Alexander Tracht

# Optimising for Win-Win Situations in Multi-Objective Decision Making with multiple Decision Makers

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

INF

FAKULTÄT FÜR
INFORMATIK

Intelligent Cooperative Systems
Computational Intelligence

# Optimising for Win-Win Situations in Multi-Objective Decision Making with multiple Decision Makers

Master Thesis

Alexander Tracht

May 19, 2022

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

Advisor: Sebastian Mai

**Alexander Tracht:** *Optimising for Win-Win Situations in Multi-Objective Decision Making with multiple Decision Makers*
Otto-von-Guericke Universität
Intelligent Cooperative Systems
Computational Intelligence
Magdeburg, 2022.

# Abstract

The field of Multi-Objective Optimisation aims to find a broad set of solutions along the optimal, non-dominated solution front, the Pareto Front. Widely used optimisation approaches to approximate the Pareto Front are Evolutionary Algorithms, which use evolutionary principles for optimisation. Multi-Criteria Decision-Making deals with Decision Makers and fulfilling their preferences, often stated as reference points, for solving optimisation problems, typically choosing a single, final, solution from a set of solutions. Combining both Evolutionary Algorithms and multiple Decision Makers' preferences, a subset of the pareto front, which is based on consensus, can be approximated to find a solution all Decision Makers can agree on.
This thesis presents two novel adaptations of NSGA-II for Teams, which integrates the concepts of fairness and gain to the Evolutionary Algorithm NSGA-II by adding a Pareto Regret based filtering step before environment selection. The first presented adaptation calculates Pareto Regret based on the Decision Maker's *partial* preference towards specific base values of a solution, whose regular objective values are aggregated on these *partial* base objective values. The resulting algorithm Win-Win NSGA-II for Teams optimises for a "win-win" consensus, wherein a Decision Maker's preference is only considered for their chosen partial objective values of the solution and the resulting solution therefore respects all Decision Maker's preferences.
The second adaptation, Cosine NSGA-II for Teams, is based on specifying reference weights instead of points, which form a reference line from which the cosine similarity to the solution vector is computed to specify the similarity between both solutions through their shared angle. Resulting is a reference line along which the Decision Maker prefers the solutions, which eliminates the need for repositioning reference points towards the Pareto Front.
Both algorithms were tested and compared to NSGA-II and (Adaptive) NSGA-II for Teams on the scalable Multi-Objective Multi-Agent Pathfinding (MOMAPF) problem, in which agents travel from one side of a map with obstacles to the other, an example relevant in the real world, as e.g. in robot path planning. Additionally, the second algorithm was tested and compared on the benchmark TNK problem.
Results indicate that, while worse in terms of "regular" fairness and gain, Win-

Win NSGA-II for Teams successfully optimise for win-win solutions, wherein all Decision Makers' preferences are aimed to be fullfilled, which also lead to a higher diversity. Regarding Cosine NSGA-II for Teams, performance is similar to regular NSGA-II for Teams, with a stronger effect of the Decision Makers' preferences and the optimisation for consensus, the effect coined as consensus pressure, while exhibiting less convergence to the Pareto Front.

# Preface

I'd like to thank a lot of people for their support while i was writing this thesis, I don't know how I could have written this without them. First I'd like to thank both my supervisor Prof. Dr.-Ing. habil. Sanaz Mostaghim and my advisor Sebastian Mai, without their tireless scientific support no matter how long I took to actually get into writing again, without them I don't think I could've finished this thesis. Second I'd like to thank my girlfriend Chrissi without whom I would have gone mad and lost my strength during the several stages of near-lockdown during the pandemic and the ever-changing circumstances under which this thesis was written. Thank you to all my friends and my family, you know who you are (especially if you're reading this) and you gave me strength during the long periods of stress and, later, apathy.

Finally a word of greeting to the reader. I hope you find this work useful and that it advances you in your endeavors, good luck with them.

# Contents

# List of Figures

# List of Algorithms

# List of Acronyms

**C-Metric** Coverage-Metric

**DM** Decision Maker

**EA** Evolutionary Algorithm

**GA** Genetic Algorithm

**HV** Hypervolume

**MAPF** Multi-Agent Pathfinding

**MCDM** Multi-Criteria Decision-Making

**MOMAPF** Multi-Objective Multi-Agent Pathfinding

**MOOP** Multi-Objective Optimisation Problem

**NSGA-II** Non-Dominated Sorting Algorithm II

**NSGA-II for Teams** Non-Dominated Sorting Algorithm II for Teams

**REA** Raw-Egg-Agent

**BA** Bread-Agent

**CAA** Cut-Avocado-Agent

# List of Nomenclature

$\phi_n$ Partial objective value for an objective value $\sigma_m$, see equation (4.1), page 34

$\sigma$ Objective value for Multi-Objective Optimisation Problems, see equation (2.1), page 8

$\succ$ Pareto Dominance Operator, see equation (2.3), page 8

$\vec{f}(x)$ Fitness values (objective values) of a solution $x$, see equation (2.7), page 16

$agg(\Phi)$ An aggregation function on a set of partial objective values $\Phi$, see equation (4.1), page 34

$apr(x, PR)$ Average Pareto Regret of solution $x$, see equation (2.6), page 13

$c(X, Y)$ C-Metric of solution front $X$ to solution front $Y$, see equation (5.2), page 45

$cd(F_i)$ Crowding Distances of solutions in front $F_i$, see equation (2.3), page 11

$cpr(f(x), R_i)$ Cosine Pareto Regret, the cosine similarity of a solution and a reference point, see equation (4.3), page 37

$epr(f(x), R_i))$ Egoistic Pareto Regret of solution $x$ with regards to the reference point $R_i$, see equation (4.2), page 35

$F$ An (approximated) front, see equation (5.1), page 44

$f_L(x)$ Length fitness of solution $x$, the average path length across all agents, see equation (2.10), page 16

$f_R(x)$ Risk fitness of solution $x$, the minimum distance across agents to obstacles and other agents, see equation (2.10), page 16

$f_T(x)$ Time fitness of solution $x$, the average travel time across all agents, see equation (2.10), page 16

$fairness\_filter(S, R)$ Fairness filter algorithm which filter solution set $S$ using fairness and gain, see equation (4.0), page 30

$fnds(S)$ Fast-Non-Dominated-Sort of solution set $S$ [4], see equation (2.3), page 11

$hv(F)$ Hypervolume of a solution front, see equation (5.1), page 44

$ipr(x, PR)$ Inequality in Pareto Regret of solution $x$, see equation (2.6), page 13

$n_{\text{gen}}$ Number of generations, see equation (5.0), page 39

$n_{\text{pop}}$ Population size, see equation (5.0), page 39

$P_g$ Population of generation $g$, see equation (2.3), page 11

$pr(x), R_m)$ Pareto Regret of solution $x$ with regards to reference point $R_m$, see equation (2.4), page 13

$R$ Set of reference points, see equation (2.4), page 13

$r_{\text{filter}}$ The rate at which $fairness\_filter$ filters the incoming solution set $S$, see equation (4.0), page 30

$x$ A solution to an optimisation problem, see equation (2.2), page 8

# 1 Introduction

Multi-Objective Optimisation and Multi-Criteria Decision-Making are important areas of research regarding the general topic of finding solutions to optimisation problems. Multi-Objective Optimisation deals with finding a set of solutions to an optimisation problem with multiple different values that are are optimised for, the objective values. On the other hand, the field of Multi-Criteria Decision-Making covers choosing a solution from a set of solutions based on the preferences of (multiple) Decision Maker(s). An optimisation method used to find a set of solutions close to the problem's Pareto Front, the set of solutions that are the respective best for their combination of objective values, is the Evolutionary Algorithm. It utilises evolutionary principles like mutation, crossover and selection to generate a population of solutions which then enters the next cycle (or generation) of evolutionary optimisation.

Utilising the preferences of the Decision Makers, the Evolutionary Algorithm can focus it's search towards the part of the Pareto Front which is the most interesting for them, improving the exploitation, the thoroughness of optimisation search, of that area. In the past [8] [6] attempts have been made to include the concepts of fairness and gain into Evolutionary Algorithms to satisfy the Decision Makers preferences and create a consensus in which all Decision Makers can to some degree agree on the found solutions. This was accomplished by having the Decision Makers state reference points, which declare their preferences as points in the space of objective values, and then filtering the population of the Evolutionary Algorithm for the solutions with the most fairness and gain, calculated through the usage of Pareto Regret, the Euclidean distance of a solution to the ideal solution (the respective reference point).

A problem of Multi-Objective Optimisation specifically in combination with Multi-Criteria Decision-Making is the common inability of Decision Makers to state their preferences for specific aspects of the solution, as a *partial* preferences, whereas objective values oftentimes are based on other aggregated base values (*partial* objective values), e.g. the average travel time of multiple agents in a pathfinding problem is not the same as the travel time of one specific agent, which the Decision Maker might be more interested in however. Optimising for creating a consensus and finding a "win-win" solution, as to that all Decision Makers satisfy their preferences, even though in regular objective space they would be in opposition, between *partial* preferences of multiple Decision Makers, using fairness and gain filtering is a novel field of research and this work's topic.

The scalable Multi-Objective Multi-Agent Pathfinding problem provides a suitable and realistic test-bed for this thesis, as it deals with multiple agents which each try to reach a goal on a map with obstacles. This problem is well suited for testing partial preferences and optimising for win-win situations, as each Decision Maker can focus on one agent's partial objective values and state their preferences as to how safe or fast they want them to be.

## 1.1 Motivation

This thesis explores new approaches to adapt the concepts of fairness, gain and Pareto Regret to optimise the consent in Multi-Objective Optimisation Problems (MOOPs) with multiple Decision Makers (DMs). The problem used to showcase this work is the scalable Multi-Objective Multi-Agent Pathfinding (MOMAPF) problem, visible in figure 1.1, based on the work of Mai et al. [23] [14] [15].

For finding a solution a DM can agree on, a common approach is to use reference points [18] [16] [5] to express their preferences beforehand (*a-priori*, see 2.2). With multiple DMs, finding consensus on their preferences is a common problem [7] [25].

Finding a consensing solution includes three main steps [8]:

1. Formulation of a goal (agreement on the model)
2. Select a subset of solutions
3. Negotiate the final solution

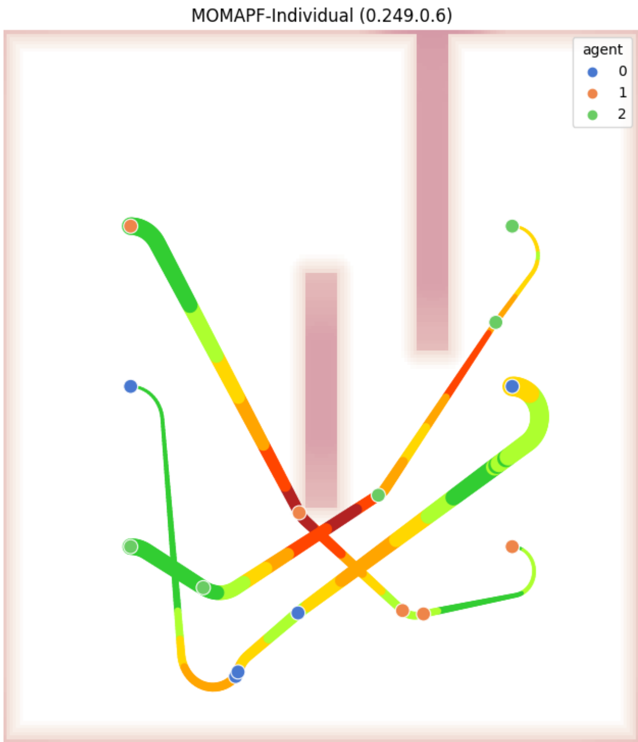


Figure 1.1: A Multi-Objective Multi-Agent Pathfinding (MOMAPF) problem solution visualisation, with agents on both sides of the map, navigating from the small end to the large end of their respective path, the colour indicates the risk of the agent based on the distance to obstacles and other agents, the path's thickness indicates the passed time

This work focuses on augmenting an existing approach for solving step 2, the selection of a subset of solutions which is tied to the DMs' preferences. The difficulty in this step lies in the possibly widely differing preferences of the DMs.

The approach this work focuses on are fairness- and gain-based selection algorithms, specifically for Evolutionary Algorithms. The primary work this is based on is a report from the 2020 Dagstuhl seminar 20031 by Emmerich et al. [8], where reference points are used to focus the selection of the solution subset under consideration of the DMs' preferences. The approach used utilises a second level of filtering solutions with fairness and gain. Both fairness and gain are computed using Pareto Regret as the underlying metric (see 2.3), which describes the similarity of solutions to a desired ideal solution.

This work introduces the differentiation of a specific kind of MOOP, a Partitionable Multi-Objective Optimisation Problem (see 4.2). Partitionable in this sense contrasts that objective functions in MOOPs oftentimes aggregate underlying metrics to reduce the number of objectives, e.g. by averaging the time of flight of multiple agents in Multi-Agent Pathfinding (MAPF) (see 2.4), instead of using each agent's value as a separate objective. This in turn, in favor of simplifying the computational problem, takes away from the ability of DMs to prefer some elements of a solution, e.g. the time of flight of a specific agent in MAPF, rather than the aggregated (e.g. the average) value of all elements, which is referred to as *partial* preference from here on. An example on why this can be beneficial is shown in figure 1.2, where each DM cares about their agents specific preference and in that sense do not agree with the presented solution, but due to the objective values being aggregated (e.g. averaged), the presented solution is regularly acceptable, as it exhibits a certain degree of compromise between high safety for the Raw-Egg-Agent (REA), high speed for the Cut-Avocado-Agent (CAA) and compromising values for the Bread-Agent (BA).

Partitionable Multi-Objective Optimisation Problems therefore are problems where the objective functions aggregate the underlying base values and the base values, from here *partial* objective values, can still be accessed for optimisation (see 4.2).

In comparison to the base work using regular fairness and gain, using the partial objective values in the calculation of fairness and gain is poised to better express specific interests of DMs (e.g. as shown in 1.2) and potentially even find better overall solutions by incorporating a "win-win" approach in the solution, as to that all DMs are satisfied by the solution, even though the DMs' preferences would be in opposition with each other in regular objective space.

An aside result of this work is an additional new way to calculate Pareto Regret, using the new concept of reference lines.

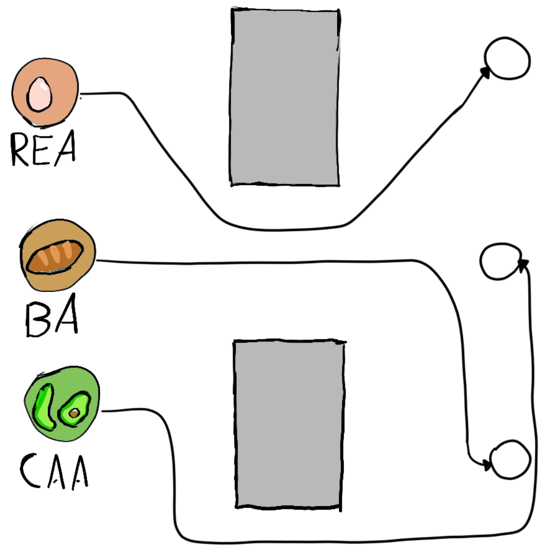


Figure 1.2: An Example MOMAPF-Scenario with DMs caring about one agents performance each, wherein the Raw-Egg-Agent should care about safety, the Bread-Agent should be able to compromise and the Cut-Avocado-Agent should care about travel time

## 1.2 Research Questions

The following research questions are proposed to examine how effective the proposed algorithms are.

- How do the presented algorithms compare to the existing algorithms on the regular objective space, measured by Hypervolume?
- How well do the presented algorithms respect Decision Makers' preference points on the regular objective space by qualitative inspection?
- How do the presented algorithms compare to the existing algorithms in terms of Pareto Regret based fairness and gain, measured by Coverage-Metric (C-Metric)?
- How do the presented algorithms compare to the existing algorithms in terms of the novel Egoistic Pareto based fairness and gain, measured by C-Metric?

- How well do the presented algorithms respect the Decision Makers' partial preferences with regards to phenotype of the solutions by qualitative inspection?

To evaluate these questions, the new algorithms are compared with algorithms proposed in the work of Emmerich et al. [8] and the work of Djartov [6] based on the Dagstuhl seminar 20031 report, by applying them to the benchmark experiment, using the Multi-Objective Multi-Agent Pathfinding (MOMAPF) problem.
The novelty way to calculate Pareto Regret using cosine similarity is additionally separately compared in a non-partitionable, more simple experiment using the benchmark TNK problem (see 2.5).

## 1.3 Thesis Structure

This thesis is structured in 7 chapters: Introduction, Basic Principles, Related Work, Methodology, Experiments, Evaluation and Conclusion and Future Work. Chapter 2 deals with basic principles like Evolutionary Algorithms and Decision Making. Chapter 3 deals with the current state of fairness and gain in Multi-Objective Optimisation and, in specific, Evolutionary Algorithms, as well as the optimisation of Multi-Agent Pathfinding. Chapter 4 explains the basics of Evolutionary Algorithms, the research this work is based on and the new approaches to fairness and gain. Chapter 5 and 6 deal with the conducted experiments and their evaluation respectively. In the last, seventh chapter, a conclusion is drawn and an outlook to potential future work is given.

# 2 Basic Principles

This chapter deals with basic, ground-laying principles, equations and algorithms, that are used throughout this thesis. First presented are Multi-Objective Optimisation, Multi-Objective Optimisation Problems and Evolutionary Algorithms in general, after which an introduction to the field of Multi-Criteria Decision-Making is given, both to introduce the general principles on which this thesis is built. The concepts and specific implementations of fairness and gain are presented next, which represent the core part of the algorithms presented in this thesis. Finally, the scalable Multi-Objective Multi-Agent Pathfinding (MOMAPF) problem, dealing with Multi-Agent Pathfinding but optimising for multiple objectives, and the much simpler TNK problem are introduced, as they are the benchmark problems on which the presented algorithms are tested.

