Tobias Benecke

Tracing the impact of the initial population in evolutionary algorithms

INF FAKULTÄT FÜR INFORMATIK

Intelligent Cooperative Systems
Computational Intelligence

Tracing the impact of the initial population in evolutionary algorithms

Master Thesis

Tobias Benecke

22.05.2020

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

Advisor: Dr. Cristian Ramirez Atencia

Tobias Benecke: *Tracing the impact of the initial population in evolutionary algorithms*
Otto-von-Guericke Universität
Intelligent Cooperative Systems
Computational Intelligence
Magdeburg, 2020.

Abstract

In this thesis, a method for tracing the influence of the initial population throughout the generations in evolutionary algorithms (EAs) is proposed and evaluated. The algorithm tracks the influence by attaching markers on genes, linking to the initial population. The algorithm is implemented for bit vector and integer vector representations. Besides the tracable evolutionary algorithm (T-EA), four metrics to measure the impact of an individual from the initial population are proposed. The concept of tracing the impact will be evaluated using three problems, the Max Ones problem, the 0/1 Knapsack problem and the (Un)bound Knapsack problem. With the evaluation of the framework, assumptions regarding the influence if the initial populations on the result of an EA are discussed and answered.

Contents

List of Figures

List of Tables

Glossary

$G_0 = \begin{pmatrix} X_0^0 \\ \vdots \\ X_n^0 \end{pmatrix}$ Generation 0 with n individuals.

$G_P = \begin{pmatrix} X_0^P \\ \vdots \\ X_n^P \end{pmatrix}$ generation P with n individuals.

$H\left(\Psi_j^P\right)$ Shannon-Entropy of the genome j.

P number of Generations.

P_k probability of traceID k occurring in a specific genome.

$X_i^P = \begin{pmatrix} x_{0,0}^P \\ \vdots \\ x_{0,m}^P \end{pmatrix}$ individual i of generation P with m genomes.

Ψ_j^P traceIDs of genome j of generation P.

$f(X_i^P)$ fitness function returning the fitness of the individual X_i^P.

k traceID, ind number of the initial generation.

m number of genomes per individual.

n number of individuals per generation.

$t(x_{i,j}^P)$ traceID of the gene j in individual i in generation P.

$x_{i,j}^P$ gene galue of the gene j in individual i in generation P.

Acronyms

AI artificial intelligence.

CI counting-based impact.

EA evolutionary algorithm.

EI entropy-based impact.

FEI fitness-entropy-based impact.

FI fitness-based impact.

NEAT neuroevolution of augmented typologies.

NN neuronal network.

T-EA tracable evolutionary algorithm.

XAI explainable artificial intelligence.

1. Introduction

Evolutionary algorithm (EA) are one of the three biology inspired methods in the field of computational intelligence. Based on the concept of evolution in nature, they try to solve complex optimization problems by modeling populations of solutions, evolving them over many generations to find an optimum. The problems to be solved can be highly diverse, ranging from optimising shapes of wind tourbines or areoplane wings, optimizing the structure and training neuronal networks (NNs), to applications in the medical domain.

EAs generally are considered to be understandable white boxes, but understanding the learning process is still challenging, especially for complex problems. While fields like explainable artificial intelligence (XAI) have already emerged for NNs, better understanding the results of EAs is still largely unexplored.

One important factor on the results of an EA is the initial population. While seeding the initial population is already an established field of research, all of these methods are solely based on the evaluation of the quality of the result. To better understand the influence of the first generation, this thesis proposes the tracable evolutionary algorithm (T-EA), a form of EA capable of tracking the influence of the initial population through the generations to the final result. Measuring the impact of an individual on the last generation, besides a simple counting approach, three additional metrics for measuring the impact of an individual are proposed, using the fitness of the individuals as well as the diversity of the population.

Traceability for EA has been implemented for bit vector and integer vector representations. It is evaluated using three different optimizations problems, the Max Ones problem, the 0/1 Knapsack problem and the (Un)bound Knapsack problem. First, a proof of concept evaluation of selected testruns is provided to show the information gain when evaluating single and multiple testruns. Finally, three hypotheses are used to test the T-EA, to compare the four impact metrics with each other and to gain some general knowledge about the relation between the initial population and the resulting impact of the individuals. For this reason, the hypothesis are aimed at general assumptions regarding the impact of individuals or mutation on the last generation.

1.1. Structure of the thesis

Completing this chapter, the structure of the thesis is presented in the form of the headline of the chapter, as well as a short description of its content.

Basics The second chapter explains the basics of the thesis, starting with the optimization problem in section (2.1), followed by an explanation of EA in section (2.2), finally describing the problems used to evaluate the thesis in section (2.3).

Motivation and Related Work explains both the related work as well as motivation for the thesis. The first section (3.1) and second section (3.2) show motivations from the field of XAI and automatic parameter tuning in EA. The section (3.3) showcase the related fields of study of seeding the initial population in EA. Closing the chapter is section (3.4), showcasing the current use of historical markers in EA.

