



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Departamento de Computación

**A New Gradient Free Continuation Method for the Treatment of
Constrained Multiobjective Optimization Problems**

Tesis que presenta

Ing. Eduardo Salinas Márquez

para obtener el grado de

Maestro en Ciencias

en Computación

Director de la Tesis

Dr. Oliver Schütze

México, D.F.

Agosto del 2013

Resumen

La optimización multiobjetivo ha sido una fuerza de empuje hacia nuevas aplicaciones en áreas tales como las matemáticas, la física y la ingeniería. En años recientes, se han desarrollado diversas técnicas para el manejo de la optimización de un solo, y múltiples objetivos, desde estrategias de búsqueda lineal, a estrategias basadas en poblaciones como los algoritmos evolutivos.

Uno de estos enfoques de búsqueda lineal, es moverse sobre el conjunto de interés, el conjunto de Pareto, por medio de métodos de continuación multiobjetivo. Estos métodos son muy efectivos localmente, sin embargo, requieren -hasta ahora- información de gradiente o de la Hessiana (derivada de segundo orden) lo cual limita su aplicación.

En este trabajo se propone un nuevo método de continuación basado en la Búsqueda Dirigida Directa (DDS por sus siglas del inglés *Discrete Directed Search*). Este nuevo algoritmo fue desarrollado para resolver problemas de optimización multiobjetivo, sin la necesidad de utilizar información de gradiente. Además, el nuevo algoritmo de continuación por DDS puede manejar restricciones de igualdad y de desigualdad. También presentamos ejemplos numéricos sobre problemas académicos, así como un problema relacionado con el diseño de misiones espaciales. Se han obtenido aproximaciones de alta calidad utilizando este método, haciéndolo una alternativa interesante para resolver problemas de optimización multiobjetivo en ingeniería.

Abstract

Multiobjective optimization has been a pulling force towards new applications in fields such as mathematics, physics, and engineering. In recent years, many new techniques have been developed to treat single and multiple objective optimization, ranging from mathematical programming techniques such as line search methods, to population based strategies such as evolutionary algorithms.

One of these line search approaches, is to move along the set of interest, the Pareto set, by means of multiobjective continuation methods. Such methods are very effective locally, however, they require -so far- gradient or even Hessian (second order derivative) information, which limits their applications.

In this work, we propose a new continuation method, based on the Discrete Directed Search (DDS) method. This new algorithm was developed to solve multiobjective optimization problems, without the need of gradient information. In addition, this new DDS Continuation Method is capable of handling equality and inequality constraints. We also present numerical examples on academic models, as well as one problem related to space mission design. High quality approximations have been achieved using this method, making it an interesting alternative to solve engineering multiobjective optimization problems.

Agradecimientos

A mis padres por motivarme, por impulsarme para seguir adelante y convertirme en una mejor persona.

A mis hermanos porque siempre han estado cuando los necesito.

A toda mi familia, ya que sin ellos no podría haber llegado a donde estoy.

A Ana por todo su amor y apoyo incondicional.

A mis amigos y compañeros de la maestría, que me apoyaron durante todo este recorrido.

A mi asesor el Dr. Oliver Schütze, porque su idea dio vida a este proyecto, por apoyarme para enviar este trabajo a congresos y revistas, y porque resolvió la mayor parte de mis dudas en todo este proceso.

A mis sinodales el Dr. Luis Gerardo de la Fraga y la Dra. Xiaou Li, cuyas acertadas observaciones y correcciones me ayudaron a mejorar mi trabajo.

A todos los profesores del Departamento de Computación del CINVESTAV, que me formaron como un mejor estudiante día a día.

A las secretarias del departamento, que nos guiaron desde que llegamos hasta que salimos.

Al CONACYT, pues sin el apoyo de mi beca de maestría no podría haber realizado mis estudios.

List of Figures

2.1	General predictor corrector method. The points u_i are the best approximated solutions to $c(s_i)$	19
2.2	Predictor orrector iteration. The predicted point v_{i+1} is corrected to w_{i+1} , which is the closest to the zero point u_{i+1}	20
2.3	Basic idea of the Directed Search Descent Method	21
2.4	Zigzag method. The “zig” step starts from an optimal stolution x_1 , moves in direction g_1 . Then the “zag” step moves to a new optimal solution x_2 using direction g , which is a projection of g_2 over g_1	28
2.5	Basic concepts for an evolutionary algorithms.	29
3.1	Directed Search Method. Due to the local nature of the gradient information, the predictor step usually steers away from the desired curve.	35
3.2	Corrector step in the Directed Search Descent Method. When we move away from the desired direction, we can shoot back using a corrector step.	36
3.3	Graphical interpretation of the curvature-based control. If the curve of the Pareto set is fixed, the step size can be computed by trigonometric equations.	42
3.4	Graphical interpretation of the cones of tolerance control. Here we see that the step size t_1 is not acceptable since it violates the tolerance β	43
3.5	Graphical representation of the Pareto dominance control. The new step size must produce a value inside the descent cone of the set of objective functions.	44
3.6	Example of the convex weight as a stopping criteria for an MOP with two objectives.	51

3.7	Numerical result for the gradient free continuation method using $\epsilon = 0.04$. The number of used function evaluations was 191.	53
3.8	Numerical result for the gradient free continuation method using $\epsilon = 0.08$. The number of used function evaluations was 95.	54
3.9	Numerical result for the gradient free continuation method, HCS and MOEA/D. Our method surpasses the other algorithms in terms of function evaluations and in the approximation of the Pareto front.	55
3.10	Numerical result for the gradient free continuation method and the HCS. Our method surpasses the other algorithm in terms of function evaluations and in the approximation of the Pareto front.	56
3.11	Numerical result for the DDS continuation method, HCS and MOEA/D. Our method surpasses the other algorithm in terms of function evaluations and in the approximation of the Pareto front.	57
4.1	Numerical result of the DDS continuation method and the HCS on a box- constrained MOP. Left are the projections of the Pareto set onto the first three coordinates and the candidate solutions and right analog figures in objective space.	63
4.2	Numerical result for the gradient free continuation method on an equality constrained MOP. Apparently, the algorithm is able to compute a well spread set of solutions with a low budget of function evaluations.	66
4.3	Numerical result for the gradient free continuation method on a nonlinear equality constrained MOP. Apparently, the algorithm is able to compute a well spread set of solutions with a low budget of function evaluations.	67
4.4	Close-up of the numerical result for the gradient free continuation method on a nonlinear equality constrained MOP. There is no corrector step and hence, the method does not ensure a perfect approximation.	68
4.5	Numerical result for the gradient free continuation method on a linear in- equality constrained MOP. Apparently, the algorithm is able to compute a well spread set of solutions with a low budget of function evaluations.	69

4.6	Numerical result for the gradient free continuation method on a nonlinear inequality constrained MOP. We can see that our method follows the active constraint.	70
5.1	Graphical representation of the three-impulse test problem. The spacecraft departs from an orbit around Earth and performs several movements until it reaches its final destination.	74
5.2	Numerical result for the gradient free continuation method on the three-impulse trajectory problem. Left is the reference front along with the two iterations of the DDS continuation method. Right, close-up to the covered areas.	76
5.3	Numerical result for the DDS continuation method, the HCS, and <i>Matlab's</i> genetic algorithm in parameter space. Left is the projection over the variables x_1, x_2 and x_3 . Right, the projection over the three decision variables x_3, x_4 and x_5	77
5.4	Numerical result for the Hill Climber with Sidestep on the three-impulse trajectory problem. Left is the reference front along with the two iterations of the HCS. Right, close-up to the covered areas.	78
5.5	Numerical result for <i>Matlab's</i> genetic algorithm on the three-impulse trajectory problem. Left is the reference front along with the two iterations of the HCS. Right, close-up to the covered areas.	78

Contents

Figure Index

1	Introduction	1
1.1	Motivation	3
1.2	The Problem	4
1.3	General and Particular Aims of the Project	5
1.4	Contributions	6
1.5	Organization of the Thesis	6
2	Background	9
2.1	Theoretical Background	10
2.1.1	Basic concepts	11
2.1.2	Optimality conditions	12
2.1.3	Descent directions	13
2.2	Related Work	14
2.2.1	Scalarization strategies	14
2.2.2	Line search strategies	16
2.2.3	Numerical continuation methods	17
2.2.4	The Directed Search method	19
2.2.5	DS: Descent method	20
2.2.6	DS: Continuation method	22
2.2.7	Discrete Directed Search: Descent method	24

2.2.8	Hill Climber with Sidestep	26
2.2.9	Zigzag search method	27
2.2.10	Evolutionary algorithms	27
2.3	Final Comments	30
3	Gradient Free Continuation Method for Unconstrained Problems	33
3.1	Gradient free line search	34
3.2	Step size control strategies	38
3.2.1	Wolfe step size control	40
3.2.2	Curvature-based step size control	40
3.2.3	Cones of tolerance	41
3.2.4	Pareto dominance control	43
3.2.5	Final comments	43
3.3	Gradient free continuation method	44
3.3.1	Steering the search	44
3.3.2	Computation of $\mathcal{F}(x)$	46
3.3.3	Step size control	47
3.3.4	Predictor corrector iteration	49
3.3.5	Stopping criteria	49
3.4	Numerical Results	52
3.4.1	Convex problem, two objectives, two variables	52
3.4.2	Convex problem, two objectives, eight variables	53
3.4.3	Witting problem	54
3.4.4	Okabe 1	55
3.4.5	Final comments	57
4	Constraint Handling	59
4.1	Box Constraints	60
4.1.1	Numerical results	62
4.2	Equality Constraints	64

4.2.1	Numerical results	65
4.3	Inequality Constraints	67
4.3.1	Numerical results	68
4.4	Final Comments	70
5	Three-Impulse Space Trajectory Design Problem	71
5.1	Three-Impulse Test Problem	72
5.2	Lambert's Problem	74
5.3	Numerical Results	75
5.4	Final Comments	79
6	Discussion and Future Work	81
6.1	Future Work	83
	Bibliography	85

Introduction

Optimization is the process of seeking to improve a certain *objective*, which depends on a set of *unknown elements*. This is, one tries to find the values of these elements to minimize or maximize the objective, which represents a quantitative measure.

Optimization is an interesting and challenging field inside mathematics. People have always sought to improve their goals, be they in their daily lives or in academic problems. It is clear that better, more efficient solutions to these problems are always needed. However, the computation of better solutions usually come with a cost, and then we are forced to ponder as to which solution is the best given our context. This is where optimization comes into play. Since the beginning of the 20th century, mathematicians were interested in finding formulas to obtain optima by means of calculus. Work done by Newton, Gauss or Lagrange boosted the interest in such techniques. It was later noticed that these problems could be translated into real-world applications and from then on, optimization was not only a mathematical concept, but an area that could help anyone in need. In the 1930's, the United States of America military applied optimization on their training schedules, as a way to maximize the time used by their troops.

Further research into this field, revealed that not only were people searching for a single answer to their needs, or more importantly one could not assign a specific “value” to the given problem. The term of multiobjective optimization came to be. Multiple objectives, or multiobjective optimization is the search for a wide range of optimal solutions, where we sacrifice the efficiency of one or more solutions, to gain or improve another one. This balance constitutes the area of interest of this work.

Multiobjective optimization has taken a fundamental role in society throughout the last years. Its application in engineering is wide. For example, in manufacturing, improving the quality of products and at the same time minimizing their cost is of great interest for manufacturers and clients. Another example is in the design of many products. This requires to take into account many variables such as the amount of material to be used, the shape that it will take, the performance and interaction with other elements.

Applications in aircraft structural design require a detailed analysis to satisfy material and security conditions. One of the main concerns in aircraft structure is the design of airfoils, i.e., the shape of the wings. The objectives here are to minimize the drag caused by the flow of air, and at the same time to maximize the lift. Many authors have attacked airfoil structure designs as well as other engineering-related problems concerning aircrafts [1–8].

Other interesting applications are in the field of space mission designs. In these situations, where a slight variation in the decisions can lead up to a huge increment of costs for a space agency. For example, changing the trajectory of a satellite going into deep space, can make a difference in years for its final destination. Usually, orbits are calculated as to use the gravity of nearby planets to propel or eject a satellite or shuttle into the desired path and thus save time and fuel for the traveling satellite. Optimization has been used as a tool in space mission designs, mainly to improve the trajectory of an object in space as to minimize the time of departure.

To name a few examples, Vasile et al. [9], formulated the problem of a mission to Mars and back to Earth as a global optimization problem, this is, to find the best solutions in the entire design space. They considered straight trajectories, along with catapulting by means of gravity pulls from said planets, and obtained a set of optimal solutions to allow the mission control to select the one that suited their needs at that particular time.

Another example, by Sánchez et al. [10], generated a mathematical model for the deviation of objects approaching Earth by means of collision, this is, to minimize the mass of the object impacting an asteroid and to maximize the deviation caused by the explosion.

There are several approaches for solving real-world problems using multiobjective optimization. The simplest one is by means of scalarization methods [11–13], which break

down the initial problems into a set of single objective ones and then attempt to minimize each one separately. Other ideas are to use nature-inspired techniques such as evolutionary strategies [14, 15], in which a group of data points “evolves” throughout time by combining promising solutions. In order to speed up convergence for this kind of algorithms, a new approach is to fuse them with local search methods. This group of algorithms is known as memetic methods [16]. Finally if a partial set of the best solutions is found, one can turn to continuation methods [17, 18], which, given an initial point (considered the best by a given function), move through objective space, obtaining solutions which are as good as the initial one. The broad set of attacks to real-world problems leads to the work presented in this thesis, in which a local search strategy is combined with a continuation method to obtain the set of best solutions for a given problem.

In this chapter, we present a quick overview of the thesis. In Section 1.1 we prese the motivation of our work, and the main difficulties arising from engineering problems, as well as the main problem attacked in this work. In Section 1.2 we briefly describe the objectives of this thesis, and in Section 1.3 we state the contributions obtained from this work. Finally a description of the structure of this document is presented in Section 1.4.

1.1 Motivation

Multiobjective optimization in engineering has evolved and new strategies have been developed in order to solve aeronautical problems. Bowen and Zhide [1] used single objective optimization to assess airfoils in several operational conditions. Chamis [8] stated an engine design problem using multiobjective optimization, and Gaiddon et al. [2] used multiobjective optimization as well on missile inlets. Both concluded that multiobjective optimization obtained better results and that obtaining a Pareto front simplifies the process of satisfying design constraints. Finally, in a real world application Chiba et al. [3] attacked the problem of the airfoil design using multiobjective optimization, and a combination of genetic algorithm and parallel processing yielded good results.

One of the most recent attempts to solve an airfoil and engine problem was developed

by Kipouros et al. [4]. The authors developed a variation of the taboo search algorithm and successfully improved the results of a single objective optimization problem (SOP) previously used in the same design problem by Harvey et al. [5]. The problem was designed for two objectives but it was later extended to three and four objectives [6]. The fact that the design space can be extended into more objectives, shows that real world problems can be attacked with different optimization approaches.

Evolutionary algorithms have also been used to attack aeronautical design problems. These algorithms use a population and generate solutions by breeding or mutating the current solutions to generate a new population, and keep the non-dominated solutions. In 2004, Emmerich and Naujoks [7] proposed a modification of two evolutionary algorithms: NSGA-II [14] and ϵ -MOEA [19]. The authors implemented metamodels which approximate function values in airfoil design. This modification was made to reduce the time in the time-intensive function calls needed to evaluate airfoil design parameters. Their results showed an improvement in the spread of solutions as well as in the rate of convergence towards the Pareto front. However one major issue, according to Emmerichs on multiobjective evolutionary algorithms (MOEAs), is that the larger the population, the greater the number function evaluations. In the context of engineering problem, function evaluations are a time-consuming task, and thus, MOEAs become infeasible.

Due to the difficulty of solving real-world problems using optimization techniques and the time-consuming tasks that convey such problems, a method that minimizes the time and cost (measured in terms of function evaluations) is of great interest. This led us to merge an academic local search method into a continuation method in aeronautical design cases, and more specifically into a space trajectory mission design.

1.2 The Problem

In aeronautical design, the large number of infeasible solutions makes it a challenging task to obtain the set of optimal solutions. In addition, the high cost in both time and computational power of each function evaluation further increase the complexity of the problem. As stated

before, multiobjective optimization can help us search towards and along the Pareto front.

Using a local search method described by Schütze et. al in [20], one can guide the search in objective space towards the Pareto front. Once the Pareto front is reached, we can apply the gradient-free version of the method to the problem and obtain a smooth approximation of the Pareto front, thus leaving the decision-makers the more useful choice given their current context. This method, compared to standard evolutionary algorithms and other continuation methods, cannot obtain a better approximation using the same budget of function evaluations.

Our aim is to obtain good approximations of the solution sets in academic problems and then attack aeronautical and space mission design problems, with a “small” budget of function evaluations using a gradient-free continuation method. Since so far, no gradient-free continuation method exists, we will demonstrate the usefulness of this method in multiobjective optimization problems.

1.3 General and Particular Aims of the Project

The aim of this project is,

- To develop a new gradient free continuation method for the treatment of multiobjective optimization problems to achieve a good approximation quality of the Pareto front using a low number of function evaluations, and to apply it to space mission design problems.

Particularly,

- To develop and analyze a new gradient free continuation method for unconstrained multiobjective optimization problems based on the Discrete Directed Search (DDS).
- To adapt this novel method for the handling of constrained optimization problems.
- To test the DDS Continuation method on academic problems.
- To apply the DDS continuation method on problems related to space mission design.

- To evaluate the results with the current state-of-the-art evolutionary algorithms and continuation methods.

1.4 Contributions

The main contribution of this work is a novel continuation method which eliminates one of the main issues of continuation methods: The use of gradient information. In this new method, we adapt the Directed Search method [21] into the gradient-free continuation method which uses a low number of function evaluations while yielding a good approximation of the Pareto front of a given problem. This method can also handle box, linear and non-linear constraints. In addition, this method can be applied to real-world problems, in particular, to engineering design optimization.

The secondary contributions of our work are as follows:

- An implementation of both the discrete directed search and the gradient free directed search continuation method.
- A good approximation of the Pareto front for the a real-world space trajectory problem.
- A software package for free academic use.
- A paper submitted to an international journal: The *IEEE Transactions on Evolutionary Computation*, and the participation in an international conference: The *EVOLVE 2013 - A Bridge Between Probability, Set-Oriented Numerics, and Evolutionary Computation*, held in Leiden, The Netherlands from July 11th to the 13th.

1.5 Organization of the Thesis

An outline of the thesis document is described as follows:

In **Chapter 2**, some basic notations for multiobjective optimization are presented, as well as related strategies for solving such problems. Also a brief overview of multiobjective

optimization in engineering is shown.

In **Chapter 3**, the proposed method is thoroughly explained: The novel gradient free Discrete Directed Search continuation method, which includes the predictor and corrector iterations, as well as the elements used for the step size control, and some initial results on academic functions.

Chapter 4 is used to describe the adaptation of the DDS Continuation method to constrained problems. Further, some results on constrained academic problems are shown.

In **Chapter 5** we present a three-impulse space trajectory problem, and the results of the gradient-free continuation method, as well as a comparison with standard evolutionary algorithms, and the result of another gradient-free continuation method.

Finally, in **Chapter 6**, some comments and a discussion are presented, which include possible applications in other areas of engineering and future work.

Background

Numerical optimization is the search for improvements of the result of a given problem. In single objective optimization, this translates into the minimization or maximization of the objective function value until we reach an optimal solution. However in multiobjective optimization there is a set of objective functions to be minimized or maximized concurrently. It is our desire to improve all objectives at once, but the functions are usually conflicting amongst each other. In the set of optimal solutions, we cannot always improve one objective without deteriorating another one. This “trade-off” generates an entire set of candidate solutions. In literature, the set of optimal solutions is called the *Pareto set*. The image of these solutions in the objective functions is called *Pareto front*.

To obtain the desired solutions, we can use evolutionary algorithms, which simulate nature’s way of improving organisms in an environment, or we can use other methods such as continuation algorithms, which iterate to better solutions using previously found good solutions.

In this chapter, a brief overview of state-of-the-art algorithms and basic definitions used in the field of multiobjective optimization are presented. The structure of this chapter is as follows. In Section 2.1, some common concepts are stated as to define the grounds for the work presented in this thesis. The concepts of multiobjective optimization problem and formal Pareto terminologies are noted, following the definitions of descent directions, and numerical continuation methods. Section 2.2 gives an overview of the variety of algorithms for the treatment of multiobjective optimization problems, and finally the algorithm used as the foundation for this work is detailed. We close the chapter with some comments in

Section 2.3.

2.1 Theoretical Background

A multiobjective optimization problem (MOP) can be defined in the following form:

$$\begin{aligned} & \text{minimize } F(x) \\ & \text{subject to } x \in S, \\ & g_i(x) \leq 0 \quad i = 1, \dots, l, \\ & h_j(x) = 0 \quad j = 1, \dots, m, \end{aligned} \tag{2.1}$$

where $F(x)$ represents a vector of at least $k \geq 2$ *objective functions* $f_i : S \rightarrow \mathbb{R}$. This map is defined as

$$F(x) = (f_1(x), f_2(x), \dots, f_k(x))^T. \tag{2.2}$$

The *decision vector* $x = (x_1, x_2, \dots, x_n)^T$ corresponds to the feasible region $S \subseteq \mathbb{R}^n$.

