of the space, since it is the point with the largest value for the first objective. The same holds for the point at $(0.25, 0.75)$ and the first objective.

However, the point located at $(0.5, 0.5)$ has to compete with the other two points to be the sole contributor for some regions. Therefore, its effect on the hypervolume is smaller, which

leads to a smaller gradient. Furthermore, if less area is dominated by the middle point alone, which can occur during the points' adjustment as the middle one moves less, then its influence becomes even smaller and it can become dominated.

It is important to highlight that this behavior does not always happen, but can occur along the iterative process, as shown in [23]. This leads to the base hypothesis for the algorithm developed in this chapter: when using the hypervolume's gradient for optimization, the competition for increasing the hypervolume among points should be avoided.

4.2 Hybrid Hypervolume Maximization Algorithm

From the discussion in Section 4.1, one can see that the major problem when optimizing the hypervolume directly using its gradient may be the competition among points. Therefore, our proposed algorithm optimizes a single solution at a time, avoiding this competition.

Theoretically, the algorithm can be described by choosing a new point that maximizes the hypervolume when taking into account the previous points, such that its recurring equation can be written as:

$$x_t = \arg\max_{x \in \mathcal{X}} H(X_{t-1} \cup \{x\}, z), X_t = X_{t-1} \cup \{x_t\}, t \in \mathbb{N}, \tag{4.1}$$

where the initial set is given by $X_{-1} = \{\}$.

Since a single point is being optimized at a time, the optimization becomes simpler and, as we will show in Section 4.3, requires less function evaluations. Moreover, one could argue that maintaining the previous set fixed reduces the flexibility allowed in comparison with a set where all the points are being concurrently adjusted. Although this may be true, we will also show in Section 4.3 that the proposed algorithm performs well despite this loss of flexibility.

The algorithm described in Eq. (4.1) is theoretically ideal, but finding the maximum is hard in practice. Therefore, the actual algorithm proposed is shown in Fig. 4.2. This algorithm performs exploration of the objective space until a new solution that is not dominated by the previous candidate solutions is found. When it happens, the hypervolume of the whole set is larger than the hypervolume when considering only previous candidate solutions.

The new candidate solution is then exploited to maximize the total hypervolume and, after convergence, is added to the existing set. It is important to highlight that the exploitation phase cannot make the solution become dominated, since that would reduce the hypervolume in comparison with the initial condition. Therefore, the problem of points becoming dominated

Input: Objectives f
Input: Design space $\mathcal{X}$
Input: Nadir point z
Output: Set of candidate solutions X
  function HYBRIDGREEDYOPTIMIZER($f, \mathcal{X}, z$)
    $Regions, X \leftarrow$ CREATEINITIALREGION($f, \mathcal{X}$)
    while not *stop condition* **and** $|Regions| > 0$ **do**
      $R \leftarrow$ POP($Regions$) ▷ Removes the region with the largest volume
      $x' \leftarrow$ EXPLOREDETERMINISTIC($f, \mathcal{X}, R, X$)
      if x' is valid **then**
        $x \leftarrow$ EXPLOIT($f, \mathcal{X}, x', X, z$)
        $NewRegions \leftarrow$ CREATEREGIONS(R, x, f)
        $Regions \leftarrow Regions \cup NewRegions$
        $X \leftarrow X \cup \{x\}$
      end if
    end while
    while not *stop condition* **do**
      $x' \leftarrow$ EXPLORESTOCHASTIC($f, \mathcal{X}, X$)
      $x \leftarrow$ EXPLOIT($f, \mathcal{X}, x', X, z$)
      $X \leftarrow X \cup \{x\}$
    end while
    return X
  end function

Figure 4.2: Hybrid algorithm that performs deterministic and stochastic exploration until a suitable solution is found, and then exploits it.

Input: Objectives f
Input: Design space $\mathcal{X}$
Input: Current exploration region R
Input: Set of candidate solutions X
Output: New initial condition x'
  function EXPLOREDETERMINISTIC($f, \mathcal{X}, R, X$)
    $x' \leftarrow$ MEAN($R.X$)
    Minimize $\|R.mid - f(x)\|$ from x' until some candidate x is not dominated by X
    if found non-dominated x **then**
      $x' \leftarrow x$
    else
      $x' \leftarrow$ some invalid state
    end if
    return x'
  end function

Figure 4.3: A deterministic exploration is performed based on some region.

during the exploitation is avoided. Furthermore, the exploitation is a traditional single-objective optimization, so that gradient methods can be used if the decision set $\mathcal{X}$ is continuous or hill-climbing methods can be used for discrete $\mathcal{X}$.

Once the exploitation finishes, the algorithm begins the exploration phase again. The

Input: Objectives f
Input: Design space $\mathcal{X}$
Output: Set of candidate solutions X
Output: Initial exploration region R
  function CREATEINITIALREGION($f, \mathcal{X}$)
    $X \leftarrow \{\}$
    $x' \leftarrow$ MEAN($\mathcal{X}$) ▷ Gets the average candidate
    for $i = 1, \ldots, |f|$ **do**
      $x \leftarrow$ MINIMIZE($f_i, x', \mathcal{X}$)
      $X \leftarrow X \cup \{x\}$
    end for
    $R \leftarrow$ CREATEREGION(X, f)
    return $\{R\}, X$
  end function

Figure 4.4: The initial region is created from the points that minimize each objective individually.

exploration can be deterministic, based on regions of the objective space defined by previous solutions, or stochastic, where a stochastic algorithm, such as an evolutionary algorithm, is used to find the new candidate. When a non-dominated candidate is found, the algorithm turns to exploitation again.

We highlight that the deterministic exploitation algorithm proposed is based on the definition of these regions, but other deterministic methods can be used. However, the algorithm must be able to establish when it is not able to provide further improvements, so that the change to the stochastic global exploration can be made. In the algorithm shown in Fig. 4.2, regions that do not provide a valid initial condition are discarded without creating new regions, so that eventually the algorithm can switch to the stochastic global exploration.

The algorithm for deterministic exploration is shown in Fig. 4.3. It combines the points used to create a given region in order to produce an initial condition and tries to minimize the distance between its objective value and a reference point. Once a non-dominated point is found, it is returned for exploitation. Although this simple optimization provided good results without requiring many function evaluations, other methods can be used to perform this exploration. Alternatively, one can also perform a stochastic exploration instead of a deterministic one, but this may have negative effects on the performance if the information provided by the output (region R) is not used, since a global search would be required.

The first region is created by finding points that minimize each objective separately, as shown in Fig. 4.4. This establishes that the initial region will have a number of candidate solutions associated with it equal to the number of objectives, so that the solutions are at the border of the region.

Input: Current explored region R
Input: Current solution x
Input: Objectives f
Output: New exploration regions $NewRegions$
  function CREATEREGIONS(R, x, f)
    $NewRegions \leftarrow \{\}$
    for X' **in** COMBINATIONS($R.X, |R.X| - 1$) **do**
      $R' \leftarrow$ CREATEREGION($X' \cup \{x\}, f$)
      $NewRegions \leftarrow NewRegions \cup \{R'\}$
    end for
    return $NewRegions$
  end function

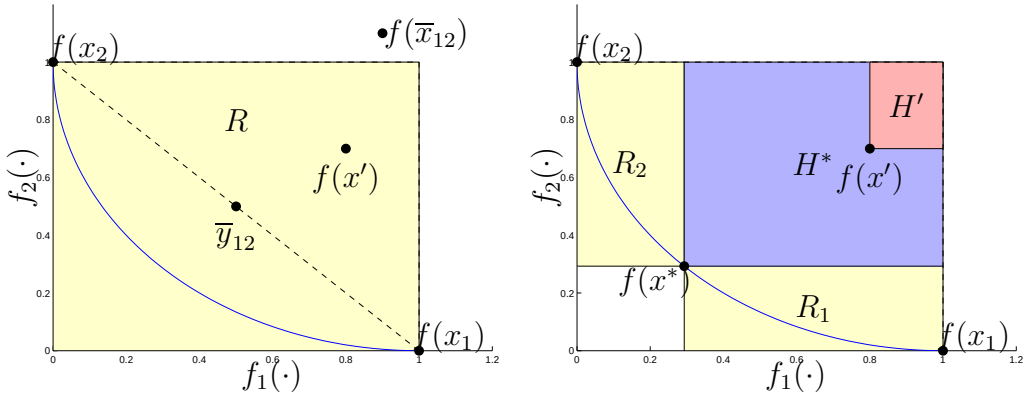
Figure 4.5: New exploration regions are created by combining the current solution with the previous region.

Input: Objectives f
Input: Set of candidate solutions X
Output: Exploration region R
  function CREATEREGION(X, f)
    $V = \prod_{i=1}^{|f|}(\max_{x \in X} f_i(x) - \min_{x \in X} f_i(x))$
    if $V > 0$ **then**
      $R.X \leftarrow X$
      $R.mid \leftarrow$ MEAN($\{f(x) \mid x \in X\}$)
      $R.V \leftarrow V$
    else
      $R \leftarrow$ null element such that $\{R\} \equiv \{\}$
    end if
    return R
  end function

Figure 4.6: An exploration region is created from a set of candidates if the region have some volume.

When new regions are created after exploitation, we ignore the solutions that created the region, one at a time, and replace it with the proposed new solution, as shown in Fig. 4.5, to create a new region. This guarantees that the number of solutions for each region is kept equal to the number of objectives.

Finally, Fig. 4.6 shows how a region is created. If a region does not have a volume, then at least one objective for two solutions is the same. Although we could allow such region to exist without modifying the rest of the algorithm, these regions tend to not provide good candidates for exploitation and delay the change to stochastic global exploration. Furthermore, one can even prohibit regions with volume smaller than some known constant, as they probably will not provide good exploitation points, and the change to stochastic global exploration happens earlier.



(a) Deterministic exploration **(b)** Exploitation

Figure 4.7: Deterministic exploration and exploitation steps of the new algorithm in an example problem. The Pareto frontier is shown in the blue line, and the regions used by the deterministic exploration are shown in yellow.

Fig. 4.7 shows a step of the algorithm in an example problem with two objectives. The deterministic exploration receives a region R, composed of the points x_1 and x_2. The mean of the points that compose the region is given by $\overline{x}_{12} = (x_1 + x_2)/2$ and its evaluation in the objective space is shown in Fig. 4.7a. The mean objective of the points that compose the region is also computed and is shown as $\overline{y}_{12} = (f(x_1) + f(x_2))/2$. The deterministic exploration is then defined by the problem

$$\min_{x \in \mathcal{X}} \|f(x) - \overline{y}_{12}\|, \tag{4.2}$$

which uses $\overline{x}_{12}$ as the initial condition for the optimization. Since $\overline{y}_{12}$ is guaranteed to be non-dominated by $f(x_1)$ and $f(x_2)$, this should guide the search to the non-dominated region of the space.

While performing this optimization, some intermediary points are evaluated, either while computing the numeric gradient or after performing a gradient descent step. The deterministic exploration stops as soon as a non-dominated point is found, which is given by $f(x')$ in the example in Fig. 4.7a. Note that this example shows $f(\overline{x}_{12})$ as being dominated by $f(x_1)$ and $f(x_2)$, but it can also be non-dominated. In this case, $x' = \overline{x}_{12}$ and no optimization step for the problem in Eq. (4.2) is performed. Supposing no non-dominated point $f(x')$ is found during the deterministic exploration, the region is simply discarded, without performing an exploitation step.

Using the point x', whose $f(x')$ is non-dominated, provided by the deterministic or stochastic exploration, the exploitation is performed. Fig. 4.7b shows the hypervolume contributions for the initial point x' and the optimal point x^*, which maximizes the total hypervolume as in Eq. (4.1). Since x' is non-dominated, its hypervolume contribution H' is positive and the hypervolume gradient relative to the objectives is non-zero. After finding x^* and if x' was pro-

vided by the deterministic exploration, new regions must be created to allow further exploration. Therefore, according to Fig. 4.5, the regions $R_1 = (x_1, x^*)$ and $R_2 = (x_2, x^*)$ are created for further exploration.

This finalizes a step of the algorithm, which is repeated until the given stop condition is not met. As at most one point is found by each step, the stop condition can be defined based on the number of desired points.

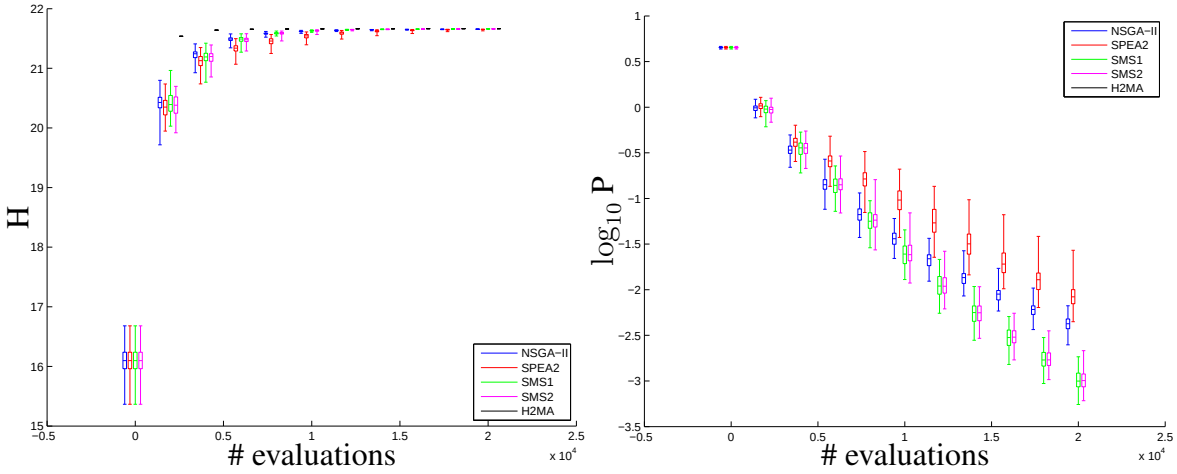
Note that all the methods used in this algorithm assume that the optimization, either for exploitation or for minimizing one objective alone, requires an initial condition. This is true for hill climbing or gradient methods, but the algorithm can easily be modified if the optimization does not require it. Furthermore, even though Fig. 4.7 shows an example for two objectives, the algorithm can be generalized for any number and each region will sill be defined by a number of points equal to the number of objectives.