## 2.1 Multi-Objective Optimisation, Evolutionary Algorithms

In this section both the general concepts of Multi-Objective Optimisation Problem (MOOP) and Evolutionary Algorithms, as well as the main algorithm on which all presented algorithms and their original base algorithms are based on, the Non-Dominated Sorting Algorithm II (NSGA-II).

### 2.1.1 Multi-Objective Optimisation Problems

A Multi-Objective Optimisation Problem consists of multiple objective values $\sigma_i$ which are to be optimised for. These typically conflict with each other, which presents additional challenges during optimisation in comparison to single-objective problems.

$$\begin{array}{l}\text{Multi-Objective Optimisation Problem:}\\ \text{minimise}\quad f(x) = (f_1(x), \dots, f_m(x)) = (\sigma_1, \dots, \sigma_m)\end{array} \tag{2.1}$$

The above equation defines a minimisation MOOP.

### 2.1.2 Pareto-Optimality and -Fronts

A solution $x$ of a MOOP is *pareto-optimal*, if no other solution $y$ exists that pareto-dominates it.

$$x \text{ is pareto-optimal} \qquad \neg\exists y \succ x \tag{2.2}$$

$$x \text{ pareto-dominates } y = x \succ y = \begin{array}{lll}\exists i & \sigma_{y,i} < \sigma_{x,i} & \wedge\\ \forall i & \sigma_{y,i} \leq \sigma_{x,i} & \end{array} \tag{2.3}$$

The concept of Pareto Dominance introduces an ordering between multiple Multi-Objective solutions. This describes how one solution can be better than another solution in every respect, if it is better (for minimisation problems smaller) in at least one objective value and at least equal in all other objective values (2.1.2).

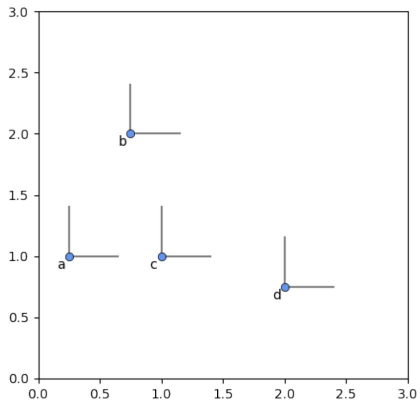


Figure 2.1: Visualisation of Pareto Dominance for Minimisation, $a$ dominates $b$ and $c$, but not $d$

The set of pareto-optimal solutions is the *Pareto Front*, which is typically the set of solutions optimisation algorithms try to approximate to present a DM with, so that they can make an informed decision and select the final solution(s).

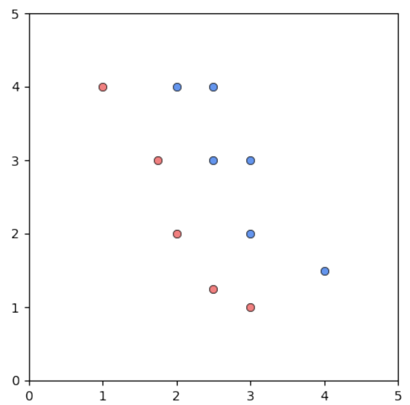


Figure 2.2: Visualisation of a Pareto Front for Minimisation

### 2.1.3 Evolutionary Algorithms

The optimisation methods used in this thesis, Evolutionary Algorithms (EAs) are a group of meta-heuristics using evolutionary principles to heuristically find good solutions to, for example, Multi-Objective Optimisation Problems. In this work, if EAs are mentioned, they refer to a subset of EAs, Genetic Algorithms (GAs). GAs takes strong inspiration from natural evolutionary principles like genotype, phenotype, the concept of fitness, genetic mutation and crossover, natural (environmental) selection, populations and generational advancements.

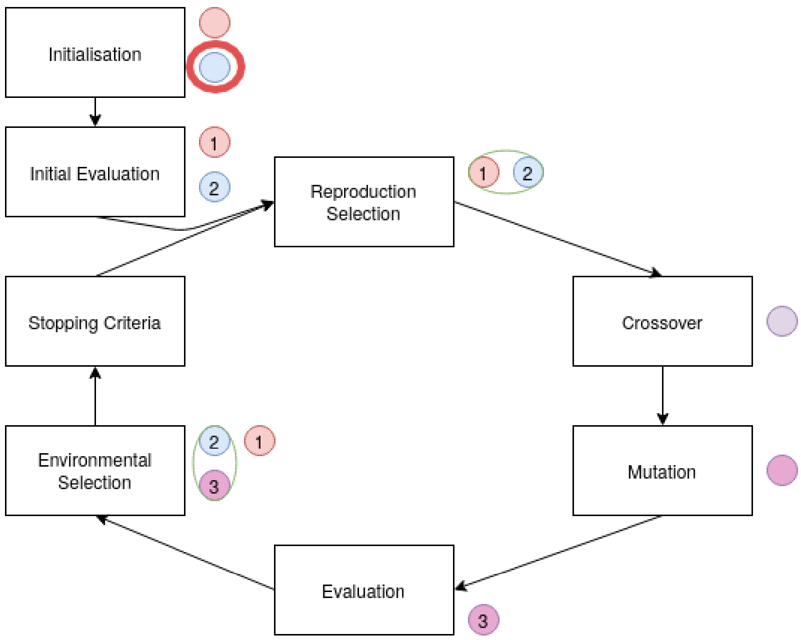


Figure 2.3: Visualisation of a generic Genetic Algorithm's elements and procedure, first a population of solutions is initialised, they are then each assigned a fitness value through evaluation, on which basis they are selected for reproduction, they create an offspring using crossover and mutation, which then gets assigned a fitness as well, the population is reduced through environmental selection based on the individuals' fitness values and finally the algorithm either enters the next generation or stops

As seen in 2.3, a basic approach to a GA is to first initialise a population, evaluate it (therefore assign a fitness to each individual), based on the fitness, select, with a mating selection method, parent individuals to create offspring with both crossover and mutation methods. After also evaluating the offspring, an environmental selection method is employed to keep the population size in check (e.g. to reduce it to the starting size). The cycle of mating selection, creating offspring and environmental selection, which is a *generation*, continues until a termination criterium (e.g. a set number of generations) is reached. Elements of GAs are (often) problem-specific, mainly encoding solutions for a genotype representation and fitness evaluation, but oftentimes also genetic mutation and crossover methods. This is not only due to problem-specific genotypes which are not necessarily able to work with any type of mutation or crossover method, but also due to the possibility to create offspring that are invalid solution, as they e.g. would break problem-specific constraints like a weight limit on a knapsack problem. For that operators oftentimes need problem-specific handling to ensure that either no invalid solutions are created, to "repair" or to penalise invalid solutions.

#### Non-Dominated-Sorting-Algorithm-II

An example of a popular, widely used EA, which also is the basis for all presented algorithms and their base algorithms, is the Non-Dominated Sorting Algorithm II (NSGA-II) [4]. Parent selection is based on binary tournament selection, where two individuals are chosen from the population at random and, based on a criterion, the better solution is added to the parent set. Pareto-Dominance (see 2.1.2) is used as the selection criterion and, as a tie-breaker, crowding-distance is utilised. As seen in algorithm 1, NSGA-II uses fast non-dominated sorting (FNDS [4]) to sort solutions into the respective pareto fronts, assigning each a rank based on the front they are sorted into. For selection, NSGA-II takes as many fronts (top-down) as possible, until there would be too many solutions selected (see 2.4). The last front, which was not added, gets sorted by highest crowding distance of each solution, so that a high diversity is ensured when adding the remaining solutions to the selected set.

## 2.2 Multi-Criteria Decision-Making

The term Multi-Criteria Decision-Making (MCDM) refers to solving or optimising MOOPs for a, typically human, Decision Maker (DM), who has goals

**Algorithm 1** The Non-Dominated Sorting Algorithm II (NSGA-II), operating on population $P_g$ for generation $g$, *parent_select* is a tournament selection, *generate_offspring* is problem-dependent [4]

$PA_g \leftarrow parent\_select(P_g)$ ▷ *Select parent individuals*
$O_g \leftarrow generate\_offspring(PA_g)$ ▷ *Generate offspring*
$F \leftarrow fnds(P_g \cup O_g)$ ▷ *Use FNDS to sort into fronts*
$P_{g+1} \leftarrow \emptyset, i \leftarrow 1$
**while** $|P_{g+1}| < n_{\text{pop}} \wedge |P_{g+1}| + |F_i| \leq n_{\text{pop}}$ **do**
    $P_{g+1} \leftarrow P_{g+1} \cup F_i$ ▷ *Add fronts until*
    $i \leftarrow i + 1$ ▷ *population would grow too large*
$CD \leftarrow cd(F_i)$ ▷ *Calculate crowding-distances for last, partially included front*
$F_i \leftarrow sort\_by(F_i, CD)$ ▷ *Sort front by crowding-distances*
$P_{g+1} \leftarrow P_{g+1} \cup F_i[1 : (n_{\text{pop}} - |P_{g+1}|)]$

or preferences as to what values are supposed to be reached in the optimisation process, which in this thesis helps to focus the optimisation search on a specific area of the Pareto Front. MCDM is commonly defined by helping the DM in their decision process, which is typically classified in 4 distinct approaches [21]: *no preference*, *a priori*, *a posteriori* and *interactive* methods. If there is no preference information of a DM available, *no preference* methods are used to select a neutral solution, compromising between objective functions. *A priori* methods utilise a DM's stated-*prior* preference information to present the DM a single / multiple preference-appropiate solution(s). *A posteriori* methods on the other hand rely on the DM to select from a subset of (pareto-)optimal solutions *after* the fact. Lastly, if a DM is able to participate during the optimisation process, *Interactive* methods are a possibility for the DM to adapt their preferences based on presented solutions and further explore the solution space.

## 2.3 Fairness and Gain

*Fairness* and *Gain*, providing the basis of the filtering algorithm the presented algorithms adapt, aredefined in the report of the Dagstuhl Seminar 20031 [8] and based on the metric of Pareto Regret (see 2.3), calculated using the preferences of a group of Decision Makers (DMs). Pareto Regret [11] is defined

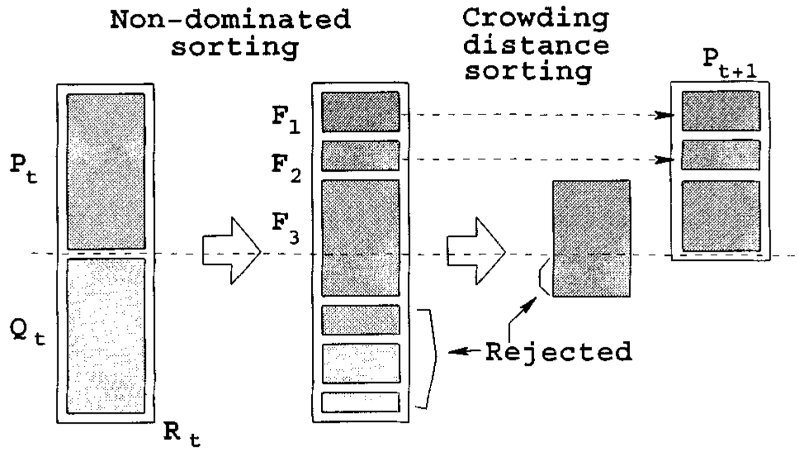


Figure 2.4: Visualisation of ranking and selection in NSGA-II, the last front is only partially included based on crowding distance [4]

as a loss metric of a given solution $x$ over an ideal solution $r$. In the report, the ideal solution is a DM-specified reference point and Pareto Regret is the Euclidean distance from $x$ to $r$.

$$pr(x, R_m) = \sum_{i=1}^{m} (f_i(x) - R_{m,i}) \tag{2.4}$$

Gain is then based on Pareto Regret ($pr$), averaging Pareto Regret (Average Pareto Regret $apr$) values of the solution $x$ in regard to each reference point. Fairness is the summed inequality (Inequality in Pareto Regret $ipr$), meaning the deviation from the average, of Pareto Regret of a solution $x$ in regard to each reference point $R_m$.

$$apr(f(x), R) = \frac{1}{d} \sum_{i=1}^{d} pr(f(x), R_i) \tag{2.5}$$

$$ipr(f(x), R) = \sum_{i=1}^{d} |pr(f(x), R_i) - apr(f(x), R)| \tag{2.6}$$

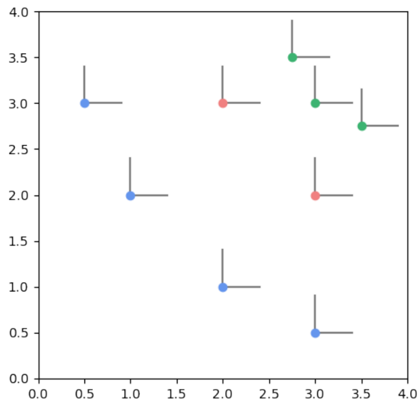


Figure 2.5: Visualisation of the fronts in NSGA-II, different colours represent different fronts

## 2.4 Multi-Agent Pathfinding

The field of MAPF deals with pathfinding problems for multiple agents, to find efficient paths without crossover or collisions to reach their goal from a set starting position, typically as a Single-Objective problem. Commonly one of the following objective functions is used [9]:

- *Summed cost*, the sum of all path's time cost
- *Makespan*, the maximum cost of a path / the time until all agents have reached their goal

Additionally, a common way to ensure solution quality is to discard solutions that include collisions with either the environment or other agents (or, the other way around: only collision-free solutions are considered valid).

### Multi-Objective Multi-Agent Pathfinding

Proposed by Mai et al. [23] [14] [15], the MOMAPF(-Vehicle-Model) problem deals with MAPF in a novel way by including multiple objectives during optimisation, with the goal to create an easily scalable benchmark problem to test optimisation algorithms against.

The MOMAPF problem referred to in this work (and used in the experiments) is based on the work by Mai and Mostaghim [15], but also modified in the sense that liberties have been taken to e.g. only use a subset of objective functions defined by the author.

In MOMAPF, in this work, multiple agents are placed on a square map, consisting of grid cells, with obstacles, each with it's respective starting and goal positions, such that agents from both sides must cross the map to reach their respective goals. A solution for MOMAPF is encoded as a set of waypoints through which the agent navigates from start to goal (see 2.6).

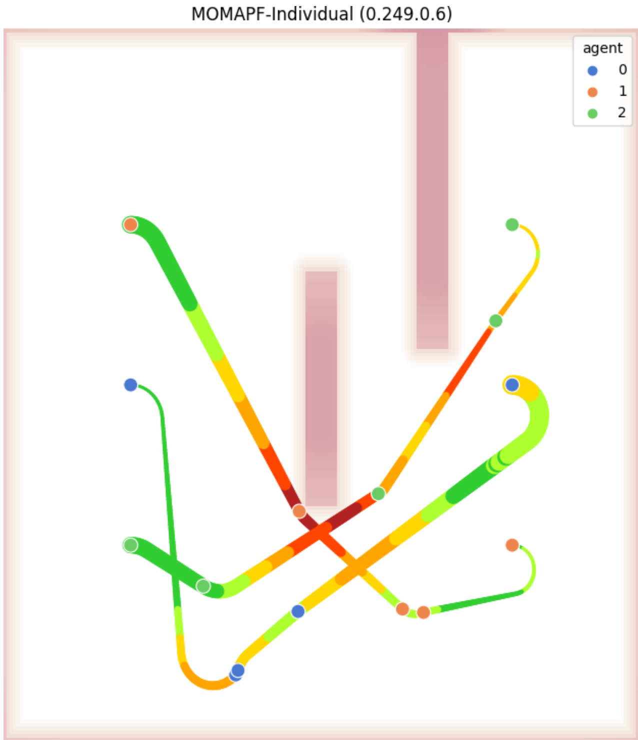


Figure 2.6: Visualisation of a solution for MOMAPF, the agents travel from one side of the map to the other, risk of each agent is indicated by how red the agent's path's colour is, passed time is indicated by thickness of the agent's path

Decoding a solution is accomplished by employing an algorithm to find the shortest path in accordance with the used vehicle model connecting the waypoints.

Different vehicle models for planning agent navigation can be used:

- *Straight*, the waypoints connect by linear movements

- *Rotate Translate Rotate*, the agent can turn in place, moves linearly to the waypoint and rotates afterwards
- *Dubins*, the agent always moves forward in straight lines and circle segments
- *Reeds-Shepp*, similar to dubins the agent can however change direction
- *Adaptive Dubins*, like *Dubins*, but with velocity based circle segments (the faster the agent goes the bigger the circle has to be)
- *Adaptive Reeds-Shepp*, analogous to *Adaptive Dubins*

A solution vector $x$ contains multiple objective values.

$$\emptyset \neq \vec{f}(x) \subseteq \{f_R(x), f_T(x), f_L(x)\} \tag{2.7}$$

For optimisation using EAs, multiple options for objective functions are defined and can be mixed and matched as defined in the equations below.

$$\text{risk} \quad f_R(x) = d_{\max} - d_A(x) \tag{2.8}$$

$$\text{time} \quad f_T(x) = \frac{\sum_{i=0}^{k} \tau_i}{k} \tag{2.9}$$

$$\text{length} \quad f_L(x) = \frac{\sum_{i=0}^{k} \sum_{t=0}^{\tau_i} |\vec{x}_i(t) - \vec{x}_i(t+1)|}{k} \tag{2.10}$$

$$\text{min. distance of agents} \quad d_A(x) = min( \begin{array}{l} \underset{\forall i \neq j, \forall t}{min} (|\vec{a}_i(t) - \vec{a}_j(t)|), \\ \frac{1}{2} \underset{\forall i,j,t}{min} (|\vec{x}_i(t) - \vec{o}_j|) \end{array} ) \tag{2.11}$$

As can be seen in figure 2.7, MOMAPF defines different map types:

- *empty*, no obstacles are on the map

- *double_gap*, a number $n_{\text{gaps}}$ of gaps are in a rectangular obstacle in the middle of the map
- *labyrinth*, two rectangular obstacles flush with either the top or the bottom to form a winding tunnel

Additionally, a *bar* of length $l_{\text{bar}}$, can be placed on the map as a rectangular obstacle.