Traceable Evolutionary Algorithms explains the concept of T-EAs. First, the concept of traceIDs for genes is presented in section (4.1), showing also first examples of the tracking process. Secondly in section (4.2), four different metrics for measuring the impact of an initial individual are proposed.

Experimental Setup first describes three hypothesis about the development of gene heritage through the EA in section (5.1). Secondly the test design for the three hypothesis as well as the configurations for the three problems used are presented in section (5.2).

Evaluation first provides a proof of concept evaluation, showing the capabilities of the T-EA for a single and multiple runs in section (6.1), then evaluates hypothesis explained in the previous chapter in section (6.2).

Conclusion and future Work Closing the thesis, this chapter summarizes the results discussed in the previous evaluation chapter in section (7.1) and gives an outlook on the potential future developments of the T-EA in section (7.2).

2. Basics

Understanding the basics of evolutionary algorithms (EAs) as well as the problems analyzed in this thesis are crucial when analyzing the impact of the initial generation on the last. Therefore, this chapter first describes the basics of an optimization problem, then explains the necessary parts of EAs for the topic of this thesis and finally focuses on the problems used to analyze the proposed metrics.

2.1. Optimization Problem

Optimization problems [25, p. 189] are a pair of a solution space (G) and an evaluation function ($g : G \to \mathbb{R}$), assigning the quality $g(x)$ to each candidate of solutions ($x \in G$). The goal is to find the globally best solution x' with $\forall x \in G : g(x) \leqslant g(x')$. An example of those two spaces can be seen in figure (2.1).

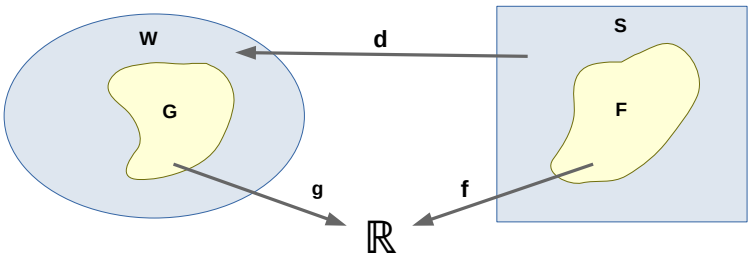

Figure 2.1.: Relation between the solution space (G) with the evaluation function (g) and the search space (S) with the optimization algorithm (f), connected by the encoding (d).

Solving an optimization problem is done by searching the best solution in the search space (S) with an optimization algorithm ($f : S \to \mathbb{R}$). Connecting the search space with the solution space is an encoding ($d : S \to G \subseteq W$).

The search space needs to include the optimal solution, but can also include unfeasible solutions ($S \nsubseteq F$), representing incorrect solutions in the solution space ($W \nsubseteq G$). One type of optimization algorithm that can solve such optimization problems are EAs, described in more detail in the next section.

2.2. Evolutionary Algorithms

EAs try to find the best solutions in optimization problems, described in the previous section. The principle of EAs is inspired by evolution in the real world. A population of solutions is evolved over many generations by recombining and altering them, filtering worse solutions by selection to gradually improve the solutions over time. Kruse et al. split EAs into the following parts [25, p. 195]:

- A **population** of *individuals*, which represents a subset of candidate solutions to the problem.
- An **encoding** for the individuals, which allows to represent candidate solutions for the given problem. Individuals can be encoded in different ways, most commonly as a data-vector, a tree or a graph. The representation of the individual is also called genotype or genome. The genotype represents a solution of the problem in the search space. The individual attributes of an individual are called genes, while a value of a gene is called an allele. The real solution in the solution space is called a phenotype.
- A **fitness function** to evaluate the individuals in the context of the problem.
- **Generations**, iterations of the EA.
- **Genetic operators** to generate new individuals, which can be done through *crossover* and *mutation*. Crossover is done by picking two individuals as the parents and combining them into new individuals, called offspring. In mutation, the genes of an individual are randomly altered to create a new one.
- Two **selection mechanisms**, one that picks the individuals for the genetic operators and another that specifies how many individuals from the old generation and how many of the offspring created by the genetic operators are picked to survive to the next generation.
- A **termination criterion** defining the conditions to end the search.

Depending on the problem, the implementation of an EA will differ. There are many types of EAs, like genetic algorithms [25, 245 ff.], evolutionary strategies [25, 257 ff.], evolutionary programming [25, 257 ff.] or many other population based approaches.

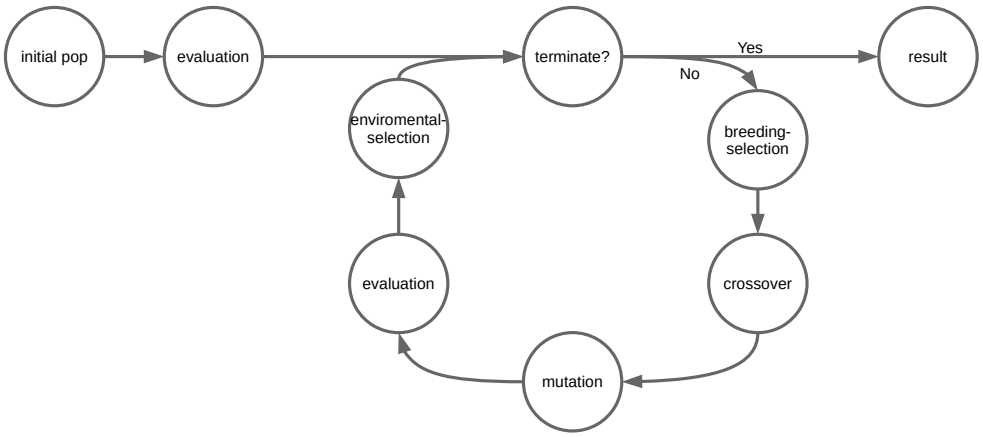

Figure 2.2.: Visual representation of the general framework of an EA.

