Boris Djartov

Multi-Criteria Decision-Making with Many Decision Makers: Integrating Fairness and Gain

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

INF

FAKULTÄT FÜR
INFORMATIK

Intelligent Cooperative Systems
Computational Intelligence

Multi-Criteria Decision-Making with Many Decision Makers: Integrating Fairness and Gain

Master Thesis

Boris Djartov

April 15, 2021

Supervisor: Prof. Dr.-Ing. habil. Sanaz Mostaghim

Boris Djartov: *Multi-Criteria Decision-Making with Many Decision Makers: Integrating Fairness and Gain*
Otto-von-Guericke Universität
Intelligent Cooperative Systems
Computational Intelligence
Magdeburg, 2021.

Abstract

Being faced with an increasingly complicated world, decision makers need to collaborate as well as consider multiple conflicting interest to tackle more difficult problems. The use of multi-objective optimisation algorithms has been geared towards solving such problems, with many works attempting to incorporate specific decision makers preferences or exploring particular areas of interest. The goal of this thesis is to provide a self contained method to aid many decision makers when they are collectively confronted with a multi-criteria optimisation problem and need to collaborate and consider each others preferences. The main ideas for this thesis come courtesy of the 2020 Dagstuhl report, Supporting Problem Solving with Many Decision Makers in Multi-objective Optimization [63]. Thus, this thesis incorporates a multi-level structure, where we have two multi-objective optimisation tasks. The initial level constitutes of the real-world problem to be solved, while the second level serves as a mechanism to incorporate an element of fairness, which provides solutions acceptable by all participating decision makers. The proposed approach is based on the NSGA-II algorithm, with the fast non-dominated sorting approach being used to handle the gain and fairness level. The approach was then employed on a number of simulated scenarios, featuring different numbers of decision makers and mindfully placed preference points to have greater understanding of the potential of the concept. The results indicate that the newly proposed Adaptive NSGA-II for Teams algorithm focuses the solutions within an area defined by the preference points of the decision makers. This results in superior convergence compared to the base NSGA-II, however, it negatively effects the diversity of the solutions. Adaptive NSGA-II for Teams, along with an additional filter to represent certain power dynamics between the decision makers, form the structure of a unified concept which may help many decision makers tackle a multi-objective optimization problem.

Contents

List of Figures V

List of Tables VII

List of Algorithms IX

List of Acronyms XI

1. Introduction 1

1.1. Motivation 1

1.2. Goals of this Work 2

1.3. Thesis Structure 3

2. Background 5

2.1. Multi-Objective optimization 5

2.2. Evolutionary Algorithms 8

2.3. Multi-Objective Evolutionary Algorithms 17

2.4. Evaluation Metrics for Multi-Objective Evolutionary Algorithms 21

2.5. Fairness and Gain 24

2.6. Scalable Test Problems for Evolutionary Multi-Objective Optimization 25

3. Related Work 27

3.1. Multi-Objective Optimisation and Fairness 28

3.2. Multi-Objective Optimisation and Preference Points 29

3.3. Multi-Objective Optimisation and Many Decision Makers 32

4. Methodology and Approach 35

4.1. NSGA-II for Teams 35

4.2. Adaptive NSGA-II for Teams 40
4.3. Some Decision Makers are Fairer than Others 44
4.4. On Real World Application 47

5. Implementation and Experiment Design **49**
5.1. Implementation 49
5.2. Experiment Design 51

6. Results **57**
6.1. Results Analysis of NSGA-II for Teams 57
6.1.1. Examining the Effect that Distances Between the Preference Points has on the Final Solutions 58
6.1.2. Examining the Effects that Differences in Terms of Domination has on the Final Solutions 62
6.2. Results Analysis for the Adaptive NSGA-II for Teams Algorithm 68
6.3. Examining the Behaviour when Faced with Complex Cases . . . 74
6.4. Summary of Analysis 76

7. Conclusion and Future Work **79**

A. Crowding Distance Assignment **83**

B. Fast Non-Dominated Sorting **85**

C. Global variables of the algorithm **87**

D. Preference Points Values for Showcased Situations **89**

Bibliography **91**

List of Figures

2.1. Representation of the mapping between decision space and objective space . . . 6
2.2. Illustration of domination and a non-dominated set for a minimization problem . . . 7
2.3. Simple representation of a one-point crossover . . . 14
2.4. The probability density function for creating offspring under an SBX . . . 15
2.5. Probability distribution for creating a mutated value for continuous variables . . . 17
2.6. GD and IGD metric . . . 22
2.7. HV metric . . . 23
2.8. The NSGA-II Population on Test Problem DTLZ2 . . . 26

4.1. Illustration of a Problematic Scenario Regarding the NSGA-II for Teams Algorithm . . . 40
4.2. Illustration of Weighted Filtering . . . 45
4.3. Flowchart on Practical Implementation . . . 48

5.1. PlatEMO Sequence Diagram of Running a General Multi-Objective Optimization Algorithm . . . 50
5.2. PlatEMO GUI . . . 50
5.3. Scenarios 1 and 2 . . . 52
5.4. Scenarios 3, 4 and 5 . . . 53

6.1. Scenario 1 . . . 60
6.2. Scenario 2 . . . 61

6.3. Scenario 3 65
6.4. Scenario 4 66
6.5. Scenario 5 67
6.6. Preference points repositioning, scenario 3 71
6.7. Preference points repositioning, scenario 4 72
6.8. Preference points repositioning, scenario 5 73

List of Tables

2.1. Biological terms as they relate to ideas and concepts in EA . . . 10

5.1. Variants for an attempted move 54

6.1. Metrics for scenario 1 60
6.2. Metrics for scenario 2 61
6.3. Metrics for scenario 3 65
6.4. Metrics for scenario 4 66
6.5. Metrics for scenario 5 67
6.6. GD values in comparison with initial concept 69
6.7. GD values in comparison with NSGA-II algorithm 69
6.8. Fairness values for the variations of the improved concept 70
6.9. Metrics, scenario 6 75
6.10. Metrics, scenario 7 75

C.1. Global variables of the algorithm 87

D.1. Preference Point Values 90

List of Algorithms

1. General Evolutionary Algorithm 9
2. Binary Tournament Selection 12
3. NSGA-II 20
4. Fairness Filter 37
5. NSGA-II for Teams 38
6. Preference Repositioning 42
7. Adaptive NSGA-II for Teams 43
8. Modified Influences 46
9. Crowding Distance Assignment 83
10. Fast Non-Dominated Sorting 86

List of Acronyms

GD	Generational distance
HV	Hypervolume
IGD	Inverted Generation Distance
SBX	Simulated binary crossover
FNDS	Fast non-dominated sorting
EA	Evolutionary Algorithms
GA	Genetic Algorithms
DM	Decision maker
DMs	Decision makers
PfP	Preference point
PfPs	Preference points
M	Number of objectives
N	Population size
D	Number of decision variables
GUI	Graphical user interface
FEs	Evaluations per run

1. Introduction

1.1. Motivation

Increasingly complex problems inevitably require multiple differing Decision makers (DMs) and multiple objectives which are to be pursued [5][63]. Thus, the methods to solve multi-objective optimization problems need to accommodate a situation when more than one Decision maker (DM) is involved in the decision making process. This additional consideration carries with it many additional challenges, some of which may be:

- Differing views on constraints and objective functions.
- Differing interests and motives for solving the problem.
- Differing negotiation tactics.
- Differences between the DMs in terms of influence or power.

In order to tackle the problem of multiple DMs in multi-objective optimization this work will examine scenarios in which the DMs involved have been able to:

- Express their preferences in terms of the objectives.
- Agreed on a problem formulation.
- Agreed to participate in a negotiation in order to reach a joint decision.

Albeit challenging, the aforementioned set of requirements should be a reasonable expectation for a group of DMs attempting to solve a particular problem [63].

The problem of many DMs tackling a multi-objective optimization problem has been addressed by other scholarly works. However, the ideas behind these approaches have been either problem specific [37] or have focused on implementing weighted methods [38]. In contrast, this approach focuses on the ideas set up by the 2020 Dagstuhl report, Supporting Problem Solving with

Many DMs in Multi-objective Optimization [63]. Thus, it uses pre-existing Preference points (PfPs) from the DMs in order to tackle the problem of multiple DMs. Additionally, the 2020 Dagstuhl report necessitates the use of a multi leveled structure, where the second level consists of another multi-objective optimization problem. Therefore, this approach will be a multi-level multi-objective optimization concept, where the multiple DMs specify their preferences as PfPs i.e. specific values for the objectives which are to be optimised. The optimization process is regulated by the second level, which is an additional mechanism to modify the behaviour of the solutions and get a set of alternatives, inline with the participants interests.

1.2. Goals of this Work

This work aims to build upon the method described in the aforementioned 2020 Dagstuhl report [63]. Henceforth, its goals are:

- To formulate a complete implementation of the approach in the 2020 Dagstuhl report [63].
- To understand the behaviour of the algorithm, with respects to various positioning of the PfPs, in terms of distance and domination.
- To propose improvements of the idea in the 2020 Dagstuhl report [63] and examine their effectiveness.
- To examine the applicability of the algorithm when dealing with complex problems consisting of many decision variables and objectives.
- To create an overall methodology for dealing with multi-objective optimization problems with multiple differing DMs, with the ideas of this thesis at its core.
- To enable this approach to also capture elementary power dynamics between DMs.

Thus, this approach aims to supply future DMs with a more flexible and modifiable set of tools, enabling them to make better decisions.

1.3. Thesis Structure

After the introduction of this thesis, Chapter 2 covers the theoretical background of the concepts used in the proposed approach. The fundamentals of multi-objective optimization are conveyed along with a general explanation of genetic algorithms. This is followed by a presentation of the evolutionary algorithms used as the back bone of the thesis. Afterwards, an overview the various metrics and test problems is presented, along with the fairness and gain concepts from the 2020 Dagstuhl report [63]. The related work, categorised by its various similarities to the concepts of this thesis is conveyed in Chapter 3. The exact implementation details and other proposed concepts are explained in Chapter 4. Chapter 5 contains the details of the various tests and experiments, as well as certain technical details regarding the implementation. Following, the results and their interpretation is covered in Chapter 6. Finally, Chapter 7 delivers a conclusion about the presented approach and the found results whilst also including avenues for future research.

2. Background

This chapter covers the fundamentals of the concepts applied in this thesis. It can be used as reference for details of the underlying methods used in the proposed approach.

2.1. Multi-Objective optimization

Given that the topic deals with multi-objective optimization it seems only fitting that we start by specifying it in greater detail. Optimization refers to finding one or more solutions which correspond to extreme values of one or multiple objectives. While single-objective optimization is concerned with optimising the value of just one objective, multi-objective optimization involves a set of objectives which are subject to either minimisation or maximisation [18, Chapter 2]. Mathematically, a multi-objective optimization problem can be defined as

$$min/max(f(\vec{x})) \qquad \text{where: } f(\vec{x}) = [f_1(\vec{x}), ..., f_m(\vec{x})], \quad \text{s.t } \vec{x} \in S, \tag{2.1}$$

where m is the number of objectives functions $f_i : S \to Z$. The $\vec{x}$ is a vector of decision variables, such that $\vec{x} \in R^n$. The set S is a feasible set formed by constraint functions and is referred to as (feasible) decision space or (feasible) search space. In multi-objective optimization the objectives constitute a multi-dimensional space in addition to the decision variable space. This additional space is called the objective space Z which is of great interest. For each solution $\vec{x}$ in the decision space there exists a point in the objective space. The aforementioned mapping takes place between the an n-dimensional decision variable vector and and m-dimensional objective vector [18, Chapter 2]. A representation of this mapping is illustrated in figure 2.1

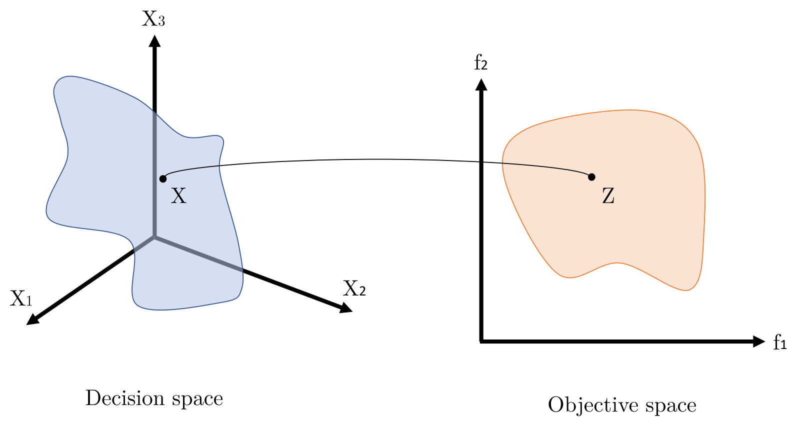


Figure 2.1.: Representation of the mapping between decision space and objective space

Although more prominent, single objective optimization involves greater information loss than multi-objective optimization. This is due to the fact that most real world problems are in fact multi-objective problems, often consisting of a great many objectives. Thus, when one uses single objective optimization it generally means that a multi-objective optimization problem was degenerated into a single-objective one, which results in certain quality loss [18, Chapter 2] [4, Chapter 1]. This focus on multiple objectives means a greater focus on trade-offs between the differing solutions, which is why the general idea behind multi-objective optimization is to generate a diverse set of solutions with the best trade-offs. These solutions can then be examined and analysed by a DM, usually human, to make a final decision [18, Chapters 1 and 2] [4, Chapter 1]. Due to the presence of multiple objectives, a simple comparison is somewhat difficult. Therefore, in order to compare the solutions and specify the set of best solutions the concept of domination is employed. Mathematically speaking, a solution $\vec{x_1} \in S$ is said to dominate a solution $\vec{x_2} \in S$ if both of the following conditions are met:

- $f_i(\vec{x_1}) \leq f_i(\vec{x_2})$ for all $i = 1...m$
- $f_j(\vec{x_1}) < f_j(\vec{x_2})$ for at least one j,

where the $f_i(\vec{x_n})$ are the objective values for the ith objective with regards to the solution $\vec{x_n}$. Additionally, the equations suppose a minimization problem i.e. where all the objectives are minimized. A solution is refereed to as a non dominated solution if there is no other solution which dominates it. A non-dominated set contains all the solutions which are not dominated by the other solutions.

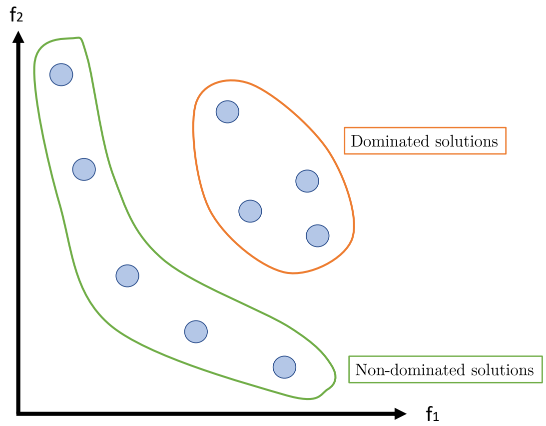


Figure 2.2.: Illustration of domination and a non-dominated set for a minimization problem

With a better understanding of the concept of domination we can now define Pareto optimality. Specifically, a solution $\vec{x^*} \in S$ is said to be Pareto optimal if there exist no other solution $\vec{x} \in S$ such that:

- $f_i(\vec{x}) \leq f_i(\vec{x^*})$ for all $i = 1...m$
- $f_j(\vec{x}) < f_j(\vec{x^*})$ for at least one j,

where again the presented equations are for a minimization problem. The non-dominated set of the entire feasible decision space S is the globally Pareto-optimal set [18, Chapter 2] [4, Chapter 1].

Generating a Pareto-optimal set of solutions is the real focus when one tries to solve multi-objective optimization problems. As stated before, due to the fact that we end up with a set of solutions, we need a DM/DMs to chose the final course of action [4, Part 2, Chapter 1]. Thus, based on the classification of the methods by Hwang and Masud [31] and taking into account the participation of the DM we have:

- Methods where no articulation of preference information is used (no-preference methods).

- Methods where a posteriori articulation of preference information is used (a posteriori methods).
- Methods where a priori articulation of preference information is used (a priori methods) .
- Methods where progressive articulation of preference information is used (interactive methods).

Although not the only kind of classification for these methods, it is a very prominent one which will additionally ease the explanation of the algorithm presented in this thesis [4, Part 2, Chapter 1].

2.2. Evolutionary Algorithms

A common approach to tackle these multi-objective optimization problems is by using Evolutionary Algorithms (EA) [35] [18, Chapters 2 and 4]. EAs are a subset of metaheuristic algorithms. Metaheuristics are fairly general computational techniques that are typically used to solve numerical and combinatorial optimization problems. When using metaheuristics one attempts to adapt the probelm onto the various structural components that make up the metaheuristic algorithms. They are usually applied to problems for which no efficient solution algorithm is known, i.e. problems, for which all known algorithms have an (asymptotic) time complexity that is exponential in the problem size. Due to the difficult nature of such problems one can only hope to find approximate solutions to the problem. In this regard, metaheuristics have usually been able to find sufficiently good solutions for these difficult problems [53, Chapter 11]. Of the broader family of evolutionary algorithms, Genetic Algorithms (GA), first introduced by J.H. Holland in 1975 seem to be one of the more popular variants [9]. Other noteworthy concepts include Evolutionary Strategies, introduced by Rechenberg and Schwefel [8][65]. Additionally, significant ideas also include Evolutionary Programming, introduced by Fogel [26], as well as Genetic Programming, which gathered great momentum thanks to John Koza [3]. As the name might imply these algorithms are inspired form the Darwinian notion of evolution and survival of the fittest. The main idea behind these approaches is to utilize a population of individuals (solutions in our case) along with various selection mechanisms in order to obtain the "fittest" solutions to a problem [22]. Although quite useful, these ideas seem to lack the popularity of the more prominent neural networks [12, Chapter 1]. Having shown that there exist a great deal of variety when it comes to evolutionary algorithms, certain mutual traits can be observed in many of these approaches. The basic structure of an evolutionary algorithms consists of:

- A decision on how to model the problem/ encoding,
- A method to create an initial population,
- A fitness function to evaluate the individuals,
- A selection method on the basis of the fitness function,
- A set of genetic operators to modify the individuals,
- A termination criterion for the search, and
- The values for various parameters [53, Chapter 13] .

The aforementioned structure can be better observed in algorithm 1, which showcases a generic evolutionary algorithm. The biological terms associated with these algorithms and their properties are displayed in table 2.1 [53, Chapters 11 and 12]. Given that EAs consist of multiple integral parts a dedicated detailing of each of them will be provided. This short presentation of the various underlying mechanisms will focus on the aspects which are most relevant to the concepts and ideas covered in this thesis.

Algorithm 1: General Evolutionary Algorithm [53, Chapter 13]

```
t ← 0                                  // initialize the generation counter
initialize pop(t)                      // create the initial population
evaluate pop(t)                        // evaluate population (compute fitness)
while NOT Termination condition do
    t ← t + 1 // count the created generation
    select pop(t) from pop(t - 1)      // select individuals based on fitness
    alter pop(t)                       // apply genetic operators
    evaluate pop(t)                    // evaluate the new population
end
return pop                             // population of evaluated individuals
```

Biological term	Meaning in context of EA
individual	solution
fitness	quality of solution
phenotype	solution in objective space
genotype	solution in decision space (representation)
gene	one part of the genotype (attribute)
allele	value of a gene
parent	individual used to produce new individuals
mutation	neighborhood move
crossover	operator used to produce new individuals by combining information from two or more parents
child/offspring	individual created by crossover/mutation
population	set of solutions
generation	iteration of an EA

Table 2.1.: Biological terms as they relate to ideas and concepts in EA [53, Chapter 11]

Encoding \Modeling

Encoding or modeling is one of the most crucial elements when approaching a problem. It refers to how the solutions/individuals of the optimization problem are represented. An unfavorable encoding may, at worst, lead to no solution/s being found or, at best, drastically increase the time it takes to arrive at an acceptable solution [53, Chapter 12]. Although many problems can be encoded in differing ways, certain encoding have been proven to be superior for specific types of problems [10]. Generally, one of the most important aspects is the relationship between the chosen encoding and the genetic operators. If the encoding has a benefit of reducing the search space but is incompatible with the genetic operators, the approach would seldom produce acceptable and desirable results. In cases such as these, having a less efficient encoding that works well with the genetic operators would result in better results overall. Having touched upon the dilemmas and difficulties with regards to encoding, there is no "cookie cutter" solution. However, there are general suggestions and rules of thumb one could follow:

- Similar phenotypes should be represented by similar genotypes.

- Similarly encoded candidate solutions should have a similar fitness.
- If possible, the search space should be closed under the used genetic operators [53, Chapter 12].

Initial Population

As one may observe from algorithm 1, initializing a population is the first step in any EA. Although various population initialization techniques have been employed in EAs [36], the initialization of the population is usually done randomly in order to ensure a wider spread of solutions and to mitigate premature convergence issues [25].