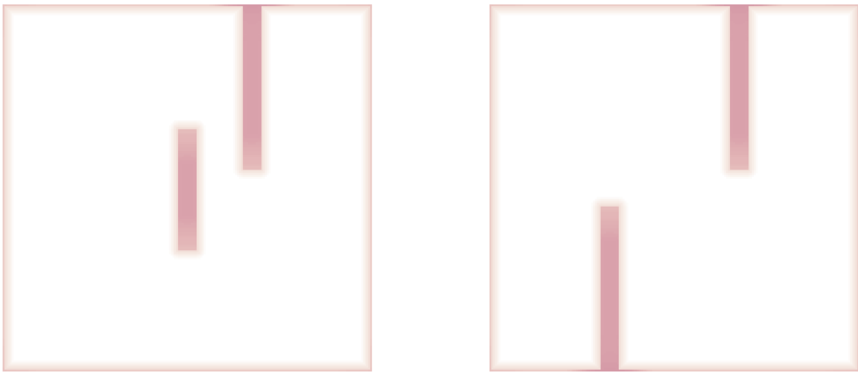

Figure 2.7: Available map types in MOMAPF, additionally an empty map can be chosen, also a rectangular obstacle (bar) can be placed on the map

## 2.5 TNK

As described by Tanaka et al. [20] in 1995, the *TNK*-problem is a multi-objective benchmark problem to test optimisation methods. Below are the objective functions and their constraints, figure 2.8 shows the Pareto Front.

$$\begin{aligned}
&\text{minimise} \quad f_1(x) = x_1,\ f_2(x) = x_2 \\
&\text{constrained by} \quad c_1: \quad x_1^2 + x_2^2 - 1 - 0.1\cos(16\arctan(\tfrac{x_1}{x_2})) \geq 0 \\
&\text{and} \quad c_2: \quad (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5 \\
&\text{and} \quad c_3: \quad 0 \leq x_1, x_2 \leq \pi
\end{aligned} \tag{2.12}$$

This work's implementation assign's a penalty term of $x_{\text{new}} = x_1 + 1, x_2 + 1$, if solutions violate constraints.

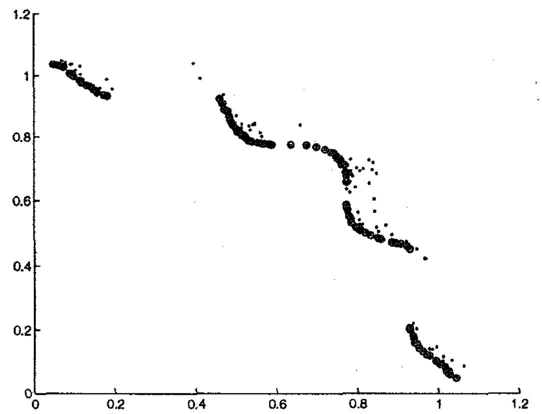


Figure 2.8: The approximated pareto front of the tnk problem [20]

# 3 Related Work

This work relates to other work in three ways: It relates to similar approaches for solving Multi-Objective Optimisation Problem (MOOP) using Evolutionary Algorithms (EAs) including a Decision Maker (DM)'s preferences (3.1), as well as Multi-Criteria Decision-Making problems with multiple DMs trying to achieve high consensus among DMs (3.2). Secondly, it relates to the base work, the report from the Dagstuhl seminar 20031 [8], and the master thesis [6], written by Djartov, introducing the explicit usage of fairness and gain in EA (3.4). Lastly, this work relates to solving pathfinding problems with EAs or by including DM preference, but explicitly the (scalable) MOMAPF problem [15] (3.3).

## 3.1 Including Decision Maker's Preferences in Evolutionary Algorithms

For including the preference of DMs in optimisation using EAs, several approaches have been made, mostly focussing on *a priori* (see 2.2) methods, to further guide the set of solutions returned by the algorithm to better represent the DM's goals. As to the author's knowledge however, no research (aside from the report from the Dagstuhl seminar, see 3.4) with respect to including and optimising for fairness and gain using EAs has been done. Some of the presented works do however have the ability to achieve some degree of consensus between DMs, which is similar to the combination of fairness and gain, as the approaches try to both achieve convergence (in that sense the "gain") and respect the preferences of all DMs ("fairness").

Thiele et al. [21] explore an interactive method for incorporating DM preference in optimisation with EAs. The author's approach employs an achievement scalarisation function, which through the use of weight values aggregates a multi-objective problem into a single-objective problem and then "projects" the reference point to the pareto front. This function, combined with an indicator-based EA (IBEA), leads to the algorithm now respecting the DM's preference (therefore named PBEA). The need for interactive approaches is emphasized, as a priori methods highly depend on the DM to express realistic and optimal preferences, for which the authors introduce an interactive procedure to use PBEA, in which the DM is asked to specify a reference point, the algorithm is being executed and the DM is then presented the solution with the smallest achievement function value and can decide to again re-specify a reference point or to end the algorithm. As the author's approach circumvents the multi-objective nature of MOOPs by using a scalarisation function, PBEA does inherently not work towards the goal of this thesis's goals, the interactive way of specifying reference points however could be incorporated into the presented algorithms and be subject of future research.

Branke et al. [2] integrate user preference a priori into an EA by defining minimum and maximum tradeoff functions, which specify the DM's desire to tradeoff between objective functions. Based on these tradeoff functions the authors specify the Guided Multi-Objective-EA (GMOEA), which changes domination criteria by introducing different (wider) domination cone angles (compare for 2.1). The change in domination leads to specific areas of the pareto front being emphasized and quicker convergence to these areas. GMOEA appears interesting, it however completely changes domination alltogether, making it less easy to change components of the employed EA to further improve on the optimisation procedure.

Mohammadi et al. [16] describe an extension to MOEA/D [27], which decomposes MOOPs into single-objective sub-problems and optimises simultaneously. The extension (the algorithm is then named R-MEAD) lies in the inclusion of (multiple) DM preference(s) to solve a shortcoming of MOEA/D, namely that there is a high number of sub-problems to approximate the entire pareto front, by only approximating a specific region of the front through the use of a smaller set of weight vectors and preference information. The smaller set of weight vectors leads to a smaller number of sub-problems, the weight vectors get adapted during evolution by calculating the update-direction through euclidean distance of solutions to the closest reference point. This results in multiple solution sets, each near a DM's reference point, it however does not lead to a consensus between DMs.

Deb et al. [5] also suggest an approach based on multiple reference points to better guide an EA towards the DM-preferred sub-set of the pareto front. The authors focus specifically on finding a set of solutions instead of a singular solution, which is a common approach in regards to combining DM-preference with MOOPs, to find the exact "best" solution with regard to the preference information (e.g. as the best solution of an achievement scalarisation function). To find a sub-set of solutions, the authors describe an extension to NSGA-II's niching process (see 1), where solutions are clustered to ensure diversity along the pareto front, which emphasizes solutions close to the reference points (euclidean distance) and clusters very similar solutions (the extension is named R-NSGA-II). The authors conclude that the approach works well to find several sub-sets along the reference points, the sub-sets are notably near each reference point however, without any consensing solutions inbetween points.

Tan et al. [19] explore a two-stage approach to integrating DM preference into the optimisation process. By defining multiple goals for the algorithms to reach (e.g. an objective value $\sigma_i < 10$), the DM's preference is considered while searching for solutions by trying to fulfill the given goals. Additionally, the goals are connected with logical connections (OR and AND) and can be assigned with different priorities. The employed EA utilises a novel ranking scheme, based on the number of goals fulfilled by a solution and their respective given priorities. While the authors do not directly compare their novel algorithm with established algorithms, they conclude that it works well on benchmark problems. Interestingly, this approach, based on the logical operation connecting goals, can achieve consensus between different preferences (with the logical and operation).

## 3.2 Maximising Consensus for Decision Makers in Multi-Objective Problems

For creating consensus for the preference of multiple DMs for MOOPs, some research exists in MCDM-literature, however not necessarily in the same sense as this work tries to pursue. Some efforts have been made towards consensus based on fuzzy preferences, but only little research has been done in the field of MCDM, as to the author's knowledge, with regards to achieving consensus, namely that all DMs' preferences are considered to some degree, during optimisation. Additionally, *partial* DM preferences, as presented in this thesis, appears to be a novel concept, without any prior research available.

Baek et al. [1] propose an algorithm to help multiple DMs choose a consensus, non-dominated solution, by assigning preference information in the form of membership functions, which quantifies how much a solution is accepted by the DM, and through the usage of a heuristic for searching weight values for the optimisation process. The algorithm is semi-interactive, iterating over values and asking the DMs if the found solution is acceptable or if to continue. Minimising the average membership function deviation from target values, the algorithm searches for a consensus between all DMs.

Emmerich et al. [7] describe a method to model consensus on decisions for multiple DMs. This is not explicitly an algorithm to search for solutions looking for a high consensus, but rather another way of modelling said consensus. Preference information is given in the form of desirability functions, which describe the probability of a DM to accept a given solution. Consensus is the cumulative probability of all DMs to accept the solution at hand, from which the expected number of DMs to accept the solution can be approximated through the use of a Monte Carlo Method. The authors give examples of how the proposed preference model can be used in the selection process in databases, but also mention how a demanding DM with an acceptance probability of 0 immediately discards the solution, regardless how well accepted it is by other DMs.

Pfeiffer et al. [18] emphasize that previous attempts at integrating multiple DM preference into EAs where lacking, especially in regards to not finding a real consensus between DMs. To elleviate this, the authors test several modifications of the existing reference point algorithm by Deb et al. [5] (see 3.1) to better solve for multi-DM-consensus. Two approaches are suggested: A ranking based approach, which copies the original ranking, based on the distance to the reference points, but modifies it to assign different values (maximum or average instead of minimum) as the crowding distance. The second approach does away with the ranking portion of the original's niching operator and only considers the maximum / average Euclidean distances towards reference points. Concluding, this approach is a modification to the original algorithm that now includes consensus between the multiple reference points, by trying to minimise the distance towards each.

As for another approach integrating consensus into the optimisation process, Xiong et al. [25] point out that it's difficult for DMs to express their preferences precisely. The authors integrate DM-preference through the means of fuzzy numbers and suggest to measure solution robustness to lessen the impact of change in DM-preferences. To optimise final solution selection towards both consensus (fuzzy distance sum) and robustness (minimum transition cost in decision space, if solution is changed in objective space), NSGA-II is used in a modified way. In addition to regular rank assignment, after assigning ranks based on dominance (see 1), all ranks are increased by one and solutions from the original first rank are placed in the new first rank, if they are also non-dominated with respect to consensus and robustness. As only little comparison to other algorithms with integrated DM-preferences is drawn, the performance of the algorithms in comparison to others is left unexplored. Resulting Solutions focus on the area indicated by reference points, however without consensus between DM-preferences, focussing more on areas near each of the reference points.

## 3.3 Multi-Agent Pathfinding

Multi-Agent Pathfinding (MAPF) is a topic of high relevance, as automated robotic paths are highly important in e.g. warehouse-like application and therefore also one of a lot of research.

Traditionally, search-based solvers are widely used in MAPF, as there exist a lot of optimal solvers, as shown in the survey of Felner et al. [9]. They define the MAPF problem to include multiple agents with starting and goal positions, with a conflict occuring if two agents are on the same vertex at the same time and the objective being either to minimise the sum of costs or the maximum cost (time) of all paths in a solution, but it is also mentioned that makespan (path length) is also a commonly used objective value. Whle sub-optimal solvers are also shown, optimal solvers include reduction-based solvers, which reduce the NP-hard problem of MAPF to another NP-hard problem and then solve that with an appropriate solver, Another type of solvers are ones based on the A*-algorithm, with the drawback of an exponential increase in time based on the number of agents, increasing cost tree search, or conflict based search approaches. The authors conclude that instead of creating new solvers, existing ones should be examined for possibilities of new optimisations, as these are already shown to perform well.

### 3.3.1 Decision-Making and Multi-Agent-Pathfinding

As for the crossover of both the fields of Multi-Agent Pathfinding and Decision-Making, most typically the decision makers are agents in the pathfinding problem and try to find compromises to achieve a high solution quality. Less common, Decision-Making in regard to Multi-Objective Pathfinding is also present in the field.

In the master thesis of Wolters [24] an algorithm is suggested to, through collective Decision-Making of the multiple agents, search for paths with a highway layout. Such a layout includes a highway, which is a path that the agents agreed upon to be fit for their own preferences and does not need to be individually changed by each agent. The Decision-Making process includes each agent voting on parts of the highway-layout, their votes based on a modified $A*$-algorithm to find their preferred next segment of the highway.

The bachelor thesis of Partes [17] deals with Decision-Making for pathfinding as a MOOP, defining an a priori algorithm to further narrow down a larger set of pareto-optimal solutions. Three approaches are suggested to better choose from the larger set: The first approach solely in objective space, utilising pareto-dominance (see 2.1), another in decision space, utilising clustering and finally a combination of the other two approaches.

### 3.3.2 Multi-Objective Multi-Agent Pathfinding

The benchmark problem in this work is the *Scalable Multi-Objective Multi-Agent Pathfinding* problem, which integrates Multi-Objective Optimisation with Multi-Agent Pathfinding, by including multiple, instead of, as is done traditionally, only one, of the typical objectives for optimisation.

In the work of Weise et al. from the [23] the original scalable MOMAPF problem is defined. It stands in contrast to most other MAPF problems as it optimises for multiple objectives instead of a single one, namely path flow-time, the average length of the agent's paths, makespan, the time needed until the last task is finished and the number of collisions. The last objective is traditionally a constraint on the optimisation problem, only solutions without collisions are typically considered valid [9], the inclusion as an objective however leads to the algorithm being able to explore different degrees of "invalidness", with partially good solutions not being discarded for being invalid. The search space for paths is restricted by a small number of waypoints, of which the encoded solutions (as a genotype) consist of (see 2.4).

Later on, in the work of Mai and Mostaghim [14], new concepts get introduced, in both scenario and agent vehicle models. The scenario of this work is defined through the agents being placed evenly on both sides of a square map with obstacles on it and the agents having to navigate from their starting position on the one side to their goal position on the other (see 1.1). Several vehicle models are introduced, the main ones being:

- *Straight*, the waypoints connect by linear movements
- *Rotate Translate Rotate*, the agent can turn in place, moves linearly to the waypoint and rotates afterwards
- *Dubins*, the agent always moves forward in straight lines and circle segments
- *Reeds-Shepp*, similar to dubins the agent can however change direction

The unpublished work of Mai [15] defines the MOMAPF-Vehicle-Model problem, which is a combination of both the MOMAPF problem and the previously defined vehicle models [14] together with additional extensions. In addition to the previously defined vehicle models, adaptive versions of both *Dubins* and

*Reeds-Shepp* are introduced, which adapt the circle segment radius to the speed of the agent. The author also introduces velocity profiles as a new factor to optimise for, as now agents can navigate in different speeds. The objectives to optimise for are average path length, average travel time (which can be different to the average path length due to velocity profiles) and risk (based on the number of collisions and distances to obstacles).

## 3.4 Fairness and Gain in Evolutionary Multi-Objective Optimisation

Fairness in regard to the preferences of multiple DMs has, as to the author's knowledge, not been explored before the report from the Dagstuhl Seminar 20031 [8] in this particular way. There has been exploration towards Multi-DM consensus (see 3.2), that is however similar but not the same as fairness, as traditional preference is not concerned with the inequality in how each DM's preference is fulfilled. Fairness in different contexts has been used as an overall goal of optimisation, e.g. for optimising the fairness of electric vehicle charging pricing [13] or for optimising the fair distribution of water [22], or as a concept in optimisation procedures, e.g. for a fair approach to create offspring in EA [12], but not as an explicit goal in the algorithm itself.

### 3.4.1 Reference Points and NSGA-II for Teams

The report of the Dagstuhl Seminar 20031 [8] explores the concept of fairness and gain as a method to better integrate multiple DM' preferences into EAs. The authors use the concept of pareto regret (see 2.3), the loss to the ideal value, in combination with reference points to both express the DM's preference and the DM's loss in regards to their preference. Fairness and Gain are defined as the Average Pareto Regret and Inequality in Pareto Regret (see 2.3) between DMs. Based on fairness and gain, the algorithm Non-Dominated Sorting Algorithm II for Teams (NSGA-II for Teams) (see 2) is described, which uses NSGA-II in combination with a fairness-and-gain-based sorting. The authors conclude through a small example benchmark that the described algorithm does work as expected, but needs further testing.

### 3.4.2 Adaptive NSGA-II for Teams

The master thesis of Djartov [6] further explores Fairness, Gain and NSGA-II for Teams, and further extends the algorithm. The author extends NSGA-II for Teams with reference points (named preference points) that wander into the direction of the pareto front (Adaptive NSGA-II for Teams), as he found that, given non-ideal reference points, the algorithm was incentiviced to converge in the neighbourhood of these reference points and not towards the pareto front (see 4.1). The algorithms were tested on benchmark functions and compared with NSGA-II, coming to the conclusion that, while better for converging on a specific, preferred area, the algorithms, by the nature of their design, were worse for overall solution diversity. The author encourages that the algorithms were tested on more (practical) benchmark problems, as to further test their usability in the field, which this work also tries to accomplish (see 5.2).