4.3 Experimental Results

To compare the algorithm proposed in Section 4.2, called Hybrid Hypervolume Maximization Algorithm (H2MA), with other existing algorithms, the ZDT family of functions [24] was chosen. These functions define a common benchmark set in the multi-objective optimization literature, since they define a wide range of problems to test different characteristics of the optimization algorithm. All functions defined in [24] have a continuous decision space $\mathcal{X}$, except for the ZDT5 which has a binary space. In this chapter, only the continuous test functions were used to evaluate the performance of the new algorithm, and their equations are shown at the end of the chapter.

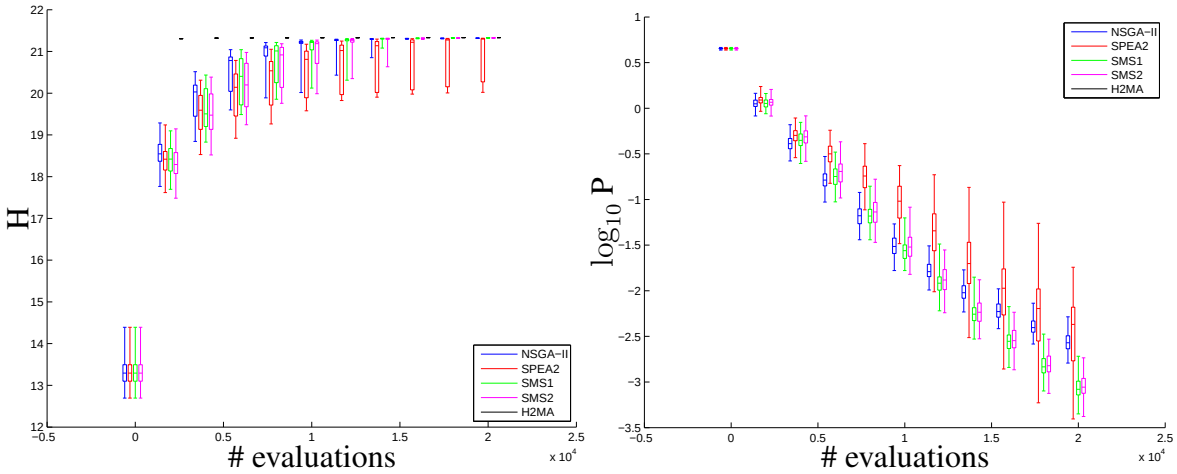
Table 4.1 provides a summary of the evaluation functions, their decision spaces, and the reference points used to compute the hypervolume. The reference points are defined by upper bounds of the objectives, which guarantees that the hypervolume computation is always valid, plus one, since not adding an extra value would mean that points at the border of the frontier would have no contribution to the hypervolume and would be avoided. In all instances, a total of $n = 30$ variables were considered, as common in the literature. The evolutionary algorithms' and evaluation functions' implementations were given by the PaGMO library [25].

Note that these reference points are not the Nadir points for each problem, usually used the reference when computing the hypervolume. The Nadir point is defined as the point with the smallest values on each coordinates that is dominated by all points in the Pareto frontier. From this definition itself, we see that we would have to know the Pareto frontier region in



(a) Hypervolume **(b)** P-distance. Zero values not shown.

Figure 4.8: 0th, 25th, 50th, 75th, and 100th percentiles every 2000 evaluations for the all algorithms on ZDT1.

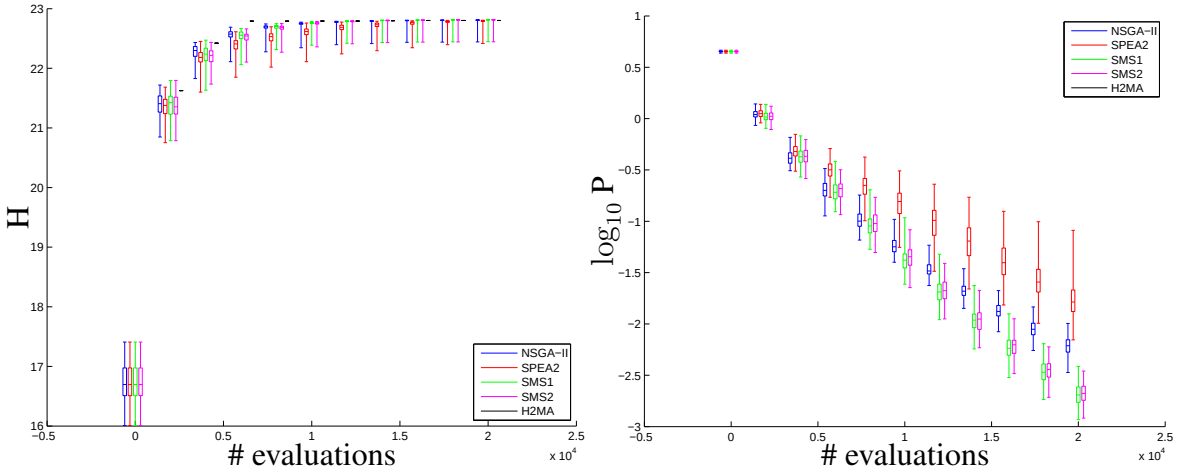


(a) Hypervolume **(b)** P-distance. Zero values not shown.

Figure 4.9: 0th, 25th, 50th, 75th, and 100th percentiles every 2000 evaluations for the all algorithms on ZDT2.

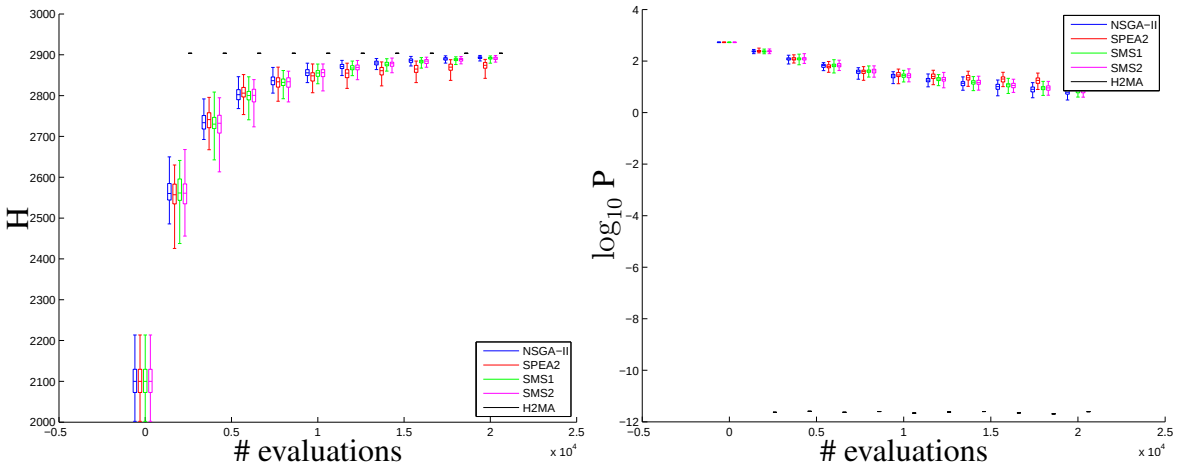
order to define the Nadir point, which would be impractical for the proposed algorithm and realistic scenarios. Therefore, we choose a reference to be used during the optimization a priori. However, following the algorithm described in Section 4.2, we first optimize each objective and then focus on exploring solutions in the areas between these first solutions, which means that the extra volume compared to the Nadir point is obtained with the first few samples and should not affect the final behavior of the algorithm significantly. Furthermore, the p-distance is not affected by the reference point, providing another quality metric for the results.

We compare our algorithm with existing reference multi-objective optimization algorithms, namely NSGA-II [8], SPEA2 [9], and SMS-EMOA [12]. All of them used a population size of 100 individuals. Tests have shown that this size is able to provide a good performance



(a) Hypervolume **(b)** P-distance. Zero values not shown.

Figure 4.10: 0th, 25th, 50th, 75th, and 100th percentiles every 2000 evaluations for the all algorithms on ZDT3.



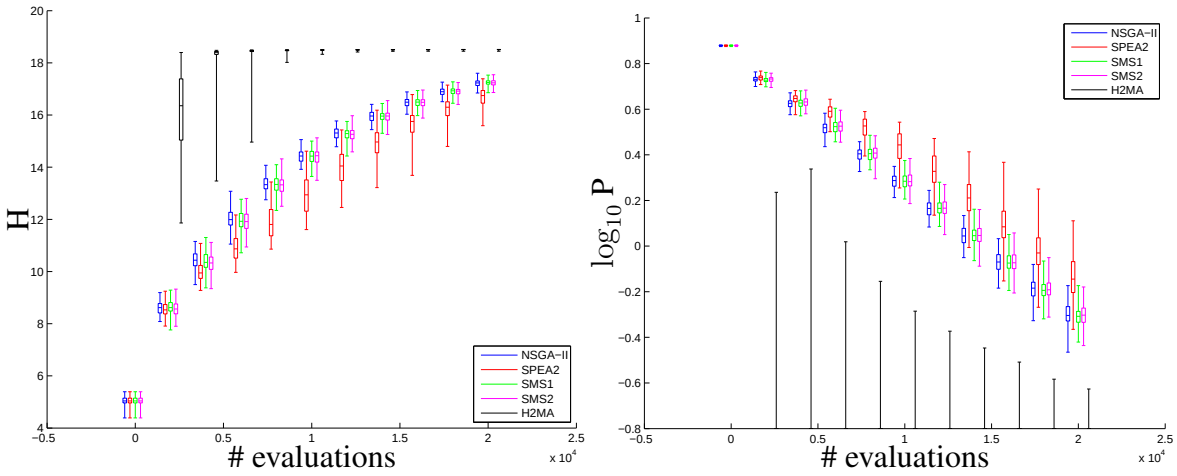
(a) Hypervolume **(b)** P-distance. Zero values not shown.

Figure 4.11: 0th, 25th, 50th, 75th, and 100th percentiles every 2000 evaluations for the all algorithms on ZDT4.

Table 4.1: Benchmark problems used for evaluation. See the Appendix for Eqs. (4.3) to (4.7).

Problem	Objectives	$\mathcal{X}$	Reference point
ZDT1	Eq. (4.3)	$[0,1]^n$	$(2, 11)$
ZDT2	Eq. (4.4)	$[0,1]^n$	$(2, 11)$
ZDT3	Eq. (4.5)	$[0,1]^n$	$(2, 11)$
ZDT4	Eq. (4.6)	$[0,1] \times [-5,5]^{n-1}$	$(2, 2 + 50(n-1))$
ZDT6	Eq. (4.7)	$[0,1]^n$	$(2, 11)$

due to balance between exploration of the space and exploitation of the individuals, with fewer individuals not providing good exploration and more not providing good exploitation. The SMS-EMOA can use two methods for selecting points in dominated fronts: the least hypervolume contribution or the domination count. Both methods were tested, with labels SMS1 and



(a) Hypervolume **(b)** P-distance. Zero values not shown.

Figure 4.12: 0th, 25th, 50th, 75th, and 100th percentiles every 2000 evaluations for the all algorithms on ZDT6.

SMS2, respectively, in the results. Note that this method only applies for the dominated fronts, since the domination count is zero for all points in the non-dominated front and the least contributor method must be used. Furthermore, the SMS-EMOA algorithm's performance presented in this chapter uses a dynamic reference point, which is found by adding one to the maximum over all points in each objective, since using the reference points presented in Table 4.1 created a very high selective pressure, which in turn led to poor exploration and performance.

Since the decision space and objectives are continuous, the exploitation and deterministic exploration methods may resort to a gradient-based algorithm. In this chapter, we used the L-BFGS-B method implemented in the library SciPy [26], which is able to handle the bounds of $\mathcal{X}$ and is very efficient to find a local optimum. As the other algorithms being compared are evolutionary algorithms, which can only access the objective functions by evaluating them at given points, the gradient for the L-BFGS-B is computed numerically to avoid an unfair advantage in favor of our algorithm.

For the stochastic global exploration, we used an evolutionary algorithm with non-dominance sorting and removal based on the number of dominating points. The population had a minimum size of 20 and was filled with the given set of previous solutions X. If less than 20 points were provided, the others were created by randomly sampling the decision space $\mathcal{X}$ uniformly. Once a new point is introduced to the non-dominated front, it is returned for exploitation because it increases the hypervolume when added to the previous solutions X. The size of this population was chosen experimentally to provide a good enough exploration of the space toward the initial conditions for the exploitation. This size is smaller than the population size for the pure evolutionary algorithms because the pure evolutionary algorithm need diversity to explore and exploit all of its population, but the stochastic part of the H2MA is already

initialized with good and diverse candidate solutions provided by the exploitation procedure, reducing its exploration requirements.

Besides computing the solutions' hypervolume, which is the metric that the H2MA is trying to maximize and that provides a good method for comparing solutions, we can compute the distance between the achieved objectives and the Pareto frontier, since the Pareto frontiers for the ZDT functions are known. This defines the P-distance [6], which is zero, or close to zero due to numerical issues, for points at the frontier.

Figs. 4.8, 4.9, 4.10, 4.11, and 4.12 present the results for the problems ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6, respectively. A maximum of 20000 function evaluations was considered, and the graphs show the 0th, 25th, 50th, 75th, and 100th percentiles for each performance indicator over 100 runs of the algorithms. Since the P-distance is shown in log-scale, some values obtained by our proposal are absent or partially present, because they have produced zero P-distance.

From ZDT1 to ZDT4, the H2MA never ran out of regions to explore, so the stochastic exploration was not used and all runs have the same performance. For the function ZDT6, the first objective, given by Eq. (4.7a), causes some problems to the deterministic exploration.

During the creation of the first region for this problem, the mean point is used as initial condition for optimizing each objective, as shown in Fig. 4.4. However, the first objective for ZDT6 has null derivative when $x_1 = 0.5$. In this case, even traditional higher-order methods would not help, since the first five derivative of $f_1(x)$ are zero. As the first objective does not change in this case and it also has local minima that are very hard to overcome, the algorithm quickly switches to using stochastic exploration. Once new regions have candidate points, the algorithm is able to exploit them.