Fitness and Selection

Fitness and how individuals are selected lies at the very heart of EAs. Seeing as though the fitness values i.e. the objective functions are very problem specific, here greater emphasis will be given on the purpose of the selection mechanisms and how they relate to the ideas covered by this thesis. The basic idea behind selection is that individuals that have better fitness values have a higher chance of reproduction. How strongly these individuals are preferred for producing offspring is referred to as selective pressure. Greater selective pressure means that even small differences in fitness values between solutions cause differing chances of procreation. A certain level of selection pressure is needed for the proper use of EAs, seeing that without it, it is more akin to a random search. Additionally, if the selection pressure is too high, we will have a fast convergence on certain individuals without having had the opportunity to explore the search space in a sufficient capacity. A good approach is to start off with lower selection pressure to cover more areas of the search space and then focus more on "zeroing in" on the right solution/s [53, Chapter 12].

In the literature there exist a great number of selection mechanisms with varying degrees of uses, advantages and disadvantages. However, here only Tournament selection [48] will be presented as it is the only selection mechanism employed in the approach covered in this thesis.

The standard tournament selection randomly samples k individuals with replacement from the current population of size n into a tournament of size k.

The idea is that individuals compete in these tournaments, where the individual who wins gets to be part of the mating pool and is involved in generating an offspring for the next generation. Given that each tournament selects one winner, n number of tournaments need to be carried out in order to select all individuals for the next generation [24]. The size of the tournament k may be used as mechanism to control the selective pressure. Smaller tournament sizes means that there is a greater chance for weaker individuals to win out and participate in the mating pool. Understandably, higher values for k means that the selective pressure is greater, as the individuals with the best fitness values will dominate most tournaments [53, Chapter 12]. A implementation of a binary tournament selection algorithm is presented in algorithm 2.

Algorithm 2: Binary Tournament Selection [48]

Data: The population $P = \{\vec{x^{(1)}}, ..., \vec{x^{(k)}}\}$,
The tournament size $S \in \{1, 2, 3, ..., k\}$
Result: population after selection $P\prime$

$P\prime \leftarrow \emptyset$ `// initialize an empty mating pool`
for $i \leftarrow k$ **do**
    $G \leftarrow s$ `// randomly chosen individuals from P`
    $g \leftarrow G\prime$ `// select the best individuals from G`
    $P\prime \leftarrow P\prime \cup \{g\}$ `// get set of winning solutions`
end
return $P\prime$ `// population of individuals for the mating pool`

The tournament selection is perhaps the most widely used selection mechanism. Its popularity is due to many factors including:

- Its selection pressure can be adjusted easily.
- It is simple to code, efficient for both non-parallel and parallel architecture [49].
- It has time complexity $O(n)$ [30].
- It does not require sorting the whole population first [40].

These factors have greatly contributed in the prominence of the tournament selection as one of the most common selection mechanisms used in EAs [24]. However, one notable issue arises when individuals are sampled with replacement, thus making it possible to have the same individual sampled multiple times, i.e. the multi-sampled issue [67]. In the paper Xie et al. [67] endeavored to get a greater understanding of the multi-sampled issue. In their analysis they concluded that for common tournament sizes of 4 or less, it is not expected to see any duplicate individuals in anything except when dealing with very small populations. Even for tournaments of size 7, it is not expected to see duplicates when the population size is greater then two hundred. Thus, one can conclude that the multi-sampled issue rarely occurs and is not of great concern, should the population size be big enough. Another wider known issue is the notsampled issue, where, due to the random nature of the process the individuals with the best fitness have a chance to never be sampled and therefore never participate in the tournaments [32]. This problem has been addressed by Sokolv and Whitley [58] with their introduction of an unbiased tournament selection method which forces the sampling of each individual at least once. This work employees the initial and base approach for its purposes.

Genetic Operators

Genetic operators are applied to a certain part of the individuals in a generation in the hopes of creating new, more favorable individuals. These genetic operators can be categorised based on the number of individuals involved, thus we have:

- Mutation or variation operators, where only one parent is involved.
- Crossover operators, when we have two parents involved.
- Recombination operators, where more than two parents participate.

An important facet of genetic operators is that they preserve the closed search space recommendation outlined previously. This is especially important to encoded permutation problems such as the traveling salesman problem [16] [71], where the genetic operators should be able to preserve this permutation property of the encoding [53, Chapter 12]. Given the depth of this topic and the limited scope of this work, a focus on crossover operators will be given, especially as how they relate to the main ideas of this thesis. The best known and simplest crossover operator is the one-point crossover. As defined in Computational Intelligence by Kruse, Mostaghim et al. [53] the one-point crossover is when "a cut point is chosen at random and the gene sequences on one side of the cut point are exchanged between the two (parent) chromosomes". A more general formulation can be a n-point crossover where we have n number of cuts. The one-point crossover can be understood in a visual capacity from figure 2.3. Simulated binary crossover (SBX) is a version

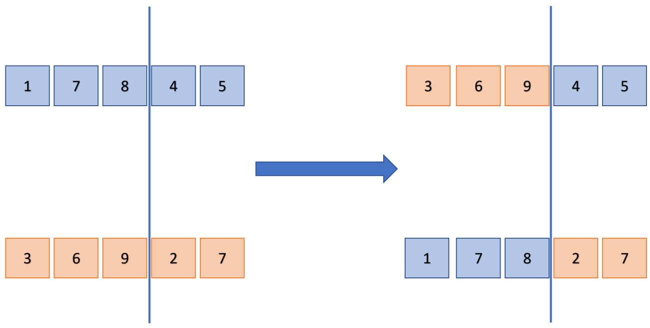


Figure 2.3.: Simple representation of a one-point crossover

of the one-point crossover for a vector of real numbers, as is the case in this thesis [20] [2]. The procedure of computing the offspring $x_i^{(1,t+1)}$ and $x_i^{(2,t+1)}$ from parents $x_i^{(1,t)}$ and $x_i^{(2,t)}$ using SBX, as represented in Agrawal and Deb [2] is:

1. A spread factor β_i is defined as the ratio of the absolute difference in offspring values of the parents:

$$\beta_i = \left| \frac{x_i^{(2,t+1)} - x_i^{(2,t+1)}}{x_i^{(2,t)} - x_i^{(1,t)}} \right| \tag{2.2}$$

2. A random value u between 0 and 1 is generated.

3. From a specified probability distribution function, β_{qi} is calculated so that the area under the curve from 0 to β_{qi} matches the number u. The probability distribution used to create an offspring is:

$$P(\beta_i) = \begin{cases} 0.5(\eta_c + 1)\beta_i^{\eta_c} & \text{if} \quad \beta_i \leq 1 \\ 0.5(\eta_c + 1)\frac{1}{\beta_i^{\eta_c+2}} & \text{otherwise} \end{cases} \tag{2.3}$$

4. The two offspring $x_i^{(1,t+1)}$ and $x_i^{(2,t+1)}$ are generated based on the following equations:

$$\begin{aligned} x_i^{(1,t+1)} &= 0.5[(1+\beta_{qi})x_i^{(1,t)} + (1-\beta_{qi})x_i^{(2,t)}] \\ x_i^{(2,t+1)} &= 0.5[(1-\beta_{qi})x_i^{(1,t)} + (1+\beta_{qi})x_i^{(2,t)}] \end{aligned} \tag{2.4}$$

The SBX has two distinct properties:

- The difference between the offspring is in proportion to the parent solutions i.e. closer parents produce closer offspring.
- Near-parent solutions are monotonically more likely to be chosen as offspring than solutions distant from the parents [20].

The first property of SBX can be modified using the distribution index η_c. Greater values of this distribution index gives a higher probability for creating solutions that are nearer to their parents, while a low value gives a higher probability for creating solutions that are further away from their parents. Another interesting characteristic of the SBX is that in the beginning we have a random search: children are very far from parents, as the parents are far from each other. However, at the end by achieving a certain convergence, we would have local search: parents are very close to each other and this leads to a local search by the children. This is a very interesting property of SBX that seems to be quite useful [20].

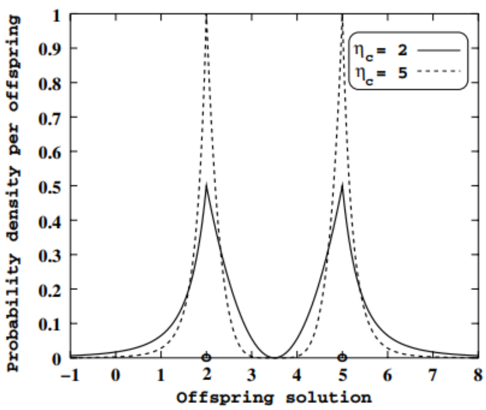


Figure 2.4.: The probability density function for creating offspring under an SBX [20]

Apart from the SBX, a mutation operator is also employed. For the purposes of this work the polynomial mutation operator will be used [19]. The probability of mutation is based on a small mutation probability p_m. Should a variable mutate, its value is changed to that of a neighbouring value. The selection of the neighbouring value is based on a polynomial probability distribution, characterised by having its mean at the current value and its distribution is based on a distribution index n. To execute a mutation a perturbance factor δ is also necessitated, such that:

$$\delta = \frac{c - p}{\triangle_{max}}, \tag{2.5}$$

where $\triangle_{max}$ represents the maximal allowed perturbance, p is the parent value and c is the mutated value. The mutated value is calculated using the perturbance factor δ:

$$P(\delta) = 0.5(n + 1)(1 - |\delta|)^n \tag{2.6}$$

The aforementioned distribution ranges $\delta \in (-1, 1)$. To create a mutated value c, one first generates a random number $u \in (0, 1)$. This number can then be used to calculate the perturbance factor δ.

$$\bar{\delta} = \begin{cases} (2u)^{\frac{1}{n+1}} - 1 & \text{if} \quad u < 0.5 \\ 1 - [2(1 - u)]^{\frac{1}{n+1}} & \text{if} \quad u \geq 0.5 \end{cases} \tag{2.7}$$

The mutated value is calculated as:

$$c = p + \bar{\delta}\triangle_{max} \tag{2.8}$$

Termination conditions

Stopping criteria similar to the encoding have been overshadowed by the more interesting elements of EAs. However, they are integral parts in the proper use of EAs. Traditionally, there have been three common stopping criteria:

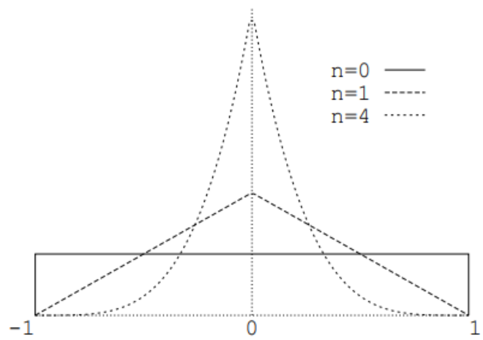


Figure 2.5.: Probability distribution for creating a mutated value for continuous variables [19]

- When a specified number of generations is reached.
- When a specified number of evaluations is reached.
- The chance of achieving significant changes in the next generations is excessively low [54].

An additional stopping criteria that is widely used is the maximal time budget criteria ,where the algorithm runs for a predetermined amount of time and returns the final solution/s. [29]. Additional stopping criteria have been used, based on:

- Derived termination criteria
- Operator-based termination criteria
- Cluster-based termination criteria
- Performance indicator criteria
- Progress indicator criteria [29]

For the purposes of this thesis a more traditional and simple stopping criteria was used, that of specified number of evaluations.

2.3. Multi-Objective Evolutionary Algorithms

In literature there a certain algorithms which have gained prominence both as adequate algorithms to solve real world problems and as benchmarks for other algorithms. These include the likes of SPEA-II introduced by Zitzler et al. [68], ϵ-MOEA introduced by Deb et al. [27], MOAE/D introduced by Zhang and Li [43]. However, as mentioned in the introduction, this thesis makes use of another well known algorithm, the NSGA-II algorithm introduced by Deb et al. [15]. The algorithm served as an improvement of its predecessor, the NSGA algorithm [60]. NSGA-II has a better time complexity of $O(mn^2)$ compared to NSGA's $O(mn^3)$, where m is the number of objectives and n the number of individuals. Additionally, the newer version has an improved mechanism to ensure better diversity. Being the backbone of the approach, the NSGA-II algorithm will be expanded upon in greater detail.

The basic premise of both NSGA and NSGA-II is the non-dominated sorting mechanism, where the algorithm derives its name from. This mechanism uses two simple ideas:

- Divide the population into several fronts.
- Select individuals from the best fronts one after the other i.e. only the solutions in the best fronts that can fit the defined population can survive [15] [53, Chapter 13].

Fast non-dominated sorting (FNDS), used in NSGA-II, computes a domination count n_p and a set of solutions S_p for each solution $\vec{x}^{(p)}$. The domination count of a solution refers to the the number of solutions which dominate the solution $\vec{x}$. The set of solutions S_p contains the solutions are dominated by the solution . After determining both n_p and S_p, solutions will with a domination count of zero are the first one added to the front F_1. Afterwards, for each member in F_1, we iterate through its set of solutions S_p and reduce the domination count for each solution in the set by one. Thus, solutions which were dominated only by the solutions contained in F_1 now themselves have a domination count of zero and are added to the front F_2. This procedure is then executed until all the solutions are assigned to a front [15] [53, Chapter 13].

Given the limited size of the population and the possible varying size of the fronts, should a front be unable to entirely fit in the population, a selection mechanism, called crowding distance, is used in order to determine which solutions should be included. Crowding distance is a measure of the density of the area (in the objective space) around the objective vector of the solution. The crowding distance is a rough indicator telling us how crowded the area (in the objective space) around the objective vector of the solution is. It is calculated by computing the normalized distances of each solution to its neighbouring solutions (in the objective space). This distance for each of the l solutions in F is stored in $D = [d_1, ..., d_l]$. Solutions with higher crowding distances are preferred as it means they are in less crowded areas and therefore easier for DMs to differentiate from other solutions. This ease of differentiation from other salutations is the main driving point behind sufficient diversity metrics and the desire to have greater diversity in the final set of solutions [15] [53, Chapter 13].

Having covered the underlying mechanisms we can now define the NSGA-II algorithm proper. NSGA-II combines non-dominated sorting and crowding distance to rank solutions into one algorithm. The NSGA-II algorithms starts off by initialising a certain, predetermined number, of solutions. As with many other EAs, it then uses tournament selection and SBX in order to generate a number of offspring $Q(t)$. The offspring are combined with the previous generation of solutions, which always contains the best individuals found so far. This new combined population $R(t)$ is however double the intended size. With non-dominated sorting solutions are ranked into different fronts F_i, The fronts are then added to the mating pool starting from the best ranked one. Should a front not be able to entirely fit due to a restriction on the population size, crowding distance is used to sort out which solutions should be added to the mating pool. The NSGA-II algorithm is shown, in pseudocode, in algorithm 3 [53, Chapter 13] [15].

The NSGA-II algorithm has, as many approaches, its own advantages and disadvantages. The introduction of a crowding distance mechanic to have greater diversity is one of the advantages of this approach, as it aids in greater differentiation of the solutions and therefore helping DMs make better decisions. One of the notable downsides is that this very diversity mechanic could cause the algorithm to lose upon convergence. Namely in later stages non-dominated and Pareto-optimal solutions could lose its place in the final population to non-dominated but also non Pareto-optimal solutions, which boasts better crowding distance. However, NGSA-II still seems to be a indispensable workhorse in the multi-objective optimization toolkit [15] [18, Chapter 6].

Algorithm 3: NSGA-II [15]

$t \leftarrow 0$ `// initialize generation counter`
$initializeP(t)$ `// create population with` μ `individuals`
$Q(t) \leftarrow \emptyset$ `// create empty offspring set`
while *termination criterion not reached* **do**
    $evaluateP(t)$ `// evaluate population based on fitness functions`
    $F_{P(t)} \leftarrow FNDS(P(t))$ `// Sort the population into front using FNDS`
    $Q(t) \leftarrow Offspring(P(t), F_{P(t)})$ `// produce offspring based on front evaluation, tournament selection, SBX and polynomial mutation`
    $evaluateQ(t)$ `// evaluate the offspring based on fitness functions`
    $R(t) \leftarrow Q(t) \cup P(t)$ `// combine parent and offspring population`
    $F \leftarrow FNDS(R(t))$ `// sort the joint population into fronts using FNDS` $F = \{F_1, F_2, ..., F_i\}$
    $P(t+1) \leftarrow \emptyset \quad \text{and} \quad i \leftarrow 1$
    while $|P(t)| + |F_i| \leq \mu$ `// do this until parent population is filled`
     do
        $P(t+1) \leftarrow P(t) \cup F[i]$ `// add the sorted solutions to the parent population starting with the best front and continuing from there`
        $i \leftarrow i + 1$ `// increase` i `to attempt to add the next front to the population`
    end
    $CD(F_i)$ `// calculate crowding distance of solutions in the front` F_i
    $Sort(F_i, CD)$ `// sort` F_i `using crowding distance`
    $P(t+1) \leftarrow P(t) \cup F_i[1 : (\mu - |P(t+1)|)]$ `// add the sorted solutions to the parent population`
    $t \leftarrow t + 1$ `// increment generation counter counter by 1`
end
return *non-dominated individuals from* $P(t)$

2.4. Evaluation Metrics for Multi-Objective Evolutionary Algorithms

In more general terms the approaches devised to tackle multi-objective optimization problems are broadly evaluated based on:

- How close the discovered non-dominated solutions are in terms of the Pareto front
- How diverse are the discovered non-dominated solutions

These two criteria are more commonly refered to as convergence and diversity [15, Chapter 9]. These two criteria are the main evaluation concepts concerning multi-objective evolutionary algorithms and ideally are both satisfied to a sufficient extent [42]. To evaluate these two characteristics many evaluation metrics have been established, measuring convergence, diversity or both. Here for the purposes of this thesis, we will be making use of all three.

Convergence based metric

In terms of convergence, Generational distance (GD) metric will be employed [64]. The premise behind this metric is to measure the average distance from the final set of non-dominated solutions to a previously known and established true Pareto front. Mathematically, GD can be defined:

$$GD = \frac{(\sum_{i=1}^{|S|} d_i^p)^{\frac{1}{\rho}}}{|S|}, \tag{2.9}$$

where $|S|$ is the number of non-dominated solutions and d_i^p is the euclidean distance from the ith non-dominated solution to the closest solution from the true Pareto front; typically the variable ρ is taken as 2. Understandably, the goal is to minimize the distance i.e. lower values for the metrics are preferred. The problem with this method is that it requires knowing the true Pareto front in advance. This issue has been tackled wit the 7-point method where generating equidistantly spaced points along each objective between the minimum and maximum possible values are generated and subsequently used to approximate the GD values [57].

Diversity based metric

For the diversity metric, a metric similar to GD will be used, namely, Inverted Generation Distance (IGD). IGD measures the average distance from the true Pareto front to the set of non-dominated solutions [13]. Mathematically, IGD is defined:

$$IGD = \frac{(\sum_{i=1}^{|P|} d_i^p)^{\frac{1}{\rho}}}{|P|}, \tag{2.10}$$

where P is the true Pareto front and d_i^p is the euclidean distance from the ith solution from the true Pareto front to the closest non-dominated solution. Here as well, ρ is commonly defined as 2. Like the GD metric, smaller values mean better diversity. The difference between GD and IGD can be visually seen on figure 2.6

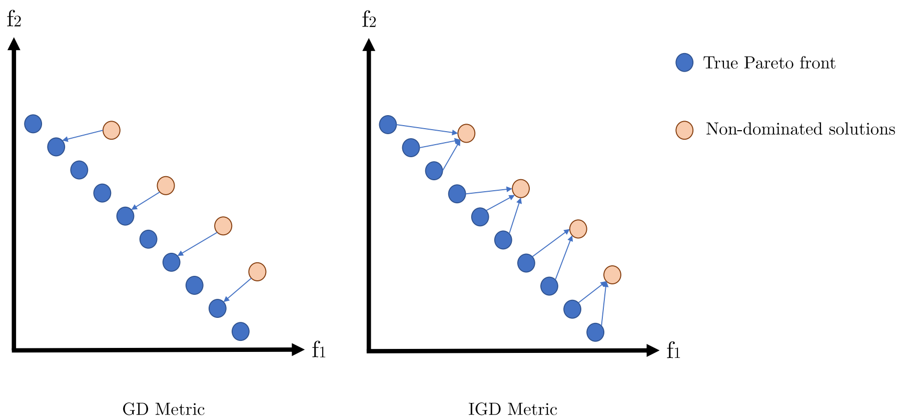


Figure 2.6.: GD and IGD metric

Convergence and Diversity based metrics

For the metric that combines both convergence and diversity the Hypervolume (HV) will be used [70]. It calculates the volume (in the objective space) covered by members of the found non-dominated set of solutions S. For each solution

$i \in S$, a hypercube is constructed with reference point W and the solution i as the diagonal corners of the hypercube [18, Chapter 9] [70]. Thereafter, a union of all HVs is calculated as follows:

$$HV = \bigcup_{i=1}^{|S|} v_i \tag{2.11}$$

The reference point W can usually be found by constructing a vector of worst objective function values. However, attention should be placed on the placement of this point as it can greatly affect the behaviour of the metric. This is one of the disadvantages associated with HV [18, Chapter 9]. For HV, unlike aforementioned GD and IGD metrics, higher values are desirable. A visual rendition of HV can be observed on image 2.7. Computing the HV, especially

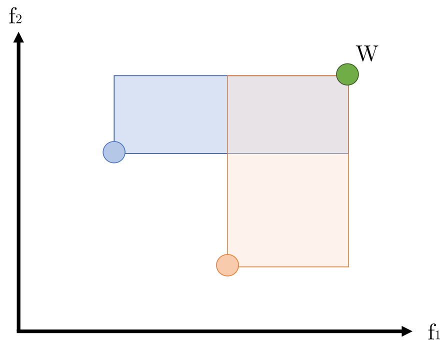


Figure 2.7.: HV metric

for higher dimensional problems can be quite difficult, due to its high computational complexity, which was proven by Bringmann and Friedric to be $\#P$-hard [28]. Given its complexity there have been a number of algorithms devised in order to more efficiently calculate this metric. For the purposes of this thesis the method of faster HV-based search using Monte Carlo sampling will be employed. The main idea behind this approach is based on the assumption that the exact HV indicator values are not vital. What is of greater importance is whether the contribution towards the HV metric of one individual is larger than that of another individual. To this end, samples of objective vectors are randomly drawn and the proportion of objective vectors that are solely dominated by a specific individual represents an estimate for the HV contribution of this individual. In their paper Ehrgott at al. show that a considerable amount of computational resources can be saved with the new approach in comparison to using algorithms for exact HV computation [21].