Figure (2.2) shows the structure of a generic EA. The algorithm needs an *initial population* as an input, which can be generated or provided by the decision maker. Before starting the loop, the initial population is evaluated and tested for termination. Starting the loop, the *breeding-selection* starts the breeding-process, selecting individuals for recombination in crossover. Several selection operators were proposed in the literature. One of the simplest is the random selection, where every individual has the same probability of being selected, regardless of its fitness. A fitness based selection mechanism would be the roulette wheel selection [25, 221 f.], where the probability of being selected depends on the fitness of the individual. Individuals with a higher fitness therefore will be picked with a higher probability than individuals with a lower fitness. Another fitness based approach is tournament selection [25, 229 f.]. In this approach, the parents are picked by randomly selecting for both parents, depending on the tournament size, k individuals from the current generation. Of those k individuals, the best one wins the tournament and is selected as a parent. This process is performed for both individuals entering *crossover*.
After the selection process, the *genetic operators* are applied to generate the offspring. The number of generated offspring can vary depending on the *environmental-selection*. It is both possible to generate more offspring than the population size allows, the exact amount needed or less than the

population size, needing to select individuals from the current generation to pass into the next generation.
Figure (2.3) shows an example of a *one-point crossover* [25, p. 236] of two individuals. The genome of both is cut at one random place and the genes on the left are swapped, creating the offspring. Extending this crossover-type, *two-point* or *n-point crossover* split the individuals according to their name in two or n pieces. Other approaches include the *uniform crossover* [25, p. 237], where an exchange probability determines if a gene is swapped or not, or the *shuffle crossover*, which shuffles the genes of the selected individuals, applies another crossover type, most commonly a form of n point crossover, and finally deshuffles the genes of the produced offspring.
After crossover, *mutation* is applied by a pre-defined probability on the genes of the individuals, altering it to a new value. Depending on the implementation this new value can be based on the old gene-value or randomly generated, independent of the old gene-value [25, 233 ff.].

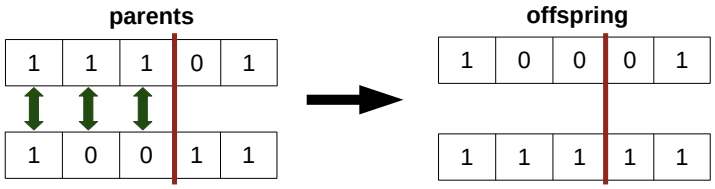

Figure 2.3.: Example of a one-point-crossover operation between two individuals, encoded as bit-vectors.

After generating the new individuals, they need to be *evaluated* with the fitness function. This function assigns a mathematical value to each individual, based on the objective of the problem. This allows the comparison of the individuals in the scope of the optimization problem. Hence optimization problems can have a single-objective or multiple-objectives. Multiple objectives can be optimized by using a separate fitness function for each objective, resulting not in a single best result but a multitude of best results in a so called pareto-front [25, 280 ff.]. However, the problems later discussed in this thesis all have a single objective. Evaluating the individuals also needs to take into account possible constraints of the problem. If a problem has constraints, invalid/unfeasible solutions can occur, meaning that a gene combination which represents a solution that is not valid and does not fulfill the constraints. Coello gives a basic overview of constraint handling in [4].

There are several options for handling such constraints, the most common being feasibility preserving methods [4, 11 ff.]. The principle of this method is to only generate feasible offspring through crossover and mutation. Other methods use a penalty function [4, 3 ff.] as an addition to the fitness function to compute the fitness. The penalty can be a fixed value, indicate the amount of violation and possibly change with the generation counter. Another possible approach is the parameter free approach [4, 15 f.]. As the name suggests, no additional penalty parameter is used, instead, feasible solution are biased through the fitness function itself, for example by strictly differentiating between feasible and unfeasible solutions assigning only positive fitness values to feasible and negative values to unfeasible solutions. This way, depending on the selection-mechanism, valid individuals have an advantage at being selected over invalid individuals.

The *environmental-selection* heavily depends on the number of generated offspring from the genetic operators. A simple selection method is the generational approach [25, p. 232], where a predefined number of the old individuals, called *elites* [25, p. 230], is passed to the next generation, while the remaining number of individuals is generated using the genetic operators, so no extra selection is required. There are several other selection mechanisms, specifying how many of the old individuals and the offspring will pass into the next generation. An example for an environmental-selection that allows generating more new individuals than needed is the *(μ+λ)-selection* [25, p. 258]. The best μ individuals among the old individuals and the offspring will pass into the next generation. λ represents the number of newly generated individuals in this case. Selecting the individuals for the next generation can also be done with the same selection methods as the breeding-selection.