Besides this issue in the deterministic exploration of the problem ZDT6, the local minima of the first objective makes some candidate solutions be sub-optimal, increasing the P-distance as shown in Fig. 4.12b. Nonetheless, the achieved P-distance is better than the evolutionary algorithms and the 75th percentile is zero. Moreover, Figs. 4.8b, 4.9b, 4.10b, and 4.11b show that the candidate solutions are always on the Pareto frontier for the problems ZDT1 to ZDT4. This allows the user to stop the optimization at any number of evaluations, even with very few function evaluations, and to have a reasonable expectation that the solutions found are efficient.

When we evaluate the hypervolume indicator, we see that, for the problems ZDT1, ZDT2, ZDT4, and ZDT6, the performance of the H2MA is much better, even for the last one using stochastic exploration. Moreover, the H2MA's worst hypervolume was always better than the

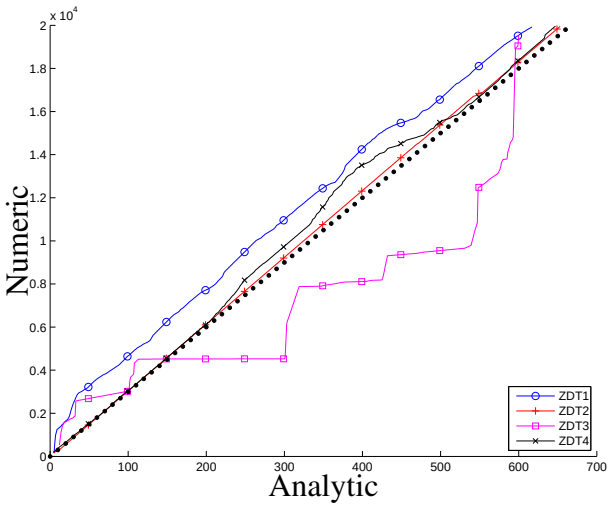


Figure 4.13: Comparison between the number of function evaluations required to achieve the same hypervolume using numeric or analytic gradient. The dotted line represents a 30-fold improvement.

best hypervolume for all evolutionary algorithms and it was able to get closer to the maximum hypervolume possible with relatively few function evaluations, being a strong indication of its efficiency.

For the problem ZDT3, whose hypervolume performance is shown in Fig. 4.10a, the H2MA was generally better than the evolutionary algorithms. The Pareto frontier for ZDT3 is composed of disconnected sets of points, which was created to test the algorithm's ability to deal with discontinuous frontiers. Since the exploitation algorithm used for the results is gradient-based, it is not able to properly handle discontinuous functions, which is the case of the hypervolume on discontinuous frontiers even when all candidates have different objective values. However, the deterministic exploration method is able to find points whose exploitation lay on the different parts of the Pareto frontier, providing the expected diversity.

Fig. 4.13 shows a comparison between the number of function evaluations required by the numeric and the analytic gradient to achieve the same hypervolume on the problems ZDT1 to ZDT4. The analytic method for computing the hypervolume's gradient is described in [22]. The comparison for ZDT6 is not shown due to its different scale, since many function evaluations are used in the global stochastic exploration because the deterministic exploration fails to find regions.

As expected, using the analytic gradient causes a 30-fold improvement in comparison to the numeric gradient, since the number of decision variables is 30. However, the gain is not linear. This can be explained by the difference in behavior during the deterministic exploration: the first non-dominated point found is used to perform the exploitation, even if this point was found during the computation of the numeric gradient. For ZDT1 and ZDT4, this causes the

new points found by the numeric gradient to be very close to the original points, reducing its performance and increasing the improvement of using the analytic gradient.

Moreover, a similar effect makes the ZDT3 performance to have a lower improvement when using the analytic gradient. Since the Pareto frontier for ZDT3 is discontinuous and this causes a discontinuity in the hypervolume, these large changes can be seen by the numeric gradient because small changes in the variables can have large effects on the hypervolume, pulling the solution if the difference is significant, while the analytic gradient is not able to provide such knowledge. Nonetheless, the analytic gradient presents at least a 15-fold improvement over the numeric one over the ZDT3.

4.3.1 Analysis of the H2MA's performance

As shown in Section 4.3, the proposed H2MA is able to surpass the reference in multi-objective optimization, based on evolutionary algorithms. Therefore, it is important to analyze the algorithm and to discuss why this improvement happened.

Evolutionary algorithms perform a guided exploration, with new individuals created based on existing high-performing individuals, which allows them to escape local minima but reduces the convergence speed. On the other hand, traditional optimization algorithms tend to find local minima quickly, but the optimal point achieved depends on the minima's regions of attraction.

These two kind of algorithms have complementary natures, which makes them good candidates for creating a hybrid algorithm: the evolutionary algorithm explores the space and provides initial conditions for the local optimization, which then finds minima quickly. Although this does create better results, it only explains the performance on the ZDT6 problem, since the other problems did not enter the stochastic phase.

In order to understand the algorithm's behavior, we must keep in mind that each new point added by the algorithm is solving a very different problem. Since the previous points that are considered during the hypervolume optimization change as more points are added, the objective surface for each new point is different from the previous ones and takes into account the already achieved portion of the hypervolume. To visualize this, suppose that the hypervolume's gradient is defined over previously found points and one previous solution is used as the initial condition for the gradient-based exploitation to find a new point to be added to the solution set. Although the initial condition was a local optimum for a previous problem, it is not a local optimum to the current problem, because any small change that creates a non-dominated point will improve the total hypervolume. Therefore, we do not need to worry about

the new optimization converging to a previous solution point because the problem landscape is different and different local minima will be found, increasing the total hypervolume. The deterministic exploration is only required because the hypervolume's gradient is not defined at the border of the hypervolume, so a new independent point must be found.

This explains the performance improvement over ZDT1 to ZDT4, because every point added improves the hypervolume as much as it can do locally, so that an improvement is guaranteed to happen. Evolutionary algorithms, on the other hand, use function evaluations without guarantees of improvement of the total hypervolume, since dominated solutions can be found.

Moreover, although a local optimum found during exploitation may not be an efficient solution due to irregularities in the objective surface, the experiments show that this is not the case most of the time, since the P-distance of the solutions found are generally zero. This result is expected, since the hypervolume is maximal when computed over points in the Pareto set, and the performance on all ZDT problems provide support to this claim.

We must highlight that we are not saying that evolutionary algorithms should not be used at all, but that they should be applied whenever traditional optimization methods are not able to solve the problem. This is the case of the ZDT6, for instance, where an evolutionary algorithm was required to provide initial conditions for the exploitation. We consider very important to have alternative methods that are better on a subset of the problems and to use them when a problem from such subset is present. This is exactly what the H2MA does: when the traditional optimization is not able to find an answer, which indicates that the problem is outside of the subset with which it can deal, an evolutionary algorithm, which is able to handle a superset class of problems, is used until the problem becomes part of the subset again, establishing a switching behavior that takes advantage of both algorithms.

4.4 Conclusion

This chapter proposed the Hybrid Hypervolume Maximization Algorithm (H2MA) for multi-objective optimization, which tries to maximize the hypervolume one point at a time. It first tries to perform deterministic local exploration and, when it gets stuck, it switches to stochastic global exploration using an evolutionary algorithm. The optimization algorithm used during deterministic optimization is problem-dependent and can be given by a gradient-based method, when the decision space is continuous, or a hill-climbing method, when the decision space is discrete. Here we have explored solely continuous decision spaces.

The algorithm was compared with reference algorithms for multi-objective optimization,

namely NSGA-II, SPEA2, and SMS-EMOA on the ZDT problems. Despite using numeric gradient for the objective functions, which increases the number of function calls, the algorithm consistently provided higher hypervolume for the same number of function evaluations when compared to the aforementioned evolutionary algorithms. Only for the ZDT3 the performance was slightly reduced due to the discontinuous nature of the Pareto frontier, which causes a discontinuity in the hypervolume, not properly handled by gradient-based methods.

Moreover, for all problems except for ZDT6, all the solutions found by the algorithm were over the Pareto frontier, which makes them efficient solutions. For the ZDT6, the median case also had all solutions over the Pareto frontier, but the use of the stochastic exploration not always guided to a solution at the Pareto frontier. Nonetheless, the obtained solutions were better than those provided by the evolutionary algorithms. Moreover, the solutions provided for ZDT1 to ZDT4 achieved high performance using only the deterministic part of the algorithm.

Evolutionary algorithms usually have better performance when their populations are larger, so that diverse individuals can be selected for crossover. However, most of the time people do not require many options, so the H2MA presents itself as an alternative choice for finding a good set of solutions at a lower computational cost in most problems, although it does not limit the computational burden and the number of points found. If the problem has more reasonable objectives than ZDT6, which was designed with an extreme case in mind, we can expect that many points will be found by the deterministic mechanisms, which makes the algorithm more reliable. Moreover, the solutions found should be efficient, which is characterized by a low P-distance, and diverse on the objectives, which is characterized by a larger hypervolume when only efficient solutions are considered.

Future work should focus on using surrogates to reduce the number of evaluations [27, 28, 29]. Although the H2MA is very efficient on its evaluations, the numeric gradient may consume lots of evaluations and be unreliable for complicated functions, as their implementation can cause numerical errors larger than the step used. Using a surrogate, the gradient can be determined directly and fewer evaluations are required.

Another important research problem is to find a new algorithm for computing the hypervolume, since existing algorithms are mainly focused on computing the hypervolume given a set of points [30]. Since the solution set is constructed one solution at a time in the H2MA, a recursive algorithm that computes the hypervolume of $X \cup \{x\}$ given the hypervolume of X should reduce the computational overhead.

Benchmark functions

ZDT1:

$$f_1(x) = x_1 \tag{4.3a}$$
$$f_2(x) = g(x)h(f_1(x), g(x)) \tag{4.3b}$$
$$g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \tag{4.3c}$$
$$h(f_1(x), g(x)) = 1 - \sqrt{f_1(x)/g(x)} \tag{4.3d}$$

ZDT2:

$$f_1(x) = x_1 \tag{4.4a}$$
$$f_2(x) = g(x)h(f_1(x), g(x)) \tag{4.4b}$$
$$g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \tag{4.4c}$$
$$h(f_1(x), g(x)) = 1 - (f_1(x)/g(x))^2 \tag{4.4d}$$

ZDT3:

$$f_1(x) = x_1 \tag{4.5a}$$
$$f_2(x) = g(x)h(f_1(x), g(x)) \tag{4.5b}$$
$$g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \tag{4.5c}$$
$$h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}} - \sin(10\pi f_1(x))\frac{f_1(x)}{g(x)} \tag{4.5d}$$

ZDT4:

$$f_1(x) = x_1 \tag{4.6a}$$
$$f_2(x) = g(x)h(f_1(x), g(x)) \tag{4.6b}$$
$$g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}(x_i^2 - 10\cos(4\pi x_i)) \tag{4.6c}$$
$$h(f_1(x), g(x)) = 1 - \sqrt{f_1(x)/g(x)} \tag{4.6d}$$

ZDT6:

$$f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1) \tag{4.7a}$$

$$f_2(x) = g(x)h(f_1(x), g(x)) \tag{4.7b}$$

$$g(x) = 1 + 9\left(\sum_{i=2}^{n} \frac{x_i}{n-1}\right)^{0.25} \tag{4.7c}$$

$$h(f_1(x), g(x)) = 1 - (f_1(x)/g(x))^2 \tag{4.7d}$$

Chapter 5

Pareto frontier approximation

A common issue for real-world MOO is the number of objectives used [18]. Since using multiple objectives allows us to evaluate trade-offs between the solutions after they are found, instead of having to define preferences a priori as discussed in Section 2.1, we would like to be able to provide as many objectives as metrics we would like to actually evaluate. However, good optimization metrics tend be expensive to compute [30] and become impractical on large number of objectives. To try to solve this problem, many approaches to define approximations or surrogates to the Pareto frontier have been defined, each with its own associated performance metric for evaluating a new solution. However, they might misbehave in certain conditions, making their use as performance metric not satisfactory.

In this chapter, we develop a theory that defines necessary and sufficient conditions for a functional description of a Pareto frontier. Based on this theory, the search for approximations of the Pareto frontier using surrogate functions should be constrained to, or at least focused on, the ones that satisfy the results. If not, the resulting manifold obtained from the function can have any shape, possibly with many dominated points, which could result in reduced performance of algorithms that use the approximation of the Pareto frontier, either during the optimization or after it.

Moreover, the theory is developed on the objective space, allowing either accurate or approximate objective evaluations to be used, without restricting the format of the objectives' surrogates. If parametric surrogate objectives are used, their association with the Pareto frontier surrogate can provide feedback on how to adjust their parameters so that the approximation is closer to the real objectives.

As an example of how to integrate the theoretical conditions in a surrogate design, we

show how to introduce the theoretical conditions as soft constraints in Gaussian processes, discussed in Section 2.4, which are nonparametric models, thus being able to adjust to variable number of samples, and whose hyper-parameters can be easily optimized.

To validate the hypothesis that surrogate methods that do not consider this theory may define invalid Pareto frontier approximations, the constrained Gaussian process is compared to a regular Gaussian process and to an existing SVM-based surrogate method [31] on a knee-shaped Pareto Frontier. Results in this experiment show that the soft constrained Gaussian process finds good approximations maximally obeying the constraints according to the degree of flexibility of the model. On the other hand, the models that do not take into account the theory can violate greatly and arbitrarily the conditions for a valid Pareto frontier. We also show how the proposed surrogate can be used as base for an approximate performance metric that is efficient to compute.

This chapter is organized as follows. Section 5.1 presents previous work on approximating the Pareto frontier and how they relate to the method proposed in this chapter. Section 5.2 introduces the notation and principles of multi-objective optimization used in this chapter, since the theory developed requires precise mathematical definitions. Section 5.3 shows the conditions that a function must satisfy to define a valid Pareto frontier. These conditions are then used in Section 5.4 to build a function to approximate a frontier given some points on it and the approximation is compared to an existing surrogate. Using this approximating method, Sections 5.5 defines a performance metric that can be used to search for solutions to the MOO problem.

5.1 Related work

The Pareto frontier is at the core of MOO algorithms, being the foundation of many methods devoted to evaluating the performance and comparing the solutions to each other, as discussed in Section 2.2. However, the frontier is defined by the objectives, which can be expensive to compute [27, 28, 29]. This leads to a variety of surrogate methods that try to approximate the objectives, e.g. [32, 33], thus saving computational resources. We divide the related work in two parts: multi-objective optimization algorithms using Gaussian process as surrogates for the objectives and other techniques for creating surrogates for the Pareto frontier itself.