2.5. Fairness and Gain

The concept of fairness and gain and their use in a upper additional layer or "level" of a problem, as described by the 2020 Dagstuhl report [63], is the main theoretical idea that the approach in this thesis builds upon. The key concepts of the report is to employ the idea of PfPs, mainly inspired from Risto et al[55], as well as employ the two new additional objectives of fairness and gain, based on similarity [63].

Within the report certain established conditions are stipulated in order to use the proposed methods. As stated before the DMs put forward PfPs, consisting of their preferred values, within objective space. How the DMs initially formulate the preference point was attempted to be answered within the confines of this thesis, however the approaches and concepts examined proved of little universal use. The report outlines certain brainstorming methods but for the purposes of this thesis the assumption will be that the DMs can more or less ascertain their preference and formulate it as a vector of values [63].

At the heart of the report is the concept of this upper level with the two objectives fairness and gain. The basis of modeling gain and fairness is the notion of losses with respect to the ideal preferred solution a DM may obtain, termed Pareto regret [39]. In the case that we define the loss by means of the euclidean distance to preference vectors, the Pareto regret can be computed as:

$$PR_j(x) = \sum_{i=1}^{m}(f_i(x) - r_j), \tag{2.12}$$

where j is the index of the DM, $j = 1, .., d$, m is the number of objectives, x is the solution selected and r is the preference point of DM j. Following, we will introduce concepts derived from the notion of Pareto regret with respect to the DMs PfPs. The average Pareto regret of a solution x is defined as the average of the Pareto regrets of all DM:

$$APR(x) = \frac{1}{d}\sum_{i=1}^{d}(PR_i(x)) \tag{2.13}$$

Inequality in Pareto Regret of a solution x is defined as:

$$IPR(x) = \sum_{i=1}^{d} |PR_i(x) - APR(x)| \tag{2.14}$$

The aforementioned APR is defined as gain and IPR as fairness. Given the formulation of these objectives, we wish to minimize both of these objectives [63]. Thus, the intent is to use the upper level objectives in order to modify the behaviour of the overall optimization process. For the purposes of this thesis this will involve a systematic removal of the dominated solutions with regards to the this fairness and gain level, from the population.

2.6. Scalable Test Problems for Evolutionary Multi-Objective Optimization

Given that the focus of this work is more towards the understanding how the gain and fairness concepts from the 2020 Dagstuhl report [63] behave, the underlying problem used for the testing will be a scalable test problem, widely used in the literature, namely DTLZ2. DTLZ2 is described in the paper by Deb et al. [1] and is an already established problem, especially as it relates to the NSGA-II algorithm. DTLZ2 mathematically is described as follows:

$$\begin{aligned}
&min(f_1(\vec{x})) = (1 + g(x_m))(\cos(x_1\pi/2) ... \cos(x_m 2\pi/2) \cos(x_m 1\pi/2), \\
&min(f_2(\vec{x})) = (1 + g(x_m))(\cos(x_1\pi/2) ... \cos(x_{m-2} 2\pi/2) \sin(x_{m-1} 1\pi/2), \\
&min(f_3(\vec{x})) = (1 + g(x_m))(\cos(x_1\pi/2) ... \cos(x_{m-2} 2\pi/2), \\
&... \\
&min(f_m(\vec{x})) = (1 + g(x_m))(\sin(x_1\pi/2), \\
&\text{with} \\
&g(x_m) = \sum_{x_i \in x_m} (x_i - 0.5)^2, \\
&0 \leq x_i \leq 1, \text{ for } i = 1, 2, ..., n
\end{aligned}$$

Where $\vec{x}$ is a vector constructed with $k = n - m + 1$ variables. The Pareto-optimal solutions correspond to $x_i = 0.5$ for all $x_i \in x_m$ and all objective function values must satisfy $\sum_{i=1}^{m} f_i^2 = 1$. The DTLZ2 as tackeld by the NSGA-II algorithm, for 3 objectives is represented visualy on figure 2.8

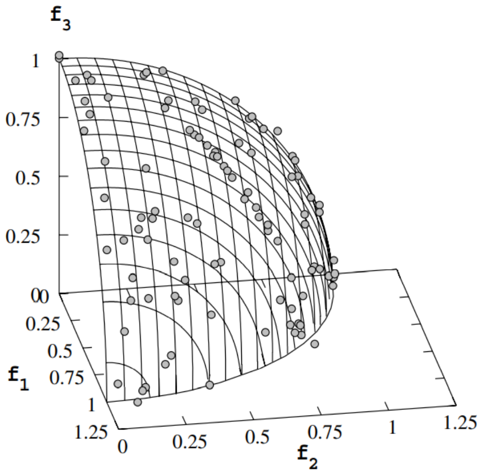


Figure 2.8.: The NSGA-II Population on Test Problem DTLZ2 [1]

3. Related Work

The concept of fairness has always been relevant, however, one can notice a trend in recent years in how this ideal has been woven into certain computational intelligence concepts. The majority of scholarly works which make use of fairness have predominantly been focused on fairness in terms of accurate representation or in terms of underlying biases against certain segments of the population. Such works include the likes of the Amazon recruiting AI that turned out to heavily favor male applicants [14], racial discrimination debate about the COMPAS recidivism risk assessment tool used by U.S. courts [23] as well as exploring biases in training data [46] [7]. However noble and noteworthy, this type of fairness is not really in the same vein as the fairness that serves as a driving mechanism in this thesis i.e. although same in spirit, quite different in many other aspects. Indeed, when one examines the directions and the nature of the proposed approach it is inevitably linked to the more established uses of evolutionary multi-objective optimisation, that of optimal resource allocation and more reasonable and fair decision making. These two concepts could be defined as the most tangible and understandable aspects of fairness which shape our daily lives to a considerable extent [34].

This promptly brings us to a very interesting and somewhat controversial claim that, at the time of writing, there really exists no other concept or idea that is comparable to the one outlined in the 2020 Dagstuhl [63] report and subsequently explored in the thesis. However, the nature and mechanisms of the approach also do not exist in a vacuum and various other aspects of it can be seen in differing methods and approaches. Of these, some employ multi-objective optimisation with a more generalized concept of fairness, other make use of fairness with regard to solving a specific problem and others still employ certain aspects of group decision making or a use of preference/reference points, in varying capacities.

3.1. Multi-Objective Optimisation and Fairness

In the work of Tia et al [62], a framework is designed to solve a negotiation problem with multiple objectives and multiple DMs. The approach is divided into two stages, where initially every DM attempts to optimize the objectives on their own. Afterwards, the fair allocation approach is employed based on Sperner's lemma [59], where every DM chooses his/her gain from the given resource allocation scheme based on his/her own preference, subject to the total quantity of the resources available and by the agreement of all other DMs [62].

In the paper by Limmer et al [44]. the authors tackle the problem of how to regulate the price of electricity at electric vehicle charging stations in a way that most people do not experience too much of a price difference within a certain time frame i.e. they defined the unfairness as the difference in price in time interval i and all of the previous intervals. Based on their experimental findings they concluded that too much difference in price in a certain time frame caused a great deal of unfairness in terms of the price the consumer pays and resulted in declining and unhappy customer base. Additionally, a focus just on fairness, understandably, had a detrimental effect on the profit. The paper makes use of a multi-objective approach using a modified version of NSGA-II, such that one of the objectives is fairness, while the other is profit. They have ascertained that by considering the fairness as a second objective, besides the profit and selecting the solution with the highest expected profit, the fairness can be significantly increased without too much of an impact on the profit. [44].

A search based approach to fairness analysis in requirement assignments to aid negotiation, mediation and decision making by Finkelstein et al. [69], the idea revolves around balancing requirements for certain features between the customers. Thus, contrary to the proposed approach that uses real values and can deal with continuous objective functions, the primary focus is on whether or not a specific feature is present i.e. it deals with a binary encoding. In the paper the authors make use of the NSGA-II algorithm in order to optimise a multi-criteria problem where the objective functions are derived based on the number, the value and the cost of the requirements fulfilled for each customer [69]. This work is particularly interesting and of note seeing as it could be modified and adapted to further expand the concept in this thesis and enable it to also tackle problems which are better modeled with a binary encoding.

In Illustration of fairness in evolutionary multi-objective optimization by Friedrich, Horoba and Neumann fairness is defined based on the concept introduced by Laumanns et al.[41]. They based it on the number of offspring a solution has created. In the paper by Laumanns et al.[41] they have claimed that having a more equalised balance between the number of offspring from all of the individuals of the population can lead to favorable results [52]. They conclude in their work that algorithms typically favor certain regions which are problematic for simple plateau functions which cannot be optimized without fairness or with fairness in the objective space, but with an approach which tackles fairness in the decision space. The approach in this work is based on establishing a certain area of interest with the use of PfPs, however, the paper by Friedrich, Horoba and Neumann [52] alludes that a restrictive nature can be detrimental when dealing with some optimisation functions.

3.2. Multi-Objective Optimisation and Preference Points

In large part the use of scalarization functions have been employed in order to convert a multi-objective optimisation problem into a single objective one. However, Kaisa Miettinen and Marko M. Mäkelä present in their paper on scalarizing functions in multi-objective optimization present fifteen scalarization function which could be implemented in various multi-objective optimisation problems [51]. Their work focus on reference point-based scalarizing functions which additionally expand the opportunities for DMs to express their preferences. It is easily seen that this paper influenced the main ideas behind the ones in the 2020 Dagstuhl report. In the paper however, Miettinen and Mäkelä additionally marry these preference based concepts with certain classification and weighted ideas. Although the concepts are similar, the one of pure preference of the DMs as one Preference point (PfP) is interestingly absent from the paper in favor of more complex scalarizing preference based concepts.

In terms of PfPs Lahdelma, Miettinen and Salminen [55] introduced the Ref-SMAA method for supporting discrete multi-objective decision making involving many DMs. Here, the aspirational goals of the DMs are represented as reference points in the reference point space and the preferences are represented by achievement functions. Additionally, they argue that that these reference points are easier for DMs to conceive and understand and that they are superior to weight vectors, given it is really difficult to give a more objective and accurate information on the weights themselves. Their presented method additionally can also easily handle problems where the criteria values are not precisely known. In this case the uncertain criteria are represented by stochastic distributions. The approach makes use of a PfP space and on the computation of a reference acceptability index in this space. It also uses a central reference point for each alternative, corresponding to the typical reference point of a DM preferring that alternative. Thus, it is able to identify good compromising solutions in a situation with multiple conflicting DMs [55]. Within the confines of this work we see again very similar concepts and ideas as in this thesis. The use of preference/reference points and a separate space where these points are additionally employed. However here we have additionally complexities in terms of the reference point space which are different to the ones from the 2020 Dagstuhl report. Based on the concept in the paper, one can generate new PfPs and then use them along with the DMs original references to devise new reference points and calculate the aforementioned index and central reference point. This novel idea of possible changing the beginning reference points in an attempt to better the overall process has been additionally incorporated into this thesis.

Mohammadi, N. Omidvar and Li [50] proposed an approach of using user-preference based points in tandem with an EA, that relies on decomposition strategies to convert a multi-objective problem into a set of single-objective problems. They also implement the PfPs in a way to focus on key areas of interest. They show that by using their approach they are able to achieve adequate solutions in a computationally more effective fashion and overcome additional drawbacks caused by domination. The algorithm R-MEAD was evaluated using two decomposition approaches, namely the weighted-sum and the Tchebychef approach [47]. Specifically, their proposed algorithm has a faster convergence as compared to MOEAD [43], especially when the number of objectives was higher. Here, we see PfPs mixed with additional scalarising functions in an effort to both combat the problems of domination and give greater importance to a specific area of interest.

Concerning reference points and their uses, the paper by Deb and Sundar [17], offers very interesting use cases. Specifically, Deb and Sundar use PfPs to isolate areas of interest, thus making it easier for the DM to reach a conclusion based on preferences. They additionally specify that a DM may specify more then one preference and finding solutions within these preferences is of greater importance to the DM. Furthermore, Deb and Sundar also show how these methods might be used in order to tackle problems with a great number of objectives (ten in the case of the paper) [17]. This scholarly work has striking similarity on the proposed approach, from its use of PfPs to its focus on areas of interests. Interestingly, it also specifies in its conclusion that future attempts should be made in the direction of focusing and defining these areas of interests and possibly employing an interactive process to fine tune them, so that the DMs can obtain better and satisfactory solutions in a faster and more efficient manner. Ideas one can find within the confines of this thesis.

The work of Branke and Deb [33] proposed a modified and controllable biases sharing approach, where by specifying a reference direction (or a linear utility function), a set of Pareto-optimal solutions near the best solution of the utility function could be ascertained. The implementation consisted of projecting all solutions on to the linear hyper-plane, where crowding distance values were calculated based on the ratio of the distances of neighbouring solutions in the original objective space as well as the projected hyper-plane. Therefore, solutions which lie on a plane parallel to the specified hyper-plane would have a comparatively large crowding distance and therefore would be preferred. It was shown that when employing this approach it converged near to the optimal solution to the utility function for some two and three-objective optimization problems. The approach required the users to specify the reference direction and a parameter which influenced the desired degree of diversity. In this approach we again see PfPs as well as a mention of an external plane, similar in some regards to the fairness and gain level discussed in the 2020 Dagstuhl report [63]. Indeed it also presented an interesting way of tackling the problem of diversity. A diversity approach based on this was additionally considered for the concept within this thesis, however it was difficult to adapt and therefore left out.

In a similar work by Branke et al. [56] we are introduced to the guided multi-objective EA (G-MOEA). This approach makes use of user preferences which were then used to modify the definition of dominance. Specifically, it allowed the DMs to specify for each pair of objectives acceptable trade-offs. Thus, one DM could specify that certain gains in one objectives were worth sacrifices in another. Although the idea works well for two objectives and was well utilized for distributed computing uses, having to provide all pair-wise information in a problem with a higher number of objectives seems difficult. One can notice that Branke and Deb have had a significant impact on this aspect of multi-objective optimisation, especially since the former was a part of the authors behind the 2020 Dagstuhl report [63]. However in this paper we are again confronted with very worthwhile concepts bottlenecked by the human DMs and their limitations. In this thesis attempts were therefore made to somewhat minimize the effect of this drawback.

3.3. Multi-Objective Optimisation and Many Decision Makers

The work of Bechikh et al. [6] focuses on a complete system of decision making with multiple DMs. Although not uncommon to see similar approaches of this nature, this one employs an interesting set of ideas in its own right. In the paper we are presented with an agent-based negotiation support system to aggregate the conflicting preferences of the DMs before the beginning of a optimisation process. This negotiation approach aids the DMs to adjust and amend their preferences through a number of negotiation rounds. The system output is a set of social preferences which will be used subsequently in a preference-based evolutionary multi-objective optimisation algorithm, such as the one presented in this work. The paper by Bechikh et al. [6] is presented as it offers some additional ways in order to come up with the intended preferences of the DMs in the initial step of the approach. What is also interesting in this approach is that it implements elements of "fuzziness" where the DMs can also specify additional regions of acceptance. In their paper Bechikh et al. [6] also bring up once again the question of areas of interest and also prioritize the optimisation and solution seeking within these areas. In short, this paper has interesting ideas that could be implemented together with the ideas presented in this thesis as a way to initially have a truly fair and reasonable set of points in order to facilitate better results.

In conclusion, there are many noteworthy works beyond the confines of these, which, for time and size considerations, were not presented. The assortment of related works here aims to shed light on the various interesting aspects embodied in the ideas covered by this work and to explore various fields in the realm of multi-objective optimisation. It also helps demonstrate the unique nature of this approach and the difficulty in finding a suitable alternative for comparison, a great testament to the creative minds behind the 2020 Dagstuhl report [63].

4. Methodology and Approach

Within the confines of this chapter are the main ideas of this thesis, along with additional details with regards to their exact structure. The concepts of fairness and gain along with their exact implementation are defined, followed by the introduction of a adaptive mechanism to overcome some of the drawbacks from the initial approach. The chapter concludes with a presentation on a concept to capture the power dynamics of the DMs as well as a primer on how one may apply these ideas in a practical decision making scenario.

4.1. NSGA-II for Teams

Understandably the exact structural details of the concept underlined of the 2020 Dagstuhl report [63] will be presented first, specifically how one determines the fairness and gain values for a solution. As in section 2.5 the fairness and gain are derived from the Pareto Regret:

$$PR_j(x) = \sum_{i=1}^{m} d(f_i(x), r_j), \tag{4.1}$$

where j is the index of the DM, $j = 1, .., d$, m is the number of objectives, x is the solution selected, r is the PfP of DM j and d is an euclidean similarity function. Following, the two objectives of Average Pareto Regret and Inequality in Pareto Regret are computed as:

$$APR(x) = \frac{1}{d} \sum_{i=1}^{d} (PR_i(x)) \tag{4.2}$$

Inequality in Pareto Regret of a solution x is defined as:

$$IPR(x) = \sum_{i=1}^{d} |PR_i(x) - APR(x)| \tag{4.3}$$

The aforementioned APR is defined as Gain and IPR as Fairness. Given the formulation of these objectives, we wish to minimize both of these objectives [63]. Having again specified the new objective functions concerning fairness, what will follow is a presentation of the steps and working mechanisms behind the modified NSGA-II for Teams Algorithm.

The basic premise is to have a initial multi-objective problem along with additional user preferences, combined into a $P_{m \times n}$ matrix, where m is the number of DMs involved or their preferences and n is the number of objectives of the base problem. This initial problem, is considered to be solved by the NSGA-II algorithm [15]. On top of this, in a sense, is another algorithm, which uses fast non-dominated sorting on the newly calculated fairness and gain objectives for each solution. Afterwards, inspired by the nature of the NSGA-II algorithms, solutions are added to a fair population $\mathcal{F}_p$. The fair population is created by adding all the solutions of a front, starting from the best front, until $\mathcal{F}_p$ has at least as many solutions as the initial population in the algorithm, an idea similar to that of NSGA-II and their front by front inclusion into the mating pool. The fair population $\mathcal{F}_p$ is then the one passed onto the initial problem solving algorithm so that it may continue solving the original problem with a population based on these, fairer solutions. This filtering process is done each generation, so the solutions get modified in an alternating fashion, being modified by both the attempt to solve the underlying optimisation problem as well as the fairness and gain objectives. The pseudocode for this specific filtering in terms of fairness is shown in algorithm 4 and the newly defined NSGA-II for Teams is presented in algorithm 5, with the changes marked in red.

Algorithm 4: Fairness Filter

Data: $F = \{\vec{f}(\vec{x}^{(1)}), ..., \vec{f}(\vec{x}^{(l)})\}$ the objective vectors for all l individuals, $P_{m \times n}$ matrix, representing the DMs preferences, inital population size N

Result: Fair population $\mathcal{F}_p$

$\mathcal{F}_p \leftarrow \emptyset$ `// initialize the filtered population as an empty set`

$PR_{l \times m} \leftarrow zeros(l, m)$ `// initialize an empty` $l \times m$ `matrix to hold all the Pareto regret values`

$GAIN_{l \times 1} \leftarrow zeros(l, 1)$ `// initialize an empty vector for the Gain values`

$FAIRNESS_{l \times 1} \leftarrow zeros(l, 1)$ `// initialize an empty vector for the Fairness values`

foreach *objective vector* i **do**

  foreach *preference* j `// this refers to the number of PfPs or DMs,` m

  do

    $PR_{ij} = Pregret(F[i], P_{j*})$ `// calculate the Pareto regret for a objective vector in terms of the specified preference, according to equation 2.12.`

  end

  $GAIN_i \leftarrow APR(PR_{i*}, m)$`// calculate the Average Pareto regret for a objective vector, according to equation 4.2.`

  $FAIRNESS_i \leftarrow IPR(PR_{i*}, GAIN_i)$`// calculate the Inequality in Pareto Regret for a objective vector, according to equation 4.3.`

end

$\mathcal{F}_{obj} \leftarrow \{\vec{f}^{(1)}_{obj}, \vec{f}^{(2)}_{obj} ..., \vec{f}^{(l)}_{obj}\}$ `//` $\vec{f}^{(i)}_{obj} = \{GAIN(f(\vec{x}^{(i)}), FAIRNESS(\vec{x}^{(i)})\}$.