Objective functions represent the outcome of selecting a decision vector from the feasible region. For example, a farmer who wants to maximize the profit by planting corn and tomato in his field. The objective function represents the money earned and the decision vector the number of acres planted with each vegetable.

In almost all optimization problems, one is faced with several constraints imposed by the decision maker or by the environment itself. We can consider two types of constraints:

- Inequality constraints, noted by $g_i(x), i = 1, \dots, l$, represent flexible or soft restrictions on the design vector. For example, a producer of a certain product may have a limit in the amount of money spent to create this product. We know that we cannot spend more money because that will result in a violation of the constraint.
- Equality constraints, noted by $h_j(x), j = 1, \dots, m$, represent rigid or hard restrictions. For example, the producer may need to generate an exact number of units of a product

each day. Any less is unacceptable, and more than that will not be able to be sold or stored. Here, we must not violate the constraint at any point.

2.1.1 Basic concepts

In the following, we present some commonly used terms in multiobjective optimization. These concepts were first presented by Francis Edgeworth [22] and then established by Vilfredo Pareto [23]. We describe the notion of optimality in the context of multiobjective problems, along with the definitions of optimal solutions and other useful definitions.

Pareto Dominance

Before going further to detail about optimality, first we must be able to compare two solutions, in order to choose the “best one”. However, since the images $F(s)$ of solutions $s \in S$ are not scalar values, but rather vectors, and we cannot compare two or more vectors as we would with scalars, we need to establish a measure to determine which solutions are good. The next definition establishes when a vector is better than another one:

Definition 2.1.1. A vector $a = (a_1, \dots, a_k) \in \mathbb{R}^k$ is said to be less than another vector $b = (b_1, \dots, b_k)$ (denoted by $a \leq_p b$) if and only if a is partially less than b , i.e., for all $i = 1, \dots, k$, $a_i \leq b_i$ and $a_j < b_j$ for some $j \in 1, \dots, k$.

Pareto Optimality

Using the new concept of Pareto dominance we can establish optimal solutions. Roughly speaking, optimal solutions are the ones which are not dominated by any other solution in the entire set.

Definition 2.1.2. A point $y \in S$ is dominated by a point $x \in S$ ($x \prec y$) with respect to MOP (2.1) if x is partially less than y , i.e. if $f_i(x) \leq f_i(y)$, for all $i \in 1, \dots, k$, and $f_j(x) < f_j(y)$ for some $j \in 1, \dots, k$. Otherwise it is non-dominated by x .

This definition states that a vector $x \in S$ will be better than $y \in S$, if its function values are at least as good for every objective function, and better in one or more objective

functions. Put simply, it means that there is no other vector which can successfully decrease one objective without the increase of another one at the same time. Finally, we can establish a definition for the “best” solutions and a formal definition of the Pareto Set and the Pareto front.

Pareto Optimal Set

Pareto optimal solutions are those solutions which corresponding objective function values cannot be improved simultaneously. This group of solutions are non-dominated among each other, and dominate all other solutions, based on their function values. They do not need to have anything else in common other than the fact that they are non-dominated.

Definition 2.1.3. *The set of Pareto optimal points \mathcal{P} ,*

$$\mathcal{P} = \{x \in S \mid \nexists y \in S : y < x\}$$

is called a Pareto set.

Pareto Front

The image $F(\mathcal{P})$ of this set is called the Pareto front. There is typically no analytical model at hand to represent the Pareto front. An interesting property of the Pareto front is that it forms a $(k - 1)$ -dimensional manifold, where k is the number of objectives. This property will be exploited in this work. The formal definition of the Pareto front is given as follows:

Definition 2.1.4. *The image $F(\mathcal{P})$ of the Pareto set is called the Pareto front \mathcal{PF}*

2.1.2 Optimality conditions

In numerical optimization, there are several conditions that a solution must satisfy in order to be optimal. These conditions are concerned with the gradients of the objective functions, and hence are called *first-order conditions*. These conditions deal with equality constrained problems but if they are extended to inequality constraints, they are referred to as the Karush-Kuhn-Tucker conditions [24].

Definition 2.1.5. Suppose x^* is a local minimum of a MOP, that satisfies regularity conditions. There exist constants $\mu_i, i = 1, \dots, l$ and $\lambda_j, j = 1, \dots, m$, called KKT multipliers that fulfill the following conditions

$$-\nabla f(x^*) = \sum_{i=1}^l \mu_i \nabla g_i(x^*) + \sum_{j=1}^m \nabla h_j(x^*), \quad (2.3)$$

$$g_i(x^*) \leq 0, \text{ for all } i = 1, \dots, l, \quad (2.4)$$

$$h_j(x^*) = 0, \text{ for all } j = 1, \dots, m, \quad (2.5)$$

$$\mu_i \geq 0, \text{ for all } i = 1, \dots, l, \quad (2.6)$$

$$\mu_i g_i(x^*) = 0, \text{ for all } i = 1, \dots, l. \quad (2.7)$$

If there are no inequality constraints, the KKT conditions become the Lagrange conditions and the KKT multipliers, the Lagrange multipliers.

2.1.3 Descent directions

In a multiobjective optimization problem, a direction $v \in \mathbb{R}^n$ is called a descent direction for a point $x_0 \in S$ if:

$$\langle \nabla f_i(x_0), v \rangle < 0, \quad i = 1, \dots, k. \quad (2.8)$$

In other words, a descent direction is a vector which reduces every objective function value by some amount. Using this equation, we can state that for any sufficiently small step size $t \in \mathbb{R}^+$, it holds that $f_i(x_0 + tv) < f_i(x_0), i = 1, \dots, k$.

Using this construction, we can define a descent cone, which represents the union of all descent directions for each objective. In the case of multiobjective optimization, a descent cone is the intersection of the cones of each individual function. We can state this using the next equation:

$$D(x_0) = \{\nu \in \mathbb{R}^n \setminus \{0\} \mid \langle \nabla f_i(x_0), \nu \rangle < 0, \forall i = 1, \dots, k\}. \quad (2.9)$$

If the descent cone at a point x_0 is empty, that means that we are in an optimal (local) point. There are several descent directions that can be used to steer a search towards the Pareto front, such as the ones proposed by Fliege and Svaiter [25], Schäffler, Schultz and Weinzierl [26] and Lara et. al [27].

Final comments

We have discussed the basic concepts to determine optimal solutions in the treatment of multiobjective optimization problems. Also, we mentioned the definition for the descent directions, which play a fundamental role in evolutionary and line search strategies, since these directions can guide the search to optimal solutions. The concepts of the Pareto set and the Pareto front are commonly used throughout multiobjective optimization and it is the properties of the Pareto front which are exploited to perform movements along the front using the Directed Search Continuation method.

2.2 Related Work

Here we present alternative methods for solving multiobjective optimization problems. First the two major algorithms from which most techniques derived are presented: the weighted sum method, and the Tchebycheff method. Then, a brief introduction to line search strategies is noted, which includes a short description of the step size control. Finally, some state of the art evolutionary algorithms are described, along with their application in engineering design problems. Later on, we state the algorithms that will be used as a basis for this thesis, and an example of surrogate models which can be used to predict the outcome of the design problem to reduce its computational cost.

2.2.1 Scalarization strategies

Scalarization methods work by merging the different objective functions into a single objective, and then minimizing this new problem. The earliest attempts to solve multiobjective optimization problems were based on scalarization strategies. However, the drawback of

scalarization methods is that at the end, there is only one solution, and this solution reflects the preference the decision maker has regarding each objective function [28]. The three most important scalarization strategies in multiobjective optimization are: The weighted sum, the ϵ -constraint, and the Tchebycheff methods.

The idea behind the **weighted sum method**, proposed by Zadeh [11] in 1963, is to associate each objective function with a weighting coefficient, and to minimize the weighted sum of the objectives. The weighting coefficients w_i are real numbers such that $w_i \geq 0$ for all $i = 1, \dots, k$. It is also assumed that the weights are normalized, i.e., $\sum_{i=1}^k w_i = 1$. The weighted sum problem is defined as follows:

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^k w_i f_i(x) \\ & \text{subject to} && x \in S \end{aligned} \tag{2.10}$$

where $w_i \geq 0$ for all $i = 1, \dots, k$ and $\sum_{i=1}^k w_i = 1$. However, according to Koski [29], it is stated that this method fails to find Pareto points in non-convex regions, and an even distribution of weights in the objective functions does not guarantee an even distribution of points in the Pareto front.

The **ϵ -constraint method** was introduced in 1971 [12]. The idea behind this method is to select an objective function to be optimized, and all the other objective functions are then converted into constraints by setting an upper bound to each of them. Thus, the new problem to be solved is as follows:

$$\begin{aligned} & \text{minimize} && f_\ell(x) \\ & \text{subject to} && f_j(x) \leq \epsilon_j \quad j = 1, \dots, k, \quad j \neq \ell, \\ & && x \in S, \end{aligned} \tag{2.11}$$

where $\ell \in \{1, \dots, k\}$. However, to obtain the Pareto front, one has to solve Problem (2.11) with different values for the upper bounds. This is thus the main drawback of this method. Also, the given values for the bounds greatly impact the resulting solution. If the bound is

too tight, then no solution will be found. On the other hand, if the bound is too loose, no region will be restricted at all. In addition, the amount of increase on for the constraints has to be small enough to avoid missing optimal solutions, but large enough as to not waste function evaluations. Since we generally do not know the proper distribution and behavior of the Pareto set, this information is not known [30]. Finally the choice of the objective function to minimize is problem-dependent, and as such, it is not clear on how will the Pareto front be affected by the choice of the objective function.

Similar to the weighted sum approach, the **Tchebycheff method** [13] converts the problem into a scalar optimization problem as follows:

$$\begin{aligned} \text{minimize } g^{tc}(x|\lambda, z^*) &= \max_{1 \leq i \leq k} \{\lambda_i | f_i(x) - z_i^*\} \\ \text{subject to } x &\in S \end{aligned} \tag{2.12}$$

where $z^* = (z_1^*, \dots, z_k^*)^T$ is the reference point, this is, $z_i^* = \min_{1 \leq i \leq k} \{f_i(x) | x \in S\}$. Then for each optimal solution x^* there is an associated weight vector λ , such that x^* is an optimal solution of Equation (2.12), and thus a Pareto point of the multiobjective optimization problem. In other words, by changing the reference point, we can generate every non-dominated point of a general multiobjective optimization problem. The drawback in this method is that it generates weakly non-dominated points, which is not usually desired [31].

2.2.2 Line search strategies

In these kind of methods, given an initial point, a descent direction is selected. Then a step size is chosen and the new iterate performs a movement along this direction with a certain step size. Equation (2.13) represents the fundamental idea of the line search methods.

$$x_{i+1} = x_i + t_i v_i, \tag{2.13}$$

where $t_i \in \mathbb{R}^+$ is a scalar representing the step length and $v_i \in \mathbb{R}^n$ represents the desired direction. It is critical for any line search method to choose a “suitable” step size and a

direction which minimizes at least one of the objective functions. There are many step size control strategies, for example the Wolfe or Goldstein conditions [24]. In this work, we will concentrate on the directional derivative as a step size control for the predictor and corrector iterates.

The **Wolfe conditions** are twofold: one represents a sufficient decrease in a given objective, while the other discards very small step sizes, to avoid unnecessary computations and “creeping” along a given direction. The first Wolfe condition is given by the following inequality:

$$f(x_i + t_i v_i) \leq f(x_i) + c_1 t_i \nabla f^T v_i, \quad (2.14)$$

for a constant $c_1 \in (0, 1)$. Typically a value of 10^{-4} is chosen for this constant. Equation (2.14) is usually called *sufficient decrease condition*.

The second Wolfe condition is given by the following inequality:

$$\nabla f(x_i + t_i v_i)^T v_i \geq c_2 \nabla f^T v_i, \quad (2.15)$$

for a constant $c_2 \in (c_1, 1)$. In this condition, the slope of the new point must be greater than c_2 times the initial slope. This suggests that a strongly negative slope will reduce significantly the function value by moving along such direction.

2.2.3 Numerical continuation methods

Let us consider a nonlinear system of equations which we wish to solve, this is:

$$F(x) = 0 \quad (2.16)$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a smooth map, i.e., it is continuous derivative. If we know any good starting points, this is, a zero point of F , then it is possible to try and compute another zero point by means of, say, a Newton approach [32]. This is:

$$x_{i+1} = x_i - \mathcal{F}_i^{-1} F(x_i), \quad i = 0, 1, \dots \quad (2.17)$$

where \mathcal{F} is an approximation of the Jacobian. In certain cases, this process will fail since we don't know the properties of the curve we wish to trace, mixed with poor starting points. According to [32], we can define a *homotopy* $H : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ such that:

$$H(x, 1) = G(x), \quad H(x, 0) = F(x), \quad (2.18)$$

where $G(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is also a smooth map. We can use this homotopy to try and trace the curve $c(s) \subset H^{-1}(0)$ from the starting point to a solution point. If the method succeeds then we obtain a zero point of F . A global homotopy can be defined as:

$$H(x, \lambda) = (1 - \lambda)F(x), \quad \lambda \in [0, 1] \quad (2.19)$$

It is unclear, however, if the curve $c(s)$ can be parametrized with the parameter λ . Usually, λ is ill-suited for this task and thus, we have to select an additional parameter to trace it. A common idea is to take the arclength s of the curve, and state the new problem as follows:

$$H(c(s)) = 0 \quad (2.20)$$

Finally we can state this as an initial value problem by differentiating Equation (2.20) with respect to the arclength.

$$\begin{aligned} H'(c)\dot{c} &= 0, \\ \dot{c} &= \frac{dc}{ds}, \\ \|\dot{c}\| &= 1, \\ c(0) &= (x_1, 1). \end{aligned} \quad (2.21)$$

We can use Equations (2.20) and (2.21) together: First, we can integrate Equation (2.21) and use Equation (2.20) to stabilize the result. This is the basic principle of predictor-corrector (PC) methods. Other ideas use piecewise linear (PL) approximations to the homotopy maps. The advantage of PL methods is that they do not require that the smoothness holds for the homotopy, which gives them a bigger area of application than PC methods. However, the difficulty in adapting step size control strategies to PL methods, gives PC the advantage when they are applicable, even in higher dimensions.

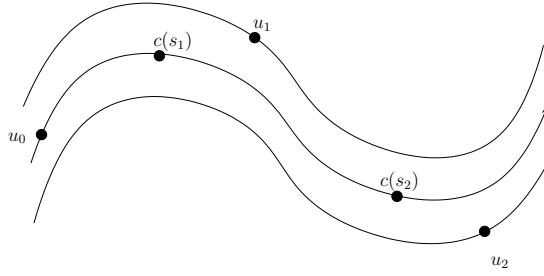


Figure 2.1: General predictor corrector method. The points u_i are the best approximated solutions to $c(s_i)$.

Predictor-Corrector (PC) methods

The idea of the PC methods is to generate a set of points $U = \{u_i = (x_i, \lambda_i) | x_i \in \mathbb{R}^n, \lambda_i \in \mathbb{R}\}$ that fit the curve we wish to trace, given by a tolerance, for example $\|H(u_i)\| < \epsilon$ for some “small” $\epsilon > 0$. We also state that an initial point $u_0 \in \mathbb{R}^n$ is given such that $H(u_0) = 0$. Figure 2.1 (borrowed from [32]) shows an example of this procedure.

First, a *predictor* point is computed, usually using numerical integration on the initial value problem of Equation (2.21). This predicted point v_{i+1} is used to compute the *corrector* point. This new point w_{i+1} is computed by solving the minimization problem described as follows:

$$\|w_{i+1} - v_{i+1}\| = \min_{H(w)=0} \|w - v_{i+1}\| \quad (2.22)$$

Figure 2.2 (borrowed from [32]) illustrates an iteration of a generic PC continuation method. Once the new point is calculated, it is used as a starting point for the next iteration. The PC continuation method applies the predictor and corrector steps iteratively to approximate the curve.

2.2.4 The Directed Search method

Here we describe the foundation of our novel algorithm, along with a rough construction of the gradient-free continuation method. The idea of the Directed Search proposed by Schütze et. al in [21], is to steer the search process into any direction in objective space. This is, given a point $x_0 \in \mathbb{R}^n$ in parameter space, and a vector $d \in \mathbb{R}^k$ in objective space, a direction

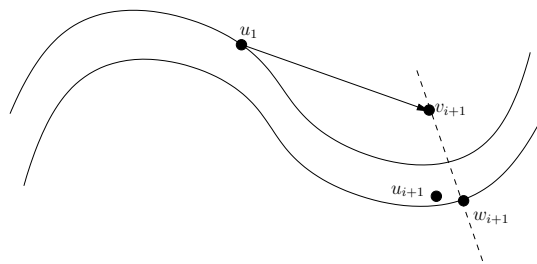


Figure 2.2: Predictor corrector iteration. The predicted point v_{i+1} is corrected to w_{i+1} , which is the closest to the zero point u_{i+1} .

vector $v \in \mathbb{R}^n$ is sought for such that:

$$\lim_{t \rightarrow 0} \frac{f_i(x_0 + tv) - f_i(x_0)}{t} = d_i, \quad i = 1, \dots, k, \quad (2.23)$$

where $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ denotes the i -th objective of the multiobjective problem. That is, the relative change of each objective value for an infinitesimal step size is given by the value of d_i . Using this construction, two methods are defined: A descent method, which finds “better” solutions leading to a boundary solution (as shown in Figure 2.3); and a continuation method which allows a movement along the Pareto set starting from a known solution.

2.2.5 DS: Descent method

The idea of the Directed Search Descent Method is to generate a curve of dominated points, with a proportion of an improvement defined by the direction α , in objective space. Using Equation (2.23), the DS method can be stated in a matrix vector notation as:

$$J(x_0)v = d, \quad (2.24)$$

where

$$J(x) = \begin{pmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_k(x)^T \end{pmatrix} \in \mathbb{R}^{k \times n}, \quad (2.25)$$

and $J(x)$ denotes the Jacobian of F at x_0 . Using this construction, a search direction v can be computed by solving this system of linear equations. Typically, the number of parameters is

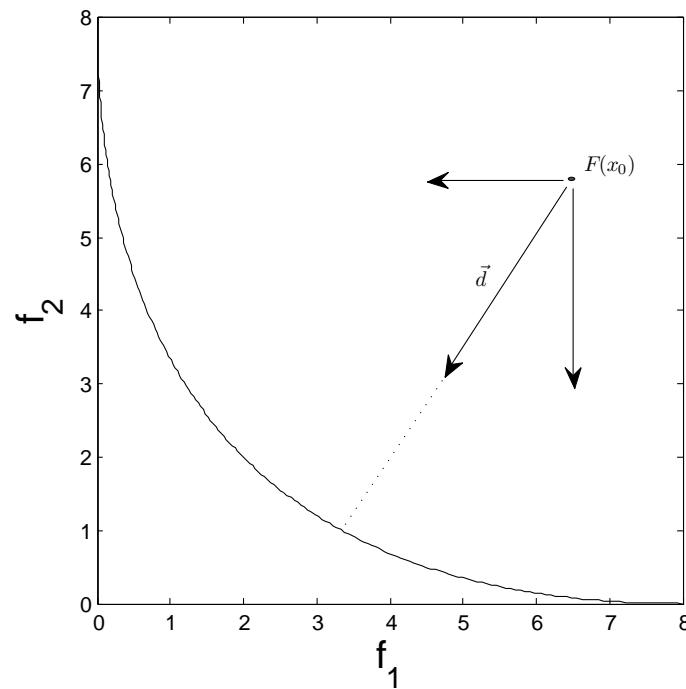


Figure 2.3: Basic idea of the Directed Search Descent Method

higher than that of the objectives, i.e., $n \gg k$, and thus the system (2.24) is underdetermined which means that there is no unique solution. One suggestion is to take

$$v_+ = J(x_0)^+ d, \quad (2.26)$$

where $J(x_0)^+ \in \mathbb{R}^{n \times k}$ denotes the pseudo inverse of $J(x_0)$ [24]. The argument for this choice is as follows: Given a candidate solution x_0 , a new solution is computed as

$$x_1 = x_0 + tv, \quad (2.27)$$

where $t \geq 0$ is a step size control and $v \in \mathbb{R}$ a vector that satisfies Equation (2.24). Of all the solutions of Equation (2.24), v_+ is the one with the smallest Euclidian norm. Hence, given a fixed (and small) value of t , a step in direction v_+ will yield the largest progress in direction d (measured in objective space). This idea is used on an initial value problem

$$\begin{aligned}x(0) &= x_0 \in \mathbb{R}^n, \\ \dot{x}(t) &= \nu_+(x(t)), t > 0.\end{aligned}\tag{2.28}$$

The drawback is that the computation of $J(x)$ requires gradient information, and in aeronautical design function evaluations are costly, both in time and resources. It is clear that a method that does not use gradient information would be of great use for this particular problem. Schütze et. al [21] go on to propose a variation of the continuation method of the Directed Search that is gradient free. The alternative presented uses a finite difference approach to solve Equation (2.24).