# 4 Methodology

For this work's methodology, first the groundlaying algorithms (NSGA-II for Teams and Adaptive NSGA-II for Teams) by Djartov [6] are described and examined (and slightly adapted), then the motivating problem and, based on that, a new categorisation for optimisaton problems is described in detail and then, finally, the algorithmic extensions and the new algorithms, Win-Win NSGA-II for Teams, it's Adaptive variant and Cosine Pareto Regret, are described.

## 4.1 NSGA-II for Teams

The Non-Dominated Sorting Algorithm II for Teams (NSGA-II for Teams) is the algorithm adapted for this work, to better include individual preferences in partitionable MOOPs (see 4.2), by using different $fairness$ and $gain$ equations. This facilitates a better match of interest or preferences in only a part (e.g. a single agent's travel time, see 1.2) of the objective value, instead of operating on aggregated objective values (see 1.1). The algorithm is originally described in Djartov's master thesis from 2021 [6] (and to a lesser extent in the report from the Dagstuhl seminar 20031 [8]), it is designed to guide the Evolutionary Algorithm (EA) towards an area of fair but good (in terms of fulfillment of any DM's preference) for all participating Decision Makers (DMs) using their preference information, specified as reference points $R$ (named preference points in the master thesis) in objective space. NSGA-II for Teams is strongly based on NSGA-II, it however adds a fairness filter taking place before sorting solutions into fronts (see 2). The filter first computes the fairness and gain values (originally only $apr$ and $ipr$, as in 2.3) of all individuals in the input set, ranks them with fast non-dominated sorting from NSGA-II [4] and then generates the filtered set using the same approach as NSGA-II (including fronts and including some individuals using crowding distance, see 1), but now based on fairness and gain values instead of the individuals' fitness.

**Algorithm 2** NSGA-II for Teams, adding fairness_filter (3) to NSGA-II[6]

1: $PA_g \leftarrow parent\_select(P_g)$ ▷ *Select parent individuals*
2: $O_g \leftarrow generate\_offspring(PA_g)$ ▷ *Generate offspring*
3: $S_g \leftarrow fairness\_filter(P_g \cup O_g, R)$ ▷ *Filter using reference points*
4: $F \leftarrow fnds(P_g)$ ▷ *Use FNDS to sort into fronts*
5: **while** $|P_{g+1}| < n_{\text{pop}} \wedge |P_{g+1}| + |F_i| \leq n_{\text{pop}}$ **do**
6: $\quad P_{g+1} \leftarrow P_{g+1} \cup F_i$ ▷ *Add fronts until*
7: $\quad i \leftarrow i + 1$ ▷ *population would grow too large*
8: $CD \leftarrow cd(F_i)$ ▷ *Calculate crowding-distances for last, partially included front*
9: $F_i \leftarrow sort\_by(F_i, CD)$ ▷ *Sort front by crowding-distances*
10: $P_{g+1} \leftarrow P_{g+1} \cup F_i[1 : (n_{\text{pop}} - |P_{g+1}|)]$

In the original algorithm, the filter would reduce the input set of individuals to the original population size $n_{\text{pop}}$, effectively side-stepping the original NSGA-II environment selection mechanism. This work adapts NSGA-II for Teams's $fairness\_filter$ to include a filter rate $r_{\text{filter}}$ for reducing the input set's size $|S|$ to $n_{\text{out}} = max(|S| \cdot r_{\text{filter}}, n_{\text{pop}})$ $(0 < r_{\text{filter}} \leq 1)$, as seen in algorithm 3.

This filter rate is utilised with the goal to reach a higher convergence rate towards the Pareto Front, as the NSGA-II selection procedure, using $fnds$ and crowding-distance, is not side-stepped this way and can apply selection pressure and elitism to the population without solely relying on good reference points to guide the approximated front. An example as to how the filter rate affects convergence, using the simple TNK benchmark-problem (see 2.5), can be seen in 5.1.

## Adaptive NSGA-II for Teams

A problem with NSGA-II for Teams observed by Djartov is the reliance on good reference points in regards to finding the true Pareto Front. As can be seen in figure 4.1, if reference points are chosen sub-optimally, solutions can potentially never converge on the true Pareto Front, as the algorithm only wants to satisfy the supplied DM's preferences and not necessarily find an approximation of the Pareto Front.

**Algorithm 3** $fairness\_filter$, operating on a set of individuals $S$, used in NSGA-II for Teams (2) [6], with a novel filter rate $r_{\text{filter}}$ to control the output size

$S$ - set of solutions, $R$ - set of reference poitns
$gain = apr$, $fairness = ipr$
$S_f, PR, G, FA \leftarrow \emptyset$ ▷ *Initialise filtered set, pareto regret, gain and fairness*
$n_{\text{out}} = max(|S| \cdot r_{\text{filter}}, n_{\text{pop}})$ ▷ *Amount of output solutions*
**for** $s \in \{1, \ldots, |S|\}$ **do**
    **for** $r \in \{1, \ldots, |R|\}$ **do**
        $PR_{s,r} \leftarrow pr(S_s, R_r)$
    $G_s \leftarrow gain(S_s, PR)$
    $FA_s \leftarrow fairness(S_s, PR)$
$FG \leftarrow \{(G_1, FA_1), \ldots, (G_{|S|}, FA_{|S|})\}$ ▷ *Zip together the respective fairness and gain values for each s*
$F \leftarrow fnds(S, FG)$ ▷ *Use fnds with values based on FG to sort S into fronts*
$i \leftarrow 1$
**while** $|S_f < n_{\text{out}} \wedge |S_f| + |F_i| \leq n_{\text{pop}}$ **do**
    $S_f \leftarrow S_f \cup F_i$ ▷ *Add fronts until*
    $i \leftarrow i + 1$ ▷ *population would grow too large*
$S_f \leftarrow S_f \cup F_i[1 : (n_{\text{out}} - |S_f|)]$

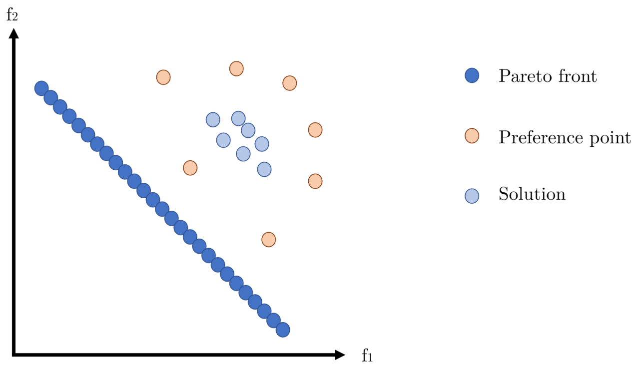


Figure 4.1: A problematic situation for NSGA-II for Teams, as it's reference points are chosen in a way that it can't converge to the true Pareto Front [6]

As DMs typically want to approximate the true Pareto Front but reference points are difficult for a DM to evaluate *a-priori*, which leads to reference points with unintentional emphasis on certain regions, Djartov [6] describes an extension to NSGA-II for Teams to reposition reference points during algorithm execution. The extended algorithm is the Adaptive NSGA-II for Teams, as seen in algorithm 4, which can reposition reference points at specified numbers of passed generations.

Solutions are ranked and sorted using $fnds$ and solutions from which to select the reference points are selected from the first few fronts. The solutions with the smallest distance to the old reference points are selected as the new reference points and the regular algorithm continues.
Adaptive NSGA-II for Teams has been adapted in this work to not reposition at only specific numbers of passed generations, but rather to reposition in a specified interval $n_{\text{gen\_repos}}$ of generations to better guide the approximated front towards the true Pareto Front, as can be seen in algorithm 5.

## 4.2 Partitionable Multi-Objective Optimisation Problems and Egoistic Pareto Regret

In order to evaluate individual aspects of a solution, aggregating objective functions, which generate objective values by aggregating several other values, are unfit, as they take away important detail for each aspect of that solution

**Algorithm 4** Adaptive NSGA-II for Teams is an extension to the regular algorithm (2) [6], adding a repositioning step for reference points with $reference\_reposition$

```
PA_g ← parent_select(P_g)                          ▷ Select parent individuals
O_g ← generate_offspring(PA_g)                     ▷ Generate offspring
R ← reference_reposition(R, P_g ∪ O_g)             ▷ Reposition reference points based on generation
S_g ← fairness_filter(P_g ∪ O_g, R)                ▷ Filter using reference points
F ← fnds(P_g)                                      ▷ Use fnds to sort into fronts
while |P_{g+1}| < n_pop ∧ |P_{g+1}| + |F_i| ≤ n_pop do
    P_{g+1} ← P_{g+1} ∪ F_i                        ▷ Add fronts until
    i ← i + 1                                      ▷ population would grow too large
CD ← cd(F_i)      ▷ Calculate crowding-distances for last, partially included front
F_i ← sort_by(F_i, CD)                             ▷ Sort front by crowding-distances
P_{g+1} ← P_{g+1} ∪ F_i[1 : (n_pop − |P_{g+1}|)]
```

**Algorithm 5** $reference\_reposition$ used in Adaptive NSGA-II for Teams ([6], named Preference Repositioning), modified for repeated repositionings based on $n_{\text{gen\_ repos}}$

```
D, F, RN ← ∅                     ▷ Initialise distances, fronts and new reference points
FNDS ← fndsS                     ▷ Get non-dominated solutions from S
i ← 1
while |F| < |R| do               ▷ Add more than the first front
    F ← F ∪ FNDS_i
    i ← i + 1
if (g mod n_gen_repos) = 0 then
    for f ∈ {1, ..., |F|} do
        for r ∈ {1, ..., |R|} do
            D_{r,f} ← dist(F_f, R_r)            ▷ Compute all distances for R and F
    for r ∈ {1, ..., |R|} do
        RN_r ← F[argmin(D_r)]   ▷ Possible reference points by shortest distance
        if RN_r ≻ R_r then                     ▷ Replace if RN_r dominates R_r
            R_r ← RN_r
```

(e.g. average travel time of a set of agents vs. the travel time of a single agent), where it can e.g. smooth away outlier values for a specific aspect. Partitionable Multi-Objective Optimisation Problems (MOOPs) are MOOPs with aggregating objective functions for each objective value $\sigma_i$, which are used to generate the objective value from sets of base values $\Phi_i$. These base values can be obtained and used in algorithms, which makes the problem *partitionable*, it is possible to take the original, *partial* (base) objective values and include them in the optimisation process.

$$\begin{aligned}&\text{Partitionable Multi-Objective Optimisation Problem:}\\&\text{minimise}\quad f(x) = (\sigma_1, \ldots, \sigma_m)\\&\sigma_i = agg_i(\Phi_i) = agg_i(\{\phi_{i,1}, \ldots, \phi_{i,n}\})\end{aligned} \tag{4.1}$$

An example of a partitionable MOOP would be the Multi-Objective Multi-Agent Pathfinding (MOMAPF) problem (see 2.4). The solution to a MOMAPF problem with 3 agents (2.6), which considers only $f_R$ and $f_T$ consists of two objective values: Risk ($f_R$) and time ($f_T$) (see 2.4). Both risk and time are objective values which have been aggregated from other, partial objective values for each agent. Risk is (in a simplified way) the minimum distance at any time-step of all three agents to either an obstacle or another agent, in a simplified way the aggregation here is $min()$ and the *partial* objective values are each the minimum distance of the agent. Time on the other hand is the average of all agents' travel times and the partial objective values are each agent's travel time.

If a DM is only interested in the risk and travel time of a specific agent, the aggregation is detrimental to how the DM can express their preference regarding a specific reference point in objective space. Therefore this work presents a novel version of Pareto Regret, named Egoistic Pareto Regret ($epr$) due to the egoistic nature of the DM, as they only care about their own, specific aspect of the solution.. It is defined on partial objective values instead of regular objective values which were aggregated of partial objective values.

$$\begin{aligned}
epr(f(x), R_i) &= epr(\{\sigma_1, \ldots, \sigma_m\}, R_i) \\
&= epr(\{(\phi_{1,1}, \ldots, \phi_{1,i}, \ldots, \phi_{1,n}), \ldots, (\phi_{m,1}, \ldots, \phi_{m,i}, \ldots, \phi_{m,n})\}, R_i) \\
&= \sum_{j=1}^{m} (\phi_{j,i} - R_{i,j}) \qquad (4.2) \\
|R| &= n
\end{aligned}$$

Egoistic Pareto Regret now represents the distance of each DM's preference to their respective aspect of the objective, or *partial* preference, e.g. a specific agent for MOMAPF, of the objective vector.

## 4.3 Win-Win NSGA-II for Teams

As to now include the DM's *partial* preference in the optimisation process, a modification to NSGA-II for Teams is applied, to now calculate pareto regret based on *partial* objective values using $epr$, therefore named Win-Win NSGA-II for Teams. Egoistic Pareto Regret replaces Pareto Regret in both $filter\_fairness$ as well as in $apr$ and $ipr$ as $fairness$ and $gain$ (see 2.3 and 3). Therefore $apr$ becomes $aepr$, $ipr$ becomes $iepr$, therefore $aepr$ replaces the gain and $iepr$ the fairness function (see 6).

**Algorithm 6** $winwin\_fairness\_filter$ Win-Win NSGA-II for Teams replaces $pr$ and the fairness and gain functions in the regular $fairness\_filter$ (see 3) [6]

$S$ - set of solutions, $R$ - set of reference points
$gain = aepr$, $fairness = iepr$
...
**for** $s \in \{1, \ldots, |S|\}$ **do**
  **for** $r \in \{1, \ldots, |R|\}$ **do**
    $PR_{s,r} \leftarrow epr(S_s, R_r)$
  ...
...

The new $winwin\_fairness\_filter$ replaces $fairness\_filter$ in NSGA-II for Teams (2) to form the new algorithm Win-Win NSGA-II for Teams.

### Win-Win Adaptive NSGA-II for Teams

Adaptive NSGA-II for Teams is adapted the same way as NSGA-II for Teams, *epr* replaces *pr*, *aepr* replaces *apr* as gain and *iepr* replaces *ipr* as fairness in $filter\_fairness$. This adaption is therefore named Win-Win Adaptive NSGA-II for Teams.

## 4.4 Cosine Pareto Regret

In a similar vein to Egoistic Pareto Regret, Cosine Pareto Regret is a modification to Pareto Regret, it is however not focussed on *partial* objective values. The goal of Cosine Pareto Regret is to replace the need for the repositioning step of Adaptive NSGA-II for Teams, as it can lead to a change in the DM's preferences, as the DM's reference point might be replaced by the closest non-dominated reference point, which could nonetheless have a different weighting of objective values (which might be undesired). Cosine Pareto Regret aims to solve this by depending on the DM to pick a reference line which represents the "weighting trajectory" on which the reference points of the DM can be imagined to be located. Additionally, this would lessen the problem of picking a specific reference point to an unknown problem topology, where the DM is unsure what dimensions of objective values are fitting to their preferences. By defining the weighting trajectory between objectives, the DM only has to understand how much they prefer one objective over the other.
The reference line is picked by the DM by specifying weights $W = \{w_1, \ldots, w_n | w_i \geq 1.0\}$, indicating how important one solution is to them in relation to other objectives, which get inverted to form the reference line $R = \{\frac{1}{w_1}, \ldots, \frac{1}{w_n}\}$.

Similarity of a solution to a reference line is calculated by calculating the line the solution's objective values are on and then calculating cosine similarity (the cosine of the angle $\alpha$ of both lines) between the two lines. To use cosine similarity in a minimisation context, it is subtracted from 1.

$$cpr(f(x), R_i) = 1 - cos(\alpha) = 1 - \frac{f(x) \cdot R_i}{||f(x)|| \cdot ||R_i||} \tag{4.3}$$

Cosine similarity is chosen to create a variant of Pareto Regret, as the angle between both the reference line and the solution vector is inherently a similarity in terms of solution trajectory during optimisation. Analogous to Egoistic Pareto Regret and *aepr* and *iepr* (see 4.2), *acpr* and *icpr* are both the Average Cosine Pareto Regret and Inequality in Cosine Pareto Regret.

## Cosine NSGA-II for Teams

Again, analogous to Egoistic Pareto Regret and Win-Win NSGA-II for Teams, Cosine NSGA-II for Teams represents the adaptation of NSGA-II for Teams for *cpr* instead of *pr*. The same way *epr* is integrated into Win-Win NSGA-II for Teams, *cpr* (and *acpr* and *icpr*) are integrated into Cosine NSGA-II for Teams by integrating them into a new $cosine\_fairness\_filter$ (see 6).

Figure 4.2 shows an example for applying Cosine NSGA-II for Teams to the TNK benchmark problem (see 2.5) with three reference lines, with evenly spaced DM-preferences. Solutions are converging towards the true front and lie in-between the three reference lines, but the population center is slanted towards the centroid of the tree lines in the second example.

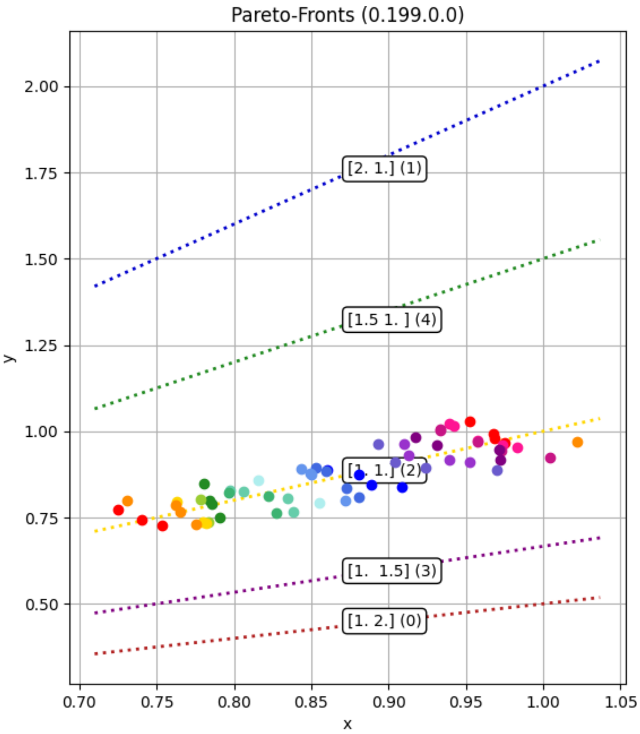


Figure 4.2: Visualisation of Cosine NSGA-II for TNK (see 2.5), shown lines are the reference lines of the DMs, solutions are coloured in based on their dominance rank

# 5 Experiments

This chapter explains the different experiments that are conducted to evaluate and compare the different methods presented. First (5.1), algorithm (and) parameter choices for the employed algorithms are presented. The second section (5.2) deals with experiment scenarios and why the Multi-Objective Multi-Agent Pathfinding (MOMAPF) (and the TNK) problem were chosen for algorithm benchmarking. After that (5.3), the chosen metrics to compare the algorithms with are explained and the rationale for choosing them is presented. Lastly (5.4), the chosen libraries and programming languages for implementing the theory are described.