$\mathcal{F} \leftarrow FNDS(F, \mathcal{F}_{obj})$ `// Use FNDS to sort the set of objective vectors` F `based on their corresponding fairness and gain objectives from` $\mathcal{F}_{obj}$. $\mathcal{F}$ `is the set of fronts, which consist of the objective vectors,` $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3 ... \mathcal{F}_l\}$

$i \leftarrow 1$

while $|\mathcal{F}_p| \leq N$ **do**

  $\mathcal{F}_p \leftarrow \mathcal{F}_p \bigcup \mathcal{F}[i]$ `// add the sorted solutions to the fair population starting with the best front and continuing from there`

  $i \leftarrow i + 1$ `// increment` i

end

return $\mathcal{F}_p$

Algorithm 5: NSGA-II for Teams

$t \leftarrow 0$ // initialize generation counter
$initialiseP(t)$ // create population with μ individuals
$Q(t) \leftarrow \emptyset$ // create empty offspring set
while *termination criterion not reached* **do**
  $evaluateP(t)$ // evaluate population based on fitness functions
  $P(t) \leftarrow \text{Fairness filter}(P(t))$ // filter the population in terms of fairness
  $F_{P(t)} \leftarrow FNDS(P(t))$ // Sort the population into front using FNDS
  $Q(t) \leftarrow Offspring(P(t), F_{P(t)})$ // produce offspring based on front evaluation, tournament selection, SBX and polynomial mutation
  $evaluateQ(t)$ // evaluate the offspring based on fitness functions
  $R(t) \leftarrow Q(t) \cup P(t)$ // combine parent and offspring population
  $F \leftarrow FNDS(R(t))$ // sort the joint population into fronts using FNDS $F = \{F_1, F_2, ..., F_i\}$
  $P(t+1) \leftarrow \emptyset \quad \text{and} \quad i \leftarrow 1$
  while $|P(t)| + |F_i| \leq \mu$ // do this until parent population is filled
  do
    $P(t+1) \leftarrow P(t) \cup F[i]$ // add the sorted solutions to the parent population starting with the best front and continuing from there
    $i \leftarrow i + 1$ // increase i to attempt to add the next front to the population
  end
  $CD(F_i)$ // calculate crowding distance of solutions in the front F_i
  $Sort(F_i, CD)$ // sort F_i using crowding distance
  $P(t+1) \leftarrow P(t) \cup F_i[1 : (\mu - |P(t+1)|)]$ // add the sorted solutions to the parent population
  $t \leftarrow t + 1$ // increment generation counter counter by 1
end
return *non-dominated individuals from* $P(t)$

Due to the nature of the approach and its dependence on the initial preferences of the DMs as well as the added element of fairness, it seems necessary to create a metric that will enable us to better understand the various nuances within the examined concept and to have a single value in order to evaluate the fairness of otherwise hard to understand situations. To that end, a metric of fairness was devised which draws its basis on the already established concepts behind the GD and IGD metrics. This metric measures the average distances in the second level, between the final population and the origin [0,0]. Given that the origin is the ideal point for the objectives of the second level, that the front is unknown to us as well as the fact that we are not concerned with the diversity in terms of fairness and gain, this seems like an adequate measure of how "fair" a given set of final solutions are. Thus the fairness value of a solution in the final population can be defined as:

$$FV(x) = d(O, x), \tag{4.4}$$

where $d(O, x)$ is an euclidean distance similarity function between the coordinate origin [0,0] and the fairness and gain values of the solution x . Following from this, the average of all the fairness values from the final population can be defined as:

$$AFV = \frac{1}{m}\sum_{i=1}^{m} FV_i, \tag{4.5}$$

where m is the size of the final population i.e. the number of solutions and FV_i is the fairness value for the ith solution. Given the nature of the metric and that it is based on the idea behind the GD and IGD metrics, it is only fitting that this is a metric which we wish to minimize.

4.2. Adaptive NSGA-II for Teams

Upon preliminary testing certain shortcomings of the initial concept were observed. Specifically, should the placement of the preferences be further distant from the front (in a negative sense) and the number of preferences sufficiently large, the effect from the fairness level will cause issues in terms of convergence.

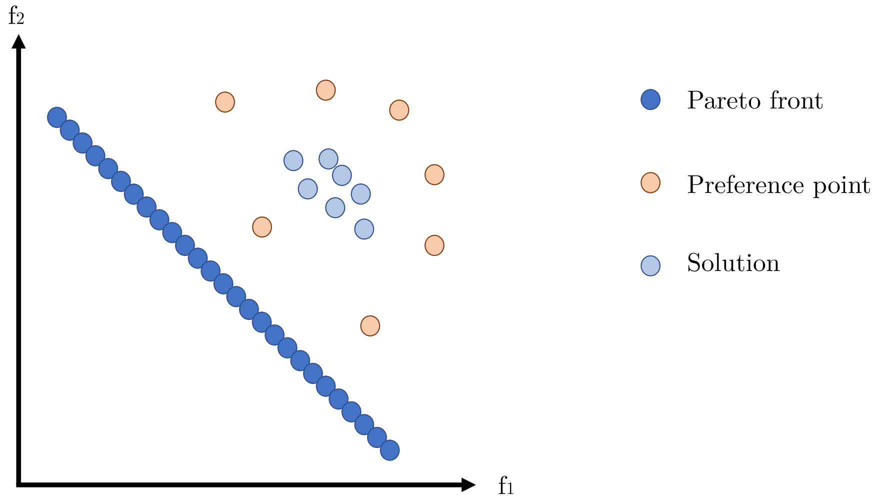


Figure 4.1.: Illustration of a Problematic Scenario Regarding the NSGA-II for Teams Algorithm

This can be visually observed in image 4.1. Although the solutions are close to the front, due to the nature of the approach and the positioning of the DMs these solutions will not move towards the front any further. This is problematic as the goal is still optimisation of the initial base problem and having the possibility of this occurring seems like a undesirable downside that must be addressed.

To this end a additional step of repositioning or moving the PfPs was included. The repositioning of the PfPs can be done at anytime, however as we will see certain times appear to be better then others. Additionally, the repositioning can be either sanctioned by the DM or be left parley to the algorithm, thus making this a interactive algorithm that can be automated in a way. The repositioning of the PfPs follows a simple idea. At a certain time we calculate the distances from the non-dominated solutions to the PfP of every DM. We then find the closest non-dominated solution to each of the PfPs. Should the closest solution be dominating the DMs PfP, it becomes that DMs new PfP. This simple idea will enable the PfPs to be pushed towards the front, thus helping with the convergence and enabling a better set of solutions to be presented to the DMs. Moreover, this should also have a positive effect on the fairness metric as DMs that are further away due to domination will inevitably get closer to the front and each other. A more detailed representation in pseudocode is given in algorithm 6 and the exact implementation within the new algorithm is given in algorithm 7.

Algorithm 6: Preference Repositioning

```
Data: F = {f(x^(1)), ..., f(x^(l))} the objective vectors for all l individuals,
      P_{m×n} matrix, representing the DMs preferences, n_v evaluation
      number, can also be a list of numbers
Result: Reference point matrix P_{m×n}
D_{k×m} ← zeros(l, m)
F_nd ← FNDS(F) // Using FNDS, get the non-dominated solutions
F_nd ← F_nd[F_1 : F_m] // Incorporate more then the first front to avoid
   convergence to a single point, should one solution be non-dominating
if evaluationnumber = n_v then
    foreach objective vector i do
        foreach preference j // this refers to the number of PfPs or DMs,
           m
         do
            D_{i,j} ← d(F_nd[i], P_{j*}) // calculate the distances from the ith
               non-dominated solution vector x^(i) to every PfPs, with the
               use of euclidean distance similarity function d
        end
    end
    foreach preference j // this refers to the number of PfPs or DMs, m
     do
        r_j^new ← F[min(D_{*j})]// identify the new possible PfPs based on
           the shortest distance
        if r_j^new ≺ P_{j*} // check the found solutions whether they dominate
           the previous PfPs
         then
            P_{j*} ← r_j^new // change the PfPs to the newfound one
        end
    end
end
return updated P_{m×n} matrix
```

Algorithm 7: Adaptive NSGA-II for Teams

```
t ← 0 // initialize generation counter
initialiseP(t) // create population with μ individuals
Q(t) ← ∅ // create empty offspring set
while termination criterion not reached do
    P_{m×n} ← Preference repositioning(P_{m×n}) // check if there exist better reference points
    evaluateP(t) // evaluate population based on fitness functions
    P(t) ← Fairness filter(P(t), P_{m×n}) // filter the population in terms of fairness
    F_{P(t)} ← FNDS(P(t)) // Sort the population into front using FNDS
    Q(t) ← Offspring(P(t), F_{P(t)}) // produce offspring based on front evaluation, tournament selection, SBX and polynomial mutation
    evaluateQ(t) // evaluate the offspring based on fitness functions
    R(t) ← Q(t) ∪ P(t) // combine parent and offspring population
    F ← FNDS(R(t)) // sort the joint population into fronts using FNDS F = {F_1, F_2, ..., F_i}
    P(t+1) ← ∅  and  i ← 1
    while |P(t)| + |F_i| ≤ μ // do this until parent population is filled
     do
        P(t+1) ← P(t) ∪ F[i] // add the sorted solutions to the parent population starting with the best front and continuing from there
        i ← i+1 // increase i to attempt to add the next front to the population
    end
    CD(F_i) // calculate crowding distance of solutions in the front F_i
    Sort(F_i, CD) // sort F_i using crowding distance
    P(t+1) ← P(t) ∪ F_i[1 : (μ − |P(t+1)|)] // add the sorted solutions to the parent population
    t ← t+1 // increment generation counter counter by 1
end
return non-dominated individuals from P(t)
```

4.3. Some Decision Makers are Fairer than Others

Given the very nature of fairness and keeping in mind the situations that this approach may be used to tackle, a way to show the various influences and power dynamics of the DMs, seems warranted. In this regard several different approaches were tried and tested. These ideas focused on attempting to have weights attached to each DM. The naive approach was to weight the distances according to the DMs influence. It was assumed that greater weights would cause greater Pareto regret values and thus the solutions would need to move closer to the more influential DM in order to minimize this. This along with a great number of variants provided disappointing results, often overemphasizing even a slightly more influential DM or barely having any effect. Therefore, a filtering approach was developed that seems to give more satisfactory results.

The approach is based on creating a weighted centroid from all of the PfPs and their respective weights (when dealing with the improved approach, the final PfPs are used). Afterwards, the distance from all solutions to this centroid is calculated. Subsequently, based on the predefined number of solutions, agreed upon by the DMs, the solutions furthest away are removed until the desired number of solutions remains. In this regard the solutions remaining will be skewed more towards the more influential DMs. A visual illustration of this is given in figure 4.3. With regards to the actual weight for the preferences/DMs certain ideas were examined, however the works such as that of Baek and Prabhu [5] along with other works in the literature provide more information on the matter. Therefore, the actual weight assignment will be let up to the discretion of the DMs. Although not necessarily complex the pseudocode is provided in algorithm 8

This additional filter applied at the end enables control of the number of alternatives needed to be discussed, when all DMs are equal. Subsequently, it also is able to capture the power dynamics of the decision makes in a very elementary way. Should the alternative be far from the most influential of DMs it is promptly eliminated and not even put up for discussion. The idea is inspired from the organisation of the Roman senate during the time of the Roman republic where speaking order and topic selection was decided upon power and seniority within the political structure [11].

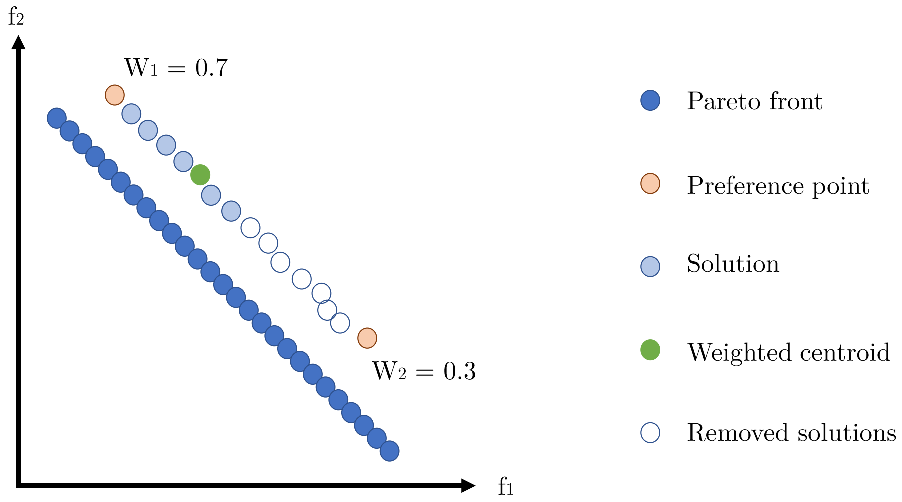


Figure 4.2.: Illustration of Weighted Filtering

With regards to the actual weights themselves, although left up to the discretion of the DMs it is nevertheless recommended that they be defined as such that their sum adds up to 1. Should the weights be derived in terms of other means that use a differing numbers scale, normalisation is advised.

Algorithm 8: Modified Influences

Data: $F = \{\vec{f}(\vec{x}^{(1)}), ..., \vec{f}(\vec{x}^{(l)})\}$ the objective vectors for all l individuals, $P_{m \times n}$ matrix , representing the DMs preferences, n_s number of solutions to remain, W weight matrix with the corresponding influence weights

Result: updated set of solutions $F = \{(\vec{x}^{(1)}), ..., (\vec{x}^{(l)})\}$

$F_{new} \leftarrow \emptyset$ `// initialize the new set of solutions as an empty set`

$C_{1\times n} \leftarrow zeros(1, n)$ `// initialize an empty vector for the centroid`

$D_{1\times n} \leftarrow zeros(l, 1)$ `// initialize an empty vector for the distances`

$C_{1\times n} \leftarrow$ the sum of all rows of $(\mathrm{P}_{m\times n} \circ \quad W)$ `// Following element wise matrix multiplication the resulting matrix is then summed up in terms of its rows so we end up with a single PfPs`

foreach *objective* i **do**

  $D_i \leftarrow d(\vec{x}^{(i)}, C_{1\times n})$ `// calculate the euclidean distance for each objective to the weighted centroid` $C_{1\times n}$

end

$F_s \leftarrow sort(F, D_i)$ `// sort the objective vectors according to the distance to the centroid`

$i \leftarrow 1$

while $/F/ < n_s$ **do**

  $F_{new} \leftarrow F_{new} \bigcup F[i]$ `// add the closest solutions`

  $i \leftarrow i + 1$ `// increment` i

end

return *updated set of solutions* $F_{new} = \{(\vec{x}^{(1)}), ..., (\vec{x}^{(l)})\}$

4.4. On Real World Application

The mechanisms described in the previous sections were envisioned to be components of a broader concept in order to support multiple DMs. This section will present how these ideas can be combined into a unified approach.

As can be observed on figure 4.3 the decision support process consist of several steps which are simple and easily explainable to the DMs. The process starts by submission of PfPs by the DMs. How these PfPs were derived is left up to the DMs and although some methods were explored in order to help the DMs come up with PfPs, this is beyond the scope of this thesis. Additionally, it is assumed that every DM submits only one PfP, however, a DM also may submit more multiple. This is a more advanced version and the DMs therefore need to additionally agree on how the influence weights would then be derived, should they be needed.

Furthermore, the DMs may specify if the preferences should be known to the other DMs or not. The obfuscation of the preferences may be useful when the DMs are leaders of different countries and wish to protect strategic information. The public display of PfPs may be useful in democracies as a way to show transparency and have more insight into the objectives of each of the political parties. Given the state of politics, should this method be used, the PfPs would still most likely remain hidden from public view.

After the preferences are given the algorithm may begin its process of searching for the optimal solutions. At a certain number of iterations or after every iteration, a change of the PfP can be attempted. Should new and better points be found the DM can be prompted if they wish to make the change. However, the DMs or some of them may also specify that they wish to accept all changes and automate this step. In this regard the concept can be either an interactive approach or it can be a priori approach, which grants greater flexibility. As soon as the algorithm finishes and delivers the final set of non-dominated solutions found, they can be filtered either in accordance with the influence of the DMs or to simply reduce the number of options discussed.

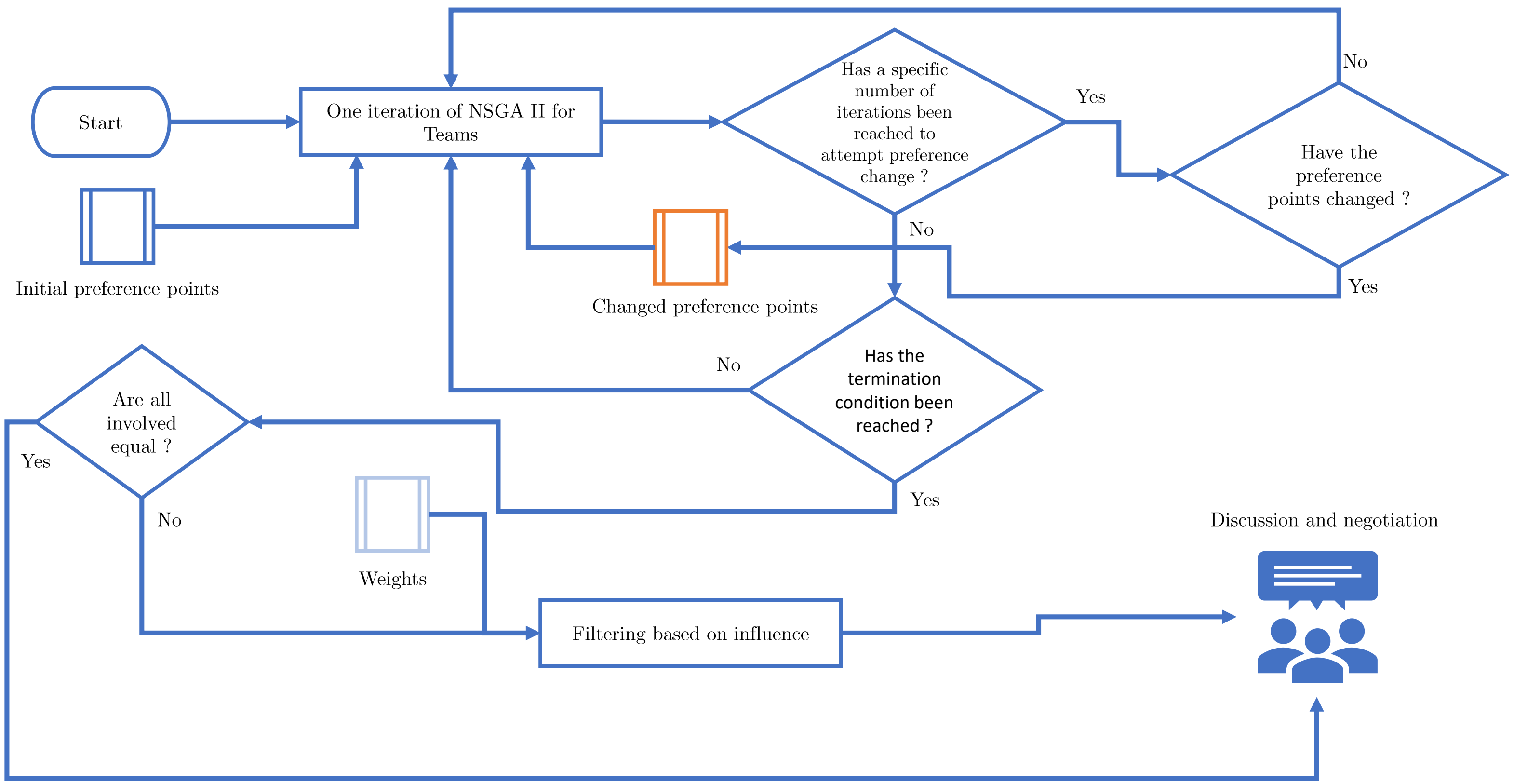


Figure 4.3.: Flowchart on Practical Implementation

5. Implementation and Experiment Design

This chapter serves to specify the implementation details of the proposed approach as well as to describe the structure and reasoning behind the experiments.

5.1. Implementation

The implementation was carried out in MATLAB R2018 version 9.4.0.813654 [45], specifically with the use of the PlatEMO platform for Evolutionary Multi-Objective Optimization version 2.6 [61].PlatEmo is a comprehensive software platform for researchers to properly benchmark existing algorithms and for practitioners to apply selected algorithms to solve real-world problems [61]. The basic sequence diagram of running a general multi-objective optimization algorithm by PlatEMO is given in figure 5.1.

Using the intuitive Graphical user interface (GUI) (shown in figure 5.2) users can additionally specify which performance indicator to calculate. The resulting mean and the standard deviation of the performance indicator value are shown. Furthermore, the best result in each row is shown in blue, and the Wilcoxon Rank-Sum test [66](with 5% significance level) result is labeled by the signs '+', '−' and '=', which indicate that the result is significantly better, significantly worst and statistically similar to the result in the control column, respectively. This way of presenting the data will also be shown throughout the result analysis as it is an efficient and convenient way to view the results of the experiments.

Although very versatile, the platform had to be additionally modified to serve the purposes of this thesis. To implement the necessary alterations, the already present NSGA-II algorithm included in PlatEmo, was augmented with the additional function in a similar vain to what is shown in the pseudocode examples in sections 4.1 and 4.2. Further alterations were made to the problem classes objects where an additional public property was added, that of the PfPs.