2.2.6 DS: Continuation method

Suppose we are given a Pareto point x_0 and its convex weight α , such that:

$$\sum_{i=1}^k \alpha_i \nabla f_i(x) = 0,\tag{2.29}$$

where $\alpha \geq 0, i = 1, \dots, k$, and $\sum_{i=1}^k \alpha_i = 1$. Also, we assume that

$$\text{rank}(J(x)) = k - 1.\tag{2.30}$$

In this case, α is orthogonal to the Pareto front [33]. This means that a search orthogonal to α could yield new predictor points. To use the directed search, one can perform a QR -decomposition of α . This is:

$$\alpha = QR,\tag{2.31}$$

where $Q = (q_1, \dots, q_k) \in \mathbb{R}^{k \times k}$ is an orthogonal matrix and each entry is a column vector, and $R = (r_{11}, 0, \dots, 0)^T \in \mathbb{R}^{k \times 1}$ and $r_{11} \in \mathbb{R} \setminus \{0\}$. By (2.31) $\alpha = r_{11}q_1$, it follows that the rest of the column vectors of Q form an orthonormal basis of the hyperplane that is orthogonal to α . Finally a well-spread set of search directions v_i may be the ones that satisfy:

$$J(x)v_i = q_i, \quad i = 2, \dots, k. \quad (2.32)$$

We know that α is not in the image of $J(x)$ and by (2.30), it follows that the vectors $q_i, i = 2, \dots, k$, are in the image of $J(x)$, and thus Equation (2.32) can be solved for each index $i = 2, \dots, k$. Note that this method does not compute the second derivate of the objective functions.

After the previous statement, a new predictor corrector variant is constructed. The authors of [21] state this method for bi-objective optimization. For higher dimensions ($k > 2$) additional considerations have to be taken into account. It is not in the scope of this project to extend to further dimensions, and thus, we will restrain ourselves to two objectives.

Given the initial Pareto point x_0 , we can choose a search direction as one of the normalized vectors $v = \pm v_2 / \|v_2\|_2$, where v_2 satisfies Equation (2.24) as stated before for $d = q_2$. To obtain the orientation, we can define the sign according to each objective, this is, the increase or decrease of one objective. To obtain the sign, we can use the respective entry of the vector q_2 . Finally, the predictor step given the direction v , is computed as follows:

$$p = x_0 - \text{sgn}(q_{i,2})tv, \quad i = 1, 2. \quad (2.33)$$

For the step size control we assume that each objective is Lipschitz continuous and that the step size t_i for the i -th objective is sufficiently small, we obtain:

$$|f_i(p) - f_i(x)| \approx L_{i,x} \|p - x\|_2. \quad (2.34)$$

After estimating the Lipschitz constant using the directional derivative, we obtain for each objective:

$$t_i = \frac{\epsilon}{w_i |\langle \nabla f_i(x), v \rangle|}, \quad i = 1, \dots, k. \quad (2.35)$$

and this yields the step size for the current iteration:

$$t = \min_{i=1, \dots, k} t_i. \quad (2.36)$$

Since the predictor step p leaves the Pareto front, it is important to return using a corrector step. The solution along the curve can be computed by using the directed search with initial point p , and the direction $d = \alpha_0$, this is the weight from the previous solution x_0 , leading to the new solution x_1 . Finally, the new weight α_1 can be computed as follows:

$$\begin{aligned} \alpha_1 \in \arg \min_{\alpha} \left\| \sum_{i=1}^k \alpha_{0,i} \nabla f_i(x) \right\|_2^2, \\ \text{s. t. } \alpha_{0,i} \geq 0, \quad i = 1, \dots, k \\ \sum_{i=1}^k \alpha_{0,i} = 1. \end{aligned} \tag{2.37}$$

The DS continuation method described in the previous section is used to move along the Pareto front, given an initial solution $x_0 \in S$ near, or on the Pareto set, but the drawback is that it requires gradient information. The goal here is to use the geometry of the Pareto front together with a predictor and corrector movement. This novel approach does not require the computation of the Hessians of the objectives. The challenge now is to eliminate the need of the Jacobian, and create a gradient free continuation method based on the Discrete Directed Search.

2.2.7 Discrete Directed Search: Descent method

Suppose we know a solution $x \in S$ and r search directions $v_i \in \mathbb{R}^n, i = 1, \dots, r$. Define the matrix $\mathcal{F}(x) \in \mathbb{R}^{k \times r}$ as follows:

$$\begin{aligned} \mathcal{F}(x) = (\langle \nabla f_i(x), v_j \rangle), \quad i = 1, \dots, k, \\ j = 1, \dots, r \end{aligned} \tag{2.38}$$

This means that every entry $m_{i,j}$ of \mathcal{F} is defined as the directional derivative of the objective f_i in direction v_j , $m_{i,j} = \nabla_{v_j} f_i(x)$. The following proposition is the basis for the gradient free directed search:

Proposition 2.2.1. *Let $x, v_i \in S, i = 1, \dots, r, \lambda \in \mathcal{R}^r$, and $v = \sum_{i=1}^r \lambda_i v_i$. Then:*

$$J(x)v = \mathcal{F}(x)\lambda. \quad (2.39)$$

The proof can be found in [21]. Furthermore, when we search for the direction v (i.e. solving Equation (2.24)) we can instead try to solve the following equation:

$$\mathcal{F}(x)\lambda = d, \quad (2.40)$$

and finally set:

$$v = \sum_{i=1}^r \lambda_i v_i. \quad (2.41)$$

Now we assume we are given a candidate solution $x_0 \in \mathbb{R}^n$ and further r points $x_i, i = 1, \dots, r$, in the neighborhood of x_0 and their function values $F(x_i), i = 1, \dots, r$. We now define:

$$v_j = \frac{x_j - x_0}{\|x_j - x_0\|_2}, \quad t_j = \|x_j - x_0\|_2, \quad j = 1, \dots, r. \quad (2.42)$$

Then all that is left is to approximate the entries of \mathcal{F} by finite differences as follows:

$$\begin{aligned} m_{i,j} &= \langle \nabla f_i(x), v_j \rangle = \lim_{t \rightarrow 0} \frac{f_i(x_0 + tv_j) - f_i(x_0)}{t} \\ &\approx \frac{f_i(x_j) - f_i(x_0)}{\|x_j - x_0\|_2} \quad i = 1, \dots, k, \quad j = 1, \dots, r. \end{aligned} \quad (2.43)$$

Analog to (2.26), we can obtain the greedy direction in the discrete setting as follows:

$$v_+^{(r)} = \sum_{i=1}^r \lambda_i v_i, \quad \text{where } \lambda = \mathcal{F}^+(x)d. \quad (2.44)$$

We can use the DDS to create a novel gradient free approach of the DS continuation method. Previous to this work, there was only one other continuation method that does not require gradient information, and is presented in the following section.

2.2.8 Hill Climber with Sidestep

The Hill Climber with Sidestep (HCS) was proposed by Schütze et al. in [18]. This algorithm allows a movement towards and along the Pareto set. This approach can be used with or without gradient information. In this section, we concentrate on the gradient free version of the HCS.

The gradient free HCS works by approximating the descent directions with neighborhood information. Given a point $x_0 \in S$, we obtain a new point \tilde{x}_1 at random from a neighborhood of x_0 , using a given radius $r \in \mathbb{R}_+^n$ which is problem dependent. If $\tilde{x}_1 \prec x_0$ then we set $\nu = \tilde{x}_1 - x_0$ as a descent direction. Then a line search is performed along that direction to obtain a better candidate solution.

If, on the other hand, $x_0 \prec \tilde{x}_1$, the same procedure is applied to the opposite direction, starting with \tilde{x}_1 . If we are far away from a local solution, then the probability is high that any of these two events will happen. If x_0 and \tilde{x}_1 are non-dominated among each other, we repeat the initial search of a neighbor, until we find a dominated solution (in which case we proceed as before) or until we reach N steps. If this happens, it means that we are probably near to an optimal solution, and thus a search along the solution set is desired.

We then need to perform a sidestep, using the accumulated knowledge obtained previously by unsuccessful trials. The proposed new direction is as follows:

$$\nu_a = \frac{1}{N} \sum_{i=1}^N s_i \frac{\tilde{x}_i - x_0}{\|\tilde{x}_i - x_0\|}, \quad (2.45)$$

where

$$s_i = \begin{cases} 1 & \text{if } f_1(\tilde{x}_i) < f_1(x_0) \\ -1 & \text{otherwise} \end{cases} \quad (2.46)$$

Note that Equation (2.47) is only useful for two objectives, where we can only move along two directions: minimizing f_1 or minimizing f_2 (this can be seen as moving “up” or “down” the Pareto front). This is due to the diversity cones created by the objectives. If there are k objectives, then there are a total of $2^k - 2$ diversity cones. So if we have three objectives, then we will have six cones (i.e. the search directions will be divided into six groups). This

means that it is less likely to find a perpendicular direction within N trials. Thus, we can use the accumulated information by averaging all search directions, and we get:

$$\nu_a = \frac{1}{N} \sum_{i=1}^N \frac{\tilde{x}_i - x_0}{\|\tilde{x}_i - x_0\|}. \quad (2.47)$$

2.2.9 Zigzag search method

Another recent and novel continuation method was created by Wang in 2012 [34]. This method works by performing a local search procedure around the Pareto front. By using the gradients of the objective functions, this method “zigzags” along the front, thus obtaining an approximated set of the Pareto optimal solutions.

The algorithm can be divided into two stages: First the “zig” step starts from an optimal solution and then performs a movement in the direction g , which is a projection of the gradient g_1 of f_1 , to the orthogonal hyperplane of the gradient g_2 of f_2 . This leads to a candidate solution near the Pareto front. Then the “zag” step is performed, using the projection of $-g_2$ to the orthogonal hyperplane of g_1 . This movement leads the candidate solution back to an optimal solution. Figure 2.4 shows the two steps of this algorithm.

The choice of these directions is described next. Given a small step size, the zig movement will find a candidate solution with a larger value of f_1 , while the value of f_2 will not change much. This is because a small step size is almost within the level set of f_2 at x_1 . If the movement is from a nonboundary of the Pareto set, then by the KKT equations, there exists a $c > 0$ such that $g_1(x_1) = -cg_2(x_1)$. For the zag movement, $-g_2$ is chosen since the candidate solution x is near the Pareto set, and this direction leads to a new optimal solution x_2 , with a fast decrease in f_2 and a small decrease in f_1 .

2.2.10 Evolutionary algorithms

Evolutionary algorithms (EAs) have their foundations in biology. Their principle idea is that individuals of a given group breed and mutate, giving birth to new individuals which in turn replace their parents [16]. An individual represents a solution to a problem, and is usually

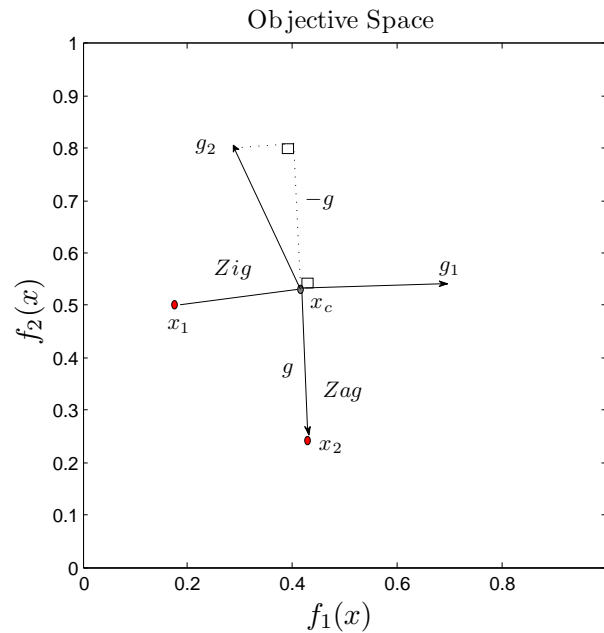


Figure 2.4: Zigzag method. The “zig” step starts from an optimal solution x_1 , moves in direction g_1 . Then the “zag” step moves to a new optimal solution x_2 using direction g , which is a projection of g_2 over g_1 .

represented by a string or vector corresponding to a biological *genotype*. This genotype is composed of *chromosomes* and each one is constructed by a series of *genes*. Using this composition, an individual is decoded into a set of parameters for the function at hand. The group of chromosomes is defined as a *population*.

When we consider an organism in nature, we know that the fittest in its environment will survive and pass on its genes. The same is true for the individuals in evolutionary algorithms. The operators in EA’s are used in the chromosomes, in order to generate new solutions with better *fitness*. Usually, three operators are used in EA’s: mutation, crossover, and selection, which work as follows:

- Mutation: The selected individual is mutated, for example, in the case of bitwise vectors, a “1” might be replaced by a “0”, and so on.
- Crossover: Just as in nature, two parents are selected from among the population and they breed to form a new solution. For example, one might combine the first segment

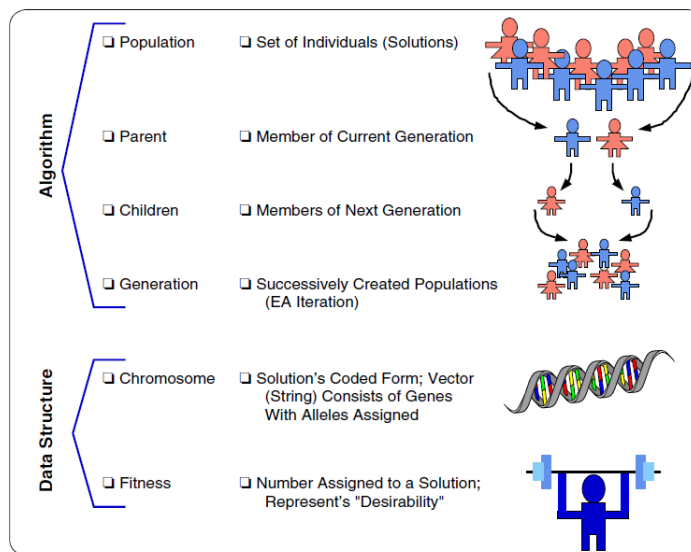


Figure 2.5: Basic concepts for an evolutionary algorithms.

of one parent with the second segment of the other.

- **Selection:** This is typically the survival of the fittest. The best individuals are chosen to go on the next generation. In this case, the individual is just copied over to the new population.

It is clear now that evolutionary algorithms must have a way to assign the fitness to each individual. In nature, fitness is defined by context, i.e., the fish which swims faster will be less likely to be eaten than a slower fish. In EA's the objective function define the optimality condition, and the fitness function is a measurement of how good does a particular solution satisfies such condition and assigns a real-value to that solution. Figure 2.5 (borrowed from [16]) shows the basic concepts of an evolutionary algorithm.

In 1994 Srinivas and Deb [35] proposed a new genetic algorithm using non-dominated sorting. This algorithm works similarly to other evolutionary algorithms, in the sense that the crossover and mutation operations remain the same. However, the selection is modified, and each solution is assigned a value based on domination. The solutions with the best values are grouped together into an initial front. Then, this procedure is repeated, generating a group of fronts using the remaining points. Finally, the reproduction is greater in the first

front since it constitutes the best solutions, and this leads the search towards the Pareto front.

Years later, the authors proposed a new method [14] called NSGA-II (Non-dominated Sorting Genetic Algorithm II). This algorithm eliminated some of the drawbacks of NSGA and constitutes one of the most widely used algorithms for multiobjective optimization. The fundamental difference is the crowding distance used to obtain a better spread of the solutions. The distance is computed by the two adjacent neighbors generating a rectangular area around a given solution. The bigger the area, the more spread between the solutions.

In 2007, Zhang et. al [15] developed a decomposition strategy for the treatment of multiobjective optimization problems. Usually, MOEAs treat the optimization problem as a whole, rather than a set of subproblems to optimize. The idea behind MOEA/D is to take the initial multiobjective optimization problem, and decompose it into a number of scalar optimization subproblems. Each problem is optimized simultaneously at each generation and the best solutions are kept. To optimize each subproblem, the information of the neighboring subproblems is taken into account.

2.3 Final Comments

In this section we have looked at the principles of numerical optimization for multiple objectives. The geometry of the Pareto front facilitates the use of the directed search to move along it, since a linearization of the front gives us a good starting direction for the local search. The concepts of continuation methods, more specifically, the predictor corrector methods let us construct our novel approach and use it to solve real world problems.

Also, we have discussed the basic principles of continuation methods: The PC method which approximates the curve by generating candidate solutions and then minimizing the error to the solution curve; and the PL methods which approach the curve by a triangulation of the space and then generating a polygonal path across such triangulation by means of its coordinates. Further in this work, we will only consider PC methods, since the proposed algorithm fits on the first type of continuation methods. Finally we mentioned two con-

tinuation methods that work in principle as the DDS continuation method. First, the DS continuation which uses gradient information, and second, the Hill Climber with Sidestep, that omits such requirements but fails to generate a good approximation of the Pareto set under a small number of function evaluations.

The two major state-of-the-art evolutionary algorithms described briefly here will be used as a comparison in both the academic functions and aeronautical problem. Because we are interested in a fair comparison between different methods, we will only compare the new gradient free continuation method with the Hill Climber with Sidestep, since it is the only other continuation method known up to the current date that does not use gradient information.

Most engineering problems have constraints, be them equality, inequality or a mixture of both types of constraints. This leads us to adapt our continuation method to the constraint cases, to fully appreciate the strength of this new algorithm. Finally we state that there is no one algorithm that is better than all others, but a wide variety of techniques to solve the same problem.

Gradient Free Continuation Method for Unconstrained Problems

In many situations in multiobjective engineering design, we have an initial Pareto optimal or nearly Pareto optimal solution which may give us a satisfying result. However, the decision maker may want to see other options along the Pareto front which are other potentially interesting solutions. In this case, we can take advantage of the starting element to generate a new set of desired solution, since the solution set forms, at least locally, a manifold. This is the basic principle of multiobjective continuation methods, such as the ones presented in [32] and [33]. The main drawback of them is that they require gradient information and usually even Hessian information.

In Section 2, we have discussed the main idea behind the Directed Search method, which given a direction in objective space, finds its corresponding direction in parameter space such that a movement in objective space is performed. The Directed Search uses gradient information in order to compute such directions, thus, the Discrete Directed Search was introduced as a possible alternative. This method applies a finite differences approach to the conventional DS in order to remove the requirement of the gradient information. It uses a group of nearby points of each solution, to approximate the search direction. Thus, the method can be used to save function evaluations.

The new Discrete Directed Search proved to be a valuable tool, both as a standalone algorithm to find boundary points in multiobjective optimization problems, as well as integrated in evolutionary algorithms to speed up convergence and spread [20]. Another advantage

of this method is that, given n neighbors, where n is the number of parameters, the DDS effectively transforms into the DS method.

In Mejía's work [36], several approaches are stated in order to reduce the time and cost to obtain the curve of dominated points until a boundary point is reached, along with a method to solve constrained MOPs. Finally, he proposes a continuation method which uses the DS to move along the optimal set to cover the entire Pareto front. In Alvarado's work [37], the DS is modified as to remove the gradient information to create the DDS. In this work, we propose an extended use of the DDS, inside the continuation method to remove the gradient requirement and to take advantage of the neighboring points at each iteration. Also, we propose a different step size control which results in a faster convergence to the boundary point for the DDS.

The remainder of the chapter is organized as follows: In Section 3.1 we present the definition of line search strategies. In Section 3.2 we present the different step size control strategies. In Section 3.3, we propose the gradient free continuation method for unconstrained MOPs, and finally, in Section 3.4 we present numerical results for several academic examples along with a comparison with MOEA/D and the HCS.

3.1 Gradient free line search

Line search strategies have been widely used either as standalone algorithms to obtain a solution of a given MOP, or inside evolutionary algorithms to build memetic strategies [38, 39]. To solve multiobjective optimization problems, several iterative methods compute a set of solutions in sequence, where each new solution $x_{i+1} \in \mathbb{R}^n$ is obtained by the previous solution $x_i \in \mathbb{R}^n$, using a direction $\nu \in \mathbb{R}^n$ and a step size $t \in \mathbb{R}^+$. We can state this iterative method as an initial value problem (IVP)

$$\begin{aligned}x(0) &= x_0 \\ \dot{x}(t) &= \nu(x(t)).\end{aligned}\tag{3.1}$$

To obtain a curve of dominated points from a random starting point to a boundary point,

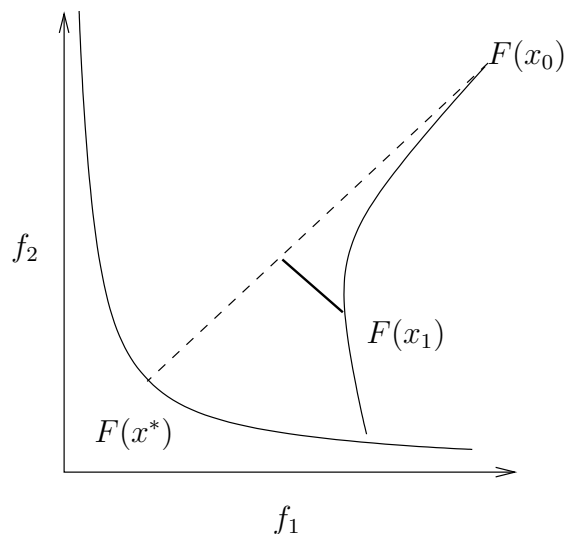


Figure 3.1: Directed Search Method. Due to the local nature of the gradient information, the predictor step usually steers away from the desired curve.

we can use the initial value problem (3.1) to compute the solution. The Directed Search can be stated as this IVP, and in Algorithm 1, we present the steps required to compute the curve of dominated points.