## 5.1 Algorithm and Problem Parameter Choices

The compared algorithms are NSGA-II, as a non-reference respecting baseline, NSGA-II for Teams and Adaptive NSGA-II for Teams as the original Pareto Regret fairness algorithms to compare against, Win-Win NSGA-II for Teams and Win-Win Adaptive NSGA-II for Teams to focus on *partial* objective values for fairness and Cosine NSGA-II for Teams, as a novel approach to Pareto Regret definition. For the optimisation using the different algorithms, the same base parameters are chosen between all of them:

- $n_{\text{pop}} = 64$ - the population size
- $n_{\text{gen}} = \{250, 200\}$ - the number of run generations, less generations were chosen for TNK, as it is a simpler problem
- $n_{\text{tournament}} = 2$ - the number of participants in a tournament, chosen as defined by the authors [4], therefore *binary* tournament selection
- Several default parameters used in MOMAPF - These were chosen in accordance with the MOMAPF-paper [15], an example being crossover and mutation probabilities

### 5.1.1 Adaptive Variants

As the slightly adapted version of Adaptive NSGA-II for Teams includes a repositioning interval, it is also described here. As proven effective in prior testing, the interval after which the *Adaptive* algorithms repositions the reference points, $n_{\text{gen\_repos}}$, is set to 25.

### 5.1.2 Filter Rate

For the $fairness\_filter$ method (see 3), a filter rate $r_{\text{filter}}$ needs to be chosen. This influences how many solutions from the population are filtered based on their respective fairness and gain values.

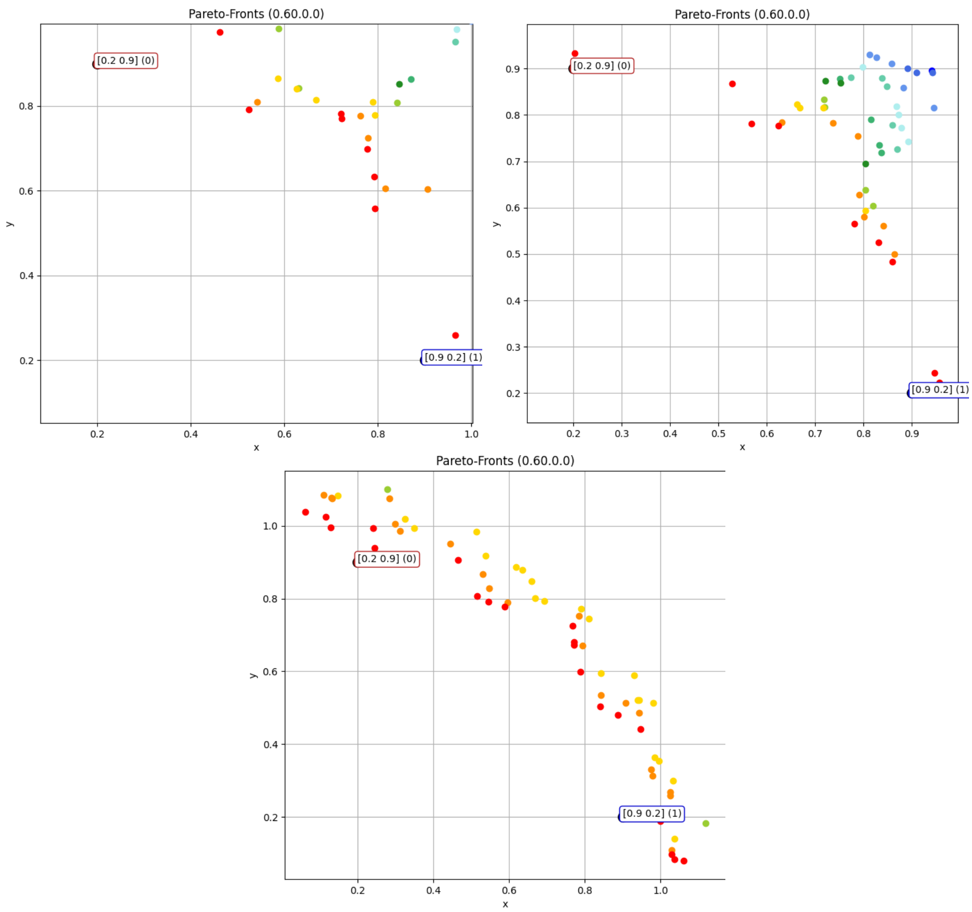


Figure 5.1: Different filter rates $r_{\text{filter}}$ for $fairness\_filter$ (see 3), from top left to bottom right: $r_{\text{filter}} = 0.5, 0.75, 1.0$ on TNK

Figure 5.1 shows the benchmark problem TNK (see 2.5) with 4 different filter rates of $0.5, 0.75$ and 1.0, which shows that on lower filter rates fairness and gain are considered and the pull from the DMs' reference points, or "consensus pressure", is the greatest, and convergence is higher, while diversity is a lot lower. On higher filter rates however, lower to none consensus pressure occurs during optimisation, which is why the filter rate for this work's experiment is set to $r_{\text{filter}} = 0.75$, which in prior testing proved to be a good middle-ground.

## 5.2 Scenarios

The main benchmark problem on which the presented algorithms are compared, is the Multi-Objective Multi-Agent Pathfinding (MOMAPF) problem, as described in the basics chapter at 2.4. This problem is chosen due to it's real-world application in e.g. logistics and it's good fit as a Partitionable MOOP that is easy to understand, as each agent has it's own preference. For objective functions, $f_R$ and $f_T$ are chosen (risk and time, see 2.4), as they represent an interesting and conflicting combination of objectives. The chosen vehicle model is Adaptive Dubins, as it most closely resembles real world vehicles. Possible map scenarios for MOMAPF are shown in figure 2.7.

Chosen scenarios include an *empty* map as a best-case scenario where obstacle interference does not interfere with the preference based optimisation, as to which e.g. a "Raw Eggs Agent" can evade other agents to navigate on a less risky path. Two *double_gap* maps are included with respectively $n_{\text{gaps}} = 2, 3$ and no bar representing real-world navigation scenarios. Finally a $double_gap$ with $n_{\text{gaps}} = 2$ but with a *bar* with length 90 (on the default $200 \times 200$ map) is also included as a "curve test" to easily qualitatively determine whether partial preferences are met, e.g. if one, risky, agent takes the inner curve and the other opts for the safer path on the outer curve.

### 5.2.1 Decision Maker Preferences and Number of Agents

As briefly explained in the introduction (see 1.2), the goal is to represent different agents with different preferences and therefore reference points $R$ (as each DM represents a single agent). For each map scenario, 6 different $n_{\text{agents}}$ (representing the number of agents) scenarios are run.

- $n_{\text{agents}} = 2$: $R = \{(70, 200), (95, 100)\}$
- $n_{\text{agents}} = 2$: $R = \{(70, 200), (95, 100), (80, 150))\}$
- $n_{\text{agents}} = 2$: $R = \{(70, 200), (95, 100), (80, 150), (60, 250)\}$
- $n_{\text{agents}} = 2$: $R = \{ \begin{array}{l} (60, 250), (60, 250), (70, 200), (70, 200), \\ (80, 150), (95, 100), (95, 100) \end{array} \}$
- $n_{\text{agents}} = 2$: $R = \{ \begin{array}{l} (60, 250), (60, 250), (60, 250), (70, 200), (70, 200), \\ (80, 150), (80, 150), (95, 100), (95, 100), (95, 100) \end{array} \}$

There are only two higher agent count scenarios, as testing beforehand concluded, that convergence on these scenarios would not be high on any of the tested algorithms, therefore these mostly serve as an example as to how MOMAPF with higher agent counts is a bigger challenge to optimisation and how fairness is harder to achieve with many DMs (in regards to MOMAPF). For each number of agents, 3 different types of agents are considered (as shown in 1.2):

- The slow but safe agent, or Raw-Egg-Agent (REA)
- The compromising agent, or Bread-Agent (BA)
- The fast and risky agent, or Cut-Avocado-Agent (CAA)

For the higher numbers of agents, a lot of agents share their type, to simplify the optimisation problem.

For Cosine NSGA-II for Teams different values are utilised, as the computation of the reference lines are different from the reference points (see 4.4). However, the logic behind the value distribution is the same as with the reference points:

- $n_{\text{agents}} = 2$: $W = \{(1, 3), (1, 1)\}$
- $n_{\text{agents}} = 3$: $W = \{(1, 3), (1, 1), (1, 2)\}$
- $n_{\text{agents}} = 4$: $W = \{(1, 3), (1, 1), (1, 2), (1, 4)\}$
- $n_{\text{agents}} = 7$: $W = \{ \begin{array}{l} (1, 4), (1, 4), (1, 3), (1, 3), \\ (1, 2), (1, 1), (1, 1) \end{array} \}$
- $n_{\text{agents}} = 10$: $W = \{ \begin{array}{l} \{(1, 4), (1, 4), (1, 4), (1, 3), (1, 3), \\ (1, 2), (1, 2), (1, 1), (1, 1), (1, 1) \end{array} \}$

### 5.2.2 TNK

The last experiment scenario is an outlier experiment which is not a Partitionable MOOP like MOMAPF, but is a benchmark problem with a known Pareto Front, TNK (see 2.5). This problem is used to showcase especially the Cosine NSGA-II for Teams variant, as it's simple nature makes it easy to show the impact of different reference points (or lines). Because it is not a partitionable MOOP however, it can not be used to showcase the Win-Win adaptations of both NSGA-II for Teams variants. 4 sets of reference points are utilised in the comparison:

- $R = \{(0.4, 0.9), (0.9, 0.4)\}$
- $R = \{(0.4, 0.9), (0.9, 0.4), (0.75, 0.75)\}$
- $R = \{(0.4, 0.9), (0.9, 0.4), (0.75, 0.75), (0.5, 0.8)\}$
- $R = \{(0.4, 0.9), (0.9, 0.4), (0.75, 0.75), (0.5, 0.8), (0.8, 0.5)\}$

And for Cosine NSGA-II for Teams (again note that these are weights, not reference points):

- $W = \{(1, 2), (2, 1)\}$
- $W = \{(1, 2), (2, 1), (1, 1)\}$
- $W = \{(1, 2), (2, 1), (1, 1), (1, 1.5)\}$
- $W = \{(1, 2), (2, 1), (1, 1), (1, 1.5), (1.5, 1)\}$

## 5.3 Evaluation and Metrics

For quantitative evaluation several metrics are utilised to compare the algorithms' performances. The metrics chosen are the Hypervolume (HV) Indicator and the coverage (or cardinal) metric (C-Metric). These metrics were chosen, as they do not require the knowledge of the problem's true Pareto Front, as for both MOMAPF and the fairness-gain space the true Pareto Front is unknown.

The Hypervolume Indicator is used to compare the algorithms in objective space, in accordance with research question 1. HV is both a metric for diversity and quality of solutions and therefore is widely used to compare algorithms [26]. The reference point for Hypervolume, $(100, 800)$, is taken from the MOMAPF paper [15].

To measure the performance of algorithms in the fairness and gain objective space, a metric that does not require any prior information is needed, as the objective space is entirely unknown and strongly dependent on the reference points. The C-Metric indicates how high the domination percentage of one set of individuals to another is. This is not an absolute measure of how much better one algorithm performs than another, but it is an indicator as to how diverse (and to a lesser extent of what quality) the solutions of one algorithm in comparison to the other are, without any prior needed information. C-Metric is utilised to compare algorithms both in "regular" fairness and gain objective space and in *partial* fairness and gain objective space, as to indicate quantitatively how well the algorithms perform in respecting both regular as well as partial DM preference.
Both HV and C-Metric are calculated the first front of the population by using $fnds$ [4].

Aside from quantitative evaluation, a qualitative approach is also taken towards evaluation, as especially with regard to solution spread and the fulfillment of DM intention in choosing their reference points (e.g. the Raw Eggs Agent to actually drive safer), metrics are not sufficient to look at.

### 5.3.1 Hypervolume Indicator

The Hypervolume (HV) Indicator is defined by the hypervolume spanned by an (approximated) Pareto Front $F$ and the chosen reference point $r_{\mathrm{hv}}$.

$$hv(F) = V(\bigcup_{\forall i \in F} volume(x_i, r_{\mathrm{hv}})) \tag{5.1}$$

As the area spanned up increases with both solution quality (closer to real Pareto Front) and diversity (less crowded solutions as crowding adds less area), HV is a hybrid metric (also see 5.2).

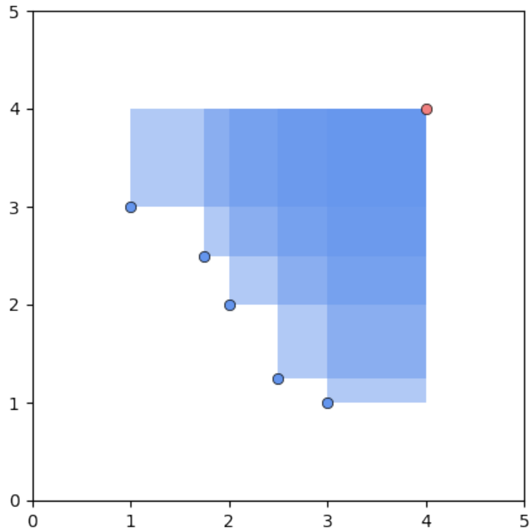


Figure 5.2: Visualisation of the Hypervolume Indicator

### 5.3.2 Coverage Metric

The coverage metric indicates how much of another non-dominated solution set $Y$ a non-dominated solution set $X$ covers (or dominates) ([10] as *Cardinal Metric*). It is defined as the percentage of all solutions in front $Y$ that are covered by solutions from $X$.

$$c(X,Y) = \frac{|\{y \in Y | \exists x \in X : x \succ y\}|}{|X|} \tag{5.2}$$

This means that the value is "asymmetric" as that both a solution set $X$ and a set $Y$ can have coverage percentages that do not sum up to 100% (see 5.3).

The C-Metric gives a rough idea as to how far one non-dominated solution set dominates another.

## 5.4 Tools, Libraries and Frameworks

The source code for implementation of the algorithms of this work (and those it is based on) are written in Python and mainly use deap [3] for implementing

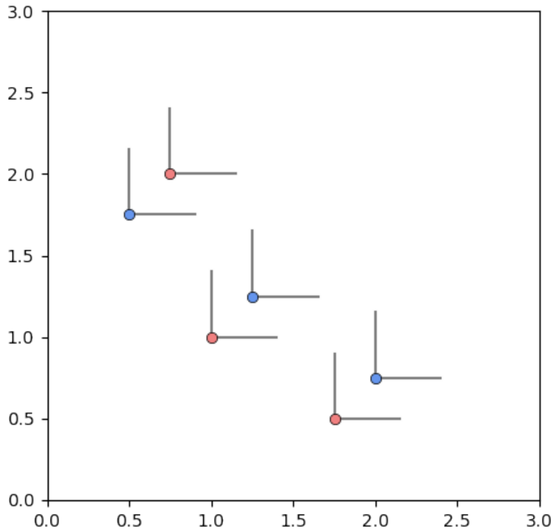


Figure 5.3: Visualisation of the Coverage-Metric (C-Metric), represented by the rate of solutions one front dominates of the other, here $c(blue, red) = \frac{1}{3}$ and $c(red, blue) = \frac{2}{3}$

the Evolutionary Algorithm and numpy for calculations. The underlying work for implementing the MOMAPF-problem was taken from the work by Mai and Mostaghim [15], also written in Python.
While working on this thesis, the extendable framework EABench was written. It aims to Implementing and comparing new algorithms for Evolutionary Algorithms using this framework takes little code and provides high extendability.

# 6 Evaluation

The experiments (each scenario combination) were run 3 times each to ensure a degree of result consistency. Shown are mostly results from the map scenario *curve*, as it proved to be the most obvious in terms of qualitatively analysing in how far a "win-win" consensus, so that the solution fullfills each Decision Maker (DM)'s partial preference, has been found by the algorithms.
As the adaptive versions of both the regular Pareto Regret and the Egoistic Pareto Regret NSGA-II for Teams are very similar to the base algorithms, they produce very similar results, if the reference points are close to the true Pareto Front or are not close to being approximated by the algorithm with the given parameters, as can be seen in figure 6.1.