The evolutionary process finally ends when the termination criteria is met. Most often this is a fixed number of generations, but it can also include other factors like reaching a fitness value or not improving for n number of generations.

2.3. Problems

This chapter explains the three problems used to evaluate the capabilities of the later proposed tracable evolutionary algorithm (T-EA). Besides the general

problem, the encoding and fitness equations are also discussed. The specific parameters for the test setup follow in a later chapter.

2.3.1. Max Ones Problem

The Max Ones Problem is a simple problem designed for testing purposes. The representation of the genotype is a bit vector or string with a fixed length, consisting of ones and zeros. The goal is to reach the maximum amount of ones in the string, so that there are only ones and no zeros. The fitness, shown in equation (2.1), is calculated by the number of ones in the vector divided by the number of genes in the genotype for normalization. The problem is unconstrained and can reach fitness values from 0 to 1.

$$f_{MaxOnes}(\vec{x}) = \frac{\sum_{i=0}^{n}(\vec{x}[i])}{n} \tag{2.1}$$

The problem is very simple and can even be solved more easily by a human than an EA. The advantage here is that the fitness value of an individual can be easily calculated by just looking at the number of ones in an individual, which makes it possible to built a specific starting population based on the fitness value. Of course, depending on the parameters, the problem can be engineered to be more difficult for EAs, for example by increasing the number of genes. The purpose for this problem is not to pose a difficult task for the evaluation, but to be able to engineer starting populations and to analyze the schema of the initial population with regards to the resulting population.

2.3.2. Knapsack Problem

The Knapsack Problem consists of the task of filling up a knapsack with items to get the maximum amount of value. The knapsack has a maximum carrying capacity and the items have a price and a weight attached to them. The goal is to find the most valuable combination of items that is still fitting into the knapsack. Some of the many variation of the Knapsack Problems are described by Kellerer et al. in [23]. The two most common types, the 0/1 Knapsack and the (Un)bounded Knapsack Problem are used in this Thesis. In the 0/1 Knapsack Problem, every item can be picked just once to be placed in the knapsack, while in the (Un)bound Knapsack Problem every item can be picked several times. The following two sections will explain the implementation of both problems with regards to the encoding, fitness function and constraints.

0/1 Knapsack Problem

In the 0/1 Knapsack Problem, also described by Sahni in [35], every item can be placed just once in the knapsack. Each possible item therefore can be represented as a bit, consisting the genome of a bit vector. The cost and weight of the items need to be stored in separate vectors, as well as the maximum weight. Figure (2.4) shows an example of the encoding of a 0/1 Knapsack Problem. The maximum weight, cost- and weight-vectors are stored globally for all individuals, while the genome of the individuals is storing whether an item is present in the knapsack or not. Since the genome is stored as a bit-vector, the values can only be 0 or 1.

Item	1	2	3	4	5
Cost	33	24	36	37	12
Weight	15	20	17	8	31

Max-weight	70

weight	fitness

Ind 1	1	0	1	1	0	40	106
Ind 2	1	0	1	1	1	71	-1
Ind 3	1	1	1	0	1	83	-17

Figure 2.4.: Example of the encoding of a 0/1 Knapsack Problem.

For handling the constraint of the maximum weight of the knapsack the beforementioned parameter-free approach [23, 187 ff.] can be used. Therefore, calculating the fitness is split into two parts: a positive fitness when the knapsack is not overweight, and a negative fitness if the knapsack is overweight. Equation (2.2) shows this fitness function, with the cost of item i being $c(i)$ and the weight of item i being $w(i)$. If the total weight in an individual ($w_{ind}(\vec{x})$) is bigger than the maximum allowed weight (w_{max}), the fitness will be the difference between w_{max} and $w_{ind}(\vec{x})$. That means, individuals who are more overweight are ranked worse than individuals being just slightly overweight. If $w_{ind}(\vec{x})$ is in the allowed range, the fitness will be the sum of the number of times an item is present in the knapsack ($\vec{x}[i]$ multiplied by the cost of the item $c(i)$). The weight of the individual is computed using equation (2.3).

$$f(x) = \begin{cases} w_{max} - w_{ind}(x) & \text{if } w_{ind}(x) \geq w_{max} \\ \sum_{i=0}^{m} x(i) \cdot c(i) & \text{otherwise} \end{cases} \tag{2.2}$$

$$w_{ind}(x) = \sum_{i=0}^{m} x(i) \cdot w(i) \tag{2.3}$$

In the example in figure (2.4), the weights of individuals is calculated with equation (2.3) and the fitness with equation (2.2). *Ind 1* has a weight lower than the maximum, therefore the fitness is calculated by the sum of the costs of the items in the knapsack. *Ind 2* and *ind 3* are both overweight, so the fitness is the difference from the actual weight to the maximum allowed weight. Since *ind 2* is less overweight than *ind 3*, its fitness value is better. That way the selection operators can differentiate between individuals which are more overweight, being less feasible than solutions which are less overweight. This ensures an improvement per generation, even if no individual of the starting population is fulfilling the weight constraint.