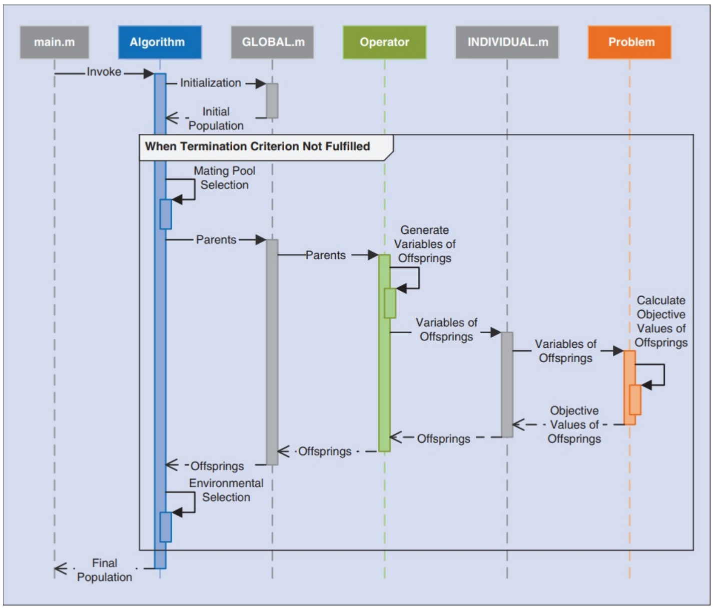


Figure 5.1.: PlatEMO Sequence Diagram of Running a General Multi-Objective Optimization Algorithm [61]

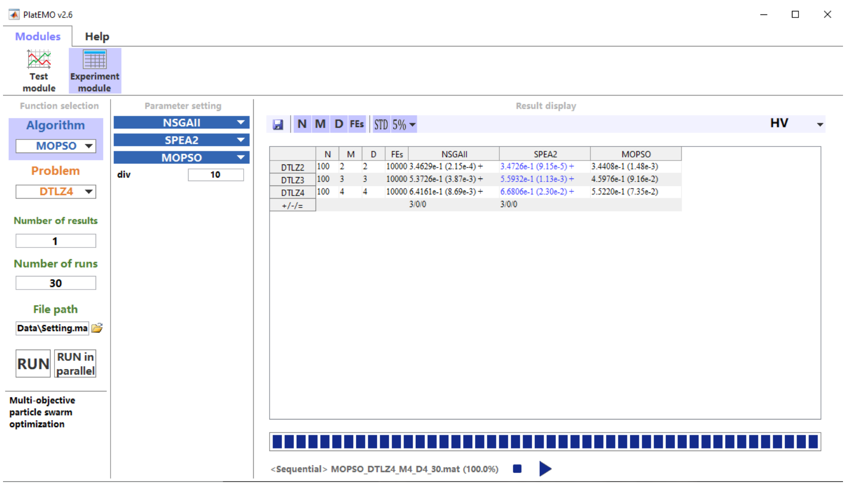


Figure 5.2.: PlatEMO GUI

5.2. Experiment Design

Given the strong relationship between the positioning of a PfP and the final outcome, it proves only prudent to try to provide answers and clarity on the relationship between the two. Specifically, greater understating of how the various positions of the PfPs to one another influence the final population of solutions is needed. To that end three general categories of experiments were conducted :

- Exploration of the behaviour of the NSGA-II for Teams algorithm.
- Comparison between the NSGA-II for Teams and Adaptive NSGA-II for Teams algorithms.
- Examining the behaviour of the Adaptive NSGA-II for Teams algorithm when faced with complex cases.

In all of these experiments, the building blocks consist of individual cases. A case is a specific arrangement of points, in an attempt to replicate specific cases that may arise during real-world implementation. These cases correspond to a single problem with regards to the PlatEMO platform. Most cases will be examined by conducting a set of 50 runs with 10,000 Evaluations per run (FEs) and an initial Population size (N) of 100. The cases that deviate from this will be promptly noted. The Number of objectives (M), Number of decision variables (D) and DMs will be specified on a case by case basis.A group of these cases that is called a scenario and it is used in order to compare the results of similar cases. In order to focus on the most important aspects the exact numerical values of the PfPs will be presented in the annex.

Exploration of the Behaviour of the NSGA-II for Teams Algorithm

The exploration of the various intricacies of the initially proposed concept will consist of examining the effects, as they relate to distances between the PfPs of the DMs, as well as their relative positioning with regards to domination. In order to simplify the analysis and not be overwhelmed by the many various cases that can occur, attention will be placed on problems with only two objectives. This allows for an easier visualisation of the PfPs and thus a better understanding of their influence on the final results. To that end five scenarios will be showcased, consisting of multiple different cases:

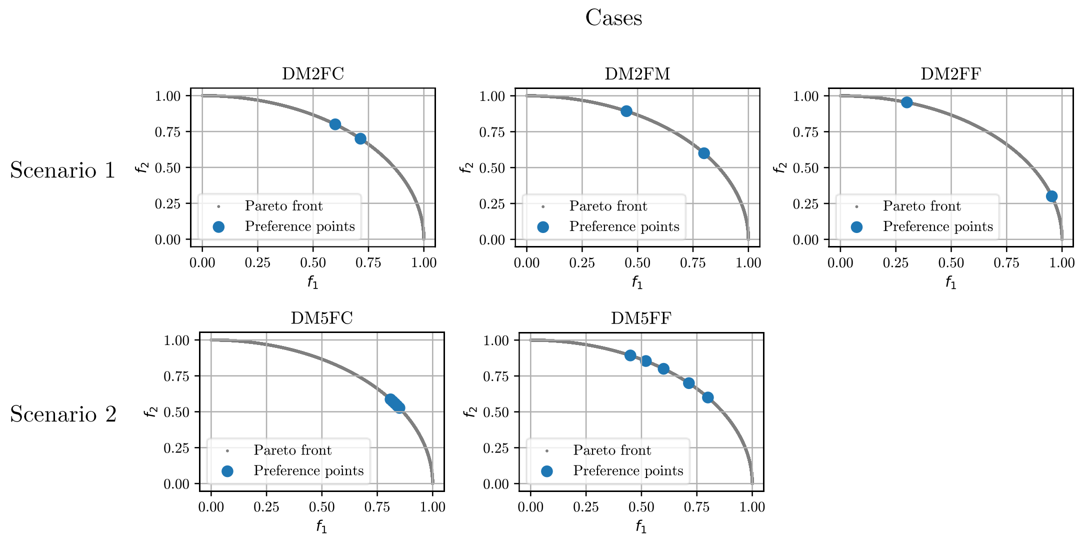


Figure 5.3.: Scenarios 1 and 2

- Scenario 1: 3 cases of 2 PfPs with varying levels of distance between them, with all the PfPs on the Pareto front.
- Scenario 2: 2 cases of 5 PfPs all on the Pareto front with different distances between the PfPs.
- Scenario 3: 3 cases of 2 PfPs with varying levels of distance between them, with 1 PfP on the Pareto front and the other one being dominated.
- Scenario 4: 2 cases of 5 PfPs, split into a case with 3 dominated PfPs and 2 dominated PfPs. The other PfPs lie on the Pareto front.
- Scenario 5: cases of 5 PfPs, split into a case with 4 dominated PfPs and 1 dominated PfP. The other PfPs are on the Pareto front.

The number of DMs has been arbitrarily capped at five DMs, as its seems like a reasonable number of DMs and a realistic number of people who could effectively negotiate and discuss without too much strain. The cases will be compared to the initial NSGA-II algorithm executed on a DTLZ2 problem

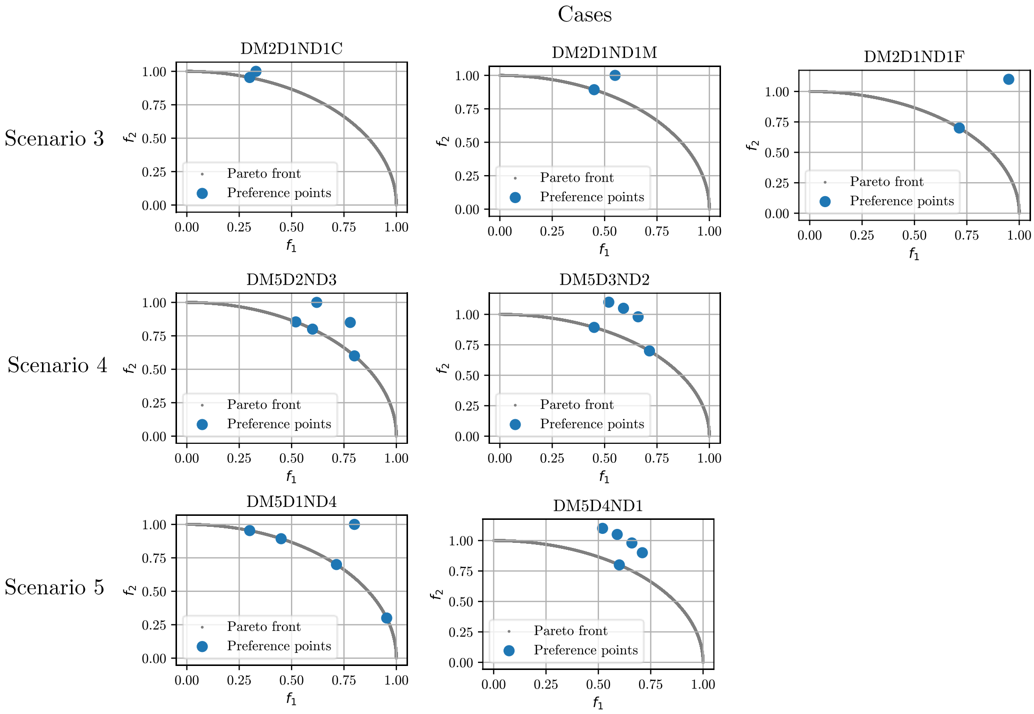


Figure 5.4.: Scenarios 3, 4 and 5

consisting of two objectives (M = 2) and two decisions variables (D = 2). The metrics employed will be the GD, IGD, HV (with a reference point at [1,1]) and the fairness metric (only applicable for the NSGA-II for Teams algorithm).

Comparison Between the NSGA-II for Teams and Adaptive NSGA-II for Teams Algorithms

Having discovered the possible negative effects on some cases and in attempt to rectify this a improved approach was suggested where the DMs could update their preferences in a logical manner. This set of experiments will only be conducted on the scenarios consisting of points that do not lie on the Pareto front and will be compared against the initial concept. The five times a repositioning attempted PfP would be made is represented in table 5.1. The scenarios will be compared to the initial algorithm executed on a the same testing scenarios.

The metrics employed will be the GD, IGD, HV (with a reference point at [1,1]), as well as the fairness metric. Given the experimental nature as well as the objectively beneficial decision to reposition ones PfPs it is assumed that all the DMs will accept the repostioning, even though they may choose not to do so when practically using the approach.

Notation	Meaning
F1	NSGA-II for Teams
FF1	Adaptive NSGA-II for Teams, move at 5000 evaluations
FF2	Adaptive NSGA-II for Teams, move at 5000 and 7500 evaluations
FF3	Adaptive NSGA-II for Teams, move at 5000, 7500 and 9370 evaluations
FFF	Adaptive NSGA-II for Teams, move after each generation

Table 5.1.: Variants for an attempted move

Examining the Behaviour of the Adaptive NSGA-II for Teams Algorithm when Faced with Complex Cases

Inspired by the paper by Deb and Sundar [17] the following experiments will be conducted by examining the effects that the new Adaptive NSGA-II for Teams algorithm when confronted with a considerable number of objectives and decision variables. The problem will again be DTLZ2 and can be organized into a broad group of two scenarios. The first one will serve as an exploration into the well known limits in terms of the concept of domination. It will thus be defined by various cases which all have six objective (M = 6) and 6 decision variables (D = 6). The cases have been organized in the following manner:

1. 5 PfPs, 4 on the Pareto front and 1 being dominated - M6DM5D1ND4.
2. 5 PfPs, 1 on the Pareto front and 4 being dominated - M6DM5D4ND1.
3. 5 PfPs, 3 on the Pareto front and 2 being dominated - M6DM5D2ND3.
4. 5 PfPs, 2 on the Pareto front and 3 being dominated - M6DM5D3ND2.

These second scenario is meant to push the algorithm to its limits. Examining the effects when we have ten objectives (M = 10) and ten decision variables (D = 10). This scenario will consist of two cases:

1. 5 PfPs, all on the front - M10DM5F.

2. 5 PfPs, randomly generated, where the value for each objective falls between [0,1] - M10DM5.

The metrics employed will be GD, IGD, HV (with a reference point at [1,1]). The goal here is to see whether the algorithm will exhibit the same behaviour, given more complex circumstances.

6. Results

This chapter covers the results from the experiments carried out as well as offering specific explanations to the outcomes observed.

6.1. Results Analysis of NSGA-II for Teams

Untangling the relationship between the placement of the reference points and the endmost results requires a detailed examinations into various different reference point scenarios. As stated before we will examine problems with two objectives in the initial level. This will enable us to visualise the reference point placement and thus draw better conclusions on the nature of their relationship to the behaviour of the algorithm as a whole.

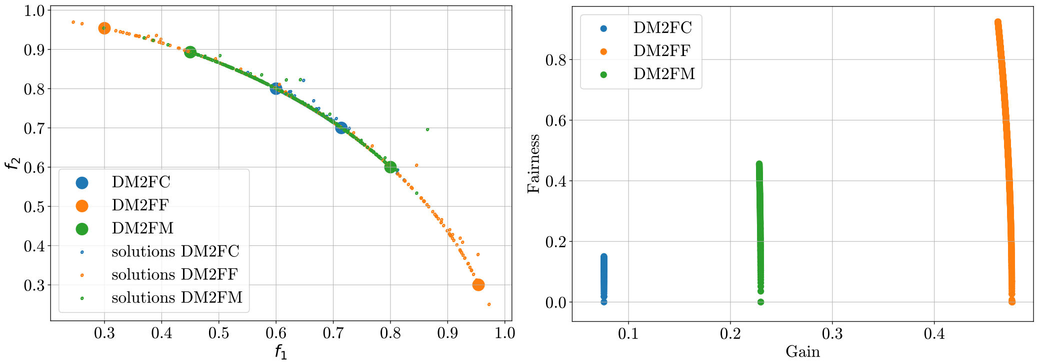
What will follow is an examination of various differently defined scenarios that were constructed in ordered to get a better understanding of the approach. The examination will encompass the five scenarios mentioned in the previous chapter. The analysis will consist of two plots, the first showcasing the positioning of the PfPs and one randomly chosen set of solutions from the 50 iterations carried out to showcase what the resulting set of solutions may look like. The second plot will consist of solutions, approximating the Pareto front of the fairness and gain level. This front was obtained by amalgamating all of the 50 runs carried out and by using FNDS, only showcasing the non-dominated solutions. This was done in an attempt to have a better understanding and to more easily showcase the Pareto front of the fairness and gain level. These plots will additionally be supplemented by a table containing the results as they relate to the GD, IGD, HV and fariness metrics. The data in the table will follow the same principle of data presentation outlined in the PlatEMO platform [61] i.e. the best result in each row is shown in blue, and the Wilcoxon Rank-Sum test (with 5% significance level) result is labeled by the signs '+', '−' and '=', which indicate that the result is significantly better, significantly worst and statistically similar to the result in the control column, respectively.

6.1.1. Examining the Effect that Distances Between the Preference Points has on the Final Solutions

The first two scenarios examined hope to shed light on the effect that distance between the reference points has. Drawing our attention on scenario 1 and examining the figure 6.1 one can see that the solutions tend to be concentrated between the PfPs, as expected. Examining the gain and fairness level, a peculiar shape of the fronts can be noted. This shape could be caused to the fact that we have two PfPs on the front which seems to be causing the solutions to gravitate in a narrower space, towards a central point between the PfPs. The accompanying table for scenario 1, depicts that the approach has a positive impact on GD values, surpassing those of the base NSGA-II algorithm, most likely caused by enabling the algorithm to focus more on a specific area of interest. An interesting observation is that the closest points do not have the best convergence values, implying that some amount of distance between the PfPs influences the convergence in a positive way. The reason behind this seems unclear, however, it might be caused by the very restrictive space coupled with the crowding distance mechanism, inherent to the NSGA-II algorithm. The diversity, showcased by IGD, seems to be greatly diminished, compared to the initial NSGA-II algorithm, an anticipated result, given the more restrictive nature of the approach. The IGD values seem to also be positively influenced by greater distances between the points, most likely due to the greater space made available to the solutions to spread over. In terms of HV values, it points that in terms of overall quality, the points furthest away from each other seem to be the best. Here, the modified algorithm, expectedly does worse then the NSGA-II algorithm. It would seem that the superior gain in terms of convergence does not really make up the shortcomings from diversity. Observing the fairness aspect of scenario 1, it seems that the fairness values of the cases where the DMs were closer translated to a better fairness value as opposed to the case where they were further apart.

Moving on to scenario 2 and focusing our attention on figure 6.2, we can again observe that the solutions tend to be gravitating towards a space enclosed by the PfPs. The two differing cases showcase that the solutions, as expected, try to find a central space between all of the PfPs. This is again illustrated in the gain and fairness levels, where a more crescent shape of the front can be seen. This shape is likely indicative of the greater area which the solutions could occupy and still be non-dominated in terms of fairness and gain. The table for scenario 2, shows that, once again, we have a positive impact on GD values, compared to the NSGA-II algorithm. Here the effects are again most likely caused by the algorithm's proclivity to focus on a constrained area of interest. The IGD values are once more inferior to the base algorithm, favoring the case with greater distance between the PfPs. The HV, as expected, is worse off than the initial algorithm, again favoring points further away from one another. Here, the fairness values, line up better with expectations, showing that the more closely positioned points have the best values.

In conclusion, the algorithm tends to restrict the solutions within an area outlined by the PfPs, as intended. The shape of the gain and fairness front seems largely indicative of the number of PfPs and the area they enclose as it relates to the true Pareto front. The fairness values seem to indicate that they are largely dependent on the position of the PfPs i.e. their distances to each other, with some minor influences by their relative position to the front and the size of the restricted area. In terms of convergence and diversity, the restrictiveness of the points, noticeably improves the GD values but causes expected worsening of the IGD values. This lack of diversity is also reflected in the HV values, which are underwhelming compared to those of the NSGA-II algorithm.