5.1.1 Gaussian processes as surrogates

In this section, we present relevant previous works that use Gaussian processes in multi-objective objective algorithms and discuss how they are related to the novel method presented in this chapter. For an extensive analysis of model-based multi-objective optimization algorithms, we refer the reader to [34].

One early approach that extended the EGO procedure, discussed in Section 2.4.4, for multi-objective optimization was the ParEGO algorithm [35], in which the objectives are transformed into a single cost function using an augmented Tchebycheff function whose coefficients are chosen randomly at each step and this new cost function is approximated by a Gaussian process. Based on this approximation, an evolutionary algorithm is used to optimize the expected improvement of the new cost function, providing the new point to be evaluated in the real system. Further study [36] showed that ParEGO is robust to noise on the function evaluations.

Later, a similar approach named MOEA/D-EGO [37] was proposed. This method also transforms the problem into a single-objective optimization task in order to use the EGO procedure, but uses a fixed set of parameters for the scalarization function instead of a random value for the parameters at each step, like ParEGO, thus creating many scalarizations of the objectives.

Alternatively, instead of performing a scalarization of the objectives and then using a Gaussian process to approximate the new function, one can create a surrogate for each objective and combine the approximations to create the scalarization. For instance, [38] and [39] adapt the SMS-EMOA [12], which is an evolutionary algorithm that selects individuals based on the hypervolume contribution as discussed in Section 2.2, to use the expected improvement on the hypervolume instead of the actual hypervolume contribution as target for the algorithm. When the real objectives' evaluations have noise, the surrogates for the objectives can also share information that might improve the approximation by using multi-task learning methods [40].

5.1.2 Pareto frontier surrogates

Among the surrogates that directly or indirectly estimate the Pareto frontier, the one introduced by Yun et al. [31] is the closest to the surrogate described in this chapter. They used a one-class support vector machine (SVM) to define a function over the objective space whose null space describes an approximation of the Pareto frontier. This function is used to select individuals, since its value increases as the candidate becomes more distant from the frontier, which are then used for crossover in a genetic algorithm.

Loshchilov et al. [41] presented a similar SVM approach, but the function learnt is defined over the decision space, which allows direct comparison with the Pareto frontier approximation without requiring evaluation of the objectives. This direct comparison can also be achieved with estimates built over the objective space by integrating surrogates for the objectives. However, contrary to the one-class SVM that learns a model to fit all samples on one side of the approximate frontier, the proposed SVM is also able to consider points that dominate the frontier being approximated, allowing approximation of multiple Pareto frontiers, each defined by a class of points in non-dominated sorting [42].

In a different approach, Loshchilov et al. [43] approximated the Pareto dominance instead of the Pareto frontier by using a rank-based SVM. In this case, instead of providing only the data points, the algorithm is also informed about the preference for an arbitrary number of sample pairs and tries to find a function where higher evaluation represents higher preference. Using the Pareto dominance to establish the preference between points and learning directly from the decision space, candidate solutions can be compared in dominance using the learnt function. However, both [41] and [43] try to estimate the Pareto frontier using generic function approximation models, which do not take into account the particularities of the Pareto frontier.

It is possible to guarantee that the Pareto frontier's estimate is valid by building conservative estimates. For instance, using a binary random field over the objective space to model the boundary between dominated and non-dominated regions, Da Fonseca and Fonseca [44] described a theory that can be used to assess the statistical performance of a stochastic optimization algorithm and compare different algorithms. The attainment function described in the paper defines the probability that a run of the stochastic algorithm will dominate the function's arguments. Although the attainment function is hard to compute, it can be approximated by multiple runs of the underlying algorithm, which makes it a good candidate for analyzing the performance statistics of the optimization algorithm and for performing hypothesis testing between MOO algorithms.

If a single run is considered, then the approximate attainment function describes a valid estimate of the Pareto frontier and it is defined as the border of the region dominated by the points provided. Although valid, this estimate is very conservative and does not interpolate between the points provided, which means it cannot provide a good idea of the frontier's shape and any evaluation of new points could be performed using only dominance comparison with the provided points.

5.2 Definitions

Although the objective space usually only makes sense when coupled with the decision space and objectives, which allows for its infeasible region and Pareto frontier to be defined, we will work only with the objective space in this chapter, which means that the results hold for any problem. We will also consider that the objective space $\mathcal{Y} = \mathbb{R}^M$, since any restriction for a specific problem is defined by means of the objectives and decision space constraints, and are handled transparently.

Using the definitions from Section 2.1, we can divide the space $\mathcal{Y}$ in three sets: an estimated frontier, the set of points strongly dominated by the estimated frontier, and the set of points not strongly dominated by the estimated frontier.

Definition 12 (Estimated Frontier)**.** A path-connected set of points $F \subset \mathbb{R}^M$ is said to be an estimated frontier if no point in it strongly dominates another point also in F, that is, $\forall y \in F, \nexists y' \in F \colon y' \prec y$, and every point in the objective space except for F either strongly dominates or is strongly dominated by a point in F, that is, $\forall y \in \mathbb{R}^M - F, \exists y' \in F \colon y \prec y' \vee y' \prec y$.

A set S is path-connected if there is a path joining any two points x and y in S and a path is defined by a continuous function $p \colon [0, 1] \to S$ with $p(0) = x$ and $p(1) = y$. Therefore, if there is a continuous path of points in S that gets from any $x \in S$ to any $y \in S$, then S is path-connected. Based on this definition, an estimated frontier F divides the objective space $\mathbb{R}^M$ in three disjoint sets: points strongly dominated by points in F, points that strongly dominate points in F, and F itself.

Definition 13 (Estimated Strict Frontier)**.** A set of points $F_s \subset \mathbb{R}^M$ is said to be an estimated strict frontier if no point in it dominates another point also in F_s, that is, $\forall y \in F_s, \nexists y' \in F_s, y' \neq y \colon y' \preceq y$, and every point in the objective space except for F_s either dominates or is dominated by a point in F_s, that is, $\forall y \in \mathbb{R}^M - F_s, \exists y' \in F_s \colon y \preceq y' \vee y' \preceq y$.

Definition 14 (Pareto Frontier)**.** An estimated strict frontier F^* is a Pareto frontier if and only if, for all points in F^*, there is no other feasible point in the objective space that dominates the point in the frontier, that is, $\forall y \in F^*, \nexists x \in \mathcal{X}, g(x) \neq y \colon g(x) \preceq y$.

The estimated frontier of Definition 12 is a generalization and an approximation of the Pareto frontier in two ways: *i*) if the Pareto frontier is discontinuous, then dominated points are added so that the estimated frontier F is path-connected while also guaranteeing that no point in it strongly dominates any other; and *ii*) the estimated frontier is simply a set of points that divide the space into dominated and non-dominated regions, without stating anything about the optimality of its points.

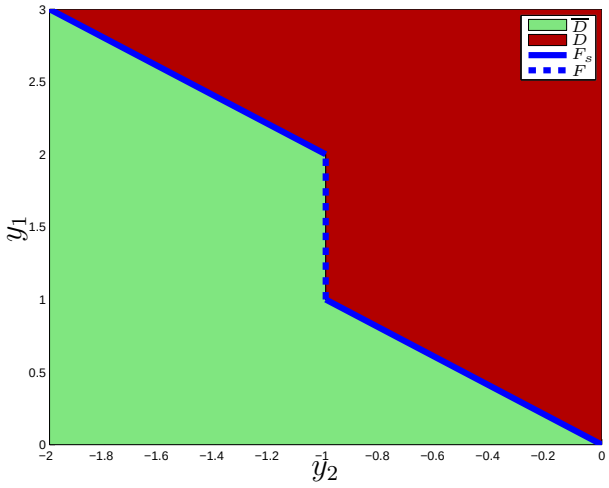


Figure 5.1: Example of the definitions for a particular multi-objective problem. The estimated strict frontier F_s is shown in a solid blue line, the estimated frontier F includes the solid and dashed blue lines, the dominated region D is shown on the top right red area, and the non-dominated region $\overline{D}$ is shown on the bottom left green area.

Consider, for instance, a problem where one of the objectives is given by

$$g_1(x) = \begin{cases} x + 1, & x > 1 \\ x, & \text{otherwise}, \end{cases}$$

and the other is given by $g_2(x) = -x$. Then the Pareto frontier F^* is given by

$$\begin{aligned} F^* = &\{(x + 1, -x) \mid x \in \mathbb{R}, x > 1\} \\ &\cup \{(x, -x) \mid x \in \mathbb{R}, x \leq 1\}, \end{aligned}$$

which clearly is not path-connected. However, if we add the set of points $\hat{F} = \{(y, -1) \mid y \in (1, 2]\}$ to F^*, then the resulting path-connected set $F = F^* \cup \hat{F}$ satisfies Definition 12, despite the fact that every point in $\hat{F}$ is dominated by $(1, -1) \in F^*$, but not strongly dominated by it.

Figure 5.1 shows an estimated strict frontier F_s, which coincides with the Pareto frontier F^* in this example, and the path-connected estimated frontier F for this problem. This makes it clear that the estimated frontier F can contain the Pareto frontier F^*, i.e. $F^* \subseteq F$, while providing a path-connected 1D manifold that splits the whole objective space $\mathbb{R}^2$. Of course, these properties of the estimated frontier are extensible to $M > 2$ objectives.

With the definition of an estimated frontier, the objective space is divided into two sets, named dominated and non-dominated sets, also shown in Fig. 5.1.

Definition 15 (Dominated Set)**.** The dominated set D for an estimated frontier F is the set of all points in $\mathbb{R}^M$ where, for each one of them, there is at least one point in F that strongly

dominates it, that is, $D = \{y \in \mathbb{R}^M \mid \exists y' \in F : y' \prec y\}$.

Definition 16 (Non-Dominated Set)**.** The non-dominated set $\overline{D}$ for an estimated frontier F is the set of all points that are not in F or D, that is, $\overline{D} = \{y \in \mathbb{R}^M \mid \exists y' \in F : y \prec y'\}$.

Note that, from the definition of strong dominance, both D and $\overline{D}$ are open and unbounded sets, with boundaries defined by the estimated frontier F. Furthermore, if F contains the Pareto frontier, then the points in $\overline{D}$ are not achievable due to the objectives' definitions.

From the partition of the objective space in three sets, one estimated frontier, one dominated and one non-dominated set, we can define a score function similarly to [41, 43].

Definition 17 (Score Function)**.** A score function $f(y) : \mathbb{R}^M \to \mathbb{R}$ for a given estimated frontier F is a function that satisfies

$$\begin{aligned} f(y) &= 0, \quad \forall y \in F, \\ f(y) &> 0, \quad \forall y \in D, \\ f(y) &< 0, \quad \forall y \in \overline{D}. \end{aligned}$$

Therefore, a score function provides a single value that places its argument in relation to the estimated frontier. Moreover, for a given estimated frontier F, there are many possible choices of score functions $f(y)$ that satisfy the definition and all of them uniquely define F based on their solution set $f(y) = 0$. This allows a score function to work as a surrogate for the estimated frontier.

5.3 Necessary and Sufficient Conditions for Surrogate Score Functions

In this section, we will show how a score function $f(y)$ can induce an estimated frontier F and the conditions it must satisfy so that the set it defines is indeed an estimated frontier, that is, no point in it strongly dominates any other point in it.

The main theory developed is based on the most general notion of a function f, but the conditions may be hard to evaluate for a general case. Therefore, we will also provide corollaries that prove the results for functions with additional constraints, like continuous derivatives. Since some of these results depend on Taylor approximations and the first derivative at the required points may be zero, we must define a generalized gradient.

Definition 18 (Generalized Gradient)**.** Let $h \in C^k$, where C^k is the class of functions where the first k derivatives exist and are continuous, with $k \geq 1$. Let $k^*(h)$ be the first non-zero derivative of h evaluated at 0, that is,

$$k^*(h) = \arg\min_{1 \leq i \leq k} \left(\left. \frac{\mathrm{d}^i h}{\mathrm{d}x^i} \right|_{x=0} \neq 0 \right),$$

where $k^*(h)$ is not defined if h is a constant function or no i satisfies the inequality. Then

$$\Delta(h) = \begin{cases} 0, & \exists a \in \mathbb{R}, \forall x \colon h(x) = a \\ \dfrac{1}{k^*(h)!} \left. \dfrac{\mathrm{d}^{k^*(h)} h}{\mathrm{d}x^{k^*(h)}} \right|_{x=0}, & \text{otherwise} \end{cases}$$

is the generalized gradient operator, which is undefined if there is no i that satisfies the inequality.

The role of the generalized gradient in the theory to be presented is to avoid issues with functions that may have null derivative at the points being evaluated but that are also increasing. Consider, for instance, the function $f(x) = x^3$, whose gradient is null at $x = 0$. This function is strictly increasing, but the first-order approximation using Taylor series is a constant. In order to consider small changes in the function's argument, we must use first non-null derivative, which is the generalized gradient, as it will dominate the approximation.

The generalized gradient can be used in the Taylor approximation as $h(\delta) = h(0) + \delta^{k^*(h)} \Delta(h) + O(\delta^{k^*(h)+1})$, where $0 < \delta \ll 1$ and $O(\cdot)$ is the Landau symbol. Since the result is based on δ being a small value, the exact power $k^*(h)$ used to compute $\delta^{k^*(h)}$ is not important for the approximation and the term $O(\delta^{k^*(h)+1})$ is dominated by the other factors.

The extensions to continuous functions f rely on the generalized gradient of a single-parameter continuous function $\hat{f}$, derived from the original f, having different signs for opposite directions. However, it does not hold for functions where $k^*(\cdot)$ is even.

For example, consider $h(x) = x^2$, which has $k^*(h) = 2$. The Taylor approximation is given by $h(\delta) \approx \delta^2 \Delta(h(x)) = 2\delta^2 = \delta^2 \Delta(h(-x)) \approx h(-\delta)$, which does not give different signs to different directions of x. Therefore, the two constraints on $\Delta(\hat{f})$ defined in the corollaries that follow can be viewed as a single constraint on $\Delta(\hat{f})$ plus the constraint that $k^*(\hat{f})$ is odd.