The disadvantage of the 0/1 Knapsack Problem in contrast to the Max Ones Problem is that it is hard to handcraft starting populations with specific characteristics. On the other hand it is a more complex, NP-hard problem, that poses a bigger challenge for EAs, potentially giving a more interesting insight when analyzing the impact of the starting population through the generations.

(Un)bound Knapsack Problem

The implementation of the (un)bound Knapsack Problem [23, p. 175] is very similar to the 0/1 Knapsack Problem. The main difference is the encoding of the individuals. Instead of a bit vector, an integer vector is used, since items can be placed more than once into the knapsack. Because integer values can be negative, and a negative amount of items is also not allowed in this problem, we need to set a min-value that the genes are allowed to reach, in this case 0 for all genes. It is also preferable to set the max-value for each gene to the maximal amount that an item can be placed into the knapsack before getting too heavy, to limit the number of times an item can be placed to a reasonable amount. Otherwise, mutation operations have a high chance of resulting in too big numbers, rendering them ineffective and hampering the improvement of the evolutionary process. An encoding of a (un)bound Knapsack Problem can be seen in figure (2.5).

Calculating both the weight (equation (2.3)) as well as the fitness (equation (2.2)) can be done with the same equations as the binary-knapsack implementation. Figure (2.5) also shows the corresponding fitness-values for each individual, similar to the previous implementation.

Item	1	2	3	4	5
Cost	33	24	36	37	12
Weight	15	20	17	8	31
Min	0	0	0	0	0
Max	4	3	4	4	2

Max-weight	70

						weight	fitness
Ind 1	2	0	1	2	0	63	106
Ind 2	0	0	0	3	1	55	123
Ind 3	1	3	0	0	0	75	-5

Figure 2.5.: Example of the encoding of a (Un)bound Knapsack Problem.

The (Un)bound Knapsack Problem was chosen to analyze a second datatype besides the bit-vector. It also proposes a harder challenge than the binary-knapsack problem, since the search-space is much bigger than in the other problem.

After presenting the basics of EAs and the problems used later in this thesis, the next chapter focuses on the motivation and related work leading to the concept of tracking genes in EAs.

3. Motivation and Related Work

Analyzing the impact of the initial population in an evolutionary algorithm (EA) is a novel approach. Though historical markers in EAs as well as the analysis of the starting population are not novel, they were never used to analyze the impact of the initial population. The practice of trying to explain the results of intelligent systems like neuronal networks is also an active field of research. This chapter first analyzes this field of explainable artificial intelligence (XAI). Then we look into the related fields of seeding the initial population and automatic parameter tuning in EAs. Finally we discuss the use of historical markers in the evolution of neuronal network topologies.

3.1. Explainable Artificial Intelligence

XAI most often referes to the interpretability problem of black box machine learning systems, especially neuronal networks (NNs). In recent years, NNs gained a lot in popularity, since advancements in computational resources, new and improved algorithms and the availability of data grew over time. Through those advancements, the application in critical areas like the medical domain, self driving cars or the finance sector become more common. With the use of such systems in those areas, it is being questioned if we can trust black-box systems like NNs. The problem most often derives from segregated input data, as Zou and Schiebinger mention in their article [45], especially in the medical domain, because generating data here is often costly. An example is a recent algorithm from Esteva et al. detecting skin cancer [10], which was trained with 129,450 images, but fewer than 5% of those images represented dark skinned individuals [45]. To regain the trust in artificial intelligence (AI), the field of XAI has emerged, trying to shine a light into the black box that is the decision process of NNs.

There are already a variety of approaches trying to explain the decisions made by AI. Dosilovic et al. categorize them into integrated interpretability, ad-hoc interpretability and ad-hoc explainability [9]. They define interpretability

as being able to humanly understand the decision-making method and explainability as explaining the decision of more complex structures through the collection of interpretable features. The three categories can be described as follows:

integrated interpretability Integrated interpretability methods are only relying on interpretable structures itself like decision rules, trees or tables. In [20], Huysmans et al. these three approaches are compared. However, limiting the structure to be humanly understandable is limiting performance for interpretability. There are also hybrid models, like [16] proposed by Gestel et al., that combine an interpretable model with some black box operators to increase performance, at the expense of interpretability.

post-hoc interpretability Post-hoc methods on the other hand try to explain more complex structures, which can not easily be interpreted by itself. Interpretability can still be achieved, e. g. through interpretable proxy models, meaning models with integrated interpretability resembling more complex structures like NNs. Zoho et al. [44] for example generated rules approximating an assembly of neuronal networks. There are also visualization-based approaches, for example Zeiler and Fergus [43] tried visualizing the intermediate layers of a NN while Cortez and Embrechts [5] are proposing the sensitivity analysis method, trying to visualize input effects on the model response and feature importance.

post-hoc explainability Explainability methods also provide an explanation in the form of text or visualizations for every decision. An example for text explanations comes from Hendricks et al. [19], proposing so called "visual classification" describing image classification. Datta et al. [7] presents a method explaining the outputs with the degree of influences from the inputs.