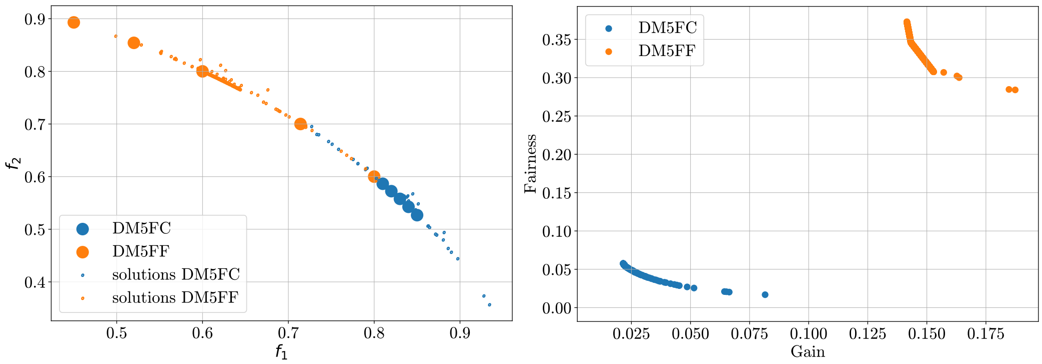
(a) PfPs, scenario 1

(b) Gain and fairness, scenario 1

Figure 6.1.: Scenario 1

Problem	N	M	D	FEs	F1	NSGAII
GD						
DM2FC	100	2	2	10000	9.7563e-6 (1.20e-5) +	8.9649e-5 (2.28e-5)
DM2FF	100	2	2	10000	1.1216e-5 (3.32e-6) +	9.5135e-5 (2.61e-5)
DM2FM	100	2	2	10000	5.9847e-6 (3.92e-7) +	8.9380e-5 (2.78e-5)
+/-/=					3/0/0	
IGD						
DM2FC	100	2	2	10000	3.2848e-1 (1.07e-1) -	5.2856e-3 (2.22e-4)
DM2FF	100	2	2	10000	8.0060e-2 (1.31e-4) -	5.2254e-3 (2.06e-4)
DM2FM	100	2	2	10000	2.3820e-1 (3.25e-4) -	5.3091e-3 (2.38e-4)
+/-/=					0/3/0	
HV						
DM2FC	100	2	2	10000	1.7799e-1 (5.43e-2) -	3.4648e-1 (2.50e-4)
DM2FF	100	2	2	10000	2.9158e-1 (9.13e-5) -	3.4648e-1 (1.82e-4)
DM2FM	100	2	2	10000	2.1890e-1 (1.51e-4) -	3.4644e-1 (2.28e-4)
+/-/=					0/3/0	
Fairness						
DM2FC	100	2	2	10000	1.24408e-1 (4.1e-3)	
DM2FF	100	2	2	10000	7.052e-1 (1.23e-2)	
DM2FM	100	2	2	10000	3.6490e-1 (5.291e-3)	

Table 6.1.: Metrics for scenario 1



(a) PfPs, scenario 2 (b) Gain and fairness, scenario 2

Figure 6.2.: Scenario 2

Problem	N	M	D	FEs	F1	NSGAII
GD						
DM5FF	100	2	2	10000	7.0966e-6 (7.23e-7) +	9.2121e-5 (2.40e-5)
DM5FC	100	2	2	10000	5.6058e-6 (2.04e-7) +	9.0107e-5 (1.74e-5)
+/-/=					2/0/0	
IGD						
DM5FF	100	2	2	10000	4.1148e-1 (8.69e-6) -	5.2146e-3 (2.38e-4)
DM5FC	100	2	2	10000	4.4516e-1 (1.15e-5) -	5.3157e-3 (2.45e-4)
+/-/=					0/2/0	
HV						
DM5FF	100	2	2	10000	1.3779e-1 (4.02e-6) -	3.4648e-1 (2.16e-4)
DM5FC	100	2	2	10000	1.2293e-1 (5.77e-6) -	3.4643e-1 (2.09e-4)
+/-/=					0/2/0	
Fairness						
DM5FC	100	2	2	10000	3.6321e-1 (6.817e-4)	
DM5FF	100	2	2	10000	5.9613e-1 (1.284e-4)	

Table 6.2.: Metrics for scenario 2

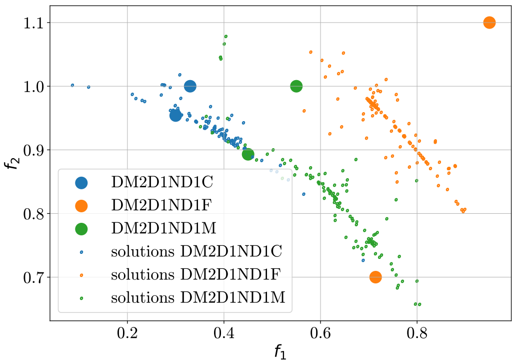
6.1.2. Examining the Effects that Differences in Terms of Domination has on the Final Solutions

Moving on from merely examining what effects distance has here we wish to also incorporate the fact that certain PfPs could also be dominating others. Beginning this analysis with scenario 3, we draw our attention to figure 6.3. This scenario serves as a transitional one, encapsulating both the influence of distance and domination. One aspect is immediately evident when examining the solutions they are noticeably further away from the front. This seems to be additionally compounded by the distance between the PfPs on and away from the Pareto front. This behavior was anticipated, as the points attempt to occupy an area between all PfPs. An interesting observation can be made whilst examining the case DM2D1ND1M, where the solutions seem to be closer to the front, than to the middle point between the points. This effect is likely caused by the influence of the base NSGA-II algorithm and its attempt to push the points towards the Pareto front of the initial problem. Observing the gain and fairness level, it seems that this front shape embodies both the crescent shape, similar to when we have multiple PfPs on the front, and the expected constrained shape, as seen by cases consisting of two PfPs. The resulting shape seems to be the result of the conflict between the push and pull from the base as well as the fairness and gain level. Examining the corresponding table 6.3 we can see that this unfavorable distancing from the Pareto front has negatively impacted the GD values in all the cases. The values are drastically worse than the ones from the NSGA-II algorithm, losing one of the favorable advantages that the approach had. The reduced values are in line with the observation from figure 6.3, where as the distances widen the GD values plummet more and more. The IGD values also exhibit the usual trend: growing better as the distances between PfP grow larger. This additionally confirms that the area available to the solutions by the PfPs is a key influencing factor to the diversity. The HV values, marred by the lack of convergence are significantly inferior to those of the NSGA-II algorithm. Finally the fairness values shows a well anticipated result, growing worse as the distances between the PfPs grows larger.

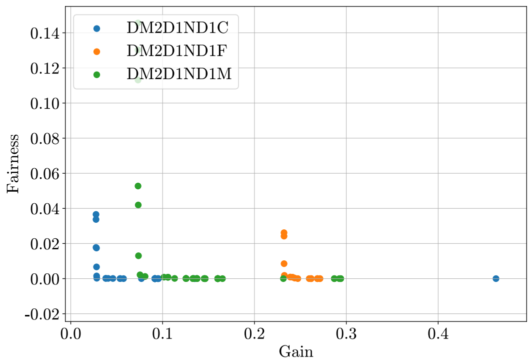
Continuing to scenario 4, we again begin by studying figure 6.4 to gain a more intuitive understanding of the scenario at hand. The solutions appear again negatively effected by the dominated PfPs. This undesirable effect seems to be additionally influenced by the ratio between the non-dominated and dominated PfPs. Drawing our attention to the gain and fairness front, we notice that the fronts have the more crescent shape, indicative of the greater area available to the solutions where they can be non-dominated in terms of gain and fairness. Continuing to the metrics presented in table 6.4, we see that the GD values have worsened compared to the NSGA-II algorithm, with the case DM5D2ND3 barely winning out. Concerning the diversity, it seems that again we have a worse result compared to the NSGA-II algorithm, with the case with less dominated PfPs having better values. The HV values as anticipated, again pail in comparison to the unaltered algorithm, where the case with fewer dominated points seems to be fairing better. The reason for this may be found in its noticeably better diversity values. A very interesting observation can be made whilst examining the fairness values. Here it is evident that the case DM5D3ND2 is fairer, a claim additionally backed up by examining the gain and fairness fronts. The cause can be found in the smaller overall area enclosed by the corresponding PfPs.

The final scenario, scenario 5, is intended to display the more extreme ratio between dominated and non-dominated PfPs, as can be seen in image 6.5. The drastic difference in convergence, due to the greater number of dominated PfPs, can also be evidenced by the GD values from table 6.5. The case with almost all points on the front seems to be exhibiting behaviors similar to those when all PfPs are on the Pareto front, besting the NSGA-II algorithm. Turning to the IGD values the DM5D4ND1 case is again better, most likely due to the greater distances between the PfPs. The same result can also be viewed in the HV values, where the better diversity and convergence values have translated to reasonable HV values. However, the NSGA-II algorithm seems to still be superior with regards to both IGD and HV values. Turning our attention towards the fairness values as well as the gain and fairness fronts, we can clearly see that the more close-knit case DM5D4ND1 has better values.

In summary the position of the PfPs, as it relates to domination, seems to greatly influence the convergence. The severity of this influence additionally seems to be dependent on the ratio between dominated and non-dominated points as well as the distance between them. In terms of fairness, this seems to have no effect as the values and fronts continue to reflect the distances between the PfPs, regardless of their relative positioning in terms of domination. In the examined scenarios it is eminently clear that the prescience of dominated points and their negative influence on the convergence must be addressed, given that the goal of this work is to provide solutions which are both fair and optimal.



(a) PfPs, scenario 3

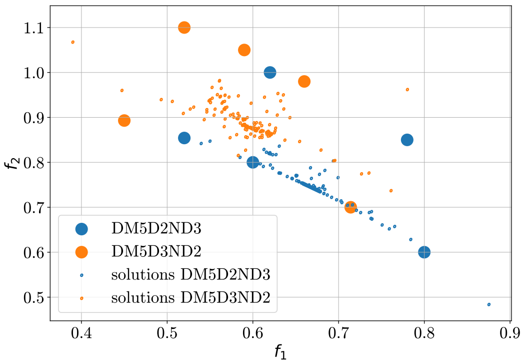


(b) Gain and fairness, scenario 3

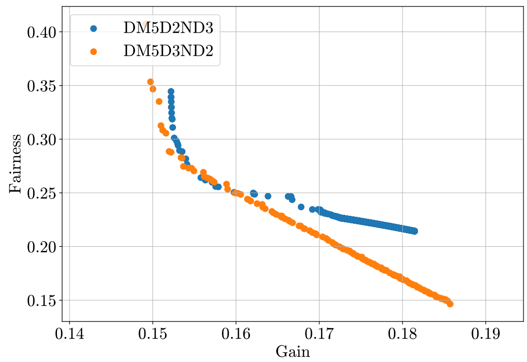
Figure 6.3.: Scenario 3

Problem	N	M	D	FEs	F1	NSGAII
GD						
DM2D1ND1C	100	2	2	10000	2.4251e-3 (3.04e-3) -	9.0013e-5 (1.99e-5)
DM2D1ND1F	100	2	2	10000	2.0630e-2 (1.20e-3) -	9.1934e-5 (2.74e-5)
DM2D1ND1M	100	2	2	10000	4.7796e-3 (1.34e-3) -	8.8846e-5 (2.21e-5)
+/-/=					0/3/0	
IGD						
DM2D1ND1C	100	2	2	10000	3.6162e-1 (8.66e-2) -	5.2552e-3 (2.36e-4)
DM2D1ND1F	100	2	2	10000	3.9429e-1 (4.49e-2) -	5.2699e-3 (2.34e-4)
DM2D1ND1M	100	2	2	10000	3.1424e-1 (4.47e-2) -	5.3077e-3 (2.14e-4)
+/-/=					0/3/0	
HV						
DM2D1ND1C	100	2	2	10000	1.5676e-1 (2.98e-2) -	3.4651e-1 (1.93e-4)
DM2D1ND1F	100	2	2	10000	1.1568e-1 (2.61e-2) -	3.4649e-1 (2.08e-4)
DM2D1ND1M	100	2	2	10000	1.6403e-1 (1.40e-2) -	3.4649e-1 (2.09e-4)
+/-/=					0/3/0	
Fairness						
DM2D1ND1C	100	2	2	10000	2.99e-2 (2.65e-2)	
DM2D1ND1F	100	2	2	10000	1.884e-1 (3.452e-2)	
DM2D1ND1M	100	2	2	10000	8.8652e-2 (5.8753e-2)	

Table 6.3.: Metrics for scenario 3



(a) PfPs, scenario 4

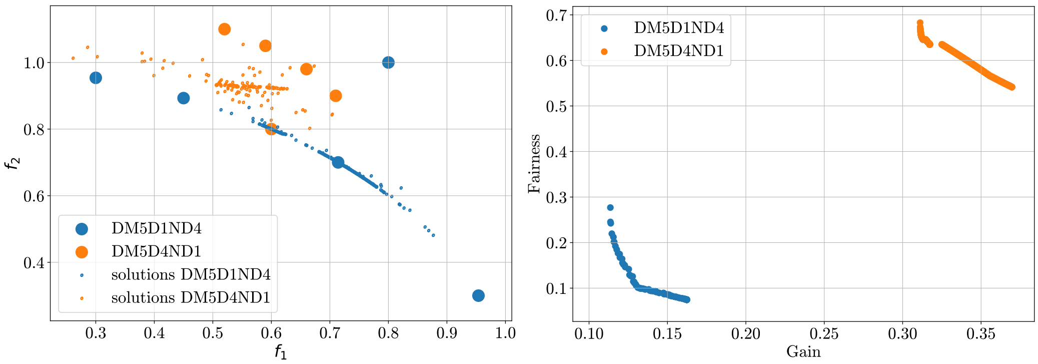


(b) Gain and fairness, scenario 4

Figure 6.4.: Scenario 4

Problem	N	M	D	FEs	F1	NSGAII
GD						
DM5D2ND3	100	2	2	10000	1.7115e+0 (1.08e-1) -	3.0568e-1 (2.88e-2)
DM5D3ND2	100	2	2	10000	1.7284e+0 (9.56e-2) -	3.1265e-1 (3.53e-2)
+/-/=					0/2/0	
IGD						
DM5D2ND3	100	2	2	10000	3.9277e-1 (2.07e-4) -	5.2322e-3 (2.09e-4)
DM5D3ND2	100	2	2	10000	4.3448e-1 (3.58e-3) -	5.3645e-3 (2.17e-4)
+/-/=					0/2/0	
HV						
DM5D2ND3	100	2	2	10000	1.4313e-1 (4.12e-4) -	3.4646e-1 (2.07e-4)
DM5D3ND2	100	2	2	10000	1.0717e-1 (2.44e-3) -	3.4642e-1 (2.69e-4)
+/-/=					0/2/0	
Fairness						
DM5D3ND2	100	2	2	10000	3.2835e-1 (1.752e-2)	
DM5D2ND3	100	2	2	10000	3.42184e-1 (4.544e-2)	

Table 6.4.: Metrics for scenario 4



(a) PfPs, scenario 5 (b) Gain and fairness, scenario 5

Figure 6.5.: Scenario 5

Problem	N	M	D	FEs	F1	NSGAII
GD						
DM5D1ND4	100	2	2	10000	3.0589e-5 (2.48e-5) +	9.4539e-5 (2.19e-5)
DM5D4ND1	100	2	2	10000	1.1479e-2 (5.67e-3) -	8.6382e-5 (2.70e-5)
+/-/=					1/1/0	
IGD						
DM5D1ND4	100	2	2	10000	3.0900e-1 (1.55e-4) -	5.2740e-3 (2.38e-4)
DM5D4ND1	100	2	2	10000	4.4944e-1 (9.55e-3) -	5.2372e-3 (1.82e-4)
+/-/=					0/2/0	
HV						
DM5D1ND4	100	2	2	10000	1.8523e-1 (1.21e-4) -	3.4645e-1 (2.35e-4)
DM5D4ND1	100	2	2	10000	9.4016e-2 (6.71e-3) -	3.4647e-1 (2.38e-4)
+/-/=					0/2/0	
Fairness						
DM5D1ND4	100	2	2	10000	8.5296e-1 (2.251e-3)	
DM5D4ND1	100	2	2	10000	1.59454e-1 (3.603e-2)	

Table 6.5.: Metrics for scenario 5

6.2. Results Analysis for the Adaptive NSGA-II for Teams Algorithm

In an effort to overcome the shortcomings of the initially concept an improvement was need. Namely, the problematic issue stemming from the positioning of the DMs and their influence on the convergence needs to be resolved. To this end an additional step of repositioning or moving the PfPs was included. Unlike the initial concept, the analysis will be focused on the GD and fairness metrics and how they behave. Here, we run the algorithm again for 10,000 evaluations just as we had before with the initial concept. The analysis consists of five different ways that the preference move can be accomplished (see chapter 5.2 for more details).

Examining table 6.6, which depicts the GD values for all the variants, one can see that the proposed augmentation seems to be working as expected. Additionally, out of all the variants, it would appear that attempting to change the PfP every generation is the best option. This is in accordance to initial expectations. It was speculated that attempting a change at specific intervals could be greatly influenced by plain luck. Furthermore, given that the changes were attempted after the half way point it is possible that the algorithm would have trouble with the convergence, given the more limited time. The attempted move after every generation seems like the best way as it should provide constant movement towards the front, which results in reaching the front sooner and having more chance to converge on the Pareto front. Observing table 6.7 shows us how the new algorithm reacts against the old NSGA-II algorithm. Here, it seems as though the improved approach is preforming as expected, beating out the original NSGA-II algorithm in every case.

Moving on to the examination of table 6.8 and examining the impact on the fairness, one can still observe improvement in all variants. Here, the the variant which attempts a change every generation, again seems to give the best results. The results of these experiments are likely due to the fact that the movement towards the front undoubtedly brings some of the PfPs together. However, as can be observed in the table, the consistency of this can be quite unpredictable as it depends greatly on random chance. Hence most variants have one or two cases where they preform poorly in comparison to the initial concept.

Problem	FFF	FF3	FF2	FF1	F1
DM2D1ND1C	4.3212e-5 (7.29e-5) +	1.0540e-4 (1.21e-4) +	9.4423e-5 (1.11e-4) +	1.3824e-4 (1.06e-4) +	2.4251e-3 (3.04e-3)
DM2D1ND1F	4.2100e-5 (1.16e-4) +	2.4770e-3 (1.66e-3) +	2.3748e-3 (1.49e-3) +	4.0100e-3 (2.70e-3) +	2.0630e-2 (1.20e-3)
DM2D1ND1M	3.1542e-5 (5.01e-5) +	4.1097e-4 (3.93e-4) +	6.5560e-4 (1.09e-3) +	1.3498e-3 (1.57e-3) +	4.7796e-3 (1.34e-3)
DM5D1ND4	1.5383e-5 (4.12e-6) +	1.4991e-5 (3.40e-6) +	1.4567e-5 (2.48e-6) +	1.5107e-5 (2.39e-6) =	3.0589e-5 (2.48e-5)
DM5D2ND3	2.7279e-5 (1.05e-4) +	1.2981e-5 (2.61e-6) +	1.2938e-5 (2.37e-6) +	7.7549e-5 (2.85e-4) +	6.5081e-4 (1.41e-4)
DM5D3ND2	1.1859e-5 (4.52e-6) +	9.7058e-3 (4.23e-3) =	9.6763e-3 (4.01e-3) =	9.8559e-3 (4.16e-3) =	9.8357e-3 (2.57e-3)
DM5D4ND1	2.2392e-5 (5.91e-5) +	5.2107e-3 (5.45e-3) +	4.3825e-3 (7.92e-3) +	8.2624e-3 (4.91e-3) +	1.1479e-2 (5.67e-3)
+/-/=	7/0/0	6/0/1	6/0/1	5/0/2	

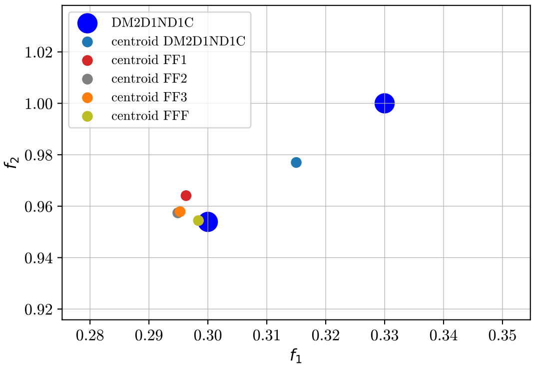
Table 6.6.: GD values in comparison with initial concept

Problem	FFF	NSGAII
DM2D1ND1C	4.3212e-5 (7.29e-5) +	9.0013e-5 (1.99e-5)
DM2D1ND1F	4.2100e-5 (1.16e-4) +	9.1934e-5 (2.74e-5)
DM2D1ND1M	3.1542e-5 (5.01e-5) +	8.8846e-5 (2.21e-5)
DM5D1ND4	1.5383e-5 (4.12e-6) +	9.4539e-5 (2.19e-5)
DM5D2ND3	2.7279e-5 (1.05e-4) +	9.0621e-5 (2.50e-5)
DM5D3ND2	1.1859e-5 (4.52e-6) +	9.6757e-5 (2.51e-5)
DM5D4ND1	2.2392e-5 (5.91e-5) +	8.6382e-5 (2.70e-5)
+/-/=	7/0/0	

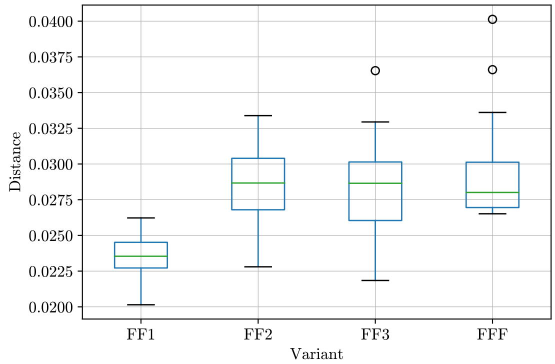
Table 6.7.: GD values in comparison with NSGA-II algorithm

Problem	FFF	FF3	FF2	FF1	F1
DM2D1ND1C	3.9764e-2 (1757e-1) -	1.094e-2 (5.631e-3) +	9.23532e-3 (6.691e-3) +	1.797e-2 (6.66712e-3) +	2.99e-2 (2.65e-2)
DM2D1ND1F	6.356e-2 (4.232e-2) +	7.4498e-2 (7.561e-2) +	8.3916e-2 (3.4343e-2) +	1.7407e-1 (4.3396-2) +	1.884e-1 (3.452e-2)
DM2D1ND1M	2.378e-2 (2.206e-2) +	2.9896e-2 (1.543e-2) +	3.4855e-2 (3.224e-2) +	6.7469e-2 (4.234e-2) +	8.8652e-2 (5.8753e-2)
DM5D1ND4	8.4155e-1 (1.6673e-2) +	8.527e-1 (3.106e-3) =	8.5343e-1 (4.182e-3) =	8.5161e-1 (2.0949e-3) =	8.5296e-1 (2.251e-3)
DM5D2ND3	3.377e-1 (2.406e-2) +	3.364e-1 (6.6475e-3) +	3.3438e-1 (6.246e-1) +	3.3303e-1 (1.8711e-2) +	3.42184e-1 (4.544e-2)
DM5D3ND2	3,0634e-1 (2.477e-2) +	3.8379e-1 (3.880e-2) -	3.7112e-1 (5.377e-2) -	3.383e-1 (3.9785e-2) -	3.2835e-1 (1.752e-2)
DM5D4ND1	2.070e-1 (3.9507e-2) -	9.6079e-2 (26569e-2) +	1.0039e-1 (2.8037e-2) +	1.6135e-1 (4.8035e-2) -	1.59454e-1 (3603e-2)
+/-/=	5/2/0	5/1/1	5/1/1	4/2/1	

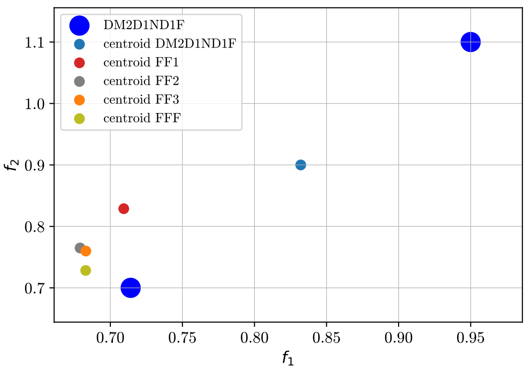
Table 6.8.: Fairness values for the variations of the improved concept