5.3.1 Necessary Conditions

The necessary conditions derived are direct applications of the estimated frontier's definition and establish the basic ground on how to define a function f from a given estimated frontier.

Lemma 4 (General Necessity)**.** *Let F be an estimated frontier. Let $f(y)\colon \mathbb{R}^M \to \mathbb{R}$ be a score function for F. Then $f(y+\delta u) > 0$ and $f(y-\delta u) < 0$ for all $y \in F$, $u \in (0,1]^M$, and $\delta \in \mathbb{R}, \delta > 0$.*

Proof. Assume there are y, u, and $\delta > 0$ such that $f(y+\delta u) \leq 0$. Let $y' = y + \delta u$, so that $y \prec y'$.

If $f(y') < 0$, then from the definition of a score function there is some $y^* \in F$ such that $y' \prec y^*$. From the transitivity of dominance, we have that $y \prec y' \prec y^*$, which is a contradiction, since the point y^* in the frontier cannot strongly dominate the point y also in the frontier. Then we must have $f(y') = 0$, which means $y' \in F$ and also creates a contradiction.

Assume that $f(y-\delta u) \geq 0$, and let $y'' = y - \delta u$. Then we can similarly prove that it also creates a contradiction.

Therefore, there are no such y, u, and δ with $f(y+\delta u) \leq 0$ or $f(y-\delta u) \geq 0$. ■

This result is intuitive, since moving δ in direction u from y we enter either D or $\overline{D}$. If the function has the required derivatives, then the following result holds.

Corollary 1 (Differentiable Necessity)**.** *Let F be an estimated frontier. Let $f(y)\colon \mathbb{R}^M \to \mathbb{R}$ be a score function for F. Let $\hat{f}^+_{y,u}(x) = f(y+xu)$ and $\hat{f}^-_{y,u}(x) = f(y-xu)$, with $x \in [0,\infty)$. Let $\Delta(\hat{f}^+_{y,u})$ and $\Delta(\hat{f}^-_{y,u})$ be defined for all $y \in F$ and $u \in (0,1]^M$. Then $\Delta(\hat{f}^+_{y,u}) > 0$ and $\Delta(\hat{f}^-_{y,u}) < 0$ for all $y \in F$ and $u \in (0,1]^M$.*

Proof. Since f satisfies all conditions from Lemma 4, we have that $f(y+\delta u) > 0$ and $f(y-\delta u) < 0$ for all y, u, and $\delta > 0$.

In particular, let $\delta \ll 1$. Approximating using Taylor series, we have that $f(y+\delta u) \approx f(y) + \delta' \Delta(\hat{f}^+_{y,u}) > 0$ and $f(y-\delta u) \approx f(y) + \delta' \Delta(\hat{f}^-_{y,u}) < 0$, where δ' is the appropriate power of δ for the expansion. Since $f(y) = 0$ and $\delta' > 0$, then $\Delta(\hat{f}^+_{y,u}) > 0$ and $\Delta(\hat{f}^-_{y,u}) < 0$ must hold. ■

Although this corollary may appear to provide weaker guarantees on f, its proof shows that the inequality constraints on the generalized gradient is equivalent to the direct inequalities on the function defined in the previous lemma.

5.3.2 Sufficient Conditions

Once defined how the estimated frontier relates to a given score function, we will show that a function that satisfies the results of the previous lemma and corollary in fact uniquely defines an estimated frontier F.

Lemma 5 (General Sufficiency)**.** *Let $f(y)\colon \mathbb{R}^M \to \mathbb{R}$ be a function. Let $F = \{y \in \mathbb{R}^M \mid f(y) = 0\}$ be a path-connected set. Let $f(y + \delta u) > 0$ and $f(y - \delta u) < 0$ for all $y \in F$, $u \in (0, 1]^M$, and $\delta \in \mathbb{R}, \delta > 0$. Then F is an estimated frontier.*

Proof. For F to be an estimated frontier, we have to prove that for any $y, y' \in F, y \neq y'$ we have $y \not\prec y'$. Assume there are y and y' in F such that $y \prec y'$.

Let $u = y' - y$ and $\delta = 1$. Then we have $f(y + \delta u) = f(y') = 0$, which violates the first inequality on $f(\cdot)$. Alternatively, we have $f(y' - \delta u) = f(y) = 0$, which violates the second inequality.

Therefore, there are no y and y' in F such that $y \prec y'$, and F is an estimated frontier. ■

The restrictions on $f(y \pm \delta u)$ may be hard to verify in general, since they must be valid for all δ. However, if the function has the appropriate derivatives, then it becomes easier to check if it satisfies the requirements.

Corollary 2 (Differentiable Sufficiency)**.** *Let $f(y)\colon \mathbb{R}^M \to \mathbb{R}$ be a function. Let $F = \{y \in \mathbb{R}^M \mid f(y) = 0\}$ be a path-connected set. Let $\hat{f}^+_{y,u}(x) = f(y + xu)$ and $\hat{f}^-_{y,u}(x) = f(y - xu)$, with $x \in [0, \infty)$. Let $\Delta(\hat{f}^+_{y,u}) > 0$ and $\Delta(\hat{f}^-_{y,u}) < 0$ for all $y \in F$ and $u \in (0, 1]^M$. Then F is an estimated frontier.*

Proof. To use Lemma 5, we must prove that $f(y + \delta u) > 0$ and $f(y - \delta u) < 0$ for all $y \in F$, $u \in (0, 1]^M$, and $\delta \in \mathbb{R}, \delta > 0$.

Suppose there is some y, u, and δ in the domain such that $f(y + \delta u) = 0$. Moreover, let δ be the smallest value for which this happens for a given y and u. Let $0 < \epsilon \ll 1$ and $\epsilon < \delta$. Then $f(y + \epsilon u) \approx f(y) + \epsilon' \Delta(\hat{f}^+_{y,u}) > 0$ and $f((y + \delta u) - \epsilon u) \approx f(y + \delta u) + \epsilon' \Delta(\hat{f}^-_{y,u}) < 0$,

where ϵ' is the appropriate power of ϵ for the approximation. However, $f(\cdot)$ cannot go from positive to negative without passing through 0 due to its continuity. Then there must be some $\delta' < \delta$ such that $f(y + \delta' u) = 0$, which contradicts the definition of δ.

Therefore, the first inequality on Lemma 5 holds. We can use a similar method to prove the second inequality, and then use the lemma. ■

Again, this corollary shows the equivalence between the inequalities on the function and on the generalized gradient.

5.3.3 Necessary and Sufficient Conditions

Since the symmetry between Lemmas 4 and 5 is clear, we can build a theorem to merge those two and provide necessary and sufficient conditions for defining an estimated frontier F from a score function $f(y)$.

Theorem 4 (General Score Function)**.** *Let $f(y)\colon \mathbb{R}^M \to \mathbb{R}$ be a function. Let $F = \{y \in \mathbb{R}^M \mid f(y) = 0\}$ be a path-connected set. Let $D = \{y \in \mathbb{R}^M \mid \exists y' \in F\colon y' \prec y\}$ and $\overline{D} = \mathbb{R}^M \backslash (F \cup D)$. Let $f(y) > 0, \forall y \in D$, and $f(y) < 0, \forall y \in \overline{D}$. Then F is an estimated frontier if and only if $f(y + \delta u) > 0$ and $f(y - \delta u) < 0$ for all $y \in F$, $u \in (0, 1]^M$, and $\delta \in \mathbb{R}, \delta > 0$.*

Proof. Assume that the constraints on f are valid. Then, from Lemma 5, we have that F is an estimated frontier. Now assume that F is an estimated frontier. Then, from Lemma 4, we have that the constraints on f are valid. ■

Instead of requiring knowledge of the sign of $f(y)$ over the sets, we can use a more strict definition, requiring continuity, to guarantee that the result holds.

Corollary 3 (Continuous Score Function)**.** *Let $f(y)\colon \mathbb{R}^M \to \mathbb{R}$ be a continuous function where there are points v_+ and v_- such that $f(v_+) > 0$, $f(v_-) < 0$, and $v_- \prec v_+$. Let $F = \{y \in \mathbb{R}^M \mid f(y) = 0\}$ be a path-connected set. Then F is an estimated frontier if and only if $f(y + \delta u) > 0$ and $f(y - \delta u) < 0$ for all $y \in F$, $u \in (0, 1]^M$, and $\delta \in \mathbb{R}, \delta > 0$.*

Proof. Assume that F is an estimated frontier. Assume that there are $y, y' \in D = \{y \in \mathbb{R}^M \mid \exists y' \in F\colon y' \prec y\}$ such that $f(y) > 0$ and $f(y') < 0$. From the continuity of f, we have that there is some $z \in D$ such that $f(z) = 0$. However, since $f(z) = 0$, it is in F. From the definition of D, there is some $z' \in F$ such that $z' \prec z$, which violates the assumption that F is

an estimated frontier. Therefore, all points in D have the same sign over f. The same can be shown for $\overline{D}$.

Since $v_- \prec v_+$, we have that $v_+ \in D$ and $v_- \in \overline{D}$. Then f satisfies all conditions from Theorem 4. ■

Again, we can replace the constraints on $f(y \pm \delta u)$ by the constraint on the generalized gradient.

Corollary 4 (Differentiable Score Function)**.** *Let $f(y)\colon \mathbb{R}^M \to \mathbb{R}$ be a function where there are points v_+ and v_- such that $f(v_+) > 0$, $f(v_-) < 0$, and $v_- \prec v_+$. Let $F = \{y \in \mathbb{R}^M \mid f(y) = 0\}$ be a path-connected set. Let $\hat{f}^+_{y,u}(x) = f(y + xu)$ and $\hat{f}^-_{y,u}(x) = f(y - xu)$. Let $\Delta(\hat{f}^+_{y,u})$ and $\Delta(\hat{f}^-_{y,u})$ be defined for all $y \in F$ and $u \in (0, 1]^M$. Then F is an estimated frontier if and only if $\Delta(\hat{f}^+_{y,u}) > 0$ and $\Delta(\hat{f}^-_{y,u}) < 0$ for all $y \in F$ and $u \in (0, 1]^M$.*

Proof. We can use Corollary 3 to show that the restrictions on $f(y \pm \delta u)$ must hold. From Corollaries 1 and 2, we know that the restrictions on $\Delta(\hat{f}^\pm_{y,u})$ are the same as the restrictions on $f(y \pm \delta u)$, so this corollary is valid. ■

5.4 Learning Surrogate Functions from Samples

After showing what conditions the function f must satisfy, one could ask how to build such function for a given problem and specially how to learn one from a given set of non-dominated points. This can be a hard question to answer in general, but we can provide an additional lemma that can help in many cases.

Lemma 6 (Strictly Increasing Sufficiency)**.** *Let $f(y)\colon \mathbb{R}^M \to \mathbb{R}$ be a strictly increasing function on each coordinate. Let $F = \{y \in \mathbb{R}^M \mid f(y) = 0\}$. Then F is an estimated frontier.*

Proof. For F to be an estimated frontier, we have to prove that for any $y, y' \in F, y \neq y'$ we have $y \not\prec y'$. Assume there are y and y' in F such that $y \prec y'$.

Let $P = (p_0 = y, p_1, \ldots, p_{M-1}, p_M = y')$ be a path between y and y' that increments only one coordinate at a time. Since f is strictly increasing, we have that $f(p_i) < f(p_{i+1})$. Thus, $f(y) < f(y')$, which contradicts the premise that $f(y) = f(y') = 0$ because they are both in the frontier.

Therefore, there are no y and y' in F where $y \prec y'$ and F is an estimated frontier. ■

Note that, because f is strictly increasing, there is no point in F that even dominates another point in F, which was allowed in Definition 12. This restriction can be relaxed to be only monotonically non-decreasing if one can guarantee that $f(y) = 0$ is only a manifold, and not a subspace with volume. If $f(y) = 0$ is a subspace, then we can find two points in it where one dominates the other, which violates the basic definition of an estimated frontier. For instance, a function that is monotonically non-decreasing and is constant in at most one dimension at a time does not create a subspace on $f(y) = 0$.

Nonetheless, this lemma can be used as a guide on how to build a function for the general case. We will build a model that tries to approximate an estimated frontier from a few of its samples using an approximated monotonically increasing function based on Gaussian processes.

5.4.1 Gaussian Processes with Monotonicity Soft Constraint as Surrogates

We consider the null mean function $\mu(x) = 0$ and the squared exponential kernel $k(x, y)$ defined in Eq. (2.12), since this is a common approach for Gaussian processes, as described in Section 2.4. Since we are mapping from the objective space $\mathbb{R}^M$ to a value in $\mathbb{R}$, according to Definition 17, the input values are the objectives y and the outputs, the scores z.

Let $Y \in \mathbb{R}^{N \times M}$ be a set of N input points and $Z \in \mathbb{R}^N$ their desired targets for training. We define the latent variable L between the two, such that

$$L|Y \sim \mathcal{N}(0, K(Y, Y)),$$

where $K(Y, Y)_{i,j} = k(y_i, y_j)$. The latent variable then produces the observed values Z through

$$Z|L \sim \mathcal{N}(L, \sigma^2 I),$$

where I is the identity matrix.

This model is the same as the one described in Section 2.4. However, only the mean prediction will be used in this section to describe the estimated frontier. Moreover, we will show how changing the allowed noise level σ affects the Pareto frontier approximation.

Besides the observations of $f(y)$ at the desired points, the GP framework also accepts observations of its derivative, since differentiation is a linear operator [13, 45], that is, the derivative of a GP is also a Gaussian process. However, since we do not know the desired

value of the gradient, only that it should be positive, from Corollary 4 and Lemma 6, forcing an arbitrary value may lead to reduced performance.

Another option is to introduce a probability distribution over the gradient in order to favor positive values, introducing monotonicity information [46]. This new distribution can be viewed as adding constraints to the Gaussian process, making it feasible to include the monotonicity information to the existing framework.

Ideally, the probability distribution over the gradient is the step function, which provides a probability of zero if the gradient is negative and the same probability for all positive gradients. However, the step function defines a hard threshold and does not allow small errors, which can cause some problems for the optimization, since it is not a differentiable function. Therefore, a smooth function that approximates the step is used to define a soft constraint over the gradient.