EAs are widely considered to be white boxes, since the learning process itself often times is understandable. However, the learning-process quickly becomes incomprehensible for humans when using complex fitness functions. Although the result is interpretable through the fitness function most of the time, and the evolutionary process is comprehensible for humans, understanding the origin of the result is hard to understand, especially for complex problems. Tracing the impact from the initial population through an EA can be seen as a first step towards a post-hoc explanation of the evolutionary dynamics when generating results with such algorithms.

3.2. Automatic parameter tuning in EAs

One motivation for tracking the impact of the initial population in EAs is to better understand the evolutionary dynamics, subsequently leading to a better understanding and improved parameter tuning. EAs have many parameters to be set beforehand, for example the population size, the mutation rate or the number of generated offspring. The reason for this configurability is the wide range of possible problems with differing optimal settings. Finding the right parameters is important for the success of the EA. While it is still common to tune the parameters based on conventions, ad hoc decisions or testruns of varius configurations, automated approaches have been developed, with irace [27] being a popular parameter tuning framework.

The name irace stands for iterative racing. It generally consists of three steps:

1. sampling new configurations according to a particular distribution

2. selecting the best configurations from the newly sampled ones by the means of racing

3. updating the sampling distribution in order to bias the sampling towards the best configurations

The term racing describes the analysis of the different configurations. A race starts with all configurations being analyzed for a fixed amount of steps. After these initial steps, the worst performing configurations are discarded and the race continues with the remaining, *surviving*, configurations, repeating the process of discarding solutions until a minimal number of configurations is met. This allows for a faster, less resource demanding runtime than entirely analyzing all the configurations. The distribution for the next generation of configurations is then altered towards the best found in the race by modifying the mean and standard deviation. This process is repeated until a termination criteria is met, similar to an EA.

However, although improvements can be shown with automatic parameter tuning [27], the underlying process is a black-box, not leading to a better understanding of why the parameters are tuned the way they are. Since the procedure is iterative, very computationally intense problems still pose a problem to runtime. Insights on the evolutionary dynamics could potentially lead to an improvement of the parameter tuning, whether by hand or automated.

Year	Authors	Approach	Result quality	Runtime	Population Diversity
2004	Maaranen, Miettinen, and Mäkelä	[28]	yes	yes	no
2005	Kimura and Matsumura	[24]	yes	yes	no
2013	Kazimipour, Li, and Qin	[22]	yes	no	no
2009	Pant, Thangaraj, and Abraham	[29]	yes	no	yes
2007	Rahnamayan, Tizhoosh, and Salama	[33]	yes	yes	no
2012	Dong et al.	[8]	yes	no	no
2007	Gao and Wang	[14]	yes	yes	no
2012	Gao and Liu	[13]	yes	yes	yes
2011	Gutiérrez et al.	[18]	yes	yes	yes
2007	Uy et al.	[40]	yes	yes	no
2012	Peng et al.	[31]	yes	no	no
2001	Leung and Wang	[26]	yes	no	no
2009	Gong et al.	[17]	yes	no	no
2008	Rahnamayan, Tizhoosh, and Salama	[34]	yes	yes	no
2010	Peng and Wang	[32]	yes	yes	no
2009	Wang et al.	[41]	yes	yes	no
2009	Dasgupta et al.	[6]	yes	yes	no
2013	Ali, Pant, and Abraham	[2]	yes	yes	no
2013	Sharma and Tyagi	[37]	no	no	no
2017	Younis, Yang, and Passow	[42]	yes	yes	no
2017	Gaina, Lucas, and Perez-Liebana	[12]	yes	no	no
2015	Friedrich and Wagner	[11]	yes	yes	no

Table 3.1.: Comparison of the evaluation of seeding techniques in literature.

3.3. Seeding the initial population in evolutionary algorithms

Studying the effect of the initial population on the result in EAs has been a popular field of study over the last decades. The process of initializing the initial generation on EAs is also known as seeding the initial population. Although EAs most of the time are initialized randomly [21], studies have been done on other seeding methods, trying to improve the performance of the EA. However, the evaluation of such seeding techniques is almost always done by studying the quality of the results and the runtime. Table (3.1) shows a collection of papers which propose or analyze seeding techniques in EAs, comparing the evaluation process. Most papers ran performance tests based on the quality of the result, with the quality being measured by the success rate or the approximation quality of the optimization. Additionally, the computational effort was also analyzed on most cases. However, only [29], [13] and [18] analyzed an additional parameter, the population diversity. All three papers analyze the initialization of particle swarm optimization, where the population diversity plays an important role for adjusting exploration and exploitation. However, no additional parameters where analyzed when comparing the results between different seeding techniques.

This work tries to close this gap by studying the impact of the starting population on the last generation. Analyzing the dynamics inside an EA through the impact of single individuals might yield new information which could improve seeding procedures in the future.