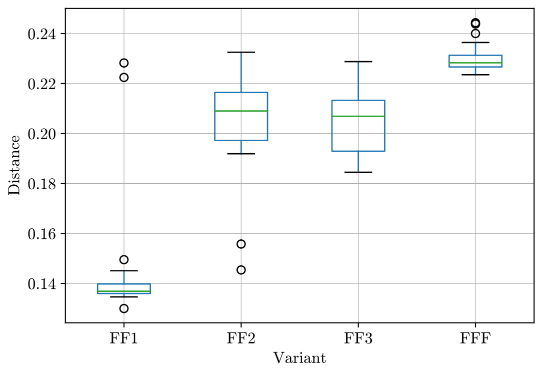
(a) PfPs centroids DM2D1ND1C



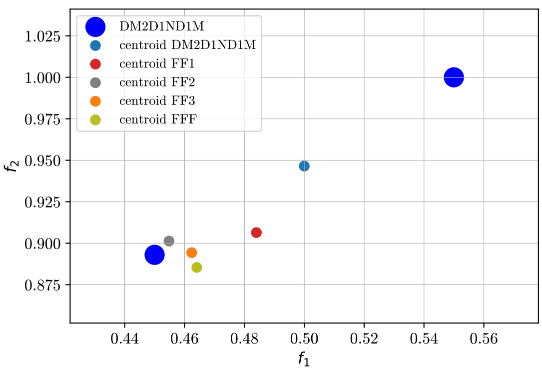
(b) Distances DM2D1ND1C



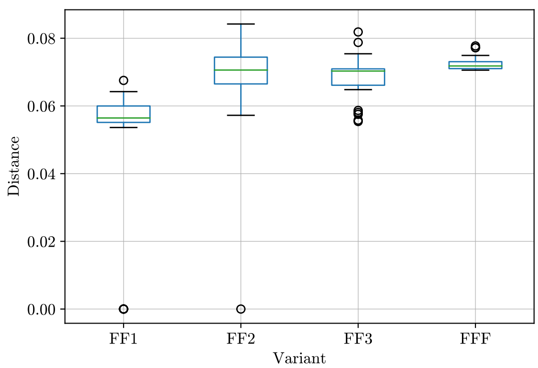
(c) PfPs centroids DM2D1ND1F



(d) Distances DM2D1ND1F

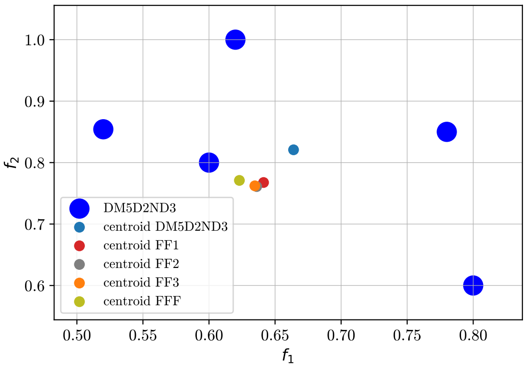


(e) PfPs centroids DM2D1ND1M

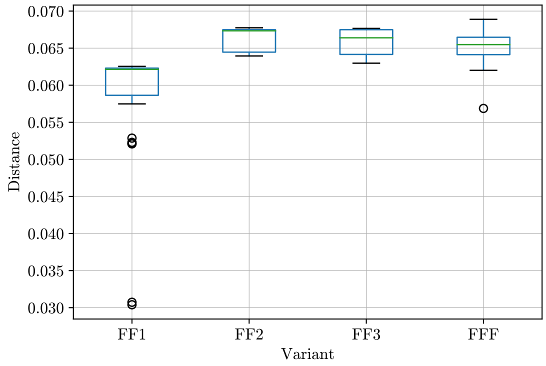


(f) Distances DM2D1ND1M

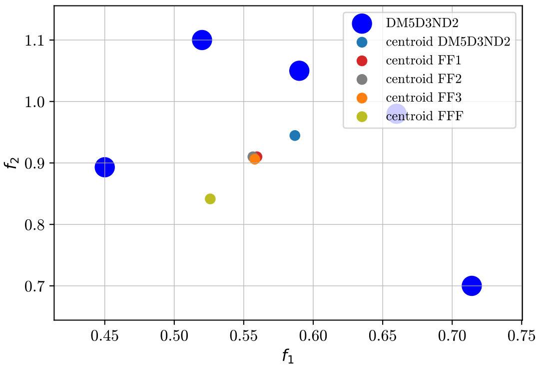
Figure 6.6.: Preference points repositioning, scenario 3



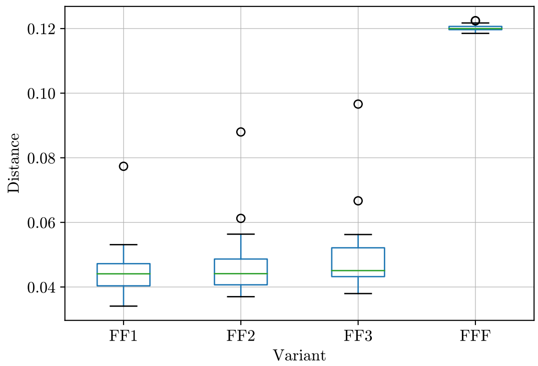
(a) PfPs centroids DM5D2ND3



(b) Distances DM5D2ND3



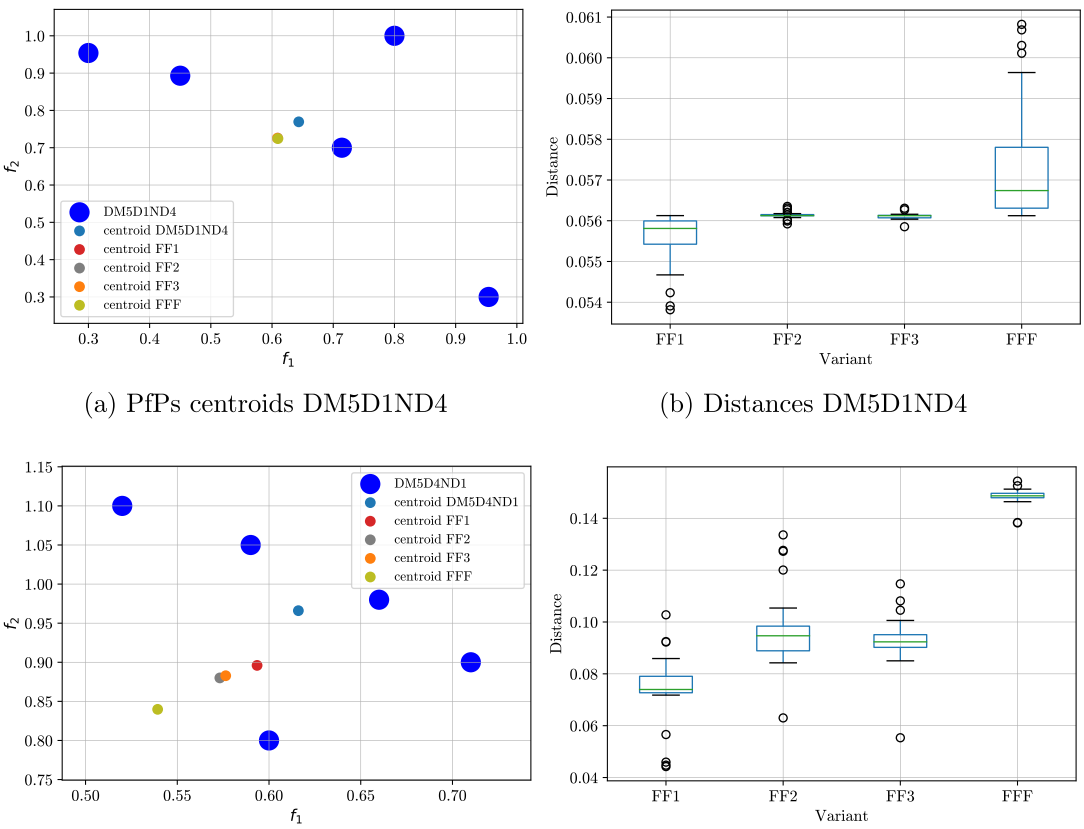
(c) PfPs centroids DM5D3ND2



(d) Distances DM5D3ND2

Figure 6.7.: Preference points repositioning, scenario 4

In an effort to have a better understanding of the effects of the PfPs repositioning, the difference between the newly determined PfPs and the original ones was investigated. The analysis consists of aggregating the final points of each variant, for each case, into a single centroid. These resulting centroids from each run were then again amalgamated into a final centroid in an attempt to have a more accurate idea of its true location. The distance from these centroids and the centroid representing the initial PfPs is calculated, to better understand the extent of the repositioning.



(a) PfPs centroids DM5D1ND4 (b) Distances DM5D1ND4

(c) PfPs centroids DM5D4ND1 (d) Distances DM5D4ND1

Figure 6.8.: Preference points repositioning, scenario 5

The end positions of the repositioned PfPs, represented by a centroid, along with their distances (euclidean) from the centroid of the original PfPs can be seen on figures 6.6, 6.7 and 6.8. Analysing the PfPs repositioning on scenario 3, we can notice that increasing the frequency of attempted repositionings, seems to correlate with further distances towards the front and better convergence, with the final variant exhibiting the best results. Moving on to scenario 4 and its PfPs changes, we can observe a similar result, which can also be noticed when examining scenario 5. What can be interesting to note specifically is the repositioing seems to make the biggest jumps initially with consecutive repositioning moving the points less and less. This can be seen especially well in figure 6.6 for case DM2D1ND1M. The biggest extreme between the variants is showcased in 6.8 for case DM5D4ND1. Here, we can clearly observe that when dealing with many dominated PfPs the proposed improvement seems to be able to bring the PfPs very close to the front, which can be also seen when examining table 6.6. However, an interesting observation can also be made from table 6.8, which shows that although the PfPs have moved to the front in a significant capacity from the FFF variant, the fairness favors the FF3 variant. With this we can conclude that although it is likely that some of the PfP will move closer when moving continuously towards the front, they also have a chance to drift further apart in an attempt to secure a better final set of solutions.

6.3. Examining the Behaviour when Faced with Complex Cases

Given the great likelihood that the concept employed would be tasked to solve more complex cases then that of only two objectives, what will follow is the examination of its behavior in more complex scenarios in terms of objectives and decision variables. The exact details concerned the cases showcased can be found in section 5.2.

Studying table 6.9 we can see that when confronted with a reasonable number of objectives, in terms of how many the concept of domination can tackle (six in this case), the results match what has been previously ascertained. The convergence values are once again better compared to the NSGA-II algorithm. In terms of IGD and HV we again see a familiar pattern, where the lack of diversity is evident, due to the more constrained nature of the approach. This would indicate that the approach seems to be operating as intended even when the number of objectives is on the higher side, for algorithms that make use of domination.

The table 6.10 on the other hand depicts extreme cases, which are hard to tackle for any algorithms that are focused around the concept of domination. We can see that should the DMs specify their PfPs to be on the front, the algorithm again behaves in an expected manner. However, should the points be more scattered and be further away from the front, the PfP repositioning seems unable to produce the same level of results as before. This is likely due to the fact that the repositioing step also uses domination. Given that domination fails to be an effective method once we have seven or more objectives it would

Problem	N	M	D	FEs	FFF	NSGAII
GD						
M6DM5D1ND4	100	6	6	10000	1.3878e-2 (1.06e-3) +	1.6429e-2 (8.51e-4)
M6DM5D2ND3	100	6	6	10000	1.3644e-2 (9.91e-4) +	1.6409e-2 (9.53e-4)
M6DM5D3ND2	100	6	6	10000	1.3486e-2 (1.35e-3) +	1.6356e-2 (7.79e-4)
M6DM5D4ND1	100	6	6	10000	1.3495e-2 (1.35e-3) +	1.6259e-2 (8.46e-4)
+/-/=					4/0/0	
IGD						
M6DM5D1ND4	100	6	6	10000	7.5976e-1 (2.18e-2) -	2.6249e-1 (1.24e-2)
M6DM5D2ND3	100	6	6	10000	7.4974e-1 (1.84e-2) -	2.6684e-1 (1.82e-2)
M6DM5D3ND2	100	6	6	10000	7.3586e-1 (2.57e-2) -	2.6333e-1 (1.25e-2)
M6DM5D4ND1	100	6	6	10000	7.7739e-1 (3.30e-2) -	2.6473e-1 (1.49e-2)
+/-/=					0/4/0	
HV						
M6DM5D1ND4	100	6	6	10000	1.0007e-1 (1.95e-2) -	5.8952e-1 (2.07e-2)
M6DM5D2ND3	100	6	6	10000	1.1031e-1 (1.49e-2) -	5.9121e-1 (2.78e-2)
M6DM5D3ND2	100	6	6	10000	1.2272e-1 (2.15e-2) -	5.8770e-1 (2.50e-2)
M6DM5D4ND1	100	6	6	10000	1.0763e-1 (2.35e-2) -	5.8854e-1 (2.48e-2)
+/-/=					0/4/0	

Table 6.9.: Metrics, scenario 6

Problem	N	M	D	FEs	FFF	NSGAII
GD						
M10DM5F	100	10	10	10000	2.8930e-2 (4.85e-4) +	3.0075e-2 (2.59e-3)
M10DM5	100	10	10	10000	4.1683e-2 (2.24e-3) -	3.0591e-2 (2.62e-3)
+/-/=					1/1/0	
IGD						
M10DM5F	100	10	10	10000	8.1252e-1 (1.57e-2) -	4.3070e-1 (5.90e-2)
M10DM5	100	10	10	10000	9.2685e-1 (2.34e-2) -	4.5651e-1 (4.43e-2)
+/-/=					0/2/0	
HV						
M10DM5F	100	10	10	10000	9.5114e-2 (1.22e-2) -	5.6950e-1 (6.36e-2)
M10DM5	100	10	10	10000	4.5974e-2 (9.31e-3) -	5.4545e-1 (6.23e-2)
+/-/=					0/2/0	

Table 6.10.: Metrics, scenario 7

seem as though the repositioing can not effectively move the PfPs closer to the front, resulting in a worsened convergence.

Given the fact that it is based on the concept of domination and the key ideas behind NSGA-II the proposed approach seems to be able to handle cases with a reasonable number of objectives in a satisfiable manner, inline with the expectations.

6.4. Summary of Analysis

The proposed NSGA-II for Teams algorithms along with its improved variant Adaptive NSGA-II for Teams seem to be behaving in an anticipated manner and capable of providing certain benefits when dealing with multi-objective optimisation with many differing DMs. The approaches exhibit greater convergence then that of the base NSGA-II algorithmdue to focusing on a specific are of interest. The specific disadvantages, caused by dominated PfPs and their tendency to pull solutions away from the front, were addressed by the reference repositioning incorporated in the adaptive version of the algorithm. With regards to fairness although the adaptive algorithm may be able to increase it in certain case, it is ultimately dependent on the similarity of the DMs preferences.

Although, excelling in convergence, the algorithm exhibits worsened results when it comes to diversity, a decline that is also evident in the HV metric. One might argue, that the bad diversity metrics could be considered a problem, however, greater diversity in this context may not be better. Diversity is needed so that the DMs are not overwhelmed from the many options presented to them and that the options can in fact appear distinct enough, to warranty the DMs consideration. But if one would consider a high stakes resource allocation problem (2020 Corona crisis is an excellent example), multiple differing groups might have different ideas how the resources should be used. This negotiation needs greater nuance then that of a business or manufacturing problem as each percent of difference from their positions could have a drastic influence on a number of human lives and can alleviate or worsen human suffering. Thus, when dealing with fairness and moral issues, especially when one needs to present practical solutions, it is important that greater nuances be available for discussion and negotiation. The problem with diversity is also partly addressed by the the ability of the DMs to specify how many solutions they want to end up with, which can enable them to keep the strong critical nuances while not being overwhelmed with alternatives.

Venturing further from fairness and many DMs it would seem that the proposed approach could also have additional uses, not necessarily bound to the initial expectations. Interestingly, it would appear that one may use the PfPs with one DM in an attempt to simply specify an area of interest of various shape and size in an effort to focus the attention of an EA. Additionally, given the modular nature of the approach it would seem rather simple to attach the fairness and gain level to an existing algorithm, different to that of NSGA-II. The approach, along with its adaptive nature may also be employed to tackle problems which seem to be difficult for other algorithms built around the concept of domination. Overall, the examined experiments and their result suggest that the NSGA-II for Teams algorithm can be used to tackle multi-objective optimisation problems with many DMs as well as other more interesting problems where focusing on an area of interest is desired.

7. Conclusion and Future Work

Given the ever growing complexity of the modern world, as well as the tendency towards fairness in a number of differing ways, methods which incorporate multiple different DMs with various perspectives and expertise are needed. Additionally, the consideration of many objectives when tackling difficult problems seems to be necessitated as such problems can rarely be transformed to a single objective problem without significant decline in the quality of the final outcome. The accurate capture of nuances in both the problems and their respective objectives is especially important when approaching problems which have a moral component.

The approach proposed in this thesis is aimed at aiding DMs when tackling these multi-criteria optimisation problems, whilst also providing mechanisms to alleviate the processes of compromise and negotiation. The process was envisioned to be as straightforward as possible, requiring only the preferences of DMs (PfPs) for each objective, their mutual agreement on the problem formulation as well as possible categorisation based on their level of influence. At the heart of the concept is the idea of a second level to the initial problem. This additional component is an accompanying multi-objective problem where the goals are that of fairness and gain. These new objectives operate on the similarity or dissimilarity of the solutions to the aforementioned preferences from the DMs. The incorporated fairness and gain level serves as an additional mechanism to influence the search for alternatives with the best possible trade-offs for the initial problem, which are also acceptable to all the participants in the decision making process.

The realisation of the ideas was executed by using the NSGA-II algorithm in order to solve the base problem, while its crucial element, fast non-dominated sorting, was fashioned to tackle the fairness and gain level. The resulting algorithm, titled NSGA-II for Teams, was then submitted to a variety of tests using the well established DTLZ2 problem, which was envisioned to replicate various possible real world scenarios, consisting of up to five DMs. In order to also have a better understanding of various situations, in terms of PfPs placement, a fairness metric was devised that focused on the fairness and gain level. This metric served as a way to gauge the fairness of certain PfPs placements, enabling us to distinguish between them with a single numerical value.

The algorithm displayed the tendency to confine solutions within an area defined by the PfPs of the DMs but also retained the influence of the initial NSGA-II algorithm to move closer to better and non-dominated solutions. Specifically the conclusion from the experiments were:

- The NSGA-II for Teams algorithm showed greater convergence capabilities, due to the focus on a smaller are of interest.
- The more confined region greatly worsened the diversity of the algorithm.
- The distances between the PfPs were the key factor with regards to the overall fairness.
- The relative positioning of the PfPs in terms of domination can greatly affect convergence, as non-dominated preference pull the solutions away from the front.

Given the negative impact on the convergence a improvement on the initial idea was devised. The new upgraded Adaptive NSGA-II for Teams implements an additional element of PfP repositioning. This repositioning can be both automated or approved by the corresponding DM. The crux of the idea was to implement a change in the PfP, swapping it for the closest non-dominated solutions, which was dominating the PfP. The logically superior new position of the DM would make it advantageous to switch whilst also moving the region of interest closer to the Pareto front, improving convergence. The examination of this adaptive version of the algorithm seems to have mitigated the problem of convergence, whilst also having a positive effect on fairness. The likely cause for the improvement in fairness is due to the repositioning bringing the points closer to the front and likely closer to one another.

The algorithm was additionally examined as how it relates when confounded with a greater number of both decision variables and objectives of the initial problem. The examined situation, consisting of six objectives and decision variables, showcased that the algorithm was still behaving in a predictable manner. However, when confronted with ten decision and objective variables, the algorithm outperformed its NSGA-II counterpart only when the PfPs were at the Pareto front. This is likely caused by the fact that the repositioning mechanism, which uses the concept of domination, was unable to effectively function due to the higher number of objectives.

The Adaptive NSGA-II algorithm was envisioned to be a centerpiece of an overall concept for many DMs. Apart from the PfPs as well as the algorithm itself the overall concepts also employs an additional filter in order to provide the DMs with the opportunity to incorporate hierarchies as well as other power dynamics in the decision making process.

This thesis only encompassed an initial examination of the original idea behind the 2020 Dagstuhl report [63], focusing on understanding its behaviour with regards to a simpler "toy" problem, which was well established in the literature. Thus, it seems only sensible to define more benchmark problems as well as attempt the use of the concept on a practical one. Furthermore, the concept could be additionally expanded to handle binary and mixed encodings, increasing the problems it can effectively solve. Conducting experiments involving live human DMs and observing their behaviour would also be an interesting avenue to explore. To better capture the psychology of humans and their various power and status dynamics game theory concepts could also be examined, as they relate to the proposed ideas of fairness and gain. The validity of the method to help individual DMs, focusing on specific areas of interest, could also be further researched. Moreover, a new method to help with the problem of diversity would also be a welcomed addition to the overall approach.