We can see that the end point results in a straight line starting from $F(x_0)$ with direction α . Normally, the gradient information is of local nature, and thus it is very likely that after a few steps, the method steers away from the desired direction, and lose the expected resulting curve, as seen on Figure 3.1. There are two options in this setting: One is to take very small step sizes to avoid losing gradient information, but this translates into an unnecessary amount of function evaluations. The alternative is to use large step sizes in an attempt to reach the end point as fast as possible, but using a corrector step, to steer us in the original desired direction. Since the purpose of this work is to reduce the number of function evaluations needed to find optimal solutions, we will demonstrate several corrector techniques to integrate within the DS and DDS.

The algorithm stops when the condition number of the Jacobian approaches a large number. In this algorithm, however, we do not have a corrector step to shoot back to the original curve. In the following, we present a basic algorithm which, given a predictor point

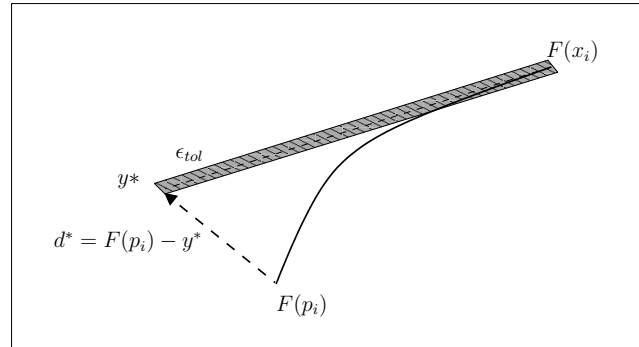


Figure 3.2: Corrector step in the Directed Search Descent Method. When we move away from the desired direction, we can shoot back using a corrector step.

Algorithm 1 Basic Directed Search Method

Require: starting point $x_0 \in \mathbb{R}^n$ with $\text{rank}(J(x_0)) = k$, $tol \in \mathbb{R}^+$, convex weight $\alpha_0 \in \mathbb{R}^k$

Ensure: $x^* \in \mathbb{R}^n$, a boundary point of an MOP

- 1: $i := 0$;
 - 2: **while** $\kappa_2(J(x_i)) < tol$ **do**
 - 3: compute $v_i := -J(x_i)^+ \alpha_{(i)}$;
 - 4: compute $t_i \in \mathbb{R}^+$ (see Section 3.2, in page 38)
 - 5: $x_{i+1} := x_i + t_i v_i$;
 - 6: choose $\alpha_{(i+1)} \in \mathbb{R}^k$;
 - 7: $i := i + 1$;
 - 8: $x^* := x_i$;
 - 9: **return** x^*
-

computed by Algorithm 1, we can correct back to the desired direction d . Figure 3.2 shows an example of this method. Algorithm 2 shows the steps required to compute the corrector step, as shown in Mejía's work [36].

Algorithm 2 Corrector step for the Directed Search Method

Require: starting point $x_0 \in \mathbb{R}^n$, predicted point computed by Algorithm 1, $p_0 \in \mathbb{R}^n$, $tol \in \mathbb{R}^+$, convex weight $\alpha_0 \in \mathbb{R}^k$

Ensure: $x_c^* \in \mathbb{R}^n$, a point in the desired direction d

- 1: $i := 0$;
 - 2: compute $t^* := \langle F(p_0) - F(x_0), -\alpha_0 \rangle$;
 - 3: compute $y^* := F(x_0) - t^* \alpha_0$;
 - 4: **while** $\|F(p_i) - y^*\|_2 > tol$ **do**
 - 5: compute $\alpha_i = F(p_i) - y^*$
 - 6: compute $v_i := -J(p_i)^+ \alpha_i$;
 - 7: $p_{i+1} := p_i + v_i$;
 - 8: $i := i + 1$;
 - 9: $x_c^* := x_i$;
 - 10: **return** x_c^*
-

We can see that the computation of new points does not require a step size control, (i.e. we set $t = 1$). This is because the method is quadratically convergence. With Algorithm 2, we can obtain a perfect curve of dominated points for the Directed Search Method.

Two things are noted in these algorithms: The first one is the use of gradient information. We already know that the Directed Search can be made gradient free using neighboring points to approximate the gradient. Algorithm 3 shows the Discrete Directed search, as stated in Alvarado's work [37].

The approximation of the gradient is by means of neighboring points, but further reductions in the function evaluations can be made. For example, one could take some previous points into account for the new Jacobian approximation. This technique is explored in the gradient free continuation method. It is noted also that there will be a similar steering away from the curve if no corrector step is performed. Algorithm 2 showed the corrector step, but

Algorithm 3 Basic Discrete Directed Search Method

Require: starting point $x_0 \in \mathbb{R}^n$ with $\text{rank}(\mathcal{F}(x_0)) = k$, $tol \in \mathbb{R}^+$, convex weight $\alpha_0 \in \mathbb{R}^k$ **Ensure:** $x^* \in \mathbb{R}^n$, a boundary point of an MOP

```

1:  $i := 0$ ;
2: while  $\kappa_2(\mathcal{F}(x_i)) < tol$  do
3:   compute neighbors  $x_i^r, i = 1, \dots, r$ ;
4:   compute  $\mathcal{F}(x_i)$ ;
5:   compute  $\lambda := -\mathcal{F}(x_i)^+ \alpha_i$ ;
6:   compute  $v_i := \sum_{i=1}^r \lambda_i v_i^r$ ;
7:   compute  $t_i \in \mathbb{R}^n$  (see Section 3.2)
8:    $x_{i+1} := x_i + t_i v_i$ ;
9:   choose  $\alpha_{(i+1)} \in \mathbb{R}^k$ ;
10:   $i := i + 1$ ;
11:  $x^* := x_i$ ;
12: return  $x^*$ 

```

required gradient information to be of use. We now remove such requirement and produce a gradient free corrector step. Algorithm 4 shows this new method.

We can see that the only difference is how the Jacobian is replaced by its approximation. Nevertheless, further reductions of the cost for the corrector can be made. For example, since we assume that the steering away from the curve is a relatively small distance, we can take the previous Jacobian matrix almost entirely (with the update of a few neighboring points). This technique is not explored in this work, but is left for future research.

3.2 Step size control strategies

In the following, we discuss potential step size control strategies for minimizing the cost of the DS and the DDS. First the Armijo conditions are presented for single objective optimization. Then, they are extended to the context of multiobjective optimization problems. The next subsection describes the curvature condition along with the cones of tolerance to create an

Algorithm 4 Corrector step for the Directed Search Method

Require: starting point $x_0 \in \mathbb{R}^n$, predicted point computed by Algorithm 1, $p_0 \in \mathbb{R}^n$, $tol \in$

\mathbb{R}^+ , convex weight $\alpha_0 \in \mathbb{R}^k$

Ensure: $x_c^* \in \mathbb{R}^n$, a point in the desired direction d

- 1: $i := 0$;
 - 2: compute $t^* := \langle F(p_0) - F(x_0), -\alpha_0 \rangle$;
 - 3: compute $y^* := F(x_0) - t^* \alpha_0$;
 - 4: **while** $\|F(p_i) - y^*\|_2 > tol$ **do**
 - 5: compute neighbors $x_i^r, i = 1, \dots, r$;
 - 6: compute $\mathcal{F}(x_i)$;
 - 7: compute $\alpha_i = F(p_i) - y^*$
 - 8: compute $\lambda := -\mathcal{F}(x_i)^+ \alpha_i$;
 - 9: compute $v_i := \sum_{i=1}^r \lambda_i v_i^r$;
 - 10: $p_{i+1} := p_i + v_i$;
 - 11: $i := i + 1$;
 - 12: $x_c^* := x_i$;
 - 13: **return** x_c^*
-

adaptive control strategy. Finally, a step size control based on dominance is proposed. We note that some of these control strategies can be combined to further improve the movement in the desired curve.

3.2.1 Wolfe step size control

The Armijo condition, together with the Goldstein condition form the Wolfe conditions [24]. The Armijo condition is popular in line search strategies, and stipulates that the resulting step size gives a sufficient decrease in the objective function f , determined by the inequality

$$f(x_i + t\nu_i) \leq f(x_i) + c_1 t \nabla f_i^T \nu_i, \quad c_1 \in (0, 1) \quad (3.2)$$

where c_1 is a fixed constant. Equation (3.2) states that for the step size t , the graph of $F(t) = f(x_i + t\nu_i)$ lies below the line $f(x_i) + c_1 t \nabla f_i^T \nu_i$. By Taylor's theorem we get

$$f(x_i + t\nu_i) = f(x_i) + t \nabla f_i^T \nu_i + O(t^2). \quad (3.3)$$

Since ν_i is a descent direction, this is, $\nabla f_i^T \nu_i < 0$, such t exists.

This condition is not sufficient to ensure convergence since it is satisfied for all small enough values of t . To discard unacceptably small step sizes, the second Wolfe condition, or curvature condition, is used

$$\nabla f(x_i + t\nu_i)^T \nu_i \geq c_2 \nabla f_i^T \nu_i, \quad c_2 \in (c_1, 1) \quad (3.4)$$

where c_2 is a fixed constant. This condition states that t must be large enough so that the slope of $f(t)$ is larger than c_2 times the slope of $f(0)$. Mejía [36] adapted the first Wolfe condition for multiobjective optimization problems, and Algorithm 5 shows this resulting strategy.

3.2.2 Curvature-based step size control

Now we present a step size control based on curvature, described thoroughly in Mejía's work [36]. However, this method requires information of two previous steps, in order to

Algorithm 5 Armijo step size control

Require: starting point $x_0 \in \mathbb{R}^n$, direction $v \in \mathbb{R}^n$, initial step size $t \in \mathbb{R}^+$, $\lambda_k \in (0, 1)$, convex weight $\alpha_0 \in \mathbb{R}^k$

Ensure: Suitable step size $t \in \mathbb{R}^+$

- 1: **while** $F(x_0 + tv)^T \alpha_0 > F(x)^T \alpha_0 - c_1 t \sum_{i=1}^k \alpha_i^2$ **do**
 - 2: $t := \lambda_k t$
 - 3: **return** t
-

approximate the curve in parameter space. For these two steps, one can use any of the other algorithms presented in this section. Figure 3.3 (borrowed from [36]) shows a graphical example of the curvature-based control.

Algorithm 6 Curvature step size control

Require: Previous iterations' step sizes $t_{i-1}, t_{i-2} \in \mathbb{R}^+$, previous iterations' Jacobian approximations $\mathcal{F}(x_{i-1}), \mathcal{F}(x_{i-2}), \epsilon \in \mathbb{R}^+$

Ensure: Suitable step size $t \in \mathbb{R}^+$

- 1: $\gamma_i := \frac{\mathcal{F}(x_{i-1}) - \mathcal{F}(x_{i-2})}{t_{i-1} - t_{i-2}}$
 - 2: $r_k := \|\gamma_i\|^{-1}$
 - 3: $t := \sqrt{2r_k t \epsilon - \epsilon^2}$
 - 4: **return** t
-

In case the gradient information is available, one can substitute $\mathcal{F}(x_{i-1})$ and $\mathcal{F}(x_{i-2})$ for the Jacobian of x_{i-1} and x_{i-2} respectively.

3.2.3 Cones of tolerance

Usually, when we are near the Pareto front, the step size must be very small to avoid overshooting. The resulting point can be very far from the original point, and in some cases it will be dominated by previous points. Mejía also proposed a step size control based on the angles between the starting point x_i and the predictor point p_i . This step size control strategy works as follows: First, a step is taken using an initial step size (which can be obtained by any other method, or as a fixed initial value). Then the angle between $F(p_i) - F(x_i)$ and

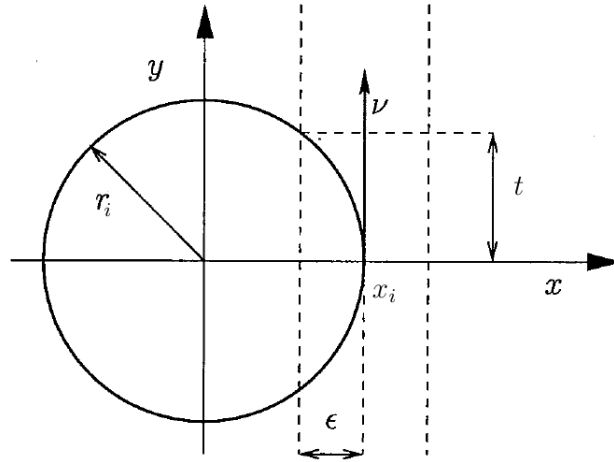


Figure 3.3: Graphical interpretation of the curvature-based control. If the curve of the Pareto set is fixed, the step size can be computed by trigonometric equations.

$F(x_i) + d$ is computed. If this angle is greater than a given tolerance β , a reduction in the step size is performed. Figure 3.4 shows the graphical interpretation of the cones of tolerance, while Algorithm 7 shows the implementation of this procedure.

Algorithm 7 Cones of tolerance step size control

Require: starting point $x_0 \in \mathbb{R}^n$, direction $v \in \mathbb{R}^n$, initial step size $t \in \mathbb{R}^+$, $\lambda_k \in (0, 1)$, convex weight $\alpha_0 \in \mathbb{R}^k$, angle of tolerance $\beta \in (0, 90]$

Ensure: Suitable step size $t \in \mathbb{R}^+$

- 1: **repeat**
 - 2: $y := x_0 + tv$;
 - 3: **if** $\angle(F(x_0 - \alpha_0), F(x_0) - F(y)) > \beta$ **then**
 - 4: $t := \lambda_k t$;
 - 5: **until** $\angle(F(x_0 - \alpha_0), F(x_0) - F(y)) \leq \beta$
 - 6: **return** t
-

It has to be noted that the cone of tolerance will only work if the corrector step is applied inside the DS or DDS. Else, the steering from the original direction may cause the cone to accept dominated solutions which will in turn worsen the result of the local search.

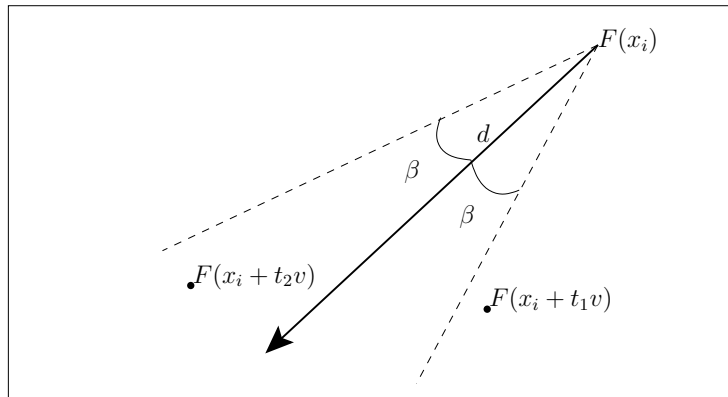


Figure 3.4: Graphical interpretation of the cones of tolerance control. Here we see that the step size t_1 is not acceptable since it violates the tolerance β .

3.2.4 Pareto dominance control

Finally, we propose a relatively simple step size control based on dominance. This is, if the new point is dominated by the original point, then a backtracking step is performed. This type of step size control works similarly to the cones of dominance, i. e., setting $\beta = 45^\circ$. Figure 3.5 shows a graphical representation of the Pareto dominance for step size control.

Algorithm 8 Dominance step size control

Require: starting point $x_0 \in \mathbb{R}^n$, direction $v \in \mathbb{R}^n$, initial step size $t \in \mathbb{R}^+$, $\lambda_k \in (0, 1)$,

Ensure: Suitable step size $t \in \mathbb{R}^+$

- 1: **while** $F(x_0 + tv) \succ F(x_0)$ **do**
 - 2: $t := \lambda_k t$;
 - 3: **return** t
-

3.2.5 Final comments

We have presented and discussed several step size control strategies to improve the DDS in the search for optimal solutions. It has to be noted that the gradient free continuation cannot use any of these strategies in the predictor step since the direction may lead to a point not necessarily in a descent direction, which breaks all the restrictions placed on these

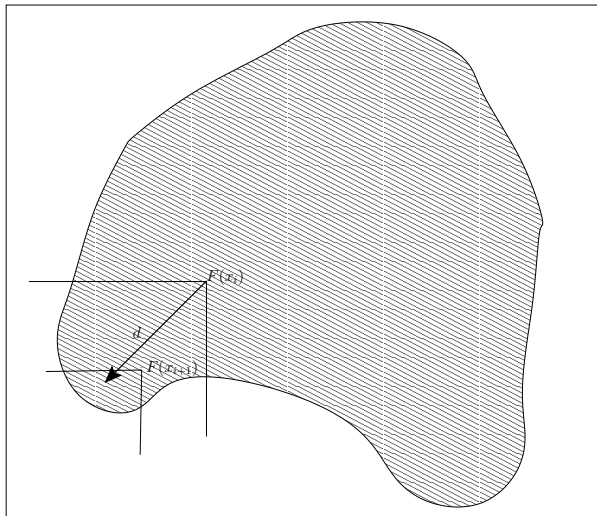


Figure 3.5: Graphical representation of the Pareto dominance control. The new step size must produce a value inside the descent cone of the set of objective functions.

methods. Thus, for the step size control in the predictor a new strategy based on directional derivatives is proposed in the following section.

3.3 Gradient free continuation method

In the following, we propose the gradient free predictor corrector method that allows us to perform a movement from a given solution x_0 along the Pareto set. The method is based on the DDS described previously, along with another observation. This is that a local search along x_0 leads with probability one to a movement from $F(x_0)$ along the front, since the dimension of the kernel of $J(x)$ is $n - k + 1$ (this is, $J(x)\nu = 0$ will happen with probability zero for a randomly chosen ν). To adapt the continuation method, we need to address several tasks that have to be performed for the gradient free version which will be presented in this section.

3.3.1 Steering the search

Crucial to the continuation method are the choices for the search directions for both the predictor and the corrector steps. In this subsection we present the possible choices for each

step and their comparison with the gradient continuation counterpart.

Predictor

In the classical DS method, a set of predictors is generated by computing a QR -factorization of the KKT weight α of the given point x_0 . These vectors form an orthonormal basis of the tangent space of the Pareto front at $F(x_0)$. The KKT weight is computed by solving

$$\begin{aligned} \alpha_1 \in \arg \min_{\alpha} \left\| \sum_{i=1}^k \alpha_{0,i} \nabla f_i(x) \right\|_2^2, \\ \text{s. t. } \alpha_{0,i} \geq 0, \quad i = 1, \dots, k \\ \sum_{i=1}^k \alpha_{0,i} = 1. \end{aligned} \tag{3.5}$$

This approach can apparently not be chosen since it requires gradient information. We can instead use the observation made in [40], namely that $J(x_0)\nu$ points along the Pareto front for almost all directions $\nu \in S$, as well as the following observation to obtain a gradient free way of an orthonormal basis of the Pareto front as well as the KKT weight. Let $\nu_i \in S$, $i = 1, \dots, k - 1$ such that $J(x_0)\nu_1, \dots, J(x_0)\nu_{k-1}$ are linearly independent. Now, we look at the QR -factorization

$$J(x_0)\nu_1, \dots, J(x_0)\nu_{k-1} = QR = (q_1, \dots, q_k)R. \tag{3.6}$$

Then we have that the set $\{q_1, \dots, q_{k-1}\}$ is the desired orthonormal basis. Then we can select $d_p = q_i, i \in \{1, \dots, k - 1\}$, and either $q_k <_p 0$ or $q_k >_p 0$, it is

$$\alpha = \text{sgn}(q_{k,1}) \frac{q_k}{\|q_k\|_1} \tag{3.7}$$

Thus, if we choose test points x_i in the neighborhood $N(x_0)$ of x_0 and we set $v_i = x_i - x_0$ one obtains a gradient free way to obtain both tangent vectors and KKT weight. The cost for this is ideally given by $k - 1$ evaluations of F , but further evaluations may occur if $J(x_0)\nu_1, \dots, J(x_0)\nu_{k-1}$ are not linearly independent, but this probability will be zero if each neighbor is chosen at random.

Since we are limiting the use of the continuation method for two objectives, we can, in addition, assume an improvement according to f_1 for example (this is, a left-up-movement along the Pareto front). We can then compute $x_1 \in N(x_0)$ and then evaluate $F(x_1)$. The desired direction given in image space is

$$d_p = \tilde{d} = F(x_1) - F(x_0), \quad (3.8)$$

in the case that $f_1(x_1) < f_1(x_0)$, otherwise we set $d_p = -\tilde{d}$.

Corrector

Since the KKT weight α of the previous Pareto point is known, we can proceed as for the gradient version of the continuation method. This is, we set $d_c = -\alpha$. Again, for two objectives, we can express this analytically: If $d_p = (a, b)^T$, then the corrector direction is given by

$$d_c = -\text{sgn}(a) \begin{pmatrix} -b \\ a \end{pmatrix}. \quad (3.9)$$

3.3.2 Computation of $\mathcal{F}(x)$

By now, it is clear that a good approximation of the Jacobian is critical for the proper behavior of the DDS. For the approximation of $\mathcal{F}(x)$, the test points chosen are equally important. We can consider points in a neighborhood for which we already know their function values, and include them in the construction of this matrix. Although, usually, additional test points will have to be sampled to improve the search direction.