3.4. Historical markers for genes in other approaches

The paper *Evolving Neuronal Networks through Augmented Topologies* [38] from Stanley and Miikkulaien published in 2001 was the first publication where historical markers were proposed to track the heritage of genes in EAs, though not for analyzing the impact of the starting population. Standley and Miikkulaien proposed a new method for generating both the weights and the topology of NNs with evolutionary algorithms. Historical markers, or *innovation-numbers* as they are called in the paper, are markers attached to the data of a gene, pointing back to its origin. To understand why innovation numbers are necessary for their method, the encoding of the NNs in the EA, the mutation and crossover operation as well as the competing conventions problem needs to be explained.

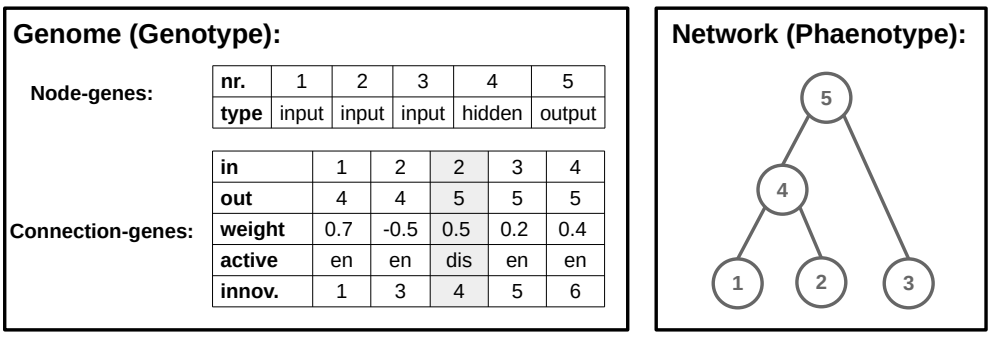

Figure 3.1.: Neuroevolution of augmented typologies (NEAT) encoding of a NN. The genome is shown on the left and the resulting network on the right. The genome is split into two vectors, the node genes, defining the nodes, and the connection genes, defining the connections of the network.

Figure (3.1) shows the encoding of an individual, with the corresponding network. The NEAT encoding, NEAT standing for neuroevolution of augmented topologies, is a direct encoding of the network structure. As the example

shows, the genome is split into two types of genes, the node genes and the connection genes. As their names suggest, the node genes define the nodes of a network, specifying the role as input-, hidden- or output nodes. The connection genes define the connections between the nodes, storing the in and out nodes that they connect, a weight, a flag if the connection is enabled or disabled, and an innovation number, the equivalent tracking number to the traceID in this thesis. The NEAT encoding does not strictly fulfill the structural requirements of a NN, so infeasible individuals are possible. Consequently, the mutation and especially the crossover operators need to satisfy these constraints to produce feasible offspring.

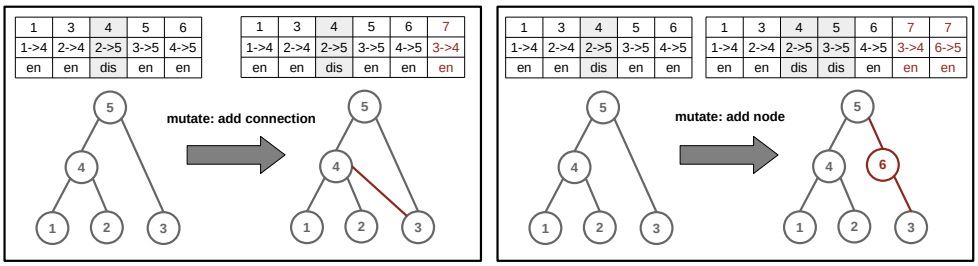

Figure 3.2.: The two types of mutation in NEAT, the add connection and add node mutation. The encoding for the connection genes is shown in a reduced form over the networks, with the innovation number on top, following the in and out nodes and the enabled flag on the bottom.

The two types of mutation operations, one for each gene type, can be seen in figure (3.2). On the left side is the *add connection* mutation, which adds a new connection to the network. The right side shows the *add node* mutation, which adds a new hidden layer node on an existing connection into the network. Adding a new node requires the deactivation of the old connection and the addition of two new connections, connecting the old "out node" with the new, as well as the new node with the old "in node". All newly added connections get an individual innovation number counted upwards. The only exception, giving two connections the same innovation number, occurs if by chance the same connection is formed in two individuals at the same time. Giving two connections the same innovation number is necessary, since they are used to differentiate them in crossover, as later described. If the same connection is added multiple times in the same generation, they should therefore have

the same innovation-number. There is also a third, non structural mutation, altering the weights of the connections.

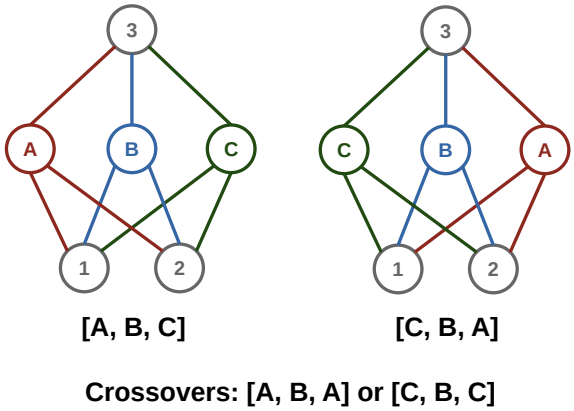

Figure 3.3.: The competing-convention-problem showing 2 out of 6 possible permutations of the same solution of a NN. Without matching the nodes, the resulting offspring will lose information.