Let $m_{d_j}^{(i)}$ be a binary value indicating that the function, when evaluated on the i-th sample, should be monotonic in the direction d_j. Then the following probability distribution can be used to approximate the step function:

$$p\left(m_{d_j}^{(i)} \left| \frac{\partial l^{(i)}}{\partial y_{d_j}} \right.\right) = \Phi\left(\zeta \frac{\partial l^{(i)}}{\partial y_{d_j}}\right) \tag{5.1a}$$

$$\Phi(v) = \int_{-\infty}^{v} \mathcal{N}(t|0,1)\mathrm{d}t, \tag{5.1b}$$

where we assume the probit function $\Phi(\cdot)$ as the derivative probability. Since the probit is a cumulative distribution function, its value ranges from 0 to 1 and it is monotonically increasing, which makes it a good approximation for the step function. The parameter ζ allows us to define how strict the distribution should be, with $\zeta \to \infty$ approximating the step function or a hard constraint. In this chapter, following the suggestion of [46], we use $\zeta = 10^6$.

Since the monotonicity probability is not normal, it has to be approximated by a normal distribution to be used in the GP framework. To understand this, first consider the problem without the monotonicity constraints, which is given by Eq. (2.14). The probability distribution of the observation is given by:

$$p(L_*|Y_*, Y, Z) = \int p(L_*|Y_*, Y, L)p(L|Y, Z)\mathrm{d}L, \tag{5.2}$$

where L is the latent variable for the training data, whose probability distribution, computed by the Bayes' rule, is

$$p(L|Y, Z) = \frac{p(Z|L)p(L|Y)}{p(Z|Y)}$$

$$p(Z|Y) = \int p(Z|L)p(L|Y)\mathrm{d}L.$$

According to the model, the prior $p(L|Y)$ and the likelihoods $p(Z|L)$ and $p(L_*|Y_*, Y, L)$ are normal distributions, which makes all integrals tractable and all other distributions defined in the closed form presented in Eq. 2.14.

Now, considering the monotonicity constraints, let $\mathcal{M}$ be the monotonicity constraints and L' be the random variable associated with the derivative of the latent variable L. Then the probability distribution in Eq. (5.1) can be written as $p(\mathcal{M}|L')$. Rewriting the posterior distribution over the latent variables, we get:

$$p(L|Y, Z, \mathcal{M}) = \frac{p(\mathcal{M}|L')p(Z|L)p(L, L'|Y)}{p(Z, \mathcal{M}|Y)} \tag{5.3a}$$

$$p(Z, \mathcal{M}|Y) = \int p(\mathcal{M}|L')p(Z|L)p(L, L'|Y)\mathrm{d}L\mathrm{d}L'. \tag{5.3b}$$

Because the distribution $p(\mathcal{M}|L')$ is not normal and every other distribution in Eq. (5.3) is normal, the integrals defined in Eqs. (5.2) and (5.3b) are intractable. Therefore, the distribution $p(\mathcal{M}|L')$ must be approximated by a normal distribution, which can be achieved using the expectation propagation algorithm [15], with the update equations described in [46]. The expectation propagation algorithm iteratively adjusts an unnormalized normal distribution to locally approximate the distribution defined by the soft constraints, such that $p(\mathcal{M}|L') \approx \tilde{Z}\mathcal{N}(L'|\tilde{\mu}, \tilde{\Sigma})$, where $\tilde{Z}$ is a normalization constant, $\tilde{\mu}$ is a mean vector with one value for each monotonicity constraint, and $\tilde{\Sigma}$ is a diagonal covariance matrix. See Sec. 2.4.3 for details on the algorithm.

Besides this monotonicity constraint, we also would like the errors between the provided values for the points z and their latent values l to be small, so that the estimated shape of the Pareto frontier is closer to the true one. This can be achieved by placing a prior inverse-gamma distribution over σ^2, whose density is given by:

$$p(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp\left(-\frac{\beta}{x}\right),$$

where $\Gamma(\cdot)$ is the gamma function. As $\beta \to \infty$, this prior is ignored, while $\beta \to 0$ indicates that there is no noise. In the results shown, we fix $\alpha = 3$ and vary β.

We define $f(y)$ as the final expected value $E[l^*|y^*, Z, Y, \theta]$, and the parameters θ are optimized to maximize the posterior probability, including gradient probability and σ^2 prior, of the training data Y and Z. We also add the monotonicity constraint on all training data for all directions, but it should be noted that we can also add only monotonicity constraint at a point

without defining its desired value. This allows us to find points that have $f(y) = 0$ but negative gradient and add the constraint on them, which in turn could improve the estimation.

To test the GP's performance as a surrogate, we consider the two test frontiers whose samples are given by $P_1 = [(0, 1), (\epsilon, \epsilon), (1, 0)]$, which is a convex frontier, and $P_2 = [(0, 1), (1 - \epsilon, 1 - \epsilon), (1, 0)]$, which is a concave frontier, both with $\epsilon = 10^{-3}$. Note that the points were purposely selected to test the ability to model very sharp frontiers. However, using only the points defined by P_1 and P_2 leads to a solution where $f(y)$ is almost 0 everywhere. To avoid this problem, we add a point $(1, 1)$, with target value 1, to P_1 and a point $(0, 0)$, with target value -1, to P_2. The parameters for the Gaussian process are found using gradient ascent in the samples' posterior probability.

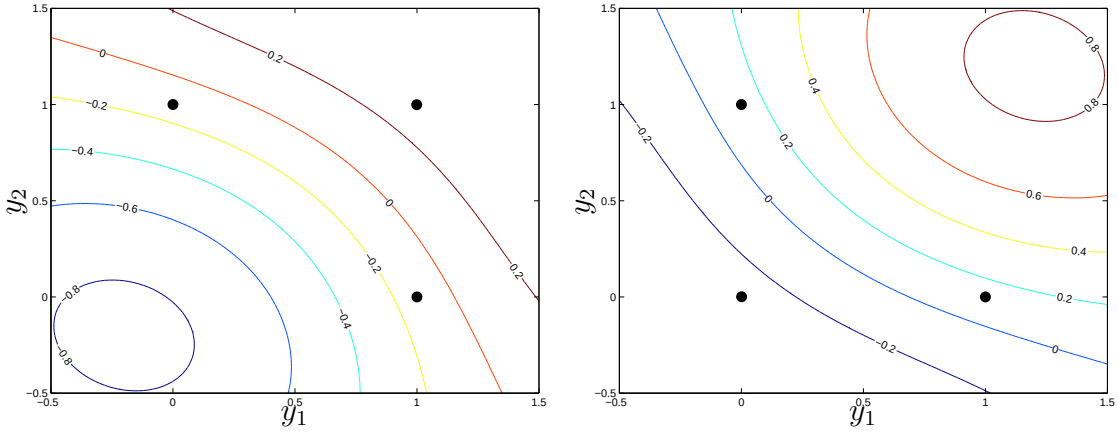
Figure 5.2 shows the resulting curves for different values of β. The first thing we notice is that, although $\beta \to \infty$ does not place any restriction on σ, which allows the observed points in the frontier to be far from their latent values that actually define the frontier, the resulting curve is still able to fit the general shape defined by the points provided.

As we reduce the value of β, the observed variance σ^2 is required to be smaller and the frontier shape gets better and better. Ideally, with $\beta = 0$, the latent points would be the same as the observed points, but this causes numeric problems due to the monotonicity information and can make it harder to satisfy the monotonicity constraint, due to the smoothness of the GP.

When we reduce the value to $\beta = 0.01$ and beyond, the resulting frontier is not valid anymore, with noticeable points with negative derivative. However, the largest difference in the concave problem is between points $(0.82, 1.055)$ and $(0.2, 0.985)$, with a total reduction in y_2 of just 0.07, and a similar result is obtained for the convex case. Therefore, this approximation is still close to the correct frontier and could be used to evaluate proposed solutions because it was built with the theoretical developments of this chapter in mind and tries to approximate them, which most likely provides better frontier estimates than methods that use traditional regression solutions, such as [31, 41, 43], where the manifold $f(y) = 0$ can have any shape.

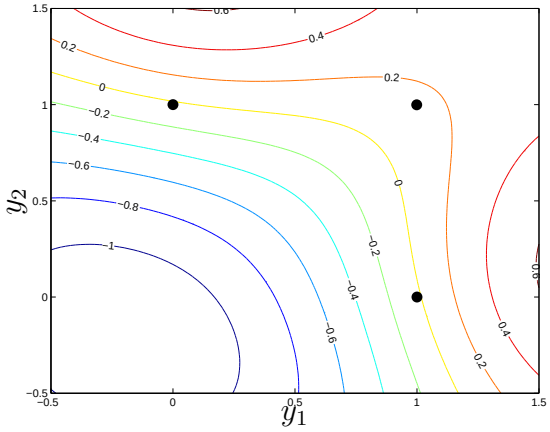
To evaluate the effect of using the gradient constraint, Fig. 5.3 shows a similar GP but without any information on the gradients. Although the expected Pareto frontier is correctly identified, there are also many points that do not belong to the frontier and where $f(y) = 0$. Since the unconstrained GP had better frontier estimates for the extreme points than the constrained GP, as all points between them and the knee satisfy the conditions, it appears that not every point benefits from the gradient constraint.

Even though both GP models failed to fully satisfy the theoretical conditions, we consider

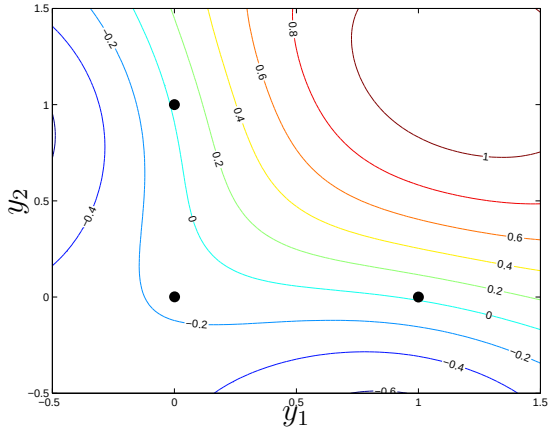


(a) Concave, $\beta \to \infty$

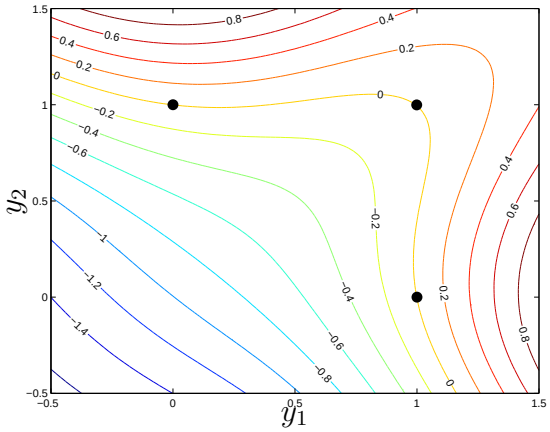
(b) Convex, $\beta \to \infty$



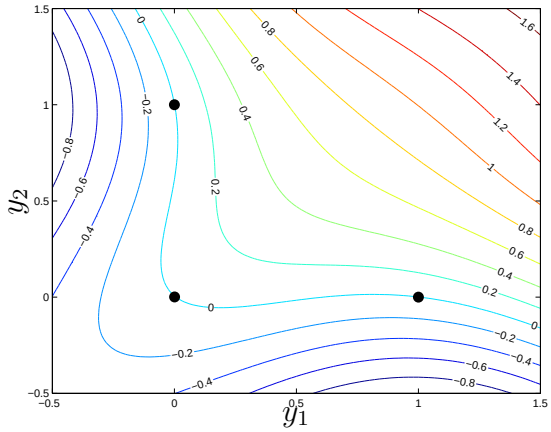
(c) Concave, $\beta = 0.1$



(d) Convex, $\beta = 0.1$

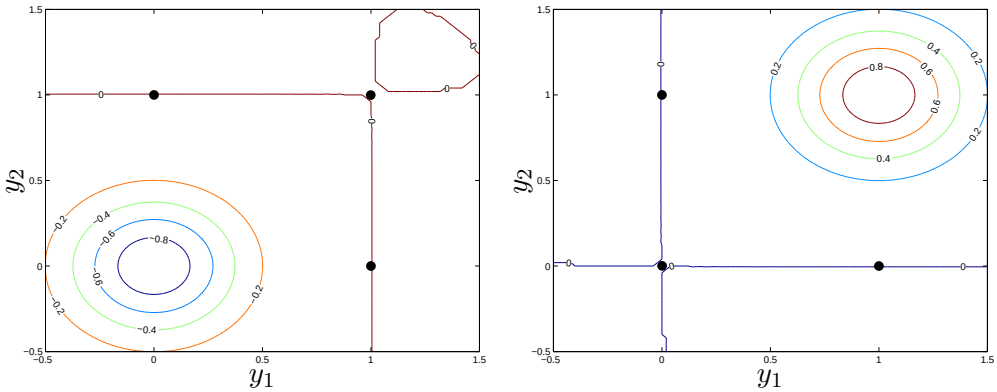


(e) Concave, $\beta \to 0$



(f) Convex, $\beta \to 0$

Figure 5.2: Contours for the $f(y)$ learned using a Gaussian process with derivative constraint. The black dots are the frontier points provided.



(a) Concave, $\beta = 0.1$ **(b)** Convex, $\beta = 0.1$

Figure 5.3: Contours for the $f(y)$ learned using standard Gaussian process. The black dots are the frontier points provided.

that the GP with derivative restriction performed better, both because there are some parameter sets that are able to satisfy the frontier conditions and because it does not violate the restrictions as much. Moreover, if the variance, which is not shown but is higher for points far from the inputs provided, is taken into account, then the violations of the GP with derivatives occur in a region with higher uncertainty than the violations of the pure GP.

Therefore, despite the minor violations of the GP with derivative constraints, this approximation is still close to the correct frontier and could be used to evaluate the proposed solutions.

5.4.2 Comparison to Existing SVM Surrogate

The surrogate method introduced in [31], like the method proposed in this chapter, is based on approximating the frontier directly from values in the objective space. This makes it a good candidate for comparison and validating the conjecture that existing methods may arbitrarily violate the conditions described in this chapter.