In closing, the conceived approach for multi-objective optimisation for many DMs, with the Adaptive NSGA-II algorithm at its core appears like a worthwhile and promising tool to help future DMs tackle ever increasingly difficult problems.

A. Crowding Distance Assignment

Algorithm 9: Crowding Distance Assignment [15]

Data: $F = \{\vec{f}(\vec{x}^{(1)}), ..., \vec{f}(\vec{x}^{(l)})\}$ the objective vectors for all l individuals in a non-dominated front F.

Result: Crowding Distance Measure

$l \leftarrow |F|$ // number of solutions in F.
$D \leftarrow \emptyset$
for i **to** l **do**
    $D \leftarrow D \cup \{0\}$ // initialize distances to each objective to 0.
end
foreach *objective* i **do**
    $F = sort(F, m)$ // sort using objective value.
    $D[1] = D[l] \leftarrow \infty$ // give high value so boundary points are always selected.
    for $i = 2$ **to** $(l - 1)$ // for all other points
    do
        $D[i] \leftarrow D[i] + (F[i+1]_m - F[i+1]_m/)$ // $F[i]_m$refers to the m-th objective function value of the i-th individual in the current non-dominated front F.
    end
end
return D

B. Fast Non-Dominated Sorting

Algorithm 10: Fast Non-dominated Sorting [15]

Data: The population P.

Result: The population sorted into multiple non-dominated fronts.

foreach $\vec{x}^{(p)} \in P$ **do**
    $S_P \leftarrow \emptyset$
    $n_P = 0$
    foreach $\vec{x}^{(q)} \in P$ **do**
        if $\vec{x}^{(p)} \prec \vec{x}^{(q)}$ `// if` $\vec{x}^{(p)}$ `dominates` $\vec{x}^{(q)}$
        . **then**
        $S_p = S_p \cup \{\vec{x}^{(q)}\}$ `// Add` $\vec{x}^{(q)}$ `to the solutions dominated by` $\vec{x}^{(p)}$`.`
        else if $\vec{x}^{(q)} \prec \vec{x}^{(p)}$ **then**
            $\eta_p = \eta_p + 1$ `// Increment domination count of` $\vec{x}^{(p)}$`.`
    end
    if $\eta_p = 0$ **then**
    end
    $\vec{x}^{(p)}$ belongs in the first front. $\vec{x}^{(p)}_{rank} = 1$
    $F_1 = F_1 \cup \{\vec{x}^{(p)}\}$
end
$i = 1$ `// Initialize the front counter`
while $F_i \neq \emptyset$ **do**
    $Q \neq \emptyset$ `// Used to store the members of the next front.`

    foreach $\vec{x}^{(p)} \in F_i$ **do**
        foreach $\vec{x}^{(q)} \in S_i$ **do**
            $\eta_q = \eta_q + 1$ **if** $\eta_q = 0$ `//` $\vec{x}^{(q)}$ `belongs to the next front.`
             then
                $\vec{x}^{(q)}_{rank} = i + 1$
                $Q = Q \cap \vec{x}^{(q)}$
            end
        end
    end
    $i = i + 1$
    $F_i = Q$
end
return $\{F_1 ... F_i\}$

C. Global variables of the algorithm

Parameter	Value
Tournament size	2
Probability of doing crossover	1
Distribution index of SBX	20
Expectation of number of bits doing mutation	1
Distribution index of polynomial mutation	20

Table C.1.: Global variables of the algorithm

D. Preference Points Values for Showcased Situations

Situation	PfPs	Objectives									
		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
DM2D1ND1C	PfP1	0.3	0.9539								
	PfP2	0.33	1								
DM2D1ND1F	PfP1	0.7141	0.7								
	PfP2	0.95	1.1								
DM2D1ND1M	PfP1	0.45	0.893								
	PfP2	0.55	1								
DM5D1ND4	PfP1	0.7141	0.7								
	PfP2	0.3	0.9539								
	PfP3	0.9539	0.3								
	PfP4	0.45	0.893								
	PfP5	0.8	1								
DM5D2ND3	PfP1	0.6	0.8								
	PfP2	0.8	0.6								
	PfP3	0.52	0.8542								
	PfP4	0.62	1								
	PfP5	0.78	0.85								
DM5D3ND2	PfP1	0.7141	0.7								
	PfP2	0.45	0.893								
	PfP3	0.52	1.1								
	PfP4	0.59	1.05								
	PfP5	0.66	0.98								
DM5D4ND1	PfP1	0.6	0.8								
	PfP2	0.52	1.1								
	PfP3	0.59	1.05								
	PfP4	0.66	0.98								
	PfP5	0.71	0.9								
M6DM5D1ND4	PfP1	0.3162	0.2	0.3	0.4	0.5	0.6				
	PfP2	0.6	0.5	0.4	0.3	0.2	0.3162				
	PfP3	0.2	0.3162	0.5	0.6	0.3	0.4				
	PfP4	0.4	0.5	0.6	0.3162	0.2	0.3				
	PfP5	0.9	0.95	0.9	0.95	0.9	0.95				
M6DM5D2ND3	PfP1	0.3162	0.2	0.3	0.4	0.5	0.6				
	PfP2	0.6	0.5	0.4	0.3	0.2	0.3162				
	PfP3	0.2	0.3162	0.5	0.6	0.3	0.4				
	PfP4	0.9	0.95	0.9	0.95	0.9	0.95				
	PfP5	0.95	0.9	0.95	0.9	0.95	0.9				
M6DM5D3ND2	PfP1	0.3162	0.2	0.3	0.4	0.5	0.6				
	PfP2	0.6	0.5	0.4	0.3	0.2	0.3162				
	PfP3	0.9	0.95	0.9	0.95	0.9	0.95				
	PfP4	0.95	0.9	0.95	0.9	0.95	0.9				
	PfP5	0.85	0.85	0.85	0.85	0.85	0.85				
M6DM5D4ND1	PfP1	0.3162	0.2	0.3	0.4	0.5	0.6				
	PfP2	0.9	0.95	0.9	0.95	0.9	0.95				
	PfP3	0.95	0.9	0.95	0.9	0.95	0.9				
	PfP4	0.85	0.85	0.85	0.85	0.85	0.85				
	PfP5	0.99	0.99	0.99	0.99	0.99	0.99				
M10DM5F	PfP1	0.1	0.135	0.145	0.42	0.32	0.65	0.21	0.12	0.35	0.5
	PfP2	0.5	0.35	0.12	0.21	0.65	0.32	0.42	0.145	0.135	0.1
	PfP3	0.35	0.12	0.21	0.65	0.32	0.42	0.145	0.135	0.1	0.5
	PfP4	0.145	0.135	0.1	0.5	0.21	0.65	0.32	0.42	0.35	0.12
	PfP5	0.145	0.135	0.5	0.21	0.32	0.42	0.12	0.35	0.1	0.65
M10DM5	PfP1	0.3162	0.2	0.3	0.4	0.5	0.6	0.6	0.5	0.4	0.3
	PfP2	0.2	0.3162	0.2	0.3162	0.5	0.6	0.3	0.4	0.4	0.5
	PfP3	0.5	0.6	0.3162	0.2	0.3	0.3	0.6	0.5	0.4	0.3162
	PfP4	0.7	0.2	0.3	0.6	0.3	0.8	0.415	0.378	0.543	0.32
	PfP5	0.1	0.9	0.9	0.43	0.2	0.3162	0.5	0.6	0.3	0.4

Table D.1.: Preference Point Values

Bibliography

[1] Lakhmi; Goldberg Robert Abraham, Ajith; Jain. *[Advanced Information and Knowledge Processing] Evolutionary Multiobjective Optimization || Scalable Test Problems for Evolutionary Multiobjective Optimization*, volume 10.1007/1-84628-137-7. 2005.

[2] Ram Agrawal, Kalyanmoy Deb, and Ram Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9, 06 2000.

[3] Peter J. Angeline. Genetic programming: On the programming of computers by means of natural selection,: John r. koza, a bradford book, mit press, cambridge ma, 1992, isbn 0-262-11170-5, xiv + 819pp.,. *Biosystems*, 33, 1994.

[4] Kaisa Miettinen PhD (auth.). *Nonlinear Multiobjective Optimization*. International Series in Operations Research Management Science 12. Springer US, 1 edition, 1998.

[5] Vittaldas Baek, Seungyup; Prabhu. An algorithm for solving multiobjective problem with multiple decision makers. *International Journal of Information and Decision Sciences*, 2008.

[6] Lamjed; Ghedira Khaled Bechikh, Slim; Ben Said. [ieee 2011 11th international conference on hybrid intelligent systems (his 2011) - melacca, malaysia (2011.12.5-2011.12.8)] 2011 11th international conference on hybrid intelligent systems (his) - negotiating decision makers' reference points for group preference-based evolutionary multi-objective optimization. 2011.

[7] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *CoRR*, abs/1607.06520, 2016.

[8] J. Born. Schwefel, h.-p., numerische optimierung von computer-modellen mittels der evolutionsstrategie. mit einer vergleichenden einführung in die

hill-climbing- und zufallsstrategien. (isr 26) basel-stuttgart, birkhäuser verlag 1977. 390 s., sfr. 48,–. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 60, 1980.

[9] Review by: Charles E. Taylor. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. complex adaptive systems.by john h. holland. *Quarterly Review of Biology*, 69, 03 1994.

[10] Richard A. Caruana. *Machine Learning Proceedings 1988 || Representation and Hidden Bias: Gray vs. Binary Coding for Genetic Algorithms.* 1988.

[11] Historia Civilis. The roman senate during the republic.

[12] David A. Coley. *An introduction to genetic algorithms for scientists and engineers*. World Scientific, har/dis edition, 1997.

[13] Carlos A. Coello Coello; Nareli Cruz Cortés. Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines*, 6, 06 2005.

[14] J. Dastin. "Amazon scraps secret AI recruiting tool that showed bias against women". online: `reuters.com`, 2018.

[15] A.; Agarwal S.; Meyarivan T. Deb, K.; Pratap. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6, 2002.

[16] Erik; Coello Coello Carlos A.; Klamroth Kathrin; Miettinen-Kaisa; Mostaghim Sanaz; Reed Patrick Deb, Kalyanmoy; Goodman. *[Lecture Notes in Computer Science] Evolutionary Multi-Criterion Optimization Volume 11411 (10th International Conference, EMO 2019, East Lansing, MI, USA, March 10-13, 2019, Proceedings) || Multi-objective Techniques for Single-Objective Local Search: A Case Study on Traveling Salesman Problem*, volume 10.1007/978-3-030-12598-1. 2019.

[17] J. Deb, Kalyanmoy; Sundar. [acm press the 8th annual conference - seattle, washington, usa (2006.07.08-2006.07.12)] proceedings of the 8th annual conference on genetic and evolutionary computation - gecco '06 - reference point based multi-objective optimization using evolutionary algorithms. 2006.

[18] Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 1 edition, 2001.

[19] Kalyanmoy Deb and Mayank Goyal. A combined genetic adaptive search (geneas) for engineering design. *Computer Science and informatics*, 1996.

[20] Karthik; Okabe Tatsuya Deb, Kalyanmoy; Sindhya. [acm press the 9th annual conference - london, england (2007.07.07-2007.07.11)] proceedings of the 9th annual conference on genetic and evolutionary computation - gecco '07 - self-adaptive simulated binary crossover for real-parameter optimization. 2007.

[21] Boris; Stewart Theodor J.; Wallenius Jyrki Ehrgott, Matthias; Naujoks. *[Lecture Notes in Economics and Mathematical Systems] Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems Volume 634 || Faster Hypervolume-Based Search Using Monte Carlo Sampling*, volume 10.1007/978-3-642-04045-0. 2010.

[22] A.E. Eiben. Genetic algorithms + data structures = evolution programs: Z. michalewicz. springer, berlin, 1996, 3rd revised and extended edition (1st edition appeared in 1992), 387 pp. (hardcover), 68 figures, 36 tables, price dm 58. *Artificial Intelligence in Medicine*, 9, 1997.

[23] J. Angwin et al. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. *PROPUBLICA,*, 2016.

[24] Yongsheng Fang and Jun li. A review of tournament selection in genetic programming. pages 181–192, 10 2010.

[25] David B Fogel. An introduction to genetic algorithms: Melanie mitchell. mit press, cambridge ma, 1996. (cloth), 270 pp. *Bulletin of Mathematical Biology*, 59, 1997.

[26] David B. Fogel. *Evolutionary Computation || Artificial Intelligence through Simulated Evolution*, volume 10.1109/9780470544600. 1998.

[27] Peter J.; Zitzler Eckart; Thiele Lothar; Deb-Kalyanmoy Fonseca, Carlos M.; Fleming. *[Lecture Notes in Computer Science] Evolutionary Multi-Criterion Optimization Volume 2632 || Towards a Quick Computation of Well-Spread Pareto-Optimal Solutions*, volume 10.1007/3-540-36970-8. 2003.

[28] Karl Bringmann; Tobias Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. *Computational Geometry*, 43, 2010.

[29] Seyyedeh Ghoreishi, Anders Clausen, and Bo Joergensen. Termination criteria in evolutionary algorithms: A survey. pages 373–384, 01 2017.

[30] David E. Goldberg. *[Foundations of Genetic Algorithms] Volume 1 // A Comparative Analysis of Selection Schemes Used in Genetic Algorithms*. 1991.

[31] J. Hirche. Hwang, ching-lai/masud, a. s. md., multiple objective decision making — methods and applications. a state-of-the-art survey. in collaboration with s. r. paidy and kwangsun yoon. lecture notes in economics and mathematical systems 164. berlin-heidelberg-new york, springer-verlag 1979. xii, 351 s., 39 abb., 35 tab., dm 35,50, isbn 3-540-09111-4. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 60, 1980.

[32] ; Andreae Peter; Johnston Mark Huayang Xie, ; Mengjie Zhang. [ieee 2008 ieee congress on evolutionary computation (cec) - hong kong, china (2008.06.1-2008.06.6)] 2008 ieee congress on evolutionary computation (ieee world congress on computational intelligence) - is the not-sampled issue in tournament selection critical? 2008.

[33] K. Deb J. Branke. *Integrating User Preferences into Evolutionary Multi-Objective Optimization*. 2004.

[34] Robert Weick Karl Marx, Friedrich Engels. *The Communist Manifesto and Das Kapital*. Knickerbocker Classics, original retail edition, 12 Feb 2019.

[35] Sumit Singh; Kumar Vijay Katoch, Sourabh; Chauhan. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 10 2020.

[36] Xiaodong; Qin A. K. Kazimipour, Borhan; Li. [ieee 2014 ieee congress on evolutionary computation (cec) - beijing, china (2014.7.6-2014.7.11)] 2014 ieee congress on evolutionary computation (cec) - a review of population initialization techniques for evolutionary algorithms. 2014.

[37] E. Xevi; S. Khan. A multi-objective optimisation approach to water management. *Journal of Environmental Management*, 77, 2005.

[38] Özgür Koksalmis, Emrah; Kabak. Deriving decision makers' weights in group decision making: An overview of objective methods. *Information Fusion*, 11 2018.

[39] Wouter M. Koolen. The pareto regret frontier. In *NIPS*, 2013.

[40] Matthew J.; Keane Martin; Bosacchi Bruno; Fogel-David B.; Bezdek James C. Koza, John R.; Streeter. Spie proceedings [spie optical science and technology, spie's 48th annual meeting - san diego, ca (sunday 3 august 2003)] applications and science of neural networks, fuzzy systems, and evolutionary computation vi - routine human-competitive machine intelligence by means of genetic programming. volume 5200, 2003.

[41] L.; Zitzler E. Laumanns, M.; Thiele. Running time analysis of multiobjective evolutionary algorithms on pseudo-boolean functions. *IEEE Transactions on Evolutionary Computation*, 8, 2004.

[42] Lothar; Deb Kalyanmoy; Zitzler Eckart Laumanns, Marco; Thiele. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10, 09 2002.

[43] Qingfu Zhang; Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11, 2007.

[44] Manuel Limmer, Steffen; Dietrich. [ieee 2018 ieee symposium series on computational intelligence (ssci) - bangalore, india (2018.11.18-2018.11.21)] 2018 ieee symposium series on computational intelligence (ssci) - optimization of dynamic prices for electric vehicle charging considering fairness. 2018.

[45] MATLAB. *version 9.4.0.813654 (R2018a)*. The MathWorks Inc., Natick, Massachusetts, 2018.

[46] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *CoRR*, abs/1908.09635, 2019.

[47] Alexander V.; Kamenev George K.; Berezkin Vadim E. Miettinen, Kaisa; Lotov. Integration of two multiobjective optimization methods for nonlinear problems. *Optimization Methods and Software*, 18, 02 2003.

[48] B. Miller and D. Goldberg. Genetic algorithms, tournament selection, and the effects of noise. *Complex Syst.*, 9, 1995.

[49] David E. Miller, Brad L.; Goldberg. Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary Computation*, 4, 06 1996.

[50] Mohammad Nabi; Li Xiaodong Mohammadi, Asad; Omidvar. [ieee 2012 ieee congress on evolutionary computation (cec) - brisbane, australia (2012.06.10-2012.06.15)] 2012 ieee congress on evolutionary computation - reference point based multi-objective optimization through decomposition. 2012.

[51] Kaisa Miettinen; Marko M. Mäkelä. On scalarizing functions in multiobjective optimization. *OR Spectrum*, 24, 05 2002.

[52] Tobias Friedrich; Christian Horoba; Frank Neumann. Illustration of fairness in evolutionary multi-objective optimization. *Theoretical Computer Science*, 412, 2011.

[53] Christian Braune Sanaz Mostaghim Matthias Steinbrecher Frank Klawonn Christian Moewes Rudolf Kruse, Christian Borgelt. *Computational Intelligence: A Methodological Introduction*. Texts in Computer Science. Imprint: Springer, Springer London, 2ed. edition, 2016.

[54] Martín Safe, Jessica Carballido, Ignacio Ponzoni, and Nélida Brignole. On stopping criteria for genetic algorithms. volume 3171, pages 405–413, 09 2004.

[55] Risto Lahdelma; Kaisa Miettinen; Pekka Salminen. Reference point approach for multiple decision makers. *European Journal of Operational Research*, 164, 2005.

[56] J Branke; T Kaußler; H Schmeck. Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software (Thomson Reuters)*, 32, 2001.

[57] Jason Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization /. 08 2005.

[58] Darrell Sokolov, Artem; Whitley. [acm press the 2005 conference - washington dc, usa (2005.06.25-2005.06.29)] proceedings of the 2005 conference on genetic and evolutionary computation - gecco '05 - unbiased tournament selection. 2005.

[59] E. Sperner. Neuer beweis für die invarianz der dimensionszahl und des gebietes. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 6, 12 1928.

[60] Kalyanmoy Srinivas, N.; Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2, 1994.

[61] Ran; Zhang Xingyi; Jin Yaochu Tian, Ye; Cheng. Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine*, 12, 11 2017.

[62] Shenglian; Liu Dedi; Pan Zhengke; Hong Xingjun Tian, Jing; Guo. A fair approach for multi-objective water resources allocation. *Water Resources Management*, 7 2019.

[63] Sanaz Mostaghim; Michael Emmerich; Jussi Hakanen; Patrick Reed; Pradyumn Kumar Shukla; Jurgen Branke; Gunter Rudolph; Heike Trautmann. Supporting problem solving with many decision makers in multi-objective optimization. *Dagstuhl 2020 Report*, 2020.

[64] David A. Van Veldhuizen. Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations. technical report. *Evolutionary Computation*, 1999.

[65] W. Vent. Rechenberg, ingo, evolutionsstrategie — optimierung technischer systeme nach prinzipien der biologischen evolution. 170 s. mit 36 abb. frommann-holzboog-verlag. stuttgart 1973. broschiert. *Feddes Repertorium*, 86, 1975.

[66] Chris Wald. The wilcoxon rank-sum test. *Department of statistics, University of Auckland*, 1997.

[67] Mengjie; Andreae Peter; Johnson Mark Xie, Huayang; Zhang. [acm press the 10th annual conference - atlanta, ga, usa (2008.07.12-2008.07.16)] proceedings of the 10th annual conference on genetic and evolutionary computation - gecco '08 - an analysis of multi-sampled issue and no-replacement tournament selection. 2008.

[68] Edmund K.; Lozano José A.; Smith Jim; Merelo-Guervós Juan Julián; Bullinaria John A.; Rowe Jonathan E.; Tiňo Peter; Kabán Ata; Schwefel Hans-Paul Yao, Xin; Burke. *[Lecture Notes in Computer Science] Parallel*

Problem Solving from Nature - PPSN VIII Volume 3242 || SPEA2+: Improving the Performance of the Strength Pareto Evolutionary Algorithm 2, volume 10.1007/b100601. 2004.

[69] Anthony Finkelstein; Mark Harman; S. Afshin Mansouri; Jian Ren; Yuanyuan Zhang. A search based approach to fairness analysis in requirement assignments to aid negotiation, mediation and decision making. *Requirements Engineering*, 14, 12 2009.

[70] L. Zitzler, E.; Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3, 1999.

[71] Göktürk Üçoluk. Genetic algorithm solution of the tsp avoiding special crossover and mutation. *Intelligent Automation Soft Computing*, 8, 01 2002.

Declaration of Authorship

I hereby declare that this thesis was created by me and me alone using only the stated sources and tools.

Boris Djartov

Magdeburg, April 15, 2021