Suppose we are given an initial point $x_0 \in \mathbb{R}^n$ as well as l neighboring solutions $x_1, \dots, x_l \in N(x_0)$. By the previous discussion, we desire that further search directions are both orthogonal to each other and to the previous ones. To compute these directions ν_{l+1}, \dots, ν_r , where $r > l$ while maintaining orthogonality, we can proceed as follows. First we compute a QR -factorization of the directions, i.e, of V , and get

$$V = QR = (q_1, \dots, q_l, q_{l+1}, \dots, q_n)R, \quad (3.10)$$

where $Q \in \mathbb{R}^{n \times n}$ is an orthogonal matrix and $R \in \mathbb{R}^{n \times r}$ is a right upper triangular matrix with nonzero diagonal elements. This is, $r_{i,i} \neq 0$ for all $i = 1, \dots, r$. Then we know by construction that $\nu_i \in \text{span}\{q_1, \dots, q_i\}$ for $i = 1, \dots, l$, and then

$$\langle \nu_i, q_j \rangle = 0, \forall i \in \{1, \dots, l\}, j \in \{l+1, \dots, r\}. \quad (3.11)$$

Now we have

$$\begin{aligned} \nu_{l+i} &= q_{l+i}, \quad i = 1, \dots, r-l \\ x_{l+i} &= x_0 + \nu_{l+i}, \quad i = 1, \dots, r-l \end{aligned} \quad (3.12)$$

For the special case that $\nu_1 = x_1 - x_0$ and $\tilde{\nu}_2 = \tilde{x}_2 - x_0$ are given such that $\{\nu_1, \nu_2\}$ are linearly independent, the second search vector ν_2 can be computed as follows

$$\nu_2 = \tilde{\nu}_2 - \langle \nu_1, \tilde{\nu}_2 \rangle \nu_1. \quad (3.13)$$

3.3.3 Step size control

In the following subsection, we propose strategies to compute suitable step sizes for the predictor as well as for the corrector.

Predictor

For the gradient based continuation, the step size control is given by [36]:

$$t_i = \frac{\epsilon}{\|v\|_2 |\langle \nabla f_i(x), v \rangle|}, \quad i = 1, \dots, k. \quad (3.14)$$

And then setting $t = \min t_i$.

To eliminate the gradient we can proceed as follows: Let $x_1 \in N(x_0)$, then we set $\nu = x_1 - x_0$ and we obtain $\|\nu\|_2 = \|x_1 - x_0\|_2$, then we get

$$\langle \nabla f_i(x_0), \nu \rangle \approx \frac{|f_i(x_1) - f_i(x_0)|}{\|x_1 - x_0\|_2}. \quad (3.15)$$

Then we can estimate t_i as follows

$$\tilde{t}_i = \frac{\epsilon}{|f_i(x_1) - f_i(x_0)|}, \quad i = 1, \dots, k. \quad (3.16)$$

And then setting $\tilde{t} = \min \tilde{t}_i$.

One potential drawback of this approximation is the fact that if $f_i(x_1)$ is too close to $f_i(x_0)$, then the step size will get very large. It is recommended to bound the overall step size by a maximum value t_{max} . Also if we use Equation (2.44) to compute the search direction ν , the above considerations for the step size control cannot be used since for ν , there does not exist a function value in that direction (since it is only known that $\nu \in \text{span}\{\nu_1, \dots, \nu_r\}$). We can instead use Equation (3.17) with the estimation of the Lipschitz constant using the neighboring information. This is, given x_0 and directions $v_i^r, i = 1, \dots, r$, and $v = V\mathcal{F}(x_0)^+d$, the directional derivative of the j -th objective in direction v is as follows:

$$\langle \nabla f_j(x), v \rangle = \langle \nabla f_j(x), \sum_{i=1}^r \lambda_i v_i \rangle = \sum_{i=1}^r \lambda_i \langle \nabla f_j(x), v_i \rangle = \sum_{i=1}^r \lambda_i m_{j,i} \quad (3.17)$$

where the $m_{j,i}$'s are the entries of \mathcal{F} . Algorithm 9 details this step size control.

Algorithm 9 Directional derivative step size control (gradient free)

Require: starting point $x_0 \in \mathbb{R}^n$, $\epsilon \in \mathbb{R}^+$, convex weight $\alpha_0 \in \mathbb{R}^k$

Ensure: Suitable step size $t \in \mathbb{R}^+$

- 1: compute neighbors $x_i^r, i = 1, \dots, r$;
 - 2: compute $\mathcal{F}(x_i)$;
 - 3: compute $\lambda := -\mathcal{F}(x_i)^+ \alpha_i$;
 - 4: compute $v := \sum_{i=1}^r \lambda_i v_i^r$;
 - 5: compute $D := \sum_{j=1}^n \sum_{i=1}^r \lambda_i m_{j,i}$;
 - 6: compute $t_i := \frac{\epsilon}{\|v\|_2 |D_i|}, i = 1, \dots, n$;
 - 7: $t = \min t_i$
 - 8: **return** t
-

Corrector

To compute suitable step sizes for the corrector, refer to Section 3.2, in page 38 of this chapter.

3.3.4 Predictor corrector iteration

In this section we present the algorithm for the continuation method that does not require gradient information. In combination with the DDS and the step size control strategies presented above, a powerful and novel tool is constructed. Further work to reduce the number of function evaluations is presented also in this section.

The algorithm works as follows. Starting from a known solution x_i , the next (approximate) solution is obtained by selecting a predictor p_i , computed by performing a step in direction d_p that points along the linearized Pareto front at $F(x_i)$. This point is only optimal if the Pareto set \mathcal{P} is not bended around x_i , but still near to \mathcal{P} if the step size is chosen sufficiently small. Thus, in the second step, p_i is corrected back to the Pareto set.

As with all continuation methods, the algorithm must start with an approximate solution and also, the search will be restricted to the connected component of the Pareto set where the point is contained in. If we have a Pareto set that has several connected components, we will need a starting point in each one if we wish to obtain the entire set. Algorithm 10 shows one iteration of the predictor corrector step.

It is important to note that the search process can be reduced by using existing information. For example, if the predictor p_i is known, and the corrector is performed via DDS, one can use the direction $\nu_1 = (x_i - p_i)/\|x_i - p_i\|$. Since the value of $F(x_i)$ of the previous candidate solution x_i is known, the use of ν_1 comes for free in the context of the DDS. This can be extended to other points in the neighborhood of p_i whose images are known.

3.3.5 Stopping criteria

Here we present some thoughts on the stopping criteria used when moving along the Pareto front. Two methods are presented, one associated with the convex weight of the KKT points,

Algorithm 10 Gradient free continuation step

Require: starting point (approximate) Pareto point $x_i \in \mathbb{R}^n$ **Ensure:** (approximate) New Pareto optimal point along the Pareto front

- 1: compute $x_1 \in N(x_i)$.
 - 2: compute $d_p := F(x_1) - F(x_i)$;
 - 3: compute $\mathcal{F}(x_i)$;
 - 4: compute $\lambda := \mathcal{F}(x_i)^+ d_p$;
 - 5: compute $v_p := \sum_{i=1}^r \lambda_i v_i^r$;
 - 6: compute t_p (see Algorithm 9).
 - 7: $p_i := x_i + t_p v_p$;
 - 8: compute d_c such that $d_p \perp d_c$.
 - 9: compute $\mathcal{F}(p_i)$;
 - 10: compute $\lambda := \mathcal{F}(p_i)^+ d_c$;
 - 11: compute $v_c := \sum_{i=1}^r \lambda_i v_i^r$;
 - 12: compute t_c (see Section 3.2, in page 38).
 - 13: $x_{i+1} := p_i + t_c v_c$.
 - 14: **return** New solution x_{i+1}
-

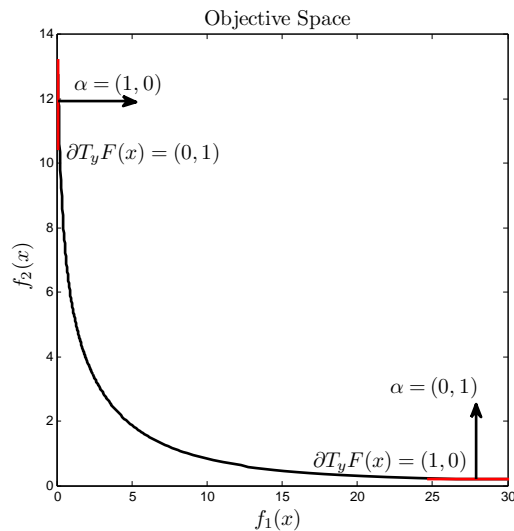


Figure 3.6: Example of the convex weight as a stopping criteria for an MOP with two objectives.

and a simple threshold in the number of iterations.

Convex weight

Different stopping criteria for the continuation method are proposed. The most straight forward criterion used is the associated convex weight in combination with the direction in objective space. Usually, the linearization of the Pareto front at the minimum (in the context of unconstrained minimization) of one function will be a line parallel to the axis. For two objectives we have the following: For f_1 , the tangent space of F at f_1 will be $(0, 1)$, and the convex weight associated with the end point will be $(1, 0)$. In this sense, we know when we are near the end of the Pareto front. Figure 3.6 shows an example of this property.

Number of iterations

We can also limit the number of steps in the continuation method when we are not sure in which direction do we need to go. It is common in engineering problems to be unaware of the geometrical form of the extreme segments of the Pareto front, hence the convex weight may not be a good stopping criteria in all cases. As a remedy, the continuation can be adapted to use a fixed number of iterations for the movement along the Pareto front.

3.4 Numerical Results

In this section we present results of several academic functions, in order to demonstrate the strength of the gradient free continuation method. A comparison between the Hillclimber with Sidestep is presented as well. It is of interest to mention that only the number of function evaluations are taken into account, but any indicator (such as Δ_p or hypervolume) can be used to assess the performance of each method. In these examples, the new method is apparently the clear winner, since the approximations of the Pareto sets differ considerably given the same (small) budget of function evaluations.

3.4.1 Convex problem, two objectives, two variables

We consider the following bi-objective optimization problem to test the gradient free continuation method.

$$\begin{aligned} & \text{minimize } f_1, f_2 : \mathbb{R}^2 \rightarrow \mathbb{R} \\ & f_1(x_1, x_2) = (x_1 - 1)^4 + (x_2 - 1)^2 \\ & f_2(x_1, x_2) = (x_1 + 1)^2 + (x_2 + 1)^2 \end{aligned} \tag{3.18}$$

The Pareto set forms an “S” shaped curve connecting the points $(-1, -1)^T$ and $(1, 1)^T$. The search has been started at $x_0 = (1, 1)^T$, which is the minimizer of f_1 . We selected to values of ϵ , the spread of the solutions throughout the Pareto front. For $\epsilon_1 = 0.04$, 63 candidate solutions were generated, spending a total of 191 function evaluations. For the

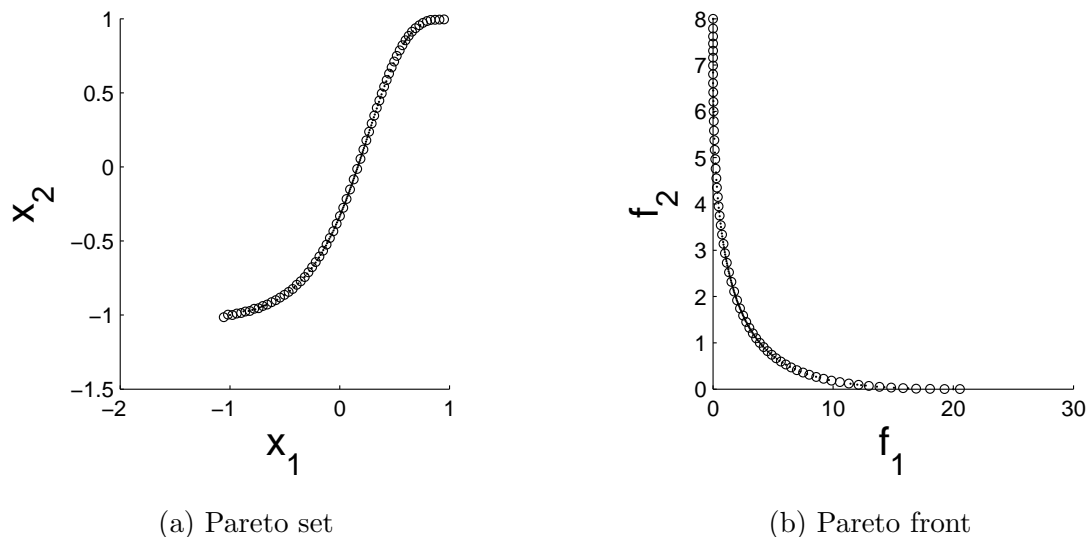


Figure 3.7: Numerical result for the gradient free continuation method using $\epsilon = 0.04$. The number of used function evaluations was 191.

larger value $\epsilon_2 = 0.08$, 32 solutions were obtained using a total of 95 function evaluations. Figures 3.7 and 3.8 show the result of the continuation method, and the difference in the choice of ϵ .

3.4.2 Convex problem, two objectives, eight variables

Next, we consider the following bi-objective problem to test the gradient free continuation method, along with the comparison between the Hill Climber with Sidestep (HCS) and MOEA/D.

$$\begin{aligned}
 &\text{minimize } f_1, f_2 : \mathbb{R}^2 \rightarrow \mathbb{R} \\
 &f_1(x) = (x_1 - 1)^4 + \sum_{i=2}^8 (x_i - 1)^2 \\
 &f_2(x) = \sum_{i=1}^8 (x_i + 1)^2
 \end{aligned} \tag{3.19}$$

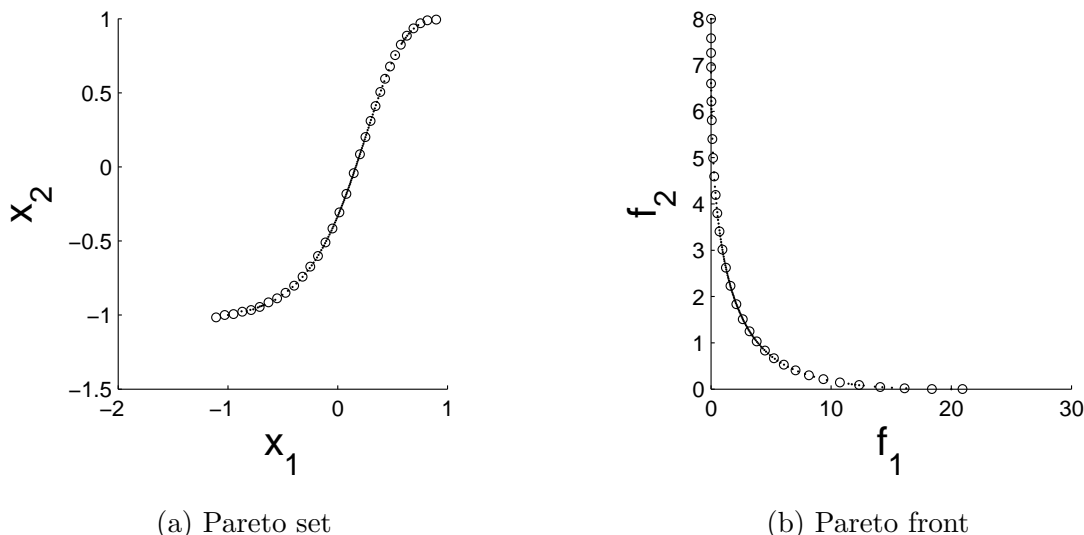


Figure 3.8: Numerical result for the gradient free continuation method using $\epsilon = 0.08$. The number of used function evaluations was 95.

In this case our gradient free continuation method spends a total of 561 function evaluations (we did not consider the number of function evaluations used to reach the front, but using the DDS, we would use around 20 or 30 additional function calls), while MOEA/D spends 588. The HCS spends over 4000 function evaluations and results in a worse approximation of the Pareto front. We can see clearly that our method beats MOEA/D in terms of approximation as well. Figure 3.9 shows the result of each method in parameter and objective space.

3.4.3 Witting problem

The following problem is also a bi-objective problem, taken from [21].

$$\begin{aligned}
 & \text{minimize } f_1, f_2 : \mathbb{R}^2 \rightarrow \mathbb{R} \\
 & f_1(x_1, x_2) = \frac{1}{2} \left(\sqrt{1 + (x_1 + x_2)^2} + \sqrt{1 + (x_1 - x_2)^2} + x_1 - x_2 + \lambda e^{-(x_1 - x_2)^2} \right) \\
 & f_2(x_1, x_2) = \frac{1}{2} \left(\sqrt{1 + (x_1 + x_2)^2} + \sqrt{1 + (x_1 - x_2)^2} - x_1 + x_2 + \lambda e^{-(x_1 - x_2)^2} \right)
 \end{aligned} \tag{3.20}$$

For $\lambda = 0.85$, the Pareto front contains a dent. The front consists of one concave and

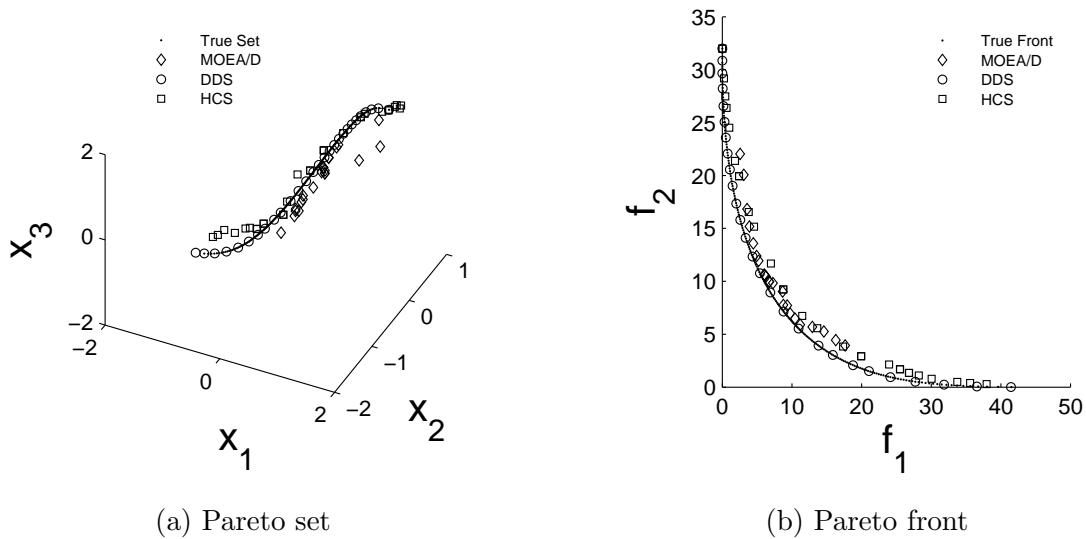


Figure 3.9: Numerical result for the gradient free continuation method, HCS and MOEA/D. Our method surpasses the other algorithms in terms of function evaluations and in the approximation of the Pareto front.

two convex parts (see Figure 3.10). We selected this example to show that the continuation method can obtain segments in concave regions of the Pareto front. On this problem our method uses 673 function evaluations, while the HCS uses 2030. Figure 3.10 shows the result of both continuation methods. The HCS obtains a good approximation but compared to the budget, the DDS continuation method outperforms the HCS in this example as well.

3.4.4 Okabe 1

The Okabe test function 1 is a benchmark model, described in [41]. It is of interest for the multiobjective optimization community since any slight change on the parameters greatly affects the values in objective space. The function is given by

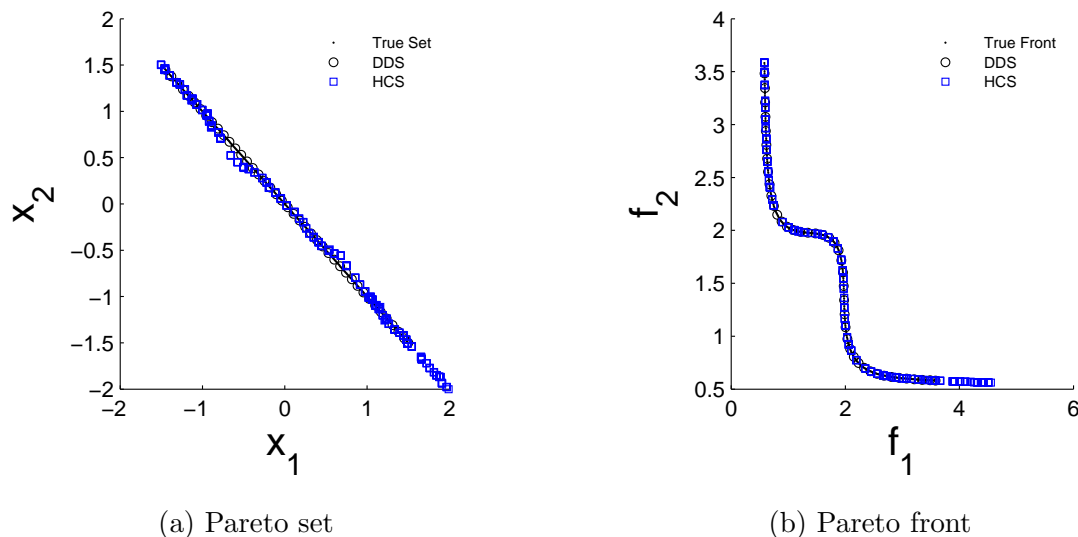


Figure 3.10: Numerical result for the gradient free continuation method and the HCS. Our method surpasses the other algorithm in terms of function evaluations and in the approximation of the Pareto front.

$$\begin{aligned}
 & \text{minimize } f_1, f_2 : \mathbb{R}^2 \rightarrow \mathbb{R} \\
 & g_1 = \cos\left(\frac{\pi}{12}\right)x_1 - \sin\left(\frac{\pi}{12}\right)x_2 \\
 & g_2 = \sin\left(\frac{\pi}{12}\right)x_1 + \cos\left(\frac{\pi}{12}\right)x_2 \\
 & f_1(x_1, x_2) = g_1 \\
 & f_2(x_1, x_2) = \sqrt{2\pi} - \sqrt{|g_1|} + 2\sqrt[3]{g_2 - 3\cos(g_1) - 3}
 \end{aligned} \tag{3.21}$$

Again, we have taken as starting point, the minimizer of f_1 . We have compared the results of the continuation method with the HCS and MOEA/D. In this settings, the DDS continuation method uses 570 function evaluations, while MOEA/D uses 675. The HCS has trouble approximating the Pareto front and uses more than 10000 function evaluations. Figure 3.11 shows the result of the DDS continuation method, the HCS and MOEA/D.