The one-class SVM used in [31] is defined by the following optimization problem:

$$\begin{aligned} \min_{w,\xi_i,\rho} \ & \frac{\|w\|^2}{2} + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i - \rho \\ \text{s.t. } & w^T\phi(x_i) \geq \rho - \xi_i \\ & \xi_i \geq 0, i \in \{1,\ldots,N\}, \end{aligned}$$

where $v \in (0, 1]$ and the feature-extraction function $\phi(x)$ is defined implicitly by the kernel

$$K(x, y) = \exp(-\gamma\|x - y\|^2),$$

which is similar to the kernel used for the GP.

One important difference between training an SVM and a GP is that the GP has a natural way to optimize its hyper-parameters by maximizing the data posterior probability, which automatically defines a trade-off between fitting the data and model complexity. For the SVM, we must use cross-validation [47], which reduces the number of points available to fit the model, since the data must be divided in the training and validation sets.

To compare the surrogate methods, we use one test problem from [3], which is also used in [31] to show the behavior of the proposed SVM surrogate. The problem is given by:

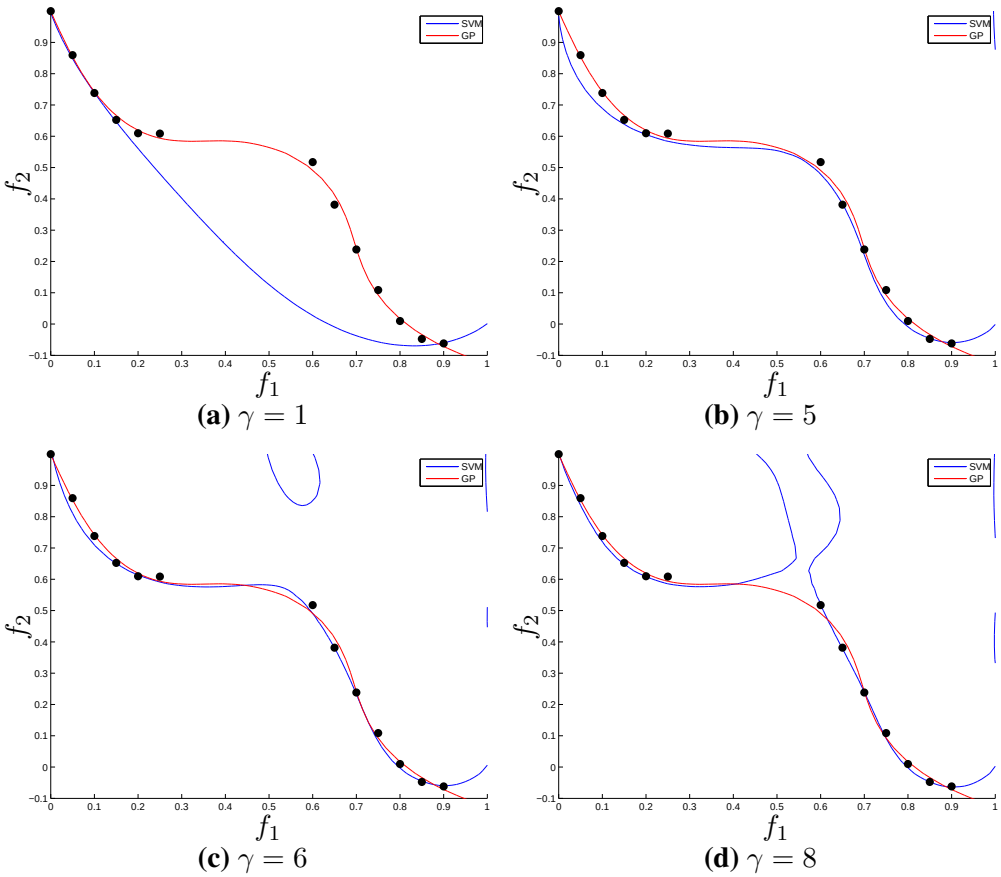
$$\begin{aligned} &\min\ f_1(x_1, x_2) = x_1 \\ &\min\ f_2(x_1, x_2) = 1 + x_2^2 - x_1 - 0.2\sin(3\pi x_1) \\ &\text{s.t. } x_1 \in [0, 1],\ x_2 \in [-2, 2]. \end{aligned}$$

We chose this problem because its Pareto frontier is discontinuous, which creates sharp changes in its associated estimated frontier, just like in Fig. 5.1, and makes it harder to approximate.

We chose $\nu = 10^{-3}$ so that the samples provided should be almost perfectly classified and we constrain the scales ρ_i in Eq. 2.12 to be equal, so that both methods can use the same features from the samples. The data set provided is composed of a grid with step 0.05 for both variables, which includes some points in the Pareto frontier. The full grid is used to fit the SVM because it provided better results than using just the non-dominated points, while only the non-dominated points and one reference with target value 1 at $(1.5, 1.5)$ are required for the GP.

Figure 5.4 shows the resulting approximations of the Pareto frontier using a GP with parameters learnt through gradient ascent in the data posterior probability, like in Section 5.4.1, and an SVM with different values of γ. The GP learns an appropriate shape from the samples provided despite the discontinuity in the frontier, but also slightly violates the constraints during the gap in $f_1 \in [0.3, 0.5]$. Moreover, in the absence of any information about the shape in the interval $f_1 \in (0.9, 1]$, because no point was provided there, the GP extrapolates a valid shape for the Pareto frontier.

The SVM is highly dependent on the parameter γ. When it is small, the shape learnt is very conservative and does not follow the shape defined by the points in the frontier. On the other hand, when it is large, the surrogate fits the points in the frontier better but also may define a function that violates greatly the conditions to be a valid Pareto frontier. The best value for γ that does not violate the constraints in the interval $f_1 \in [0, 0.9]$ is $\gamma = 5$. However, for this value the GP provides a better approximation of the Pareto frontier, as shown in Fig. 5.4b. Increasing



(a) $\gamma = 1$ **(b)** $\gamma = 5$

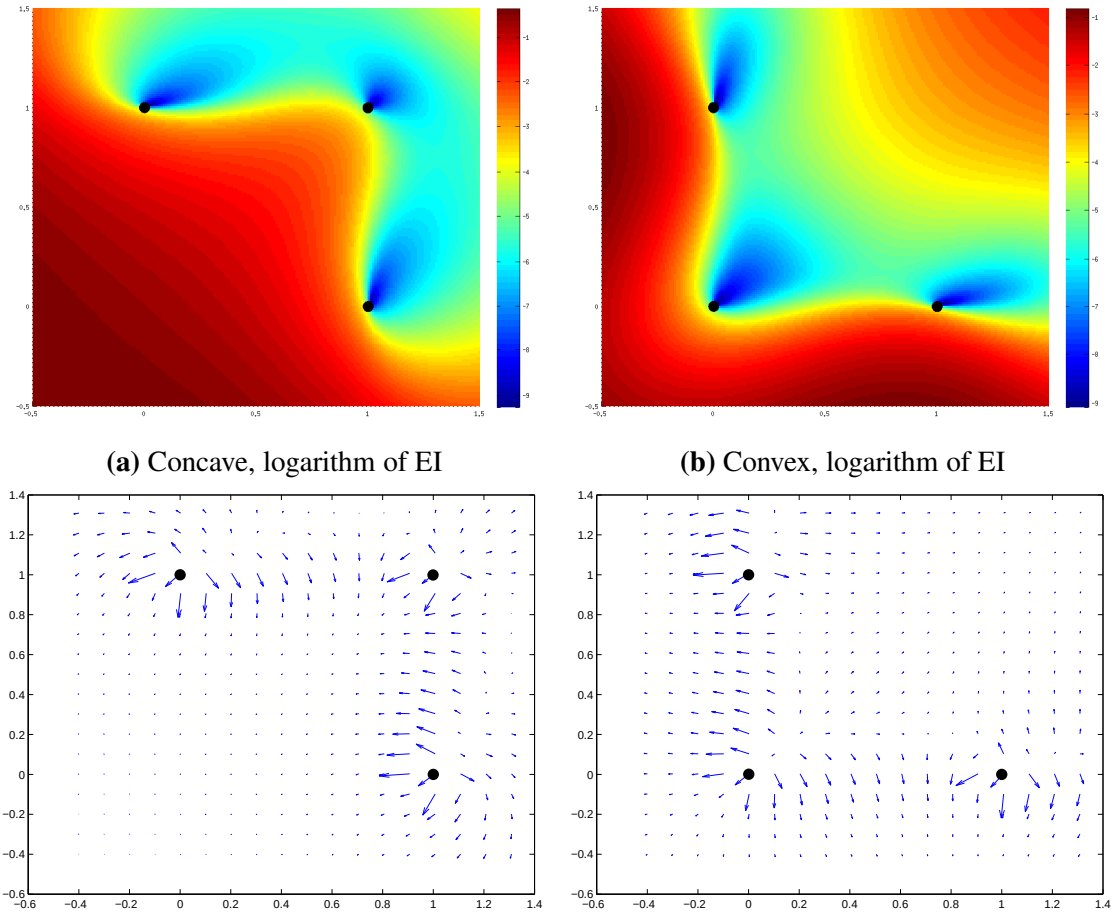
(c) $\gamma = 6$ **(d)** $\gamma = 8$

Figure 5.4: Estimated frontiers using SVM with different values of γ and Gaussian process. The points in the data set that belong to the Pareto frontier are shown as dots.

γ provides a better approximation, achieving a quality comparable to the GP, but also creates regions that violate the conditions to be a valid Pareto frontier more than the GP. Furthermore, $\gamma = 6$ defines a region that the SVM believes is part of the Pareto frontier but actually is very distant from it and inside the dominated region, as shown in Fig. 5.4c.

Besides these issues, the SVM also does not extrapolate well to the region $f_1 \in (0.9, 1]$. Close inspection shows that the dominated region defined by the SVM is finite, that is, it is described by a region in the objective space that is surrounded by an infinite region that the SVM believes is not dominated. This behavior shows that the learnt model carries no concept of the problem it is solving, which is to approximate a Pareto frontier, but describes a generic function approximation. The results in Fig. 5.4 provide evidence for the conjecture that existing methods proposed in the literature may arbitrarily violate the conditions described in this chapter.

Furthermore, if only the points at the Pareto frontier were provided for learning, then the region defined by the SVM would enclose only these points and would ignore the dominated region. Thus, the SVM method requires data in the dominated region while the GP method only requires the points at the frontier.



(a) Concave, logarithm of EI

(b) Convex, logarithm of EI

(c) Concave, gradient of EI

(d) Convex, gradient of EI

Figure 5.5: Expected improvement for the frontiers with $\beta \to 0$ in Fig. 5.2.

5.5 Expected improvement metric

According to Definition 17, the function $f^*(y)$ that describes the frontier has a value 0 on any point in the frontier, all the dominated region is positive and all the non-dominated region is negative. Using this definition, we can describe the multi-objective optimization problem as a mono-objective given by

$$\min_{y \in \mathcal{Y}} f^*(y), \tag{5.4}$$

where $\mathcal{Y}$ is the space of feasible solution in the objective space. As discussed in Chapter 1, it is more common to want to find multiple solutions, but we use this interpretation which allows us to look for one point at a time, similar to Chapter 4.

As shown in the previous section, we can build a surrogate $f(y) \approx f^*(y)$ for the Pareto frontier using Gaussian processes with the monotonicity constraint. Since this approximation is built on top of this definition, any point we can find with $f(y) < 0$ is a possible improvement over the current estimate, since it belongs to a region the approximation considered infeasible.

Therefore, with a given surrogate and using the expected improvement metric described in Section 2.4.4, we can define a metric for multi-objective optimization given by

$$EI(y) = P(1[f(y) < 0]), \tag{5.5}$$

which can be computed efficiently as analyzed in Section 2.4.4. Using this metric, the problem becomes finding a new point that maximizes the expected improvement given the current surrogate and is non-dominated by the current solutions. Note also that Eq. (5.5) only uses information from the objective space, allowing it to be used both with real evaluation of objectives or with their approximations.

Figure 5.5 shows the heatmap for the two best surrogates for the example frontiers in Figure 5.2. Clearly, in the regions close to the best solutions so far, the metric favors samples that can reach the non-dominated region, as one would expect. This is further confirmed by the gradient, whose vector field is also show in Figure 5.5. Starting the search close to the frontier and following the gradient moves the proposed solution to the non-dominated region and incentivizes spreading over the objective space, which indicates optimizing this metric through classic methods like gradient descent has the desired properties of algorithms solving MOO problems.

However, we must also highlight that some dominated regions have high value for the expected improvement. This is caused both by the absence of samples in that region, which increases the variance and thus the uncertainty about the value in those areas, the zero mean used to avoid biasing the surrogate, which means the function returns to zero in infinity, and the constraint placed on the Gaussian process, which can only represent local information. Therefore, following the metric blindly could make one waste some evaluations in dominated regions.

A natural way to minimize this issue is to also use dominated solutions when defining the surrogate. Using an algorithm like non-dominated sorting [42], we can group previously evaluated solutions on sets such that, if all points in the sets with lower order are removed, the points in a set satisfy the conditions of a Pareto frontier. Since Gaussian processes are generic function approximators and the monotonicity constraint can be applied anywhere, we can use the domination count as the regression target, satisfying the original definition that the Pareto frontier should have value zero, and we can also apply the constraint to all samples found, since they describe the general shape of the solution space.

5.6 Conclusion

In this chapter, we have introduced the necessary and sufficient conditions that functions must satisfy so that their solution spaces describe estimated frontiers. These conditions follow from the definition of an estimated frontier and are extended for differentiable functions, which allows easier verification of the conditions.

Based on these conditions, a Gaussian process (GP) was tested on challenging toy problems with very sharp Pareto frontiers. The GP was extended to include the theoretical conditions as soft probabilistic constraints and a regularization term was added to avoid large deviations between the points and their latent values. The mean latent value is used as surrogate for the Pareto frontier, and some values of the regularization constant allowed a correct frontier estimate to be found.

However, when the regularization becomes too strong, the surrogate violates the constraints that define a valid estimated frontier on some points, but this occurs far from the given inputs and the deviation is small. This suggests that, even under these conditions, the proposed function could be used to provide insight on the shape of the Pareto frontier, and possibly provide more realistic estimates than other methods that do not take the restrictions into consideration during their design.