Since the network's growth is unconstrained in the NEAT encoding, the crossover should make sure that it only produces valid new individuals without unconnected nodes or connections with no in or out node. Not getting stuck in local optima, the resulting networks also should represent good solutions. Besides unfeasible networks, information loss is a big issue in the so called competing conventions problem, shown in figure (3.3). The example consists of two permutations of a solution representing the same result. On the left side, the hidden-layer-nodes are ordered [A, B, C], and on the right side [C, B, A]. Crossing over two permutations of the same solution almost always results in information loss, resulting in the networks with hidden-layers [A, B, A] or [C, B, C]. Comparing the topology with a dedicated algorithm is computationally expensive, therefore the paper proposes a way to match genes by their historical information through the innovation number. In the crossover operation, genes can be exactly matched, since the same innovation number implies the same heritage. Figure (3.4) shows the matching of the genes using this information. Non matching genes are classified *disjoint* if they occur inside the range of the innovation numbers of both parents, and *excess* if they are outside the range of one parent. Since the weights of the individuals could have developed independently, the same innovation numbers do not imply the same weights in both individuals. Matching genes are therefore chosen randomly

from both parents, while the disjoint and excess genes are chosen from the parent with the higher fitness. In the example, the same fitness is assumed, so the disjoint and excess parts of both individuals are chosen. This way, the resulting genotype represents a valid solution with no notable information loss, and without computationally intense operations slowing the evolutionary process.

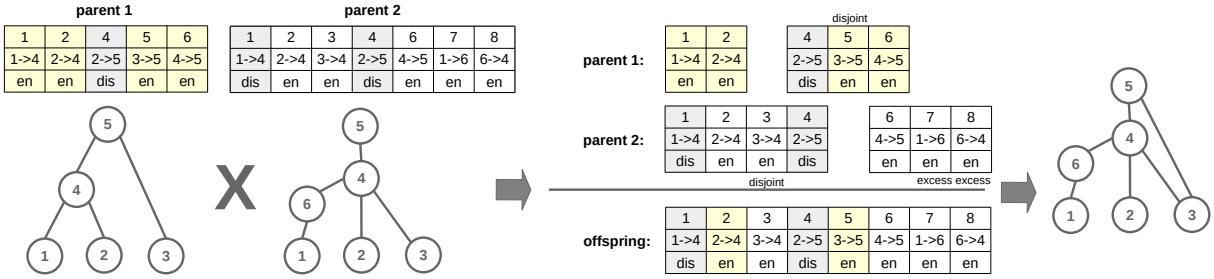

Figure 3.4.: Example crossover operation in the NEAT encoding. Two individuals are combined by sorting the genes into matching, disjoint and excess by their innovation number. Matching genes then are chosen randomly, disjoint and excess are chosen from the better fitness parent. The individuals are assumed to have the same fitness in this example, therefore the genes from both parents where picked.

The NEAT encoding later was expanded by Stanley and Miikkulaien to HyperNEAT [15], generating connective compositional pattern producing networks (connective CPPNs) for the generation of complex connectivity patterns with regularities and symmetries. Thereafter HyperNEAT was further developed to the ES-HyperNEAT [3].
Also in the original NEAT encoding, L. Pastorek and M. O'Neill proposed and compared a context-free approach to assign innovation-numbers, meaning that the same connections are always assigned the same innovation-number, not just if the same connection appears in the same generation [30].

As a historical marker, the innovation-number differs in a few ways from the traceIDs in this thesis, mainly since the innovation-numbers are not directly linked to the weight of a connection, with the only purpose of tracking the heritage for non computationally intense comparison reasons in crossover. In contrast, the traceIDs explained in the next section are directly linked to the data of a genome, changing if they are randomly mutated, since the goal is the computation of the impact of the starting individuals.

4. Traceable Evolutionary Algorithms

This chapter contains the description and definition of the tracable evolutionary algorithm (T-EA). The T-EA operates like a normal evolutionary algorithm (EA), with the addition of traceIDs, a historical marker pointing back to the origin of each gene. From these markers, the impact of the first generation into the last can be computed. The following sections provide an explanation of traceIDs and the four different impact metrics. At first traceIDs are described, how they are implemented and how they work in the EA. Secondly, this chapter focuses on the calculation of the impact from the traceIDs, as well as discussing differences, advantages and disadvantages of each metric.

4.1. TraceIDs

To be able to trace the ancestry of genomes of an individual, each gene needs an identifier linking it to the initial population. This Identifier is called traceID. To link traceIDs and data, they are stored in a tuple, called traceable datatype. In this case, we have a traceable boolean for bit vectors and a traceable integer for integer vectors. All operations can be performed like in a typical EA, using the data part of the tuple.