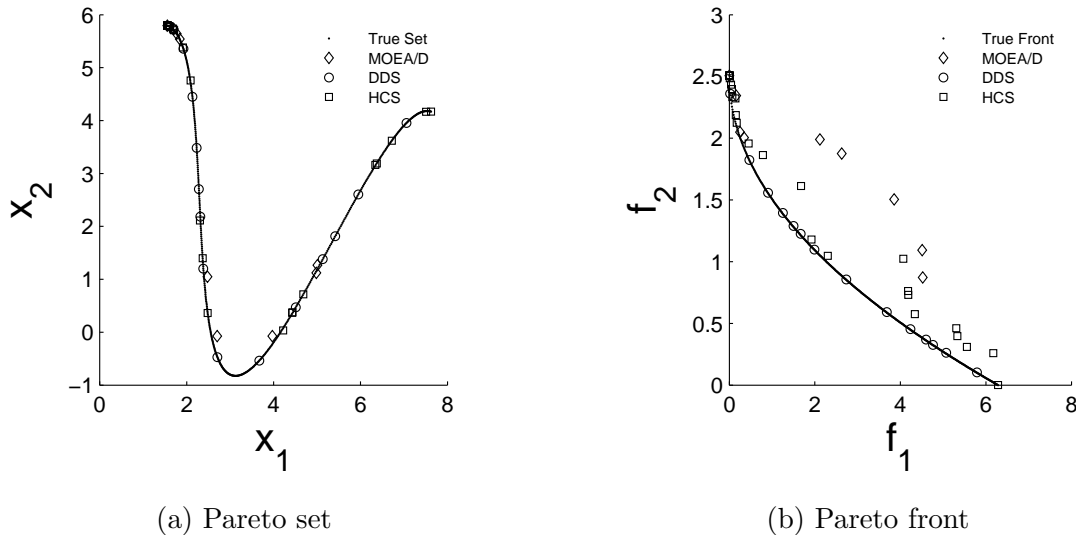


Figure 3.11: Numerical result for the DDS continuation method, HCS and MOEA/D. Our method surpasses the other algorithm in terms of function evaluations and in the approximation of the Pareto front.

3.4.5 Final comments

We have compared our gradient free continuation method with the Hill Climber with Sidestep, another continuation method which does not require gradient information. The results show that the novel approach described in this work outperforms in both function evaluations and the quality of the Pareto set approximation. It is clear that our new proposed method is useful in academic examples, but it is left to test it on a real world problem in order to test its strengths and limitations.

Also, we have tested our continuation method against MOEA/D, a state-of-the-art evolutionary algorithm. In order to be as fair as possible in our comparison, we fixed the number of elements in the population to match the solutions found in the continuation method. The fact that the function budget is limited, hinders the performance of this algorithm.

In order to adapt our method to fit any given problem, we need to address constraints, since they are commonly found on engineering applications. In the following chapters we will show an improvement on the continuation method to handle such constraints.

Constraint Handling

Constraints represent one of the major concerns in most optimization problems. The fact that the design space is somehow restricted to certain aspects makes it difficult to find feasible solutions. But constraints are not only found in academic functions. Nearly all engineering problems have a budget in costs, in raw materials, hours of work, etc. A farmer cannot plant an unlimited number of crops in an infinite patch of land. In the same way, wings of aircrafts cannot be one inch in thickness, even though that would reduce in great measure the costs.

So, we, as problem solvers, need to attack constraints to produce optimal solutions while staying inside the feasible regions for the decision makers to accept our results. Most evolutionary algorithms use *penalty functions* which usually lowers the fitness of an individual depending on how much are the constraints violated [42]. Other approaches have been considered, such as repairing procedures, which turn infeasible solutions to feasible ones using heuristic procedures; separation of objectives and constraints, such as in co-evolution strategies; or hybrid mechanisms which use a combination of different elements of other techniques to avoid some of their weaknesses. A very good reference on constraint handling for evolutionary algorithms can be found in [43].

For continuation methods, there is not much research done in constraint handling techniques, hence their use in engineering problems is, so far, diminished. This led us to adapt our method to address such issues. The constraint handling done in our work can be adapted to other continuation methods by performing some small modifications, thus a groundwork is set for further research on this topic.

In this section, we present the basic idea for handling linear constraints. Initially, some

basic concepts are presented. Then, a basic construction of a search direction is stated as to handle box constraints. Further on, we present a method to compute a promising search direction while handling linear inequality constraints which can be extended to equality constraints. Finally, we extend such methods to nonlinear constraints.

4.1 Box Constraints

Box constraints represent the most basic type of constraints inside optimization problems, and have the following form:

$$\begin{aligned} & \text{minimize } F(x) \\ & \text{such that } l \leq x \leq u \end{aligned} \tag{4.1}$$

where l and u are fixed vectors, and the inequalities are taken component by component. There are several types of methods to solve box-constrained optimization problems. The most common one is the gradient projection method [44], which is useful in simple problems but suffers from slow convergence and is not as interesting for high accuracy solutions [45].

For the treatment of box constraints we first perform a movement along the linearized Pareto front without using gradient information. For this, let (x, α, μ) be a KKT point of a MOP that contains the equality constraints $h_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \dots, p$. That is, it holds:

$$\sum_{i=1}^k \alpha_i \nabla f_i(x) + \sum_{i=1}^p \mu_i \nabla h_i(x) = 0 \tag{4.2}$$

where $\alpha_i \geq 0, i = 1, \dots, k$ and $\sum_{i=1}^k \alpha_i = 1$. Further, we assume that the rank of $J(x)$ is $k - 1$. It is known that in this case α is orthogonal to the linearized Pareto front at $F(x)$. Hence, a movement orthogonal to α is desired. For this, let $v \in \mathbb{R}^n$ such that $\langle \nabla h_i(x), v \rangle = 0$ for all $i = 1, \dots, p$. Then we obtain:

$$\begin{aligned}
 \langle J(x)\nu, \alpha \rangle &= \langle \nu, J(x)^T \alpha \rangle = \langle \nu, \sum_{i=1}^k \alpha_i \nabla f_i(x) \rangle = \langle \nu, - \sum_{i=1}^p \mu_i \nabla h_i(x) \rangle \\
 &= - \sum_{i=1}^p \mu_i \langle \nu, \nabla h_i(x) \rangle = 0
 \end{aligned} \tag{4.3}$$

And so, for such a direction it is either (i) $J(x)\nu = 0$ or (ii) a movement along ν in parameter space leads to a movement along the linearized Pareto front in objective space. That is, if we randomly pick a direction ν that is orthogonal to all gradients of the equality constraints at x , it will lead to a movement along the Pareto front with probability one since the dimension of the kernel of $J(x)$ is $n - k + 1$. Thus, if we choose a point $y \in N(x)$ at random, where $N(x)$ is a small neighborhood of x we can also expect that $F(x + t\nu) - F(x)$ is nearly orthogonal to α .

The only remaining element to fulfill is the orthogonality property. This is an easy task for box constraints: An equality constraint coming from an active inequality constraint is of the following form

$$h_i(x) = \pm x_{j_i} + a_{j_i} = 0, \quad i = 1, \dots, p, \tag{4.4}$$

where $a_{j_i} \in \mathbb{R}$, and thus

$$\nabla h_i(x) = (0, \dots, 0, \pm 1, 0, \dots, 0)^T, \quad i = 1, \dots, p. \tag{4.5}$$

Hence, directions ν with $\nu_{j_i} = 0$ fulfill the orthogonality property. In other words, points $y \in N(x)$ should be sampled with $y_{j_i} = x_{j_i}$ (this is, the coordinate value of the corresponding active constraint is not changed).

If we, on the other hand, desire to move along the front, the handling of box constraints can be viewed as a projection over the active constraint. This is, if the constraint is active, this means that we cannot leave the box without violating it, but in every other direction, we can shoot regularly. This translates to the following equation:

$$\langle h_i(x), \nu_i \rangle = 0 \Rightarrow \nu = \begin{bmatrix} \nu_1 \\ \vdots \\ \nu_{i-1} \\ 0 \\ \nu_{i+1} \\ \vdots \\ \nu_n \end{bmatrix} \quad (4.6)$$

And then, the movement in direction ν will not affect the current active constraints.

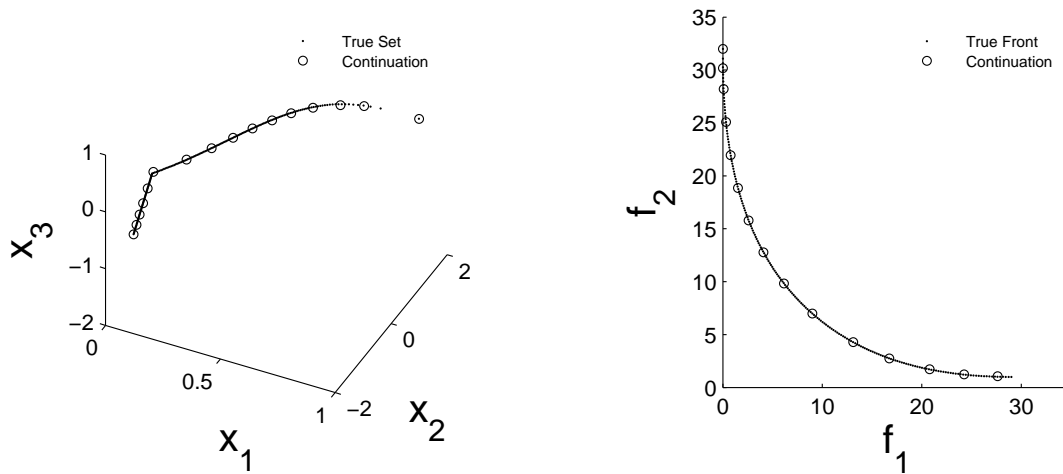
4.1.1 Numerical results

Now we turn our attention to an example to demonstrate the movement along a box-constrained Pareto set.

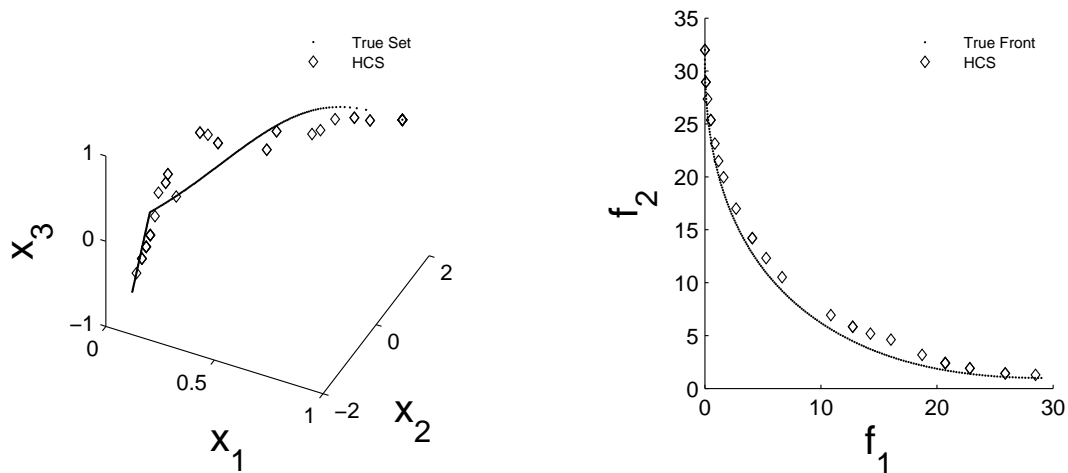
We consider the following bi-objective MOP to test the gradient free continuation method, along with the comparison between the Hill Climber with Sidestep.

$$\begin{aligned} & \text{minimize } f_1, f_2 : \mathbb{R}^2 \rightarrow \mathbb{R} \\ & f_1(x) = (x_1 - 1)^4 + \sum_{i=2}^8 (x_i - 1)^2 \\ & f_2(x) = \sum_{i=1}^8 (x_i + 1)^2 \\ & \text{such that } x_1 \geq 0. \end{aligned} \quad (4.7)$$

Figure 4.1 shows the results on this example. The DDS continuation method uses 702 function evaluations, and generated 16 solutions. The HCS spends 7995 function evaluations leading to 92 solutions (although most of them are clustered around several points). In this case we can see that the DDS continuation method is better in terms of convergence and overall cost of the procedure.



(a) Result of the DDS Continuation method for a box-constrained MOP.



(b) Result of the HCS for a box-constrained MOP.

Figure 4.1: Numerical result of the DDS continuation method and the HCS on a box-constrained MOP. Left are the projections of the Pareto set onto the first three coordinates and the candidate solutions and right analog figures in objective space.

4.2 Equality Constraints

Equality constrained optimization constitutes one of the two types of constraints in mathematics. An equality constrained optimization problem has the following form:

$$\begin{aligned} & \text{minimize } F(x) \\ & \text{s. t. } H(x) = 0 \end{aligned} \tag{4.8}$$

where $H(x) = (h_1(x), \dots, h_m(x))$, $h_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is a set of equality constraints. Many methods exist to solve equality constrained problems. Usually the objective function set is treated as a quadratic programming problem, but this is restricted to quadratic functions and linear constraints [46]. Other methods, solve the problems numerically using Lagrange multiplier [47]. To attack equality constraints, we focus on quadratic programming as well, adapted in the context of the discrete directed search.

The issue in attacking constraints is, as usual, the search for the direction ν that satisfies the desired movement along the Pareto set, as well as fulfilling the active constraints. Since we are looking at equality constraints, we know that the set of constraints is always active. In this sense, our work is facilitated.

We can state the direction search as a general least squares problem. This is, the problem is restated as follows:

$$\begin{aligned} & \text{minimize } \|Ax - b\|_2^2 \\ & \text{subject to } Cx = d \end{aligned} \tag{4.9}$$

where A is an $n \times k$ matrix, b is an $n \times 1$ vector, C is an $m \times k$ matrix and d is an $m \times 1$ vector. In the context of Directed Search, we can adapt this equation to the general search direction problem and obtain

$$\begin{aligned} & \text{minimize } \|J(x)\nu - d\|_2^2 \\ & \text{subject to } H(x) = 0 \end{aligned} \tag{4.10}$$

Since we are not looking for an optima, but rather the search direction, we need to solve for ν . The resulting equation for computing a promising search direction is given as follows:

$$\begin{bmatrix} 2(J(x)^T J(x)) & H^T \\ H & 0 \end{bmatrix} \begin{bmatrix} \nu \\ \lambda^* \end{bmatrix} = \begin{bmatrix} J(x)^T d \\ 0 \end{bmatrix} \quad (4.11)$$

Then we can solve for ν and obtain

$$\begin{bmatrix} \nu \\ \lambda^* \end{bmatrix} = \begin{bmatrix} 2(J(x)^T J(x)) & H^T \\ H & 0 \end{bmatrix}^+ \begin{bmatrix} J(x)^T d \\ 0 \end{bmatrix} \quad (4.12)$$

In this equation, H is the set of active constraints, J is the Jacobian at the current solution and d the direction of the DDS. Note that this does not cost additional function evaluations, since we already have the value of $J(x)$. For the gradient free approach, we simply replace $J(x)$ for $\mathcal{F}(x)$.

4.2.1 Numerical results

In the following, we consider an academic bi-objective optimization problem. Note that there is no comparison here with the Hill Climber with Sidestep since it is not adapted to handle constraints. However, for classical PC methods, equality constraints can be easily adapted [24], although they require Hessian information.

$$\begin{aligned} & \text{minimize } f_1, f_2 : \mathbb{R}^2 \rightarrow \mathbb{R} \\ & f_1(x) = (x_1 - 1)^2 + (x_2 - 1)^2 \\ & f_2(x) = (x_1 + 1)^2 + (x_2 + 1)^2 \\ & \text{such that } x_2 = \frac{1}{2}x_1. \end{aligned} \quad (4.13)$$

Figure 4.2 shows the result of the continuation method applied on MOP (4.13). The number of function evaluations for this example is 116. The starting point used for this example was $x = (1, 1/2)^T$. As we can see, by solving the Least Squares problem we are able to obtain a suitable direction for the continuation method.

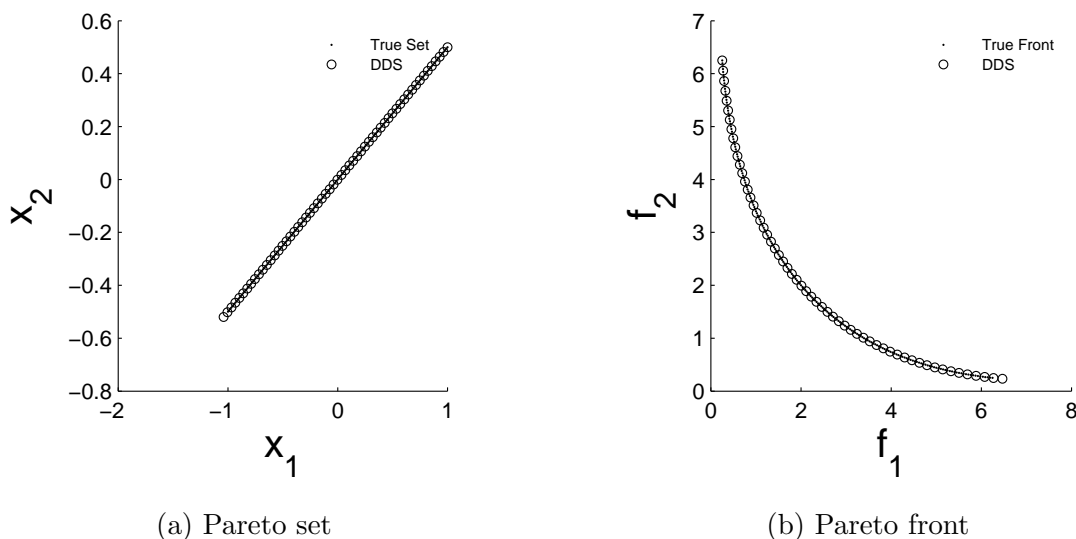


Figure 4.2: Numerical result for the gradient free continuation method on an equality constrained MOP. Apparently, the algorithm is able to compute a well spread set of solutions with a low budget of function evaluations.

For nonlinear equality constraints, we can proceed as in the linear case. The matrix H will hold the set of active constraints as well, even if they are quadratic. Since we evaluate this matrix at the current point, no additional consideration has to be taken.

In this case we take the same MOP as before, only changing the constraints to nonlinear ones.

$$\begin{aligned}
 & \text{minimize } f_1, f_2 : \mathbb{R}^2 \rightarrow \mathbb{R} \\
 & f_1(x) = (x_1 - 1)^2 + (x_2 - 1)^2 \\
 & f_2(x) = (x_1 + 1)^2 + (x_2 + 1)^2 \\
 & \text{such that } h_1(x) = \sum_{i=1}^2 (x_i - c_i)^2 = r^2
 \end{aligned} \tag{4.14}$$

where $c \in \mathbb{R}^2$ and $r \in \mathbb{R}$.

The resulting set is the top left segment of a circle, going from $(1, 1)$ to $(-1, -1)$. Figure 4.3 shows the result of the continuation method, using as starting point $x = (1, 1)^T$. Figure 4.4 shows a close-up of the Pareto optimal solutions. In this case, the method moves slightly

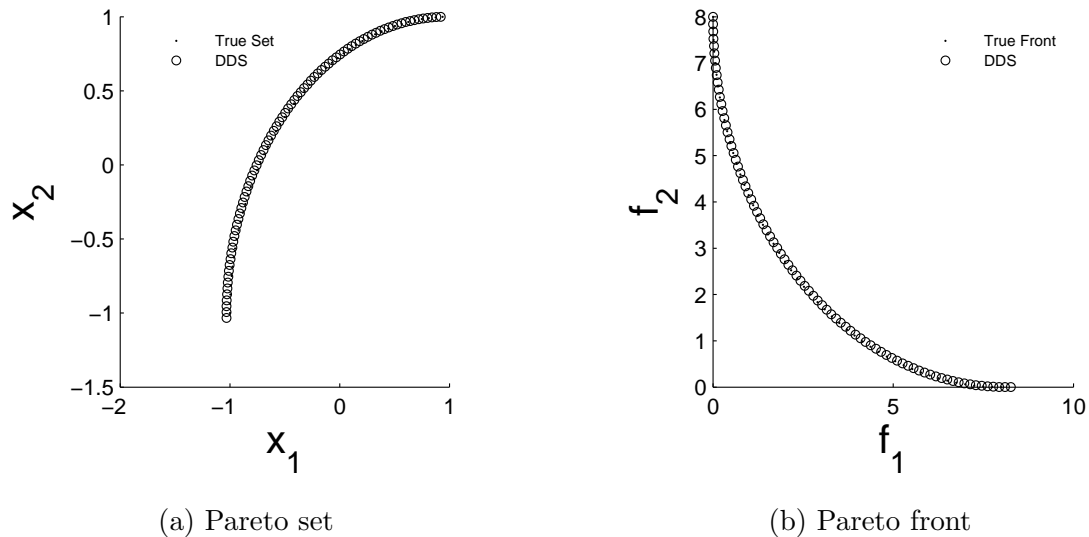


Figure 4.3: Numerical result for the gradient free continuation method on a nonlinear equality constrained MOP. Apparently, the algorithm is able to compute a well spread set of solutions with a low budget of function evaluations.

afar from the set, since there is no correction step taking place.