To validate this hypothesis and the conjecture that existing surrogate methods may violate the conditions described in this chapter, we compared the proposed GP with a one-class SVM used in [31] on one of the test problems described in the same paper. We showed that the GP again violates the constraints by small values and provides a good estimate for the Pareto frontier, while the SVM defined a worse estimate or violated the conditions more than the GP. Furthermore, the dominated region defined by the SVM is bounded by what it represents as the non-dominated region, while the GP correctly divides the space in two infinite areas.

Besides being a better surrogate for the Pareto frontier, the GP has the data posterior probability as an innate measure that can be used to optimize its hyper-parameters and only requires data at the frontier. On the other hand, the SVM must use some method, like cross-validation [47], to optimize its hyper-parameters and it requires data in the dominated region to define better approximations.

We highlight that, although GPs were used together with the theory on this chapter to approximate the Pareto frontier, the theory is general and does not depend on the specific choice of the function descriptor. Therefore, other models that are able to deal with the constraints

imposed by the theory, in either a soft or hard way, should be able to learn the desired shape of the Pareto frontier too. Nonetheless, we are not aware of any other method to create the score function in which the constraints are as easy to include as in the GP. Additionally, a GP provides robustness to changing the number of points used in the estimation and an inherent performance metric, the expected improvement.

Further investigations involve studying the behavior of the GP to approximate the Pareto frontier with real benchmarks and using some multi-objective optimization algorithm, such as NSGA-II [8], to provide the points. Since the objectives tend to be smoother than in the example frontier provided [48], we expect the estimated frontier described by a GP to fit the Pareto frontier even better in these problems.

Moreover, since the only requirement for the surrogate is that the Pareto frontier is approximated by the null space and the exact value on other parts of the objective space are not relevant, the GP could be used to fit a regression model on the individuals of a population where the target value is monotonically increasing in the objective space. Standard performance measures in multi-objective optimization, such as the class in non-dominated sorting [8] and the dominance count [12], satisfy this property and can be used as targets of the regression. In this case, the GP would not only define the Pareto frontier, but would also define a measure of the distance between a given point and the approximated Pareto frontier.

Additionally, by creating a surrogate for the Pareto frontier using a Gaussian process like the one proposed in this chapter, we are performing a scalarization of the objectives like ParEGO [35] and MOEA/D-EGO [37], but without requiring the additional parameters used to combine the objectives to create the new function. In our case, we use the values of the objectives directly to create the surrogate, without first creating a scalar function and then estimating a surrogate for it. Nonetheless, the EGO procedure [16] can be used in this surrogate, so optimization algorithms can be used to find the point with higher expected improvement to be evaluated in the real objectives.

Since this approach is independent from the model of the objectives themselves, they can be individually approximated by Gaussian processes, making use of side-information [14] and multi-task learning methods [40], and the uncertainty provided by the surrogates of the objectives can be propagated to the Pareto frontier approximation [49, 50]. Moreover, computing the expected improvement for the hypervolume, like in [38] and [39], can be expensive [51], but computing the mean and variance of a prediction using a surrogate for the Pareto frontier is polynomial on the number of points and objectives considered, which may provide a speedup for running the evolutionary algorithm with many objectives, as many evaluations of the surrogates are performed.

Finally, another interesting line of research is to evaluate when the derivative constraints on the points provided is beneficial, since in some points it avoids incorrect association of other points with the frontier, like around the knee in the unconstrained GP shown in this chapter, and in others it may make the estimated shape not satisfy the constraints, like the points in the constrained GP also shown in this chapter. This could not only provide better fit, but may also increase the fitting speed, since fewer constraints need to be evaluated, which reduces the size of the GP and the number of expectation propagation steps required. Therefore, an iterative algorithm that adds the constraints as needed should be pursued.

Chapter 6

Conclusion

The research presented in this thesis has pushed the boundaries of multi-objective optimization problems in 3 different directions with practical contributions:

- showing how reinterpreting single-objective problems can lead to higher performance;
- showing how gradient-based methods can be used for optimizing multi-objective problems faster than evolutionary algorithms; and
- showing how to build an approximation to the Pareto frontier that allows merging information from all objectives into one performance metric, based on techniques already commonly used in single-objective optimization.

Some of these direction applications are built on top of theoretical results, also presented. These results provide significant contributions to the scarce body of research on the theory of multi-objective optimization and its associated metrics. Our hope is that, by connecting to known areas in single-objective optimization, other researchers will have greater incentive to either investigate the use of results into multi-objective or adapting multi-objective views into single-objective, providing a two-way street of cooperation.

6.1 Open questions and future work

This work is just the beginning of many possible future investigations. We highlight some of the most interesting directions below for completeness.

6.1.1 What other optimization analysis can we bring to the hypervolume indicator?

Section 3.1 presented bounds on both directions connecting the mean loss and the hypervolume indicator, which is a very common metric in multi-objective optimization. Although there is some analysis of its behavior [6], it is mainly qualitative (e.g. what kind of guarantees it has when used to compare two sets of solutions), which is different from the quantitative bounds commonly found in single-objective.

With this initial bound in place and the proposed method to reach it, it should be possible to either apply other bounds, apply similar techniques to reach bounds or expand the results here for multiple solutions.

6.1.2 How to choose the parameter for using the single-solution hypervolume optimization?

Section 3.2 showed that the hypervolume optimization can be a good replacement to the mean loss. However, it introduces another hyperparameter that needs to be tuned: the reference point. The experiments show that pressing the optimization too much by setting a low value can give bad results, but slightly relaxing it can improve performance considerably.

This appears to indicate a trade-off between being too susceptible to noisy or wrong samples and trying to find a model that fits all data well. Although scheduling can help to avoid this degradation and even improve the performance across the board, the initial setting is still important.

6.1.3 What is the impact of the reference in the hybrid multi-objective optimization?

Chapter 4 presented an algorithm for performing hybrid optimization for the multi-objective formulation, which achieved considerably better results than the evolutionary algorithms it was compared to. However, the new algorithm requires knowledge of a reference point which will be used when computing the hypervolume contribution.

Although this point is not relevant in many cases, since we first optimize the individual objectives and the hypervolume contribution of new points would be the same no matter the reference, it can have an impact if the initial optimization does not work well, as shown in

Figure 4.12. The effect of choosing this reference should be further evaluated when applying the algorithm.

6.1.4 How to integrate objectives' surrogates with the approximated Pareto frontier?

The method presented in Section 5.4 to approximate the Pareto frontier performs an optimization in the posterior likelihood of the objective values. However, if the real objective functions are unknown and surrogates are used instead, we might be able to integrate their optimization such that the final estimate is better.

Instead of optimizing the approximated Pareto frontier and the objectives' surrogates isolated, one can link them by also optimizing the Pareto frontier approximation when the surrogates are used at the evaluation points instead of the real objective. Since the conditions upon which the frontier approximation is built should hold true, as we proved, objective surrogates that do not respect them should be penalized. This research direction has considerable potential, as it regularizes the surrogates while also linking information between them.

6.1.5 How to integrate historical evaluations with the approximation?

The method used in Section 5.4 used only the points candidates to the Pareto frontier. However, since the underlying problem being solved is a function approximation, one might be able to extract information about the shape of the problem even from previous evaluations already dominated.

One reasonable approach is to use the domination count, which gives a general sense of the direction in which the curve grows and is consistent with the monotonic constraint added. On the other hand, this can lead to big difference in target values if the evaluations are close to each other, since it is just a discrete count. This in turn could lead to bad behavior on the Gaussian process due to its smoothness and such condition should be investigated.

6.1.6 How well does using the expected improvement as performance metric work?

Section 5.5 showed how a metric could be derived from the Pareto frontier approximation using the expected improvement, a common metric when optimizing surrogates for single-objective problems. Moreover, with appropriate choices of mean and kernel functions, it can be differentiable, according to Equation (2.18), allowing it to be optimized using gradient methods.

Considering that this metric is cheaper to compute than the hypervolume, since it is cubic in the general case while the hypervolume is exponential [30], it should be interesting to evaluate its performance as metric to be optimized on every step. Even if it leads to slower convergence than other metrics in terms of steps, its faster computation might lead to overall better performance.

Bibliography

[1] X. Gandibleux, *Multiple criteria optimization: state of the art annotated bibliographic surveys*, ser. International Series in Operations Research & Management Science. Springer US, 2006.

[2] K. Miettinen, *Nonlinear multiobjective optimization*. Springer US, 1999.

[3] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001.

[4] C. R. B. Azevedo, "Anticipation in multiple criteria decision-making under uncertainty," Ph.D. dissertation, University of Campinas, 2014.

[5] K. Deb, "Multi-objective optimization," in *Search methodologies*. Springer, 2014, pp. 403–449.

[6] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *Evolutionary Computation, IEEE Transactions on*, vol. 7, no. 2, pp. 117–132, 2003.

[7] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[8] K. Deb, A. Pratap, S. Agarwal, and T. A. M. T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.

[9] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," 2001.

[10] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: on the design of Pareto-compliant indicators via weighted integration," in *Evolutionary multi-criterion optimization*. Springer, 2007, pp. 862–876.

[11] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point," in *Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*. ACM, 2009, pp. 87–102.

[12] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

[13] C. E. Rasmussen, J. M. Bernardo, M. J. Bayarri, J. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, "Gaussian processes to speed up hybrid Monte Carlo for expensive Bayesian integrals," in *Bayesian Statistics 7*, 2003, pp. 651–659.

[14] *Gaussian process for machine learning*.

[15] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2001, pp. 362–369.

[16] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[17] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[18] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-objective optimization: an engineering design perspective." in *EMO*, vol. 5. Springer, 2005, pp. 14–32.

[19] D. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999.

[20] D. Luenberger and Y. Ye, *Linear and nonlinear programming*. Springer US, 2008.

[21] P. A. N. Bosman, "On gradients and hybrid evolutionary algorithms for real-valued multiobjective optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 16, no. 1, pp. 51–69, 2012.

[22] M. Emmerich and A. Deutz, "Time complexity and zeros of the hypervolume indicator gradient field," in *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation III*. Springer, 2014, pp. 169–193.

[23] V. A. S. Hernández, O. Schütze, and M. Emmerich, "Hypervolume maximization via set based Newton's method," in *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*. Springer, 2014, pp. 15–28.

[24] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[25] F. Biscani, D. Izzo, and C. H. Yam, "A global optimisation toolbox for massively parallel engineering optimisation," *arXiv preprint arXiv:1004.3824*, 2010.

[26] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–, [Online; accessed 2017-07-01]. [Online]. Available: http://www.scipy.org/

[27] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft computing*, vol. 9, no. 1, pp. 3–12, 2005.

[28] J. Knowles and H. Nakayama, "Meta-modeling in multiobjective optimization," in *Multiobjective Optimization*. Springer, 2008, pp. 245–284.

[29] I. Voutchkov and A. Keane, "Multi-objective optimization using surrogates," in *Computational Intelligence in Optimization*. Springer, 2010, pp. 155–175.

[30] N. Beume, C. M. Fonseca, M. López-Ibáñez, L. Paquete, and J. Vahrenhold, "On the complexity of computing the hypervolume indicator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1075–1082, 2009.

[31] Y. Yun, H. Nakayama, and M. Arakava, "Generation of Pareto frontiers using support vector machine," in *International Conference on Multiple Criteria Decision Making*, 2004.

[32] B. Liu, Q. Zhang, and G. Gielen, "A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 2, pp. 180–192, 2014.

[33] H. Karshenas, R. Santana, C. Bielza, and P. Larranaga, "Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 4, pp. 519–542, 2014.

[34] D. Horn, T. Wagner, D. Biermann, C. Weihs, and B. Bischl, "Model-based multi-objective optimization: taxonomy, multi-point proposal, toolbox and benchmark," in *Evolutionary Multi-Criterion Optimization*. Springer, 2015, pp. 64–78.

[35] J. Knowles, "ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 1, pp. 50–66, 2006.

[36] J. Knowles, D. Corne, and A. Reynolds, "Noisy multiobjective optimization on a budget of 250 evaluations," in *Evolutionary Multi-Criterion Optimization*. Springer, 2009, pp. 36–50.

[37] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with gaussian process model," *Evolutionary Computation, IEEE Transactions on*, vol. 14, no. 3, pp. 456–474, 2010.

[38] M. Emmerich, K. C. Giannakoglou, and B. Naujoks, "Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels," *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 4, pp. 421–439, 2006.

[39] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, "Multiobjective optimization on a limited budget of evaluations using model-assisted $\mathcal{S}$-metric selection," in *Parallel Problem Solving from Nature–PPSN X*. Springer, 2008, pp. 784–794.

[40] E. V. Bonilla, K. M. Chai, and C. Williams, "Multi-task Gaussian process prediction," in *Advances in neural information processing systems*, 2007, pp. 153–160.

[41] I. Loshchilov, M. Schoenauer, and M. Sebag, "A mono surrogate for multiobjective optimization," in *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*. ACM, 2010, pp. 471–478.

[42] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to nondominated sorting for evolutionary multiobjective optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 19, no. 2, pp. 201–213, 2015.

[43] I. Loshchilov, M. Schoenauer, and M. Sebag, "Dominance-based Pareto-surrogate for multi-objective optimization," in *Simulated Evolution and Learning*. Springer, 2010, pp. 230–239.

[44] V. G. da Fonseca and C. M. Fonseca, "The attainment-function approach to stochastic multiobjective optimizer assessment and comparison," in *Experimental methods for the analysis of optimization algorithms*. Springer, 2010, pp. 103–130.

[45] A. O'Hagan, "Some Bayesian numerical analysis," *Bayesian statistics*, vol. 4, pp. 345–363, 1992.

[46] J. Riihimäki and A. Vehtari, "Gaussian processes with monotonicity information," in *International Conference on Artificial Intelligence and Statistics*, 2010, pp. 645–652.

[47] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[48] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 5, pp. 477–506, 2006.

[49] A. Girard, C. E. Rasmussen, J. Quinonero-Candela, and R. Murray-Smith, "Gaussian process priors with uncertain inputs: application to multiple-step ahead time series forecasting," 2003.

[50] A. Girard, "Approximate methods for propagation of uncertainty with Gaussian process models," Ph.D. dissertation, University of Glasgow, 2004.

[51] M. Emmerich and J. Klinkenberg, “The computation of the expected improvement in dominated hypervolume of pareto front approximations,” Tech. Rep., 2008.