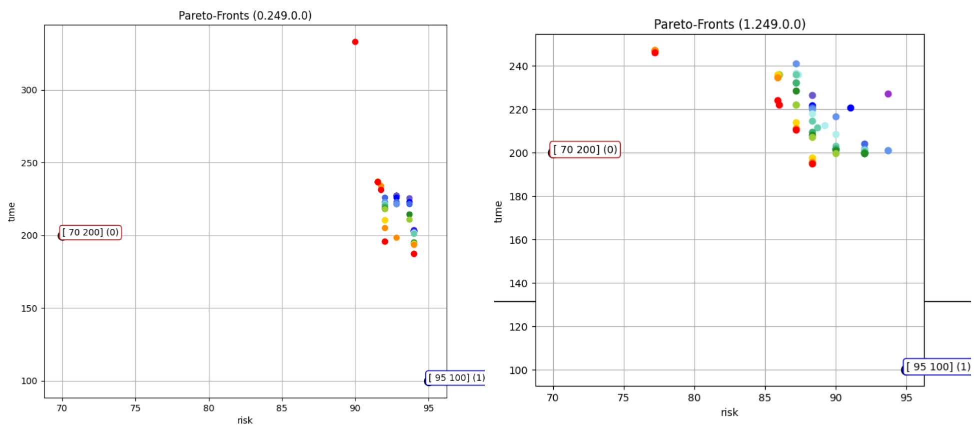


Figure 6.1: Similarity of the Pareto Front of Adaptive and regular NSGA-II for Teams, when choosing reference points near the Pareto Front

Similarly, Cosine NSGA-II for Teams qualitative results are very similar to those of regular NSGA-II for Teams, observable in figure 6.2.

As such, the results of the Adaptive versions and Cosine NSGA-II for Teams are not shown for qualitative inspection.

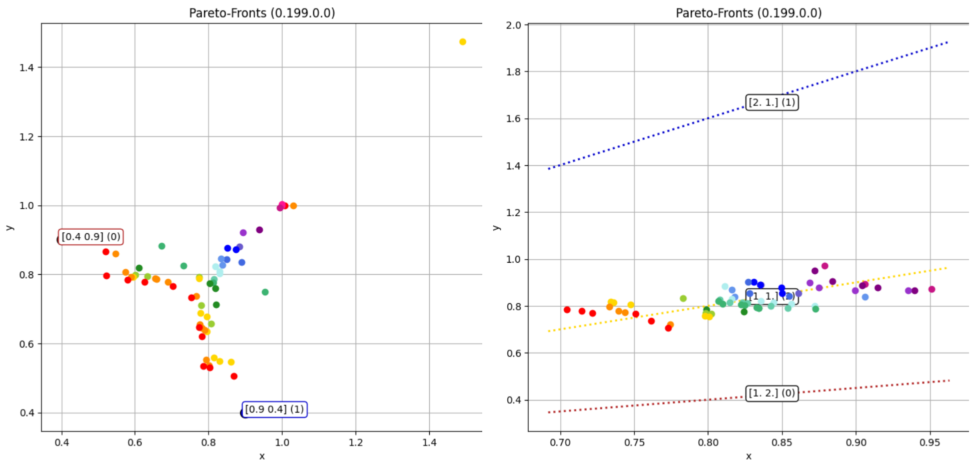


Figure 6.2: Similarity of the Pareto Front of regular and Cosine NSGA-II for Teams when choosing reference points near the Pareto Front

## 6.1 Objective Space Performance

Performance in objective space is evaluated by Hypervolume (HV) Hypervolume (see 5.3.1) of each algorithm, sorted by map and $n_{\text{agents}}$ scenario.

As can be seen in figure 6.3, overall, while there is some variation, the hypervolume decreases with a higher number of agents, which is expected, as the problem's difficulty increases at the same time, which can also be seen in figure 6.6 for NSGA-II, as to that solutions for problems with higher agent numbers get increasingly more complicated and for both $n_{textagents} = 7, 10$ solutions generated are either very risky or not valid (include collisions), as all algorithms do mostly not find valid solutions within the set $n_{\text{gen}}$. The increasing problem difficulty can also be seen when looking at the generated approximated Pareto Fronts in figure 6.4 and figure 6.5, which show that with increasing $n_{\text{agents}}$ the Pareto Front found by NSGA-II are positioned more in direction of risk, even mostly in invalid ($f_R \geq 100$) space for $n_{\text{agents}} = 7, 10$, while those found for both regular and Win-Win NSGA-II for Teams are also further away from the reference points.
Interestingly, which algorithm performs the best in terms of HV is dependent on the map scenario. NSGA-II is the best for each map scenario, which is expected as it tries to approximate the whole Pareto Front, which leads to much higher HV. For *empty* and *curve*, the Win-Win algorithms are the second highest performing algorithms, while for *double_gap* and *triple_gap*,

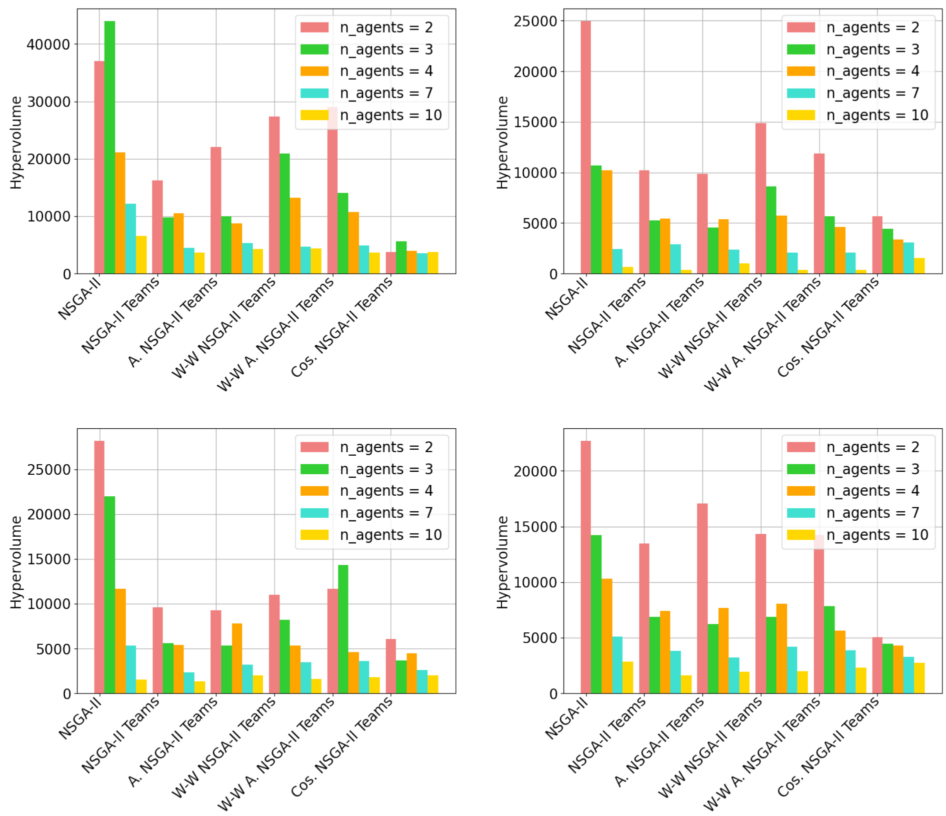


Figure 6.3: Hypervolumes of algorithms on the MOMAPF problem, on different map scenarios (left-to-right, top-to-bottom): *empty*, *curve*, *double_gap* and *triple_gap*

the gap between algorithms is much smaller or, while on *triple_gap*, they get outperformed by Adaptive NSGA-II for Teams.
As for Cosine NSGA-II for Teams, the hypervolume is a lot lower than the other algorithms, which is due to the stronger effect of the DMs' preferences, from here known as consensus pressure, also observable for TNK in figure 6.2, where the only solutions generated are in the "most fair" region of the Pareto Fronts. As the diversity is lower, Hypervolume will always be lower for Cosine NSGA-II for Teams.

For qualitative inspection of how well the algorithms respect the stated reference points, the Pareto Fronts shown in figure 6.5 (and figure 6.2 for Cosine NSGA-II for Teams) show that both regular NSGA-II for Teams as well as

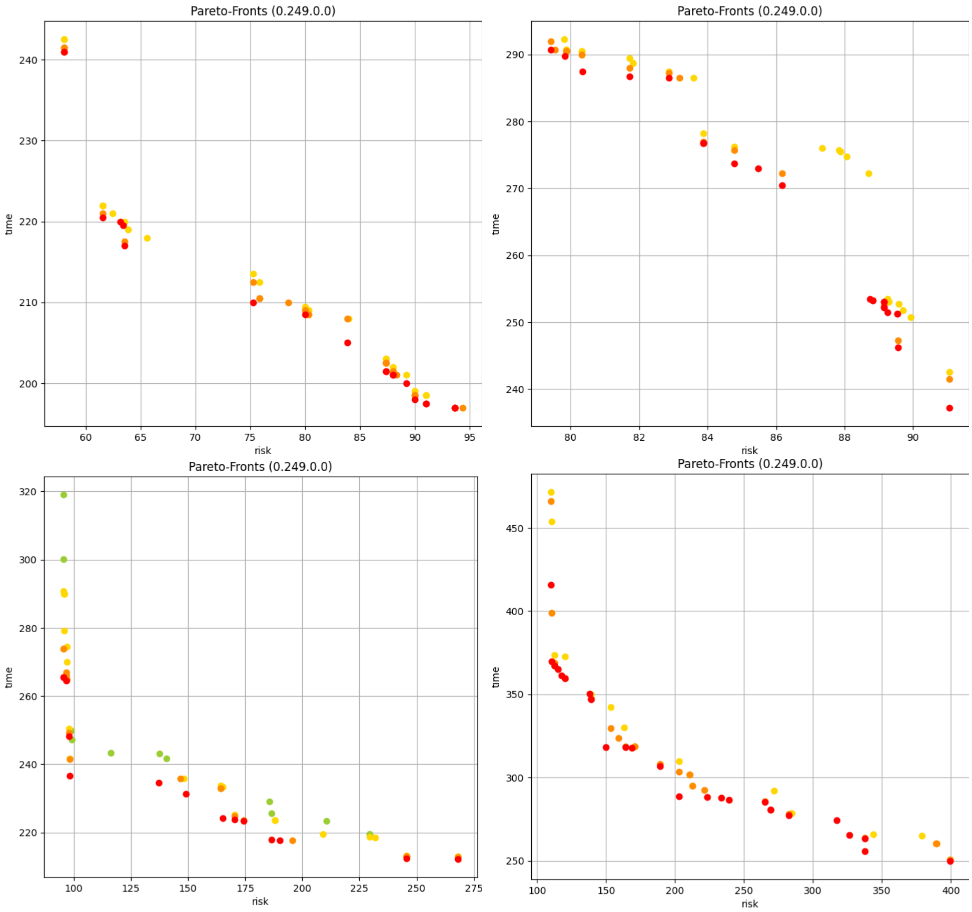


Figure 6.4: Pareto Fronts of MOMAPF, $n_{\text{agents}} = 2, 4, 7, 10$, on *curve*, generated by NSGA-II, different colours represent different fronts, looping from red to purple

Win-Win NSGA-II for Teams respect the DMs' reference points, with solutions positioned in-between them.

Win-Win NSGA-II for Teams does not optimise for regular fairness and gain, which means that in the case of the MOMAPF problem, solutions that are fair in the regular sense and therefore in-between the reference points in objective space, are also fair to some degree in respect to the partial preference. It can be seen in figure 6.5 however, that the found solutions are positioned differently than in NSGA-II for Teams, both less safe (higher $f_R$) and with less speed (higher $f_T$), which also is expected, as, if a "win-win" situation is found, a solution both encompasses (multiple) REAs, which lead to higher average travel time, and CAAs, which lead to a lower minimum distance across agents,

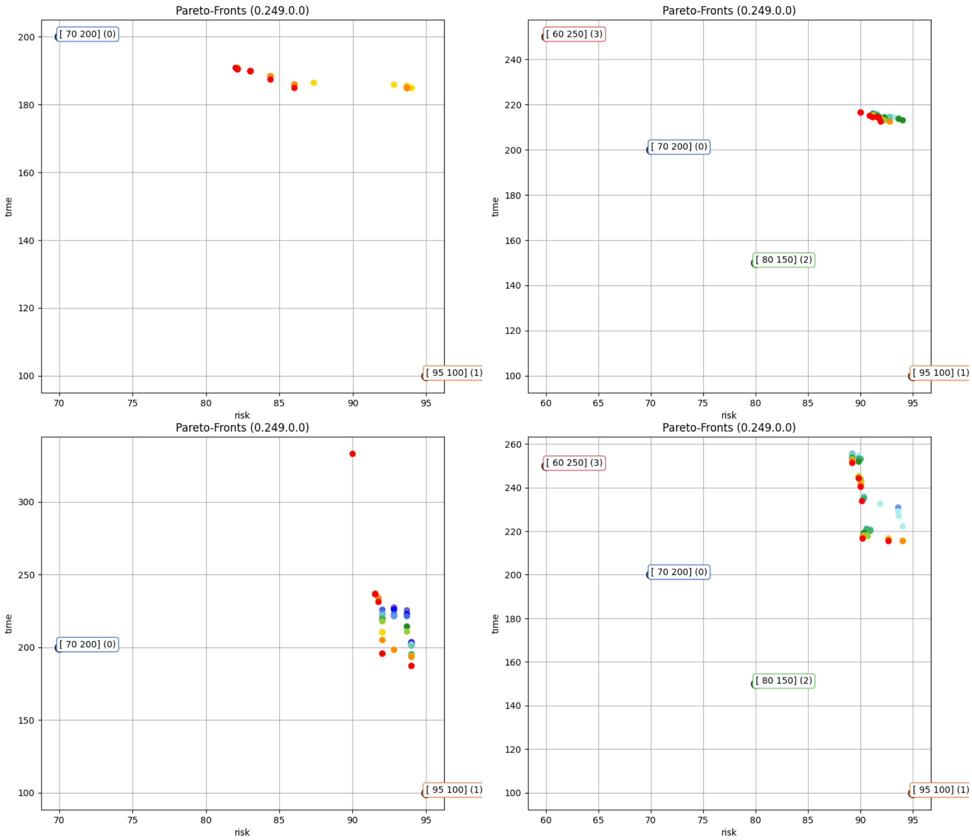


Figure 6.5: Pareto Fronts of MOMAPF, $n_{\text{agents}} = 2, 4$, on *curve*, generated by reference point respecting algorithms (top-to-bottom): NSGA-II for Teams and Win-Win NSGA-II for Teams, different colours represent different fronts, looping from red to purple

leading to higher risk values. This makes it all the more interesting, that, while achieving solutions with win-win situations, diversity and therefore HV still is generally higher than for regular NSGA-II for Teams, which then seems to experience more consensus pressure, only focussing on the small optimal (in regards to the reference points) part of the Pareto Front.

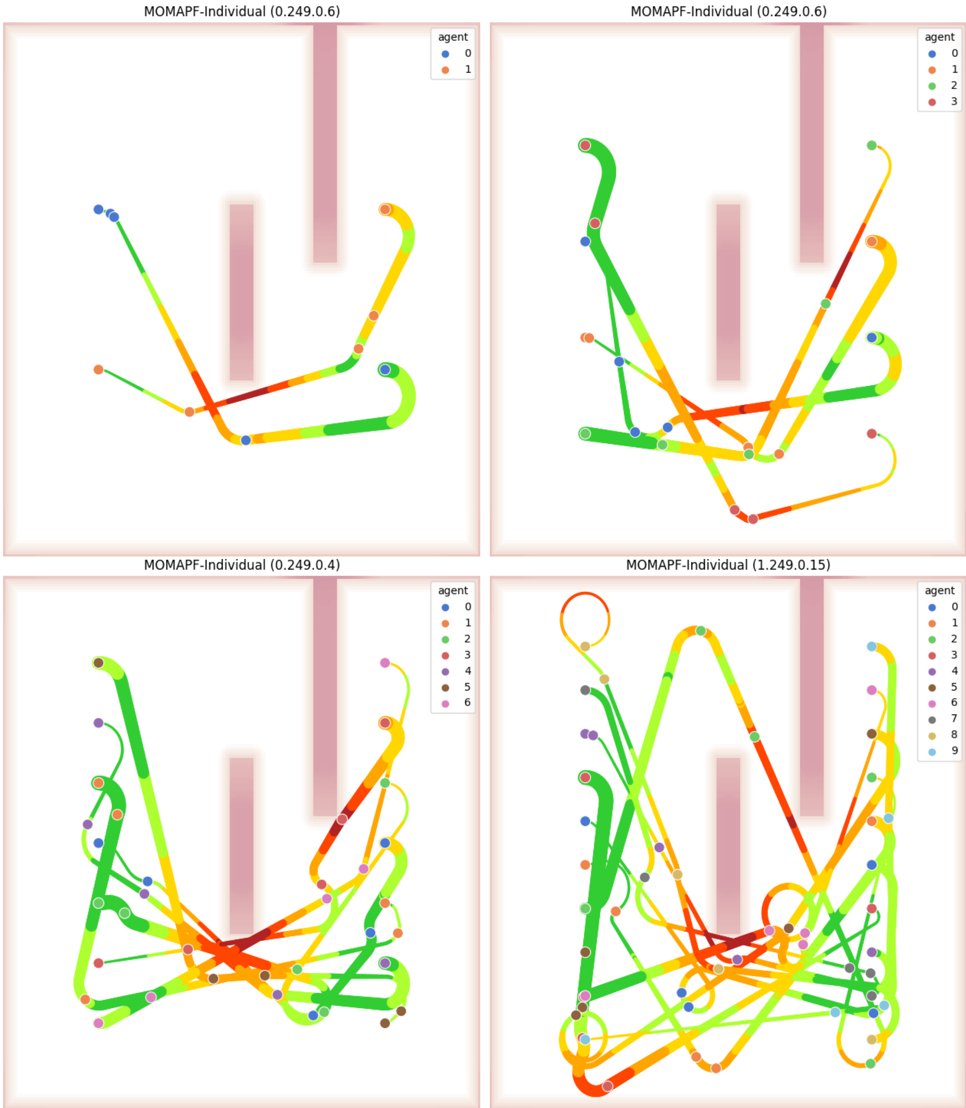


Figure 6.6: Solutions with $n_{\text{agents}} = 2, 4, 7, 10$, on *curve*, generated by NSGA-II

## 6.2 Regular Fairness and Gain

As can be seen in the confusion matrices from figure 6.7, like for HV, the resulting C-Metric values vary on different $n_{\text{agents}}$, however, a trend is discernable. Regular fairness and gain based algorithms ((Adaptive) NSGA-II for Teams and Cosine NSGA-II for Teams) dominate the field, while the presented Win-Win adaptations stay behind.