4.3 Inequality Constraints

Problems involving inequality constraints on control variables have been treated since the early stages of optimization [48]. It is of interest for any decision maker to be able to limit the available parameters at hand. Inequality constrained optimization problems have the following form

$$\begin{aligned}
 & \text{minimize } F(x) \\
 & \text{s. t. } G(x) \leq 0
 \end{aligned}
 \tag{4.15}$$

where $G(x) = (g_1(x), \dots, g_p(x))$ is a set of inequality constraints. As with equality constraints, techniques to solve them involve extended Lagrange multipliers and gradient projection methods. In this work, we propose to use the same method as with equality constraints.

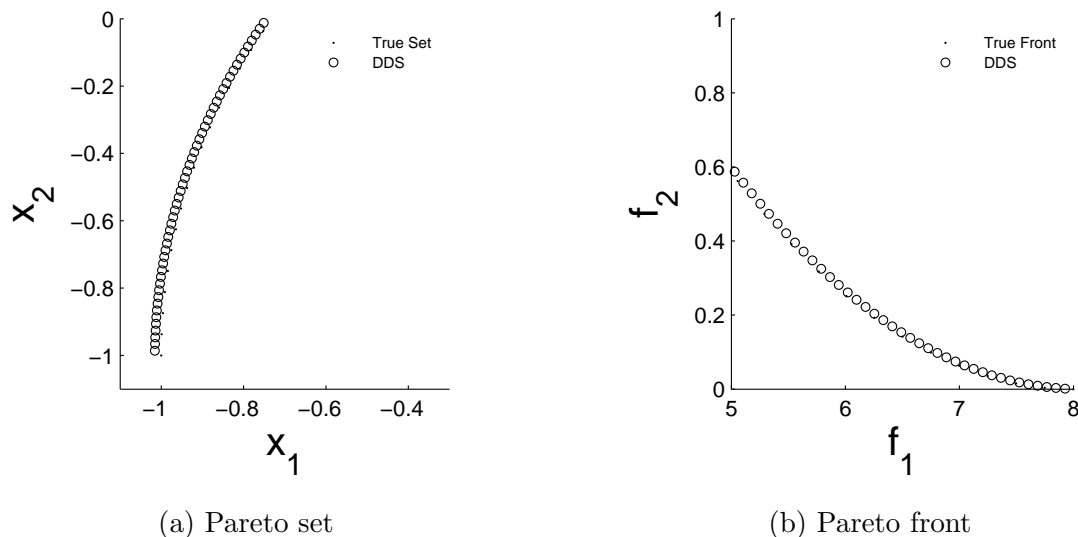


Figure 4.4: Close-up of the numerical result for the gradient free continuation method on a nonlinear equality constrained MOP. There is no corrector step and hence, the method does not ensure a perfect approximation.

We have to take into consideration the set of active constraints, this is, when and if, any constraint is violated. If no constraint is active, we can proceed as in the unconstrained case.

4.3.1 Numerical results

In the following, we consider the same example as before, a bi-objective optimization problem. Again there is no comparison here with other methods, since no other continuation method is adapted to handle this type of constraints.

$$\begin{aligned}
 & \text{minimize } f_1, f_2 : \mathbb{R}^2 \rightarrow \mathbb{R} \\
 & f_1(x) = (x_1 - 1)^2 + (x_2 - 1)^2 \\
 & f_2(x) = (x_1 + 1)^2 + (x_2 + 1)^2 \\
 & \text{such that } x_2 \leq \frac{1}{2}x_1.
 \end{aligned} \tag{4.16}$$

Figure 4.5 shows the result for the continuation method on MOP (4.16). Using the evaluation of G at the point x , we are able to determine if the constraint is active or not. In

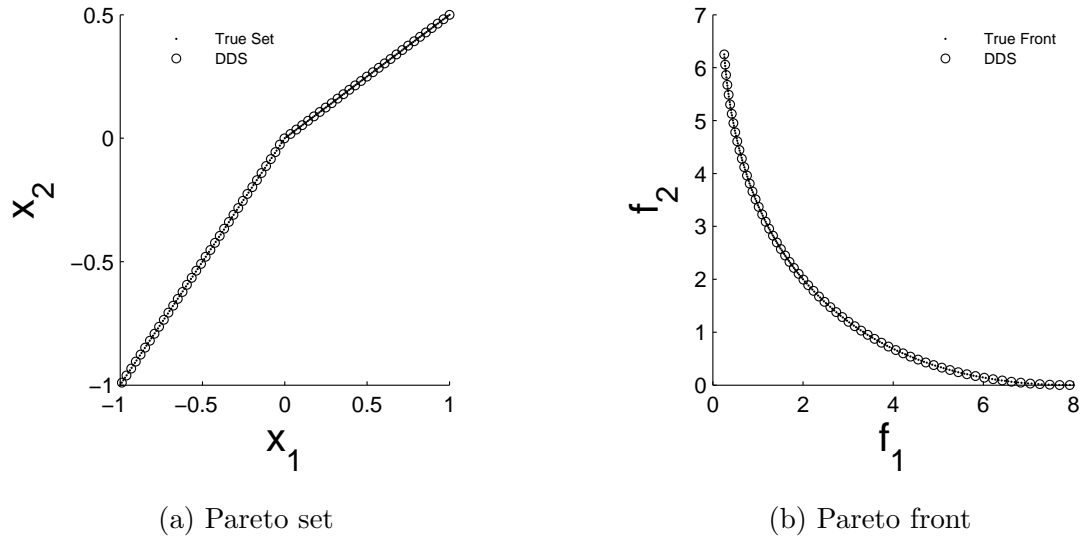


Figure 4.5: Numerical result for the gradient free continuation method on a linear inequality constrained MOP. Apparently, the algorithm is able to compute a well spread set of solutions with a low budget of function evaluations.

this case, when $x = (0, 0)^T$, the method proceeds as for the unconstrained case.

Finally, we test our gradient free continuation method in a nonlinear inequality constraint example.

$$\begin{aligned}
 &\text{minimize } f_1, f_2 : \mathbb{R}^2 \rightarrow \mathbb{R} \\
 &f_1(x) = (x_1 - 1)^2 + (x_2 - 1)^2 \\
 &f_2(x) = (x_1 + 1)^2 + (x_2 + 1)^2 \\
 &\text{such that } g_1(x) = \sum_{i=1}^2 x_i^2 \leq r^2
 \end{aligned} \tag{4.17}$$

where $r \in \mathbb{R}$.

Figure 4.6 shows the result on this academic example. We can see that there the DDS continuation method is capable of following even nonlinear inequality constraints. The lack of a corrector step generates an increasing movement away from the set, but if the distance between each solution is chosen to be relatively small, this error will be insignificant.

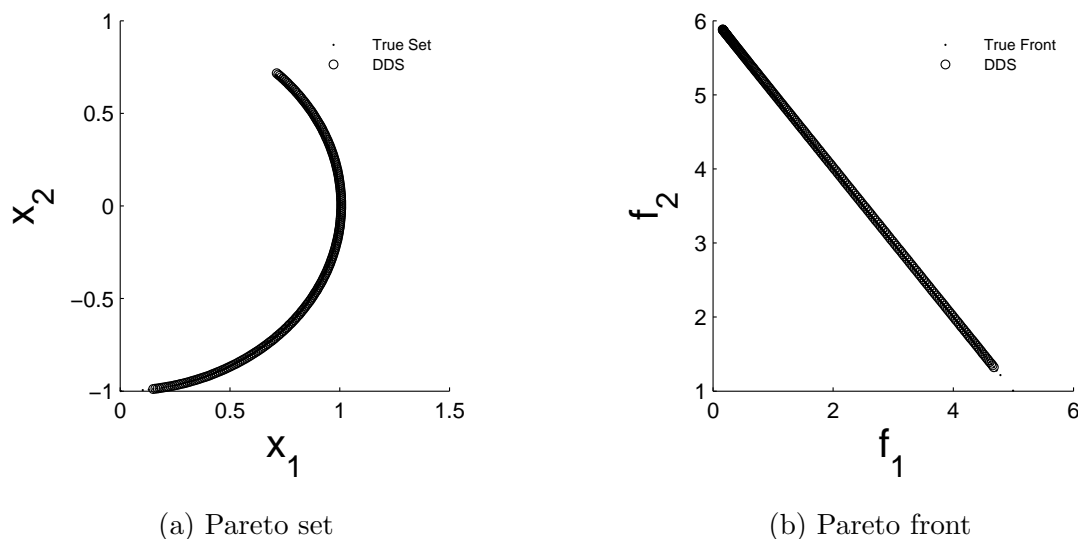


Figure 4.6: Numerical result for the gradient free continuation method on a nonlinear inequality constrained MOP. We can see that our method follows the active constraint.

4.4 Final Comments

We have demonstrated the adaptation of the DDS continuation method into different types of constraints. The fact that no other continuation method is fitted with constraint handling other than equality constraints [24], further shows the strength of this novel algorithm.

Even if the gradient free continuation method performs acceptably in the academic examples, we are left to test this on an engineering application. The following chapter shows the results of a space trajectory design problem, which fully shows that the continuation method is useful in both academic and real-world models.

Three-Impulse Space Trajectory Design Problem

Space mission designs have been an issue for aerospace institutes around the world. The costs associated with each launch have to be carefully analyzed and any change in the trajectory after a launch can reflect a delay in time which could potentially affect the entire mission. Optimization has been used to minimize the time from launch to destination, or to select the best path for an orbiting object to efficiently use the gravitational pull of nearby planets.

Many authors have proposed different approaches within optimization for this space trajectory design problems. One of the first works related to this subject was proposed by Lawden [49], who set the foundations of optimization of space trajectories. Following his work, Edelbaum [50] created a model for a constant gravitational field problem, as well as defining two-impulse trajectory designs.

Work by Schütze et al. [51] addressed the problem of a low-thrust gravity-assist trajectories. They used subdivision techniques to find the approximation of the Pareto set, which are useful in engineering problems due to the disconnected components of the feasible regions. Other work by Schütze et al. [52] uses NSGA-II together with an archiving strategy to obtain the set of approximate solutions, which gives the decision maker some freedom in the selection of optimal and near-optimal solutions.

It is clear that this area of research has been thoroughly studied, but there is still much to do. Evolutionary algorithms have commonly been used, as well as set-oriented algorithms for the discontinuity of the optimal solutions. To our knowledge, no continuation method

has been used to address space trajectory designs.

5.1 Three-Impulse Test Problem

This problem, taken from [53], represents a three-impulse Low Earth Orbit (LEO) to Gravitational Earth Orbit (GEO) transfer. The objectives are to minimize the fuel spent, and the time it takes to reach the final destination, the rendezvous with a target spacecraft on the final orbit.

This movement is a three-impulse transfer between circular orbits. Given the geometry of the initial and final orbits, a third impulse is allowed during the transfer, which is thus divided into two arcs. In such a case, the solution space to be spanned becomes wider. Constraints on the eccentricity and minimum distance from the planet surface are enforced for both arcs. The vector of decision variables x is defined as follows:

- The vector is parametrized by means of the solution of Lambert's problem between the initial and final position, along two coast arcs, $x = (t_W, t_{T_1}, r_1, \Delta\theta_1, t_{T_2})^T \in \mathbb{R}^5$, where r_1 and $\Delta\theta_1$ assign the position of the intermediate impulse, while t_{T_1} and t_{T_2} are the transfer times along the first and second coast arc, respectively.

In this setting, the spacecraft departs at time t_W from a circular orbit around the Earth with radius $r_1 = 7,000$ km, and after a transfer time $T = t_{T_1} + t_{T_2}$ is injected into a circular orbit with radius $r_f = 42,000$ km. An intermediate maneuver is performed at time $t_W + t_{T_1}$ and at position defined in polar coordinates by the radius r_1 and the angle $\Delta\theta_1$. The objective functions are the total transfer time T and the sum of the three impulses Δv_{tot} . The search space is defined in Table 5.1. The values have been normalized from 0 to 1 inside the continuation method.

Figure 5.1 shows the graphical representation of this test problem, along with its design variables. The Pareto front used in the numerical examples is taken from [54].

Table 5.1: Definition of parameters for the three-impulse test case.

Parameter	Min Value	Max Value	Description
t_W	0	1.62	Departure of the spacecraft from the Low Earth Orbit.
t_{T_1}	0.03	21.54	Transfer time along the first coast arc.
r_1	7010	105410	Radius of the polar coordinate that assigns the position of the intermediate impulse.
$\Delta\theta_1$	0.01	$2\pi - 0.01$	Angle of the polar coordinate that assigns the position of the intermediate impulse.
t_{T_2}	0.03	21.54	Transfer time along the second coast arc.

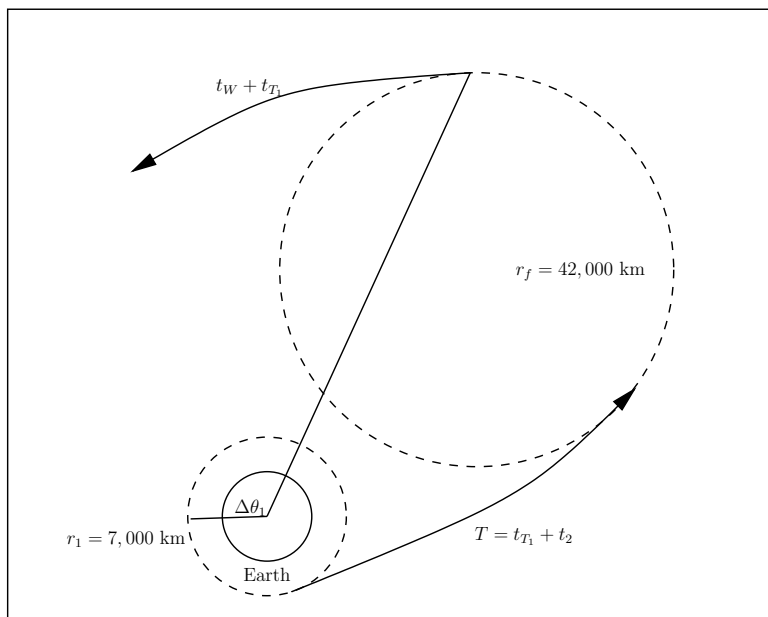


Figure 5.1: Graphical representation of the three-impulse test problem. The spacecraft departs from an orbit around Earth and performs several movements until it reaches its final destination.

5.2 Lambert's Problem

The following description of the Lambert problem is taken from [53].

The solution of the two-point boundary value problem for Keplerian motion is known as Lambert's problem, and is represented by the determination of the orbit parameters of an orbit having a specified transfer time t_{des} between two prescribed positions in space, P_1 and P_2 . Each coast arc of the transfer trajectory is identified by means of the assumed initial and final positions and transfer time. The total number of design variables is thus equal to $3(N - 2) + 2$.

The required velocity ΔV is evaluated from the knowledge of the orbit parameters, by means of a simple vector operation,

$$\Delta \vec{v}_i = \vec{v}_i^I - \vec{v}_{i-1}^F, \quad i = 1, 2, \dots, N, \quad (5.1)$$

where \vec{v}_i^I and \vec{v}_i^F are the initial and final velocities at the edges of the i -th coast arc, for

$1 < i < N - 1$, while $\vec{v}_0^F = \vec{v}_{LEO}$ and $\vec{v}_N^I = \vec{v}_F$ are the velocities on the initial LEO and final target orbit, respectively.

Lambert's algorithm simplifies greatly the structure of the feasible set, because even when inequality constraints apply to each on of $N - 1$ coast arcs, the constraints on the final position of the spacecraft are satisfied inherently. The cost is increased in the computational cost of each function evaluation, but this trade-off is of great use and does not hurt our continuation method.

5.3 Numerical Results

In this section, we present the numerical results of the continuation method applied on the three-impulse trajectory problem. Since the Pareto front of this problem falls into different connected components, it is necessary for the continuation method to start on a KKT point in each of those segments. We selected the two major components of the Pareto front as starting elements in our test run. The initial points were obtained through Matlab's genetic algorithm. Even though the points are not Pareto optimal points, they are near enough to the set as to enable the DDS continuation method to work properly.

Figure 5.2 shows the results of the continuation method using two different starting points in two components of the Pareto front. In the upper part of the front, we see that the continuation covered almost all that segment, but as the problem suggests, the DDS continuation gets trapped in a local front, thus avoiding the entire coverage of the tail segment of the front. The second iteration of the DDS continuation method managed to cover the entire area successfully. The reference front was taken from [54], and is the result of 2,000 runs of the evolutionary algorithms used in that work. The number of function evaluations for the continuation method was approximately 7,000 for each segment. We rounded to this number since the continuation method continued to find dominated solutions (not present in the figures) along each segment. This budget is compared with NSGA-II and MACS (Multi-Agent Collaborative Search [54]) which used a total of 30,000 function evaluations to obtain this front. In Figure 5.3, we made a projection of the first three components of

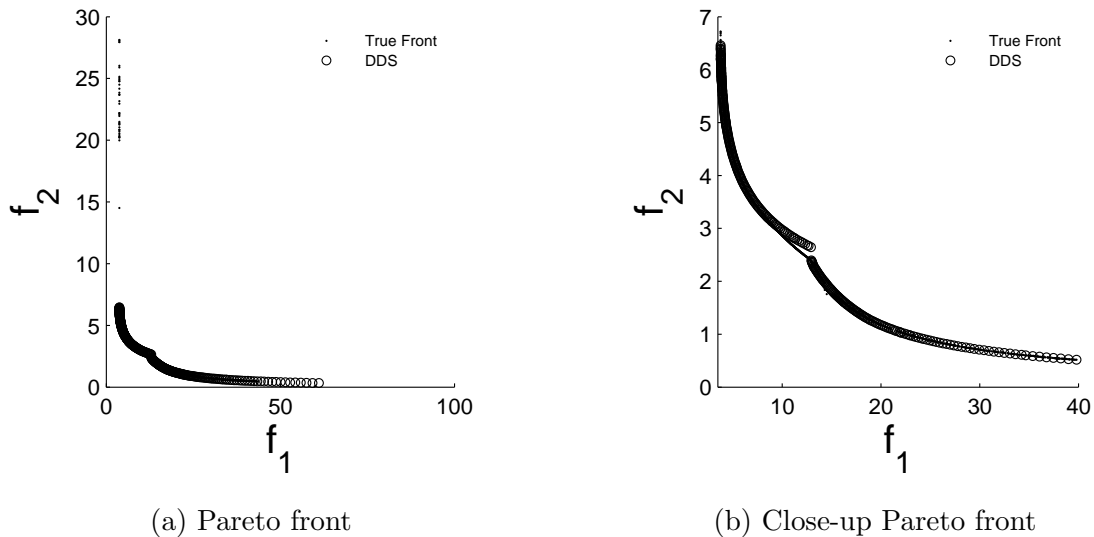
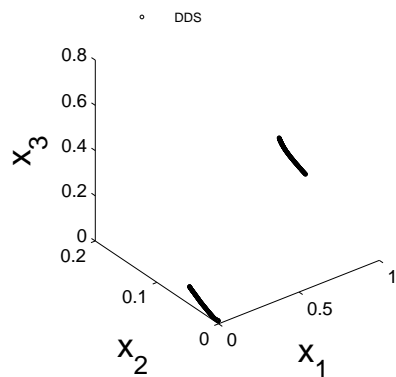


Figure 5.2: Numerical result for the gradient free continuation method on the three-impulse trajectory problem. Left is the reference front along with the two iterations of the DDS continuation method. Right, close-up to the covered areas.

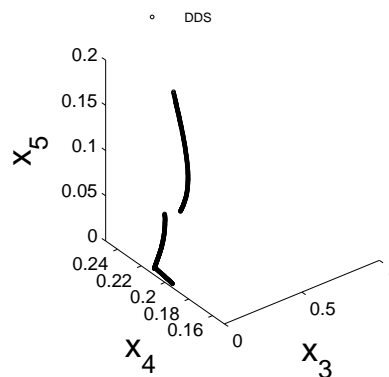
the decision variables, and on the last three components as well. This demonstrates that the Pareto set is disconnected, and thus, several starting points will be required if we wish to find each segment of the Pareto set.

Figure 5.4 shows the result of the Hill Climber with Sidestep using the same budget of number evaluations as the continuation method, and the same starting points for each run. We can see that the HCS fails to cover the second segment of the front, as well as performing poorly on the first segment. As with the DDS continuation method, the candidate solutions shown in this figure are only the non-dominated ones.