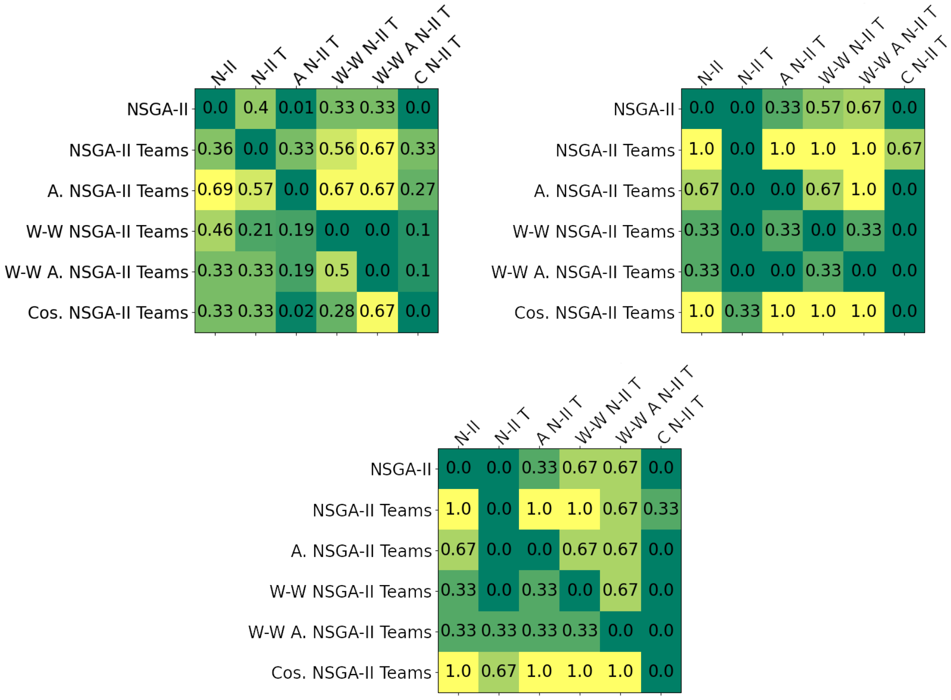

Figure 6.7: C-Metric confusion matrices based on regular fairness and gain of MOMAPF of all algorithms on *curve*, (left-to-right, top-to-bottom) $n_{\text{agents}} = 2, 3, 4$, shows the C-Metric of row versus column, $cm(\text{row}, \text{column})$

These results are expected, as the Win-Win adaptations do not optimise for regular fairness and gain, however, as the partial preferences line up in the objective space to some extent, as discussed in section 6.1, the algorithms still are performant in terms of regular fairness and gain.

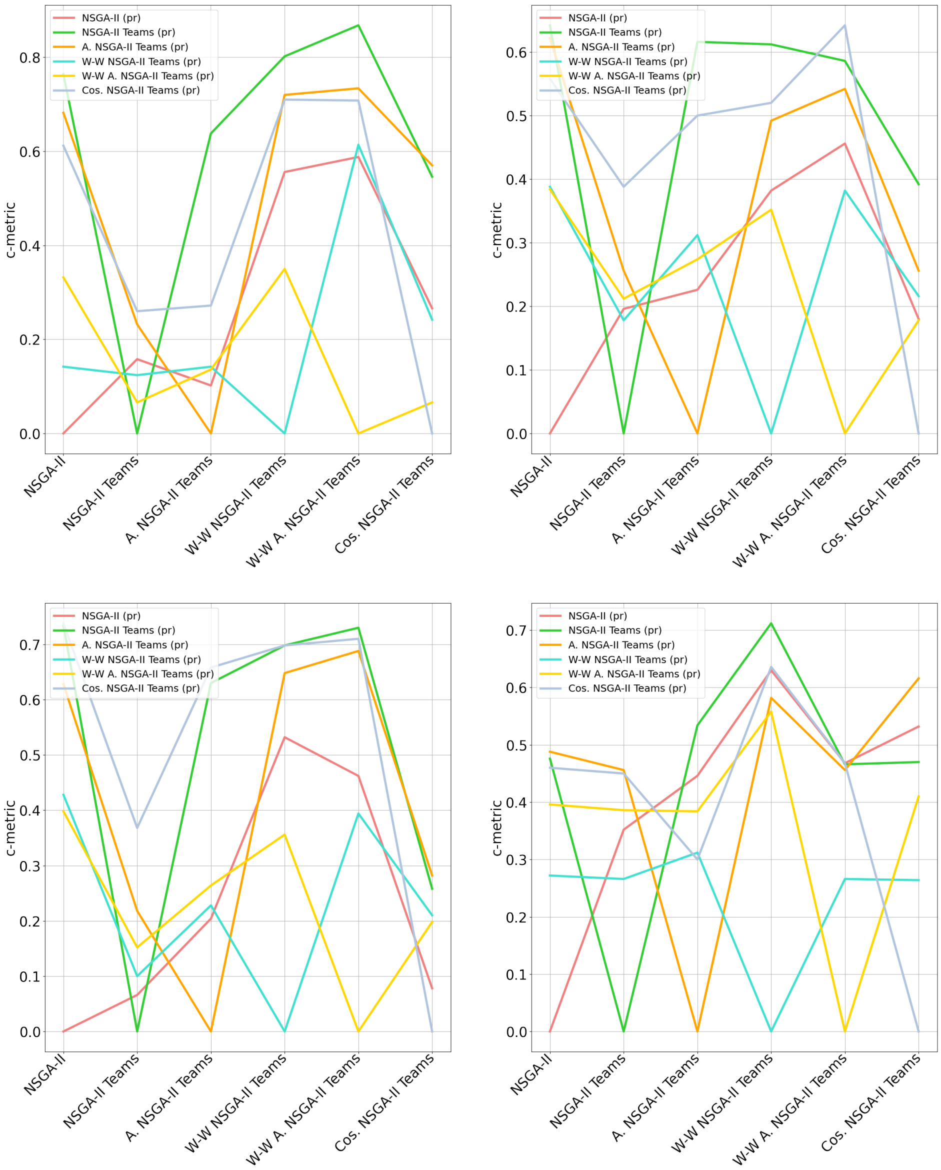


Figure 6.8: Line plots of C-Metric values for regular fairness and gain fronts of all algorithms on the MOMAPF problem, averaged over all $n_{\text{agents}}$ on all map scenarios (left-to-right, top-to-bottom): *empty*, *curve*, *double_gap* and *triple_gap*

This strongly depends on both the aggregation function utilised in calculating the regular objective values and the reference points set, as, like when calculating the risk using $f_R$, e.g. the minimum is calculated from the partial objective

values, the reference point with the lowest value (or for $f_R$ highest) strongly influences the solution's position in objective space, which on the other hand influences Pareto Regret, regular fairness and gain and the C-Metric values.
The unexpected performance of NSGA-II can be explained due to the set reference points being close to the found Pareto Front, as NSGA-II aims for a diverse solution set across the front, solutions with high fairness and gain are bound to be included.
In the line plots from figure 6.8, the average C-Metric c-metric values across all $n_{\text{agents}}$ are shown, indicating, again, differences across the different map scenarios. The general trend mirrors the confusion matrices' results, as both (Adaptive) NSGA-II for Teams and Cosine NSGA-II for Teams perform well above both NSGA-II and (Adaptive) Win-Win NSGA-II for Teams. Some exceptions are present however, both NSGA-II and Adaptive Win-Win NSGA-II for Teams perform a lot better in $triple_gap$. While NSGA-II's performance increase likely occured due to the map scenario making it easy to find consensing solutions due to the three gaps, as Win-Win NSGA-II for Teams does not exhibit the same performance increase, the Adaptive variant just experiences some variance.

## 6.3 Partial Fairness and Gain

To quantitatively analyse the performance with regards to finding solutions with high partial fairness and gain, figure 6.9 shows the confusion matrices of the C-Metric values on the partial fairness and gain front, which show again that results vary based on different values of $n_{\text{agents}}$. The trend here is however that both Win-Win based algorithms do perform better than the other algorithms, however, interestingly not necessarily everwhere, as there are spots for $n_{\text{agents}} = 3$ where for example neither NSGA-II for Teams nor Win-Win NSGA-II for Teams dominate each other meaningfully. NSGA-II for Teams does dominate it's Adaptive variant, which on the other is not dominated as much by Win-Win NSGA-II for Teams, leading to the conclusion that in this specific scenario NSGA-II for Teams and the Win-Win adaptations represent different parts of the approximated Pareto Front in fairness and gain space. Additionally, while it does perform well with regards to regular fairness and gain, Cosine NSGA-II for Teams does not perform well with regards to partial fairness and gain, which is expected, as it only explores a small front due to high consensus pressure.

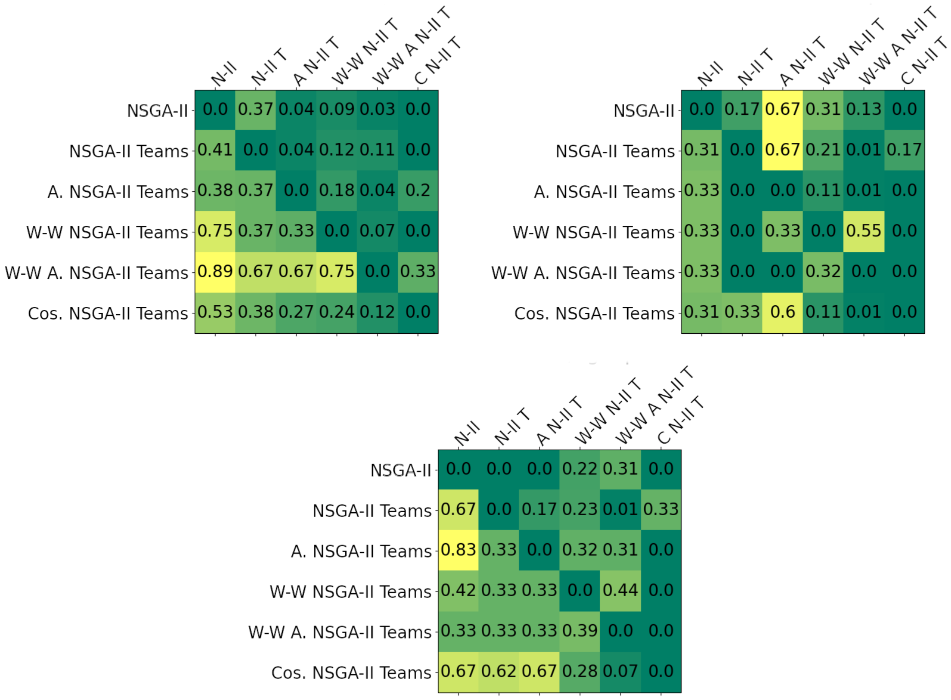


Figure 6.9: C-Metric confusion matrices based on partial fairness and gain of MOMAPF of all algorithms on *curve*, (left-to-right, top-to-bottom) $n_{\text{agents}} = 2, 3, 4$, shows the C-Metric of row versus column, $cm(\text{row}, \text{column})$

As for exploring the algorithms' performance across all map scenarios, figure 6.10 shows the C-Metric values averaged across all $n_{\text{agents}}$, showing that for partial fairness and gain values vary across different map scenarios as well. Remarkable is that, while not optimsing for it, both (Adaptive) NSGA-II for Teams and Cosine NSGA-II for Teams perform rather well across the different map scenarios, which is explained by their high performance for higher number of agents ($n_{\text{agents}} = 7, 10$), where Win-Win NSGA-II for Teams falls off, as optimisation for win-win situations with a lot of agents is significantly more difficult.

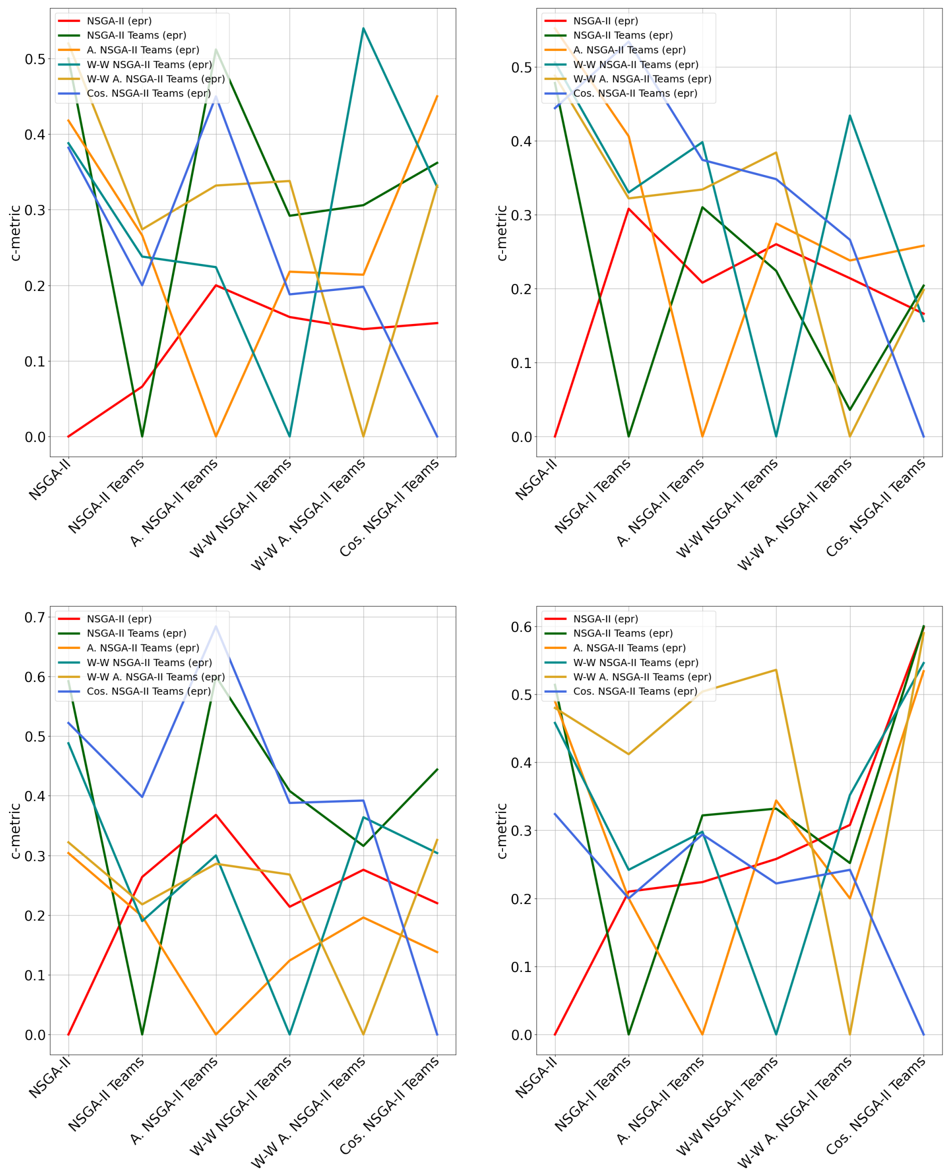


Figure 6.10: Line plots of C-Metric values for *epr*-based fairness and gain fronts of all algorithms on the MOMAPF problem, averaged over all $n_{\text{agents}}$ on all map scenarios (left-to-right, top-to-bottom): *empty*, *curve*, *double_gap* and *triple_gap*

Figure 6.11 gives insight into why the partial fairness and gain performance of non-Win-Win algorithms is so high, as it showcases an "accidentally fair" solution, where the left solution does not respect the partial preference (the blue agent is a REA, the orange agent is a CAA), the right one clearly does. Both were taken from the fronts ranked first respectively using $fnds$, which makes it clear that finding solutions with high partial fairness and gain is accidental for the algorithms not presented in this work. While NSGA-II, (Adaptive) NSGA-II for Teams and Cosine NSGA-II for Teams do not optimise for partial fairness and gain, they can however find well-performing solutions, and, due to the closeness in objective space of solutions with high regular fairness and gain and those with high partial fairness and gain, as explained in section 6.1, these can also be performant in terms of partial fairness and gain.

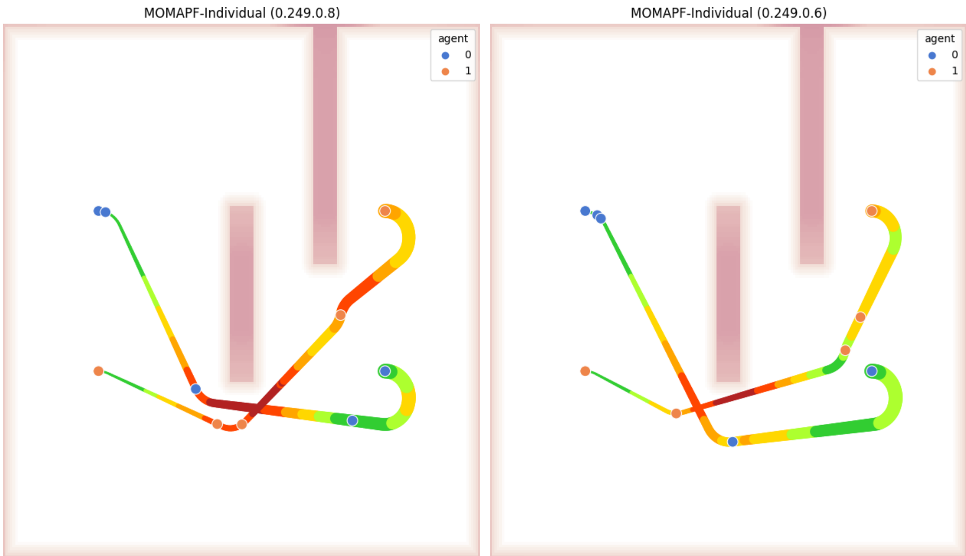


Figure 6.11: Showcase of accidentally found solutions with high partial fairness and gain on *curve* with $n_{\text{agents}} = 2$, generated by NSGA-II for Teams and NSGA-II

To explain the unexpectedly weak performance of (Adaptive) Win-Win NSGA-II for Teams, considering that the algorithms specifically optimise for partial fairness and gain, a methodical flaw is observable due to the placement of solutions in objective space, as seen in figure 6.5. While for computing the C-Metric of regular fairness and gain it is sensible to choose the solutions from the best ranked front using $fnds$, as the distance is computed in objective space, performing well in objective space does not necessarily equal performing well on the partial fairness and gain front. As explained in section 6.1, well performing solutions for both are close to each other in objective space,

however they are not necessarily on the approximated Pareto Front.
Looking at figure 6.11 showcase qualitatively whether (Adaptive) Win-Win NSGA-II for Teams is able to achieve solutions with win-win situations, it is clear that these are solutions with actual win-win situations. You can see that in the top left the REA cuts the corner, as the CAA takes a safer detour, or that in the bottom left the CAA comes very close to the obstacle, the REA dodges it and the BA finds a compromise.

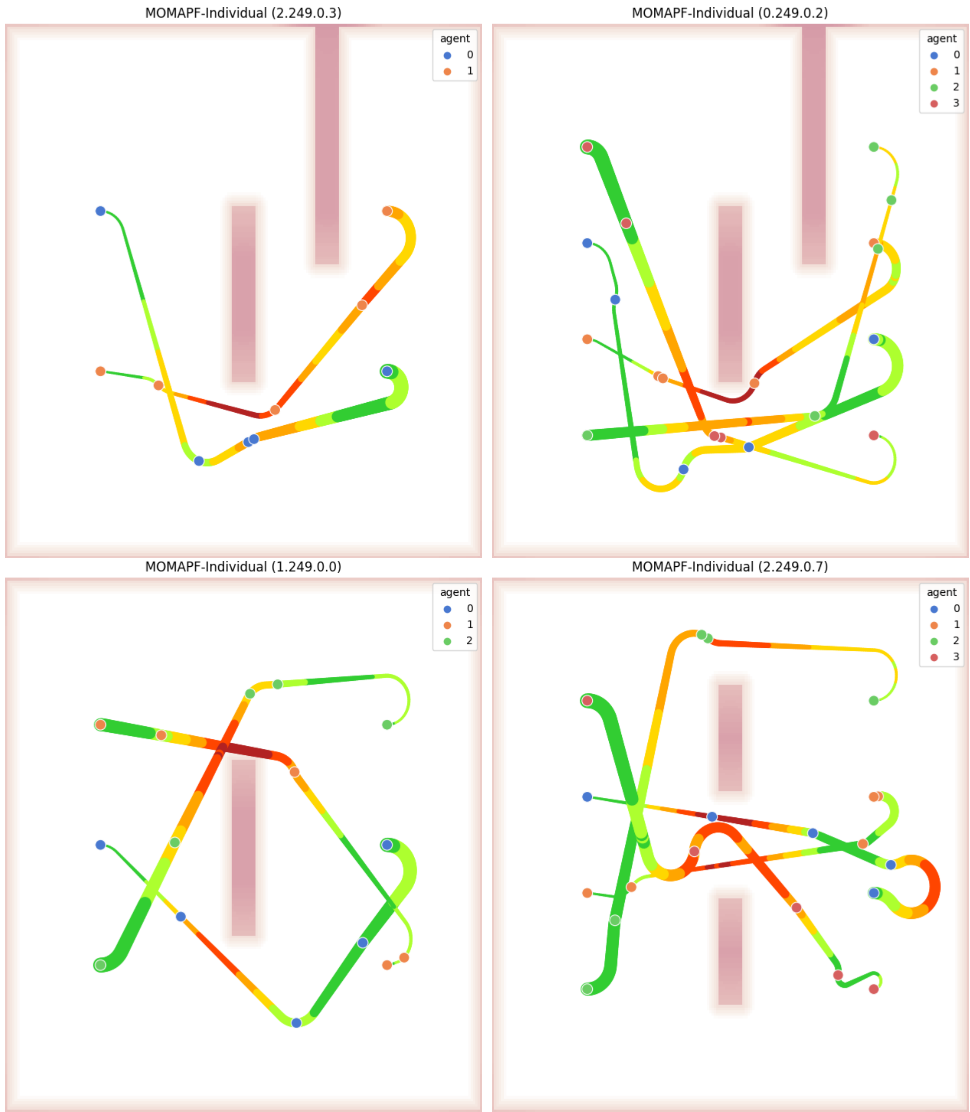


Figure 6.12: Showcase of solutions with high partial fairness and gain generated by (Adaptive) Win-Win NSGA-II for Teams (left-to-right, top-to-bottom): *curve*, $n_{\text{agents}} = 2$; *curve*, $n_{\text{agents}} = 3$; $double_gap$, $n_{\text{agents}} = 3$, $triple_gap$, $n_{\text{agents}} = 4$

# 7 Conclusion and Future Work

In this work two novel adaptations of the original Non-Dominated Sorting Algorithm II for Teams (NSGA-II for Teams) were presented.
The first adaptation deals with the problem of Paret-Regret and "regular" objective functions being too general and loosing information valuable to the Decision Maker's decision. To handle this, a new type of problem is introduced: A Partitionable Multi-Objective Optimisation Problem, which is defined by it's objective values being composed of multiple aggregated *partial* objective values. The adapted algorithm, Win-Win NSGA-II for Teams and it's Adaptive counterpart deal with the Decision Maker preferring not a certain aggregated solution, but rather only caring for one aspect of that solution, *partial* objective values, which is not possible to specify in a regular reference point based on regular, aggregated, objective values. The measure with which to compute similarity based on partial objective values is named Egoistic Pareto Regret and is strongly akin to regular Pareto Regret, but considers distance of the respective aspect of a solution vector instead of the whole solution vector when computing distance to the reference point.
The second algorithm deals with the major problem for choosing reference points, the reliance on the DM to choose reference points on or near the unknown true Pareto Front. For NSGA-II for Teams this problem was addressed with a reference point repositioning, which can lead to the DM's original preference to be lost. The presented algorithm, Cosine NSGA-II for Teams deals with this problem through the use of reference lines instead of reference points, weights get chosen by the DM and a line through objective space is generated. The Pareto Regret of NSGA-II for Teams is replaced through the use of cosine similarity between the solution vector and the generated reference line. This version of Paret-Regret is named Cosine Pareto Regret.
Both algorithms (and the first algorithm's Adaptive version) were tested and compared with NSGA-II, NSGA-II for Teams and Adaptive NSGA-II for Teams on the scalable Multi-Objective Multi-Agent Pathfinding (MOMAPF) Problem with 4 different map scenarios, each with 5 different

values for $n_{\text{agents}}$, describing the amount of agents in the optimisation problem. The problem was chosen as it is close to real world use cases for specifying preference in regards to a specific aspect of a solution, in this case the objective values of a specific agent, disregarding the other agents. For MOMAPF specifically this means that each DM corresponds to one agent and it's individual performance in regards to both time of fligt $f_T$ and risk $f_R$.
The second algorithm was also tested (and compared with the non-Win-Win algorithms) on the benchmark problem TNK to demonstrate the advantages and differences of the algorithm on a known and simple objective space topology.

The experiment results show that both the novel Win-Win algorithms as well as Cosine NSGA-II for Teams are promising for different use cases.
The presented Win-Win algorithms are shown to compare favorably to the existing algorithms in terms of Hypervolume (HV), as they achieve the same or even higher performance in that space, while qualitatively also converging to a similar front as (Adaptive) NSGA-II for Teams. Similar to NSGA-II for Teams however, they do not compare favourably to NSGA-II, which is expected, as they do not intend to approximate the whole Pareto Front. As for Cosine NSGA-II for Teams, it is shown to not perform well in terms of regular objective space or HV, as it explores a much narrower front due to a stronger effect of the Decision Makers' preferences, coined "consensus pressure". It also experiences little convergence due to the non-convergent nature of Cosine Pareto Regret in direction of the Pareto Front, which for regular NSGA-II for Teams is not a problem, as it's front converges towards the reference points, which are supposed to be chosen close to the Pareto Front.
In terms of regular, Pareto Regret based fairness and gain fronts, compared to the other algorithms and their performance measured by the C-Metric, (Adaptive) Win-Win NSGA-II for Teams are shown to not perform as well, which is expected, as it does not optimise for regular fairness and gain. Cosine NSGA-II for Teams is shown to perform very well with regards to regular fairness and gain, often out-performing regular NSGA-II for Teams, as it does not exhibit a lot of convergence however, the solutions' quality is lacking.
Regarding partial, Egoistic Pareto Regret based fairness and gain fronts, again compared to the other algorithms and the performance measured using C-Metric, the Win-Win adaptations fare a lot better than with regards

to regular fairness and gain. These results are flawed however, as the computation of C-Metric inherits a flaw at choosing the solution set, as the first front as ranked by $fnds$ is not necessarily the best set of solutions with regards to partial fairness and gain, which indicates that the real performance is a lot higher. Regular (Adaptive) NSGA-II for Teams does perform better than expected for partial fairness and gain in some map scenarios, which is mainly due to good performance at higher numbers of agents, where solutions with win-win situations are a lot harder to optimise for. NSGA-II for Teams however can find these accidentally by optimising for regular objectives, as solutions with win-win situations and solutions with high regular fairness and gain performance are close to each other for the MOMAPF problem with the chosen reference points. Cosine NSGA-II for Teams is shown to not perform particularly well for partial fairness and gain aside from higher number of agents, similar to regular NSGA-II for Teams.
By qualitative inspection, partial preferences are shown to be well respected by the Win-Win adaptations of NSGA-II for Teams in the solution phenotype, mostly exhibiting win-win situations as preferred by the DMs.

The results of this thesis indicate some arguments to further research both partial preferences for a "Win-Win" consensus as well as reference lines and vector-like preferences to deal with the problem of specifying reference points near an unknown true Pareto Front.
As to improve on the results of this thesis, a longer evaluation with a much higher sample count of each scenario-combination with more and more different reference points would be able to solidify the results and conclusions drawn here,also, as mentioned in chapter 6, the testing methodology of this thesis encompasses some weaknesses regarding the chosen individuals for computing metrics, which could be improved upon in future works. Additionally, once a good approximation to the true Pareto Front of the MOMAPF problem is known, different metrics like generational distance or inverse generational distance can be used to further inspect the algorithms' performances. Regarding the MOMAPF problem in general, the impact of the chosen map scenario and the number of agents on the ability to find fair and partially fair solutions is another subject of future research, as these results showed strong variations based on the combinations of scenarios.
Concerning the adapted Win-Win algorithms more research is needed to remove a limitation of this work's implementation, namely that each DM focusses

on only one partial aspect of the aggregated objective values, as per the used benchmark problem MOMAPF on one agents fitness. For future research, possible avenues would be to allow DMs to state multiple preferences and aggregate or rank these for fairness and gain, as this would make it possible to partly respect a DM's preferences, e.g. fully respecting the second and third preference but not the first. A problem discovered in this work is the ability to achieve partial objective value consensus, as per the MOMAPF problem some maps showed more ability to achieve a consensus regarding agents' fitnesses, while others did less so. Potentially adapting the DMs' preferences to the scenario and having a DM compromising on their own, potentially unfeasible, preferences would be, in addition to measuring the ability to achieve (partial) consensus, opportunities for more future research.
As for the Cosine NSGA-II for Teams algorithm, more research is needed regarding the concept of consensus pressure, of how the different measure underlying fairness and gain reflects on the diversity of solutions after filtering for fairness and gain. Contrary to the high consensus pressure, Cosine NSGA-II for Teams showed low convergence to the true Pareto Front of the benchmark problem TNK, which is another possible area of future research to improve on the presented algorithm. Additionally, Cosine NSGA-II for Teams as presented is not able to optimise for partial preferences of DMs.
Less specifically, as this work is an example of, novel approaches to both adaptations of Pareto Regret or metrics with the same goal of modifying the fairness and gain approach are topics of research that still have further improvements to be made.

# Bibliography

[1] Seungyup Baek and Vittaldas Prabhu. An algorithm for solving multiobjective problem with multiple decision makers. *International Journal of Information and Decision Sciences*, 1(2):178–193, 2008.

[2] Jürgen Branke, Thomas Kaußler, and Harmut Schmeck. Guidance in evolutionary multi-objective optimization. *Advances in engineering software*, 32(6):499–507, 2001.

[3] François-Michel De Rainville, Félix-Antoine Fortin, Marc-André Gardner, Marc Parizeau, and Christian Gagné. Deap: A python framework for evolutionary algorithms. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*, pages 85–92, 2012. Github: `https://github.com/DEAP/deap` at May 19, 2022.

[4] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[5] Kalyanmoy Deb and J Sundar. Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 635–642, 2006.

[6] Boris Djartov. Multi-criteria decision-making with many decision makers: Integrating fairness and gain. Master's thesis, Otto von Guericke Universität, April 2021. URL: `https://www.is.ovgu.de/is_media/Master+und+Bachelor_Arbeiten/MasterThesis_BorisDjartov-p-6730.pdf` at May 19, 2022.

[7] Michael Emmerich, André Deutz, Longmei Li, Asep Maulana, and Iryna Yevseyeva. Maximizing consensus in portfolio selection in multicriteria group decision making. *Procedia Computer Science*, 100:848–855, 2016.

[8] Michael Emmerich, Jussi Hakanen, Patrick Reed, Pradyumn Kumar Shukla, Jürgen Branke, Günter Rudolph, Sanaz Mostaghim, and Heike Trautmann. Supporting problem solving with many decision makers in multi-objective optimization, January 2020.

[9] Ariel Felner, Roni Stern, Solomon Shimony, Eli Boyarski, Meir Goldenberg, Guni Sharon, Nathan Sturtevant, Glenn Wagner, and Pavel Surynek. Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges. In *International Symposium on Combinatorial Search*, volume 8, 2017.

[10] Michael Pilegaard Hansen and Andrzej Jaszkiewicz. *Evaluating the quality of approximations to the non-dominated set.* Citeseer, 1994.

[11] Wouter M Koolen. The pareto regret frontier. *Advances in Neural Information Processing Systems*, 26, 2013.

[12] Marco Laumanns, Lothar Thiele, and Eckart Zitzler. Running time analysis of evolutionary algorithms on vector-valued pseudo-boolean functions. In *IEEE Trans. Evolutionary Comput*, 2003.

[13] Steffen Limmer and Manuel Dietrich. Optimization of dynamic prices for electric vehicle charging considering fairness. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2304–2311. IEEE, 2018.

[14] Sebastian Mai and Sanaz Mostaghim. Modeling pathfinding for swarm robotics. In *International Conference on Swarm Intelligence*, pages 190–202. Springer, 2020.

[15] Sebastian Mai and Sanaz Mostaghim. Multi-objective multi-robot pathfinding problem with novelty search optimization. To be published at Ants 2022 (`ants2022.uma.es`), November 2022.

[16] Asad Mohammadi, Mohammad Nabi Omidvar, and Xiaodong Li. Reference point based multi-objective optimization through decomposition. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE, 2012.

[17] Lukas Partes. Decision making for multi-objective pathfinding problems. Master's thesis, Otto von Guericke Universität, September 2021. URL: `https://www.is.ovgu.de/is_media/Master+und+Bachelor_Arbeiten/` at May 19, 2022.

[18] Jella Pfeiffer, Uli Golle, and Franz Rothlauf. Reference point based multi-objective evolutionary algorithms for group decisions. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 697–704, 2008.

[19] KC Tan, Tong H Lee, and EF Khor. Evolutionary algorithms with goal and priority information for multi-objective optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 1, pages 106–113. IEEE, 1999.

[20] Masahiro Tanaka, Hikaru Watanabe, Yasuyuki Furukawa, and Tetsuzo Tanino. Ga-based decision support system for multicriteria optimization. In *1995 IEEE international conference on systems, man and cybernetics. Intelligent systems for the 21st century*, volume 2, pages 1556–1561. IEEE, 1995.

[21] Lothar Thiele, P Miettinen, Pekka J Korhonen, and Julian Molina. A preference-based interactive evolutionary algorithm for multobjective optimization. *Helsinki School of Economics*, 2007.

[22] Jing Tian, Shenglian Guo, Dedi Liu, Zhengke Pan, and Xingjun Hong. A fair approach for multi-objective water resources allocation. *Water Resources Management*, 33(10):3633–3653, 2019.

[23] Jens Weise, Sebastian Mai, Heiner Zille, and Sanaz Mostaghim. On the scalable multi-objective multi-agent pathfinding problem. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2020.

[24] Vivianne Wolters. Promoting cooperative behavior with collective decision-making in highway-based multi-agent pathfinding. Master's thesis, Otto von Guericke Universität, November 2021. URL: `https://www.is.ovgu.de/is_media/Master+und+Bachelor_Arbeiten/` at May 19, 2022.

[25] Jian Xiong, Xu Tan, Ke-wei Yang, and Ying-wu Chen. Fuzzy group decision making for multiobjective problems: tradeoff between consensus and robustness. *Journal of Applied Mathematics*, 2013, 2013.

[26] Gary G Yen and Zhenan He. Performance metric ensemble for multi-objective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 18(1):131–144, 2013.

[27] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.

# Declaration of Authorship

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

Alexander Tracht — Magdeburg, May 19, 2022