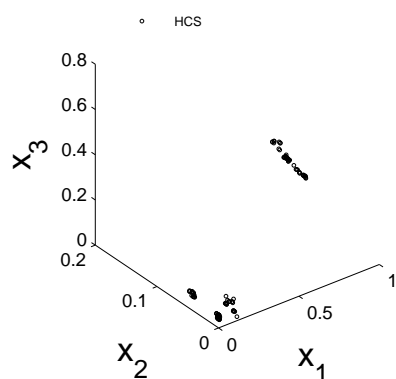
Figure 5.5 shows the numerical result of Matlab’s genetic algorithm. The budget of function evaluations for this method was 30,000. The population was set to 500, as to match the number of solutions found by our DDS continuation method. Matlab’s genetic algorithm fails to obtain any solution in the second part of the front, and also misses most of the first segment.



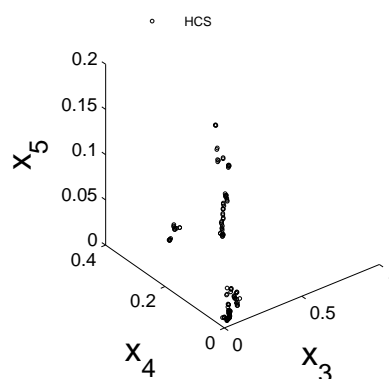
(a) Parameter space, first three coordinates



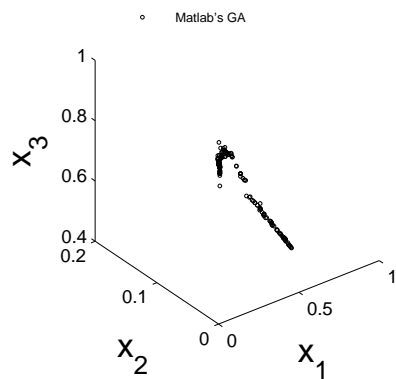
(b) Parameter space, last three coordinates



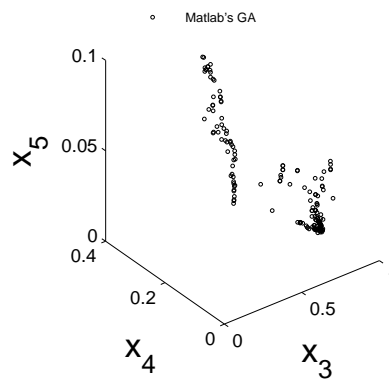
(c) Parameter space, first three coordinates



(d) Parameter space, last three coordinates



(e) Parameter space, first three coordinates



(f) Parameter space, last three coordinates

Figure 5.3: Numerical result for the DDS continuation method, the HCS, and *Matlab's* genetic algorithm in parameter space. Left is the projection over the variables x_1, x_2 and x_3 . Right, the projection over the three decision variables x_3, x_4 and x_5 .

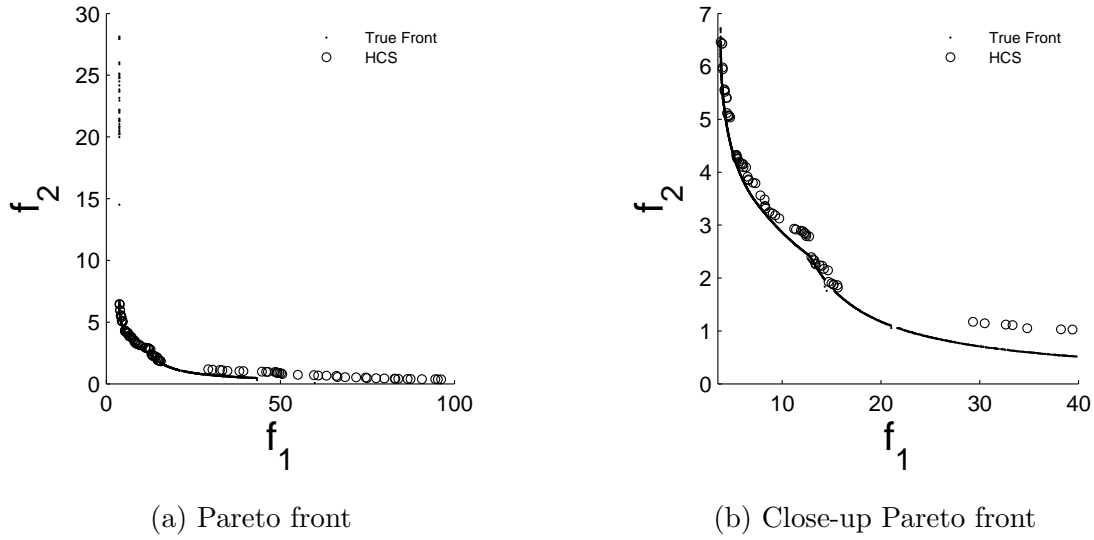


Figure 5.4: Numerical result for the Hill Climber with Sidestep on the three-impulse trajectory problem. Left is the reference front along with the two iterations of the HCS. Right, close-up to the covered areas.

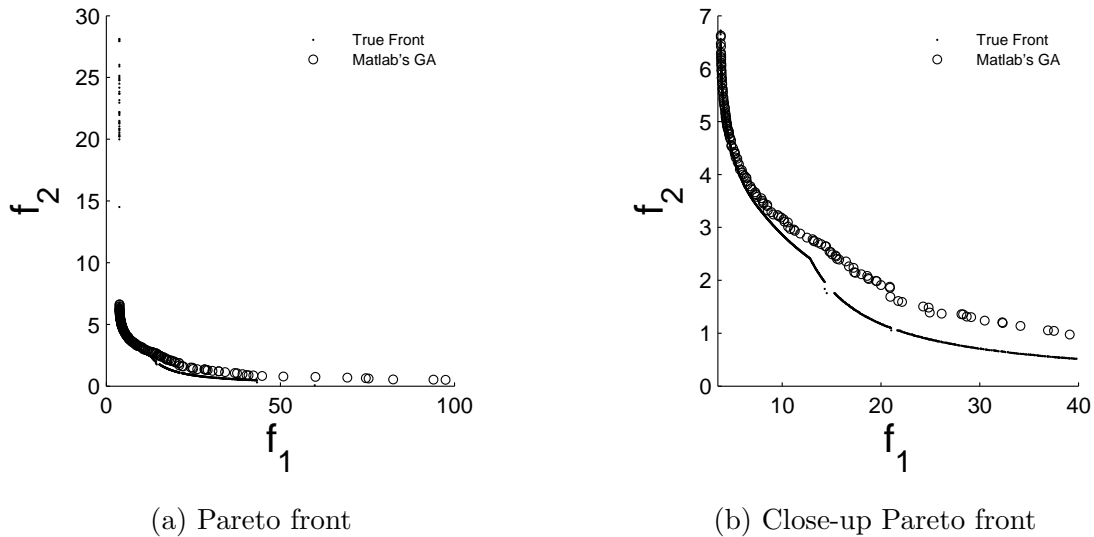


Figure 5.5: Numerical result for Matlab's genetic algorithm on the three-impulse trajectory problem. Left is the reference front along with the two iterations of the HCS. Right, close-up to the covered areas.

5.4 Final Comments

We have tested the DDS continuation method on a three-impulse trajectory design for an object leaving Earth and reaching a spacecraft. The results, though acceptable, need further improvements. Our method managed to find several connected components of the Pareto set, while lowering the total number of function evaluations, as compared with other evolutionary algorithms.

When compared with the other gradient free continuation method, the DDS continuation managed to outperform the HCS in both, spread and convergence of the solutions. It is clear that continuation methods can be used to solve engineering problems, even with high complexity as this space trajectory optimization problem. Further tests have to be performed in order to improve the results obtained by our novel approach.

Discussion and Future Work

In this work, a novel gradient free continuation method was proposed. This method is capable of moving along the set of optimal solutions, the Pareto set, using a predictor corrector strategy. This algorithm surpasses other continuation methods since it eliminates one of the major drawbacks of general continuation methods: The requirement of first order and second order gradient information. The other only continuation method that does not use gradient information is the Hill Climber with Sidestep.

We tested this method on a number of academic functions, and in each case, managed to improve the results compared with other state-of-the-art algorithms, including the evolutionary algorithm MOEA/D and the HCS, which is another gradient free continuation method. We were able to fully adapt the DDS continuation method to handle different types of constraints: From the bound constraints to the more complex nonlinear inequality constraints. It is important to notice that continuation methods can handle equality constraints, but this normally requires Hessian information. The DDS continuation method can handle equality constraints, as well as inequality constraints without any gradient information.

When applied to a real-world example, a three impulse space trajectory optimization for a satellite leaving Earth's orbit, the DDS continuation method managed to get a nice approximation of the Pareto front. Compared to NSGA-II, a powerful evolutionary algorithm, our method reduced the number of function evaluations used by half. Although the approximation was not as good as the one from this evolutionary algorithm, we think that with additional parametrization, the DDS continuation method could improve the results significantly. Since the reference Pareto front used to compare these algorithms has a non

differentiable segment, it is believed that in fact there are several local fronts near each other, which would explain the result of the continuation method.

The handling of constraints is an important factor to considerate when comparing against other continuation algorithms since no other algorithm of this kind can handle the wide variety of constraints present in almost all engineering problems. We managed to lower the number of function evaluations in every considered MOP, in addition to obtaining a nice spread of solutions along the front. Up until the work done in this thesis, the DDS continuation algorithm is the only gradient free continuation method that can handle equality and inequality constraints and obtains good approximations of Pareto sets with relatively few function evaluations.

During the development of this thesis:

- A paper was submitted to an international journal, the *IEEE Transactions on Evolutionary Computation*. The title of this paper is *The Directed Search Method for Multiobjective Optimization Problems Part II: Toward Memetic Strategies*, and its authors: Oliver Schütze, Adriana Lara, Sergio Alvarado, Eduardo Salinas Márquez, Carlos A. Coello Coello, Shaul Salomon and Gideon Avigad. In this paper, the gradient free continuation method for the treatment of unconstrained MOP's is presented, and the potential advantages of this method are discussed.
- The work was also presented in the international conference *EVOLVE 2013 - A Bridge Between Probability, Set-Oriented Numerics, and Evolutionary Computation*, held in Leiden, The Netherlands from July 11th to the 13th. The title of of this work is *A Gradient-free Continuation Method for Box-Constrained Multiobjective Optimization Problems*, and its authors: Eduardo Salinas Márquez and Oliver Schütze. In this work, the DDS continuation method is adapted for the handling box constraints, and later on, to general constraints.

There is still much research to be done, but we have proven empirically that continuation methods are a powerful tool to solve constrained multiobjective optimization problems. The low number of function evaluations and possibility of neglecting gradient informations,

makes it useful for non differentiable problems and engineering applications whose function evaluations are costly in time and computational power.

6.1 Future Work

There are several interesting options regarding the DDS continuation method. Some future improvements are presented as follows:

- The most important aspect to improve is the adaptation for more than two objectives. We have shown that the algorithm works when $k = 2$, but it is left to modify it when the Pareto front is of higher dimensions. A box recovery approach was done in Mejía's work [36], using gradient information, thus for $k = 3$, there are not many modifications to accomplish this task.
- We tested the gradient continuation method in constrained academic functions, however, it is of interest to see the behavior of our method in a real-world example with several types of constraints.
- Another interesting improvement for the continuation method would be to work on disconnected Pareto set. This is, to start on one segment, and then follow a dominated path to another segment and restart the continuation steps, and so on. The stopping criteria would need to be changed, and the step size control strategies as well, in the dominated segments of the Pareto front, for example in ZDT3.
- There is still work to be done for the constraint handling modifications. More precisely, the corrector step of the DDS. We used the DDS to correct back to the Pareto front, by utilizing the convex weight. However, in the constrained case, the direction ν in parameter space in the corrector may not correspond to the correction to the active constraint set. Thus, the DDS would need to be adapted to shoot back to the active constraints, without using gradient information.

Bibliography

- [1] Zhong Bowen and Qiao Zhide. Multiobjective optimization design of transonic airfoils. *19th ICAS Congress*, 94(2), 1994.
- [2] Gaiddon A., Knight D. D., and Poloni C. Multicriteria design optimization of a supersonic inlet based upon global missile performance. *Journal of Propulsion and Power*, 20(3):542 – 558, 2004.
- [3] Kazuhisa Chiba, Shigeru Obayashi, Kazuhiro Nakahashi, and Hiroyuki Morino. High-fidelity multidisciplinary design optimization of wing shape for regional jet aircraft. In *Evolutionary Multi-Criterion Optimization*, volume 3410 of *Lecture Notes in Computer Science*, pages 621–635. Springer Berlin / Heidelberg, 2005.
- [4] Timoleon Kipouros. Stochastic Optimisation in Computational Engineering Design. *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, 175:475 – 490, 2013.
- [5] Harvey S. A., Dawes W. N., and Gallimore S. J. An automatic design optimisation system for axial compressors, part i: software development. *ASME Turbo Expo 2003 GT2003-38115*, 2003.
- [6] Timoleon Kipouros, Daniel Jaeggi, Bill Dawes, Geoff Parks, Mark Savill, and P. J. Clarkson. Insight into high-quality aerodynamic design spaces through multi-objective optimization. *CMES: Computer Modeling in Engineering and Sciences*, 37(1):1 – 23, 2008.

-
- [7] Michael Emmerich and Boris Naujoks. Metamodel assisted multiobjective optimisation strategies and their application in airfoil design. *Adaptive Computing in Design and Manufacture VI*, 6:248 – 260, 2004.
- [8] Chamis Christos. Coupled Multidisciplinary Optimization of Engine Structural Performance. *Journal of Aircraft*, 36(1):190 – 199, 1999.
- [9] Leopold Summerer Massimiliano Vasile and Paolo De Pascale. Design of earth-mars transfer trajectories using evolutionary-branching technique. *Acta Astronautica*, 56(8):705 – 720, 2004.
- [10] M. Vasile J.P. Sánchez, C. Colombo and G. Radice. Multi-criteria comparison among several mitigation strategies for dangerous near earth objects. *Journal of Guidance, Control and Dynamics*, 32(1):121 – 142, 2009.
- [11] Lofti Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE transactions on Automatic Control*, 8:59 – 60, 1963.
- [12] Yacov Y. Haimes, Leon S. Lasdon, and David A. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE transactions on System, Man, and Cibernetics*, 1(3):296 – 297, 1971.
- [13] Kaisa M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer, 1999.
- [14] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182 – 197, April 2002.
- [15] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on*, 11(6):712–731, 2007.
- [16] Gary B.Lamont Carlos A. Coello Coello and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation Series. Springer, 2007.

- [17] Deniz Yuret and Michael de la Maza. Dynamic hill climbing: Overcoming the limitations of optimization techniques. In *The Second Turkish Symposium on Artificial Intelligence and Neural Networks*, pages 208–212. Citeseer, 1993.
- [18] Oliver Schuetze, Gustavo Sanchez, and Carlos A Coello Coello. A new memetic strategy for the numerical treatment of multi-objective optimization problems. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 705–712. ACM, 2008.
- [19] Kalyanmoy Deb, Manikanth Mohan, and Shikhar Mishra. A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions. *KanGAL report*, 2003002, 2003.
- [20] Adriana Lara, Sergio Alvarado, Shaul Salomon, Gideon Avigad, Carlos A. Coello Coello, and Oliver Schütze. The gradient free directed search method as local search within multi-objective evolutionary algorithms. In Oliver Schütze, Carlos A. Coello Coello, Alexandru-Adrian Tantar, Emilia Tantar, Pascal Bouvry, Pierre Del Moral, and Pierrick Legrand, editors, *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, volume 175 of *Advances in Intelligent Systems and Computing*, pages 153–168. Springer Berlin Heidelberg, 2013.
- [21] Oliver Schütze, Adriana Lara, and Carlos Coello Coello. The directed search method for unconstrained multi-objective optimization problems. *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, 1(1):1 – 32, 2011.
- [22] Francis Edgeworth. *Mathematical Psychics*. 1881.
- [23] Vilfredo Pareto. *Cours D'Economic Politique*, volume I and II. 1896.
- [24] Jorge Nocedal and S Wright. Numerical optimization, series in operations research and financial engineering. *Springer, New York*, 2006.
- [25] J. Fliege and B. F. Svaiter. Steepest Descent Methods for Multicriteria Optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

-
- [26] S. Schäffler and R. Schultz and K. Weinzierl. A stochastic method for the solution of unconstrained vector optimization problems. *Journal of Optimization, Theory and Application*, 114(1):209–222, 2002.
- [27] A. Lara López. *Using Gradient Based Information to build Hybrid Multi-objective Evolutionary Algorithms*. PhD thesis, 2012.
- [28] Eckart Zitzler, Lothar Thiele, Eckart Zitzler, Eckart Zitzler, Lothar Thiele, and Lothar Thiele. *An evolutionary algorithm for multiobjective optimization: The strength pareto approach*. Citeseer, 1998.
- [29] Juhani Koski. Defectiveness of weighting method in multicriterion optimization of structures. *Communications in applied numerical methods*, 1(6):333–337, 1985.
- [30] Marco Laumanns, Lothar Thiele, and Eckart Zitzler. An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169(3):932–942, 2006.
- [31] Kerstin Dächert, Jochen Gorski, and Kathrin Klamroth. An adaptive augmented weighted tchebycheff method to solve discrete, integer-valued bicriteria optimization problems. Technical report, Technical Report BUW-AMNA-OPAP 10/06, University of Wuppertal, FB Mathematik und Naturwissenschaften, 2010.
- [32] Eugene L. Allgower and Kurt Georg. *Introduction to Numerical Continuation Methods*, volume 45 of *Classics in Applied Mathematics*. 1990.
- [33] Claus Hillermeier. *Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach*. Birkhäuser, 2001.
- [34] Honggang Wang. Zigzag search for continuous multiobjective optimization. *INFORMS Journal on Computing*, 2012.
- [35] Nidamarthi Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248, 1994.

- [36] Erick Ignacio Mejía Estrada and Oliver Schütze. The directed search method for constrained multi-objective optimization problems. Master's thesis, CINVESTAV-IPN, 2012.
- [37] Sergio Jesús Alvarado García and Oliver Schütze. El punto a punto de las técnicas de búsqueda local para algoritmos de optimización multiobjetivo. Master's thesis, CINVESTAV-IPN, 2012.
- [38] P. Moscato and C. Cotta. A gentle introduction to memetic algorithms. *Handbook of Metaheuristics*, pages 105–144, 2003.
- [39] C. Cotta P. Moscato and A. Mendes. *New Optimization Techniques in Engineering*. Springer-Verlag, 2004.
- [40] Oliver Schütze, Adriana Lara, and Carlos Coello Coello. The directed search method for multi-objective optimization problems part I: Local search. *submitted to IEEE Transactions on Systems, Man and Cybernetics, Part B*, 2013.
- [41] Tatsuya Okabe, Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. On test functions for evolutionary multi-objective optimization. In Xin Yao, EdmundK. Burke, JoséA. Lozano, Jim Smith, JuanJulián Merelo-Guervós, JohnA. Bullinaria, JonathanE. Rowe, Peter Tiño, Ata Kabán, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 792–802. Springer Berlin Heidelberg, 2004.
- [42] M. H. Ghiasi A. Abedian and B. Dehghan-Manshadi. Effect of a linear-exponential penalty function on the ga's efficiency in optimization of a laminated composite panel. *International Journal of Computational Intelligence*, pages 5–11, 2006.
- [43] Carlos A. Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of state of the art. *Computer methods in applied mechanics and engineering*, 191:1245–1287, 2002.

-
- [44] J. B. Rosen. The gradient projection method for nonlinear programming. part i. linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8:181–217, 1960.
- [45] Suvrit Sra Dongmin Kim and Inderjit S. Dhillon. Tackling box-constrained optimization via a new projected quasi-newton approach. *SIAM Journal on Scientific Computing*, 30(6):3548–3563, 2010.
- [46] J. E. Dennis Jr. M. R. Celis and R. A. Tapia. A trust region strategy for equality constrained optimization. Technical report, Mathematical Sciences Department, Rice University, Houston, Texas, 1984.
- [47] Dimitri P Bertsekas. Constrained optimization and lagrange multiplier methods. *Computer Science and Applied Mathematics, Boston: Academic Press, 1982*, 1, 1982.
- [48] Arthur E Bryson, Walter F Denham, and Stewart E Dreyfus. Optimal programming problems with inequality constraints. *AIAA journal*, 1(11):2544–2550, 1963.
- [49] Derek F Lawden. *Optimal trajectories for space navigation*. Butterworths, 1963.
- [50] TN Edelbaum. How many impulses? *Astronautics and Aeronautics*, 5(11):64–69, 1967.
- [51] Oliver Schütze, Massimiliano Vasile, Oliver Junge, Michael Dellnitz, and Dario Izzo. Designing optimal low-thrust gravity-assist trajectories using space pruning and a multi-objective approach. *Engineering Optimization*, 41(2):155–181, 2009.
- [52] Oliver Schütze, Massimiliano Vasile, and Carlos A Coello Coello. Approximate solutions in space mission design. In *Parallel Problem Solving from Nature–PPSN X*, pages 805–814. Springer, 2008.
- [53] E.A. Minisci and G. Avanzini. Orbit transfer manoeuvres as a test benchmark for comparison metrics of evolutionary algorithms. In *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, pages 350–357, 2009.

- [54] Massimiliano Vasile and F Zuiani. Multi-agent collaborative search: an agent-based memetic multi-objective optimization algorithm applied to space trajectory design. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 225(11):1211–1227, 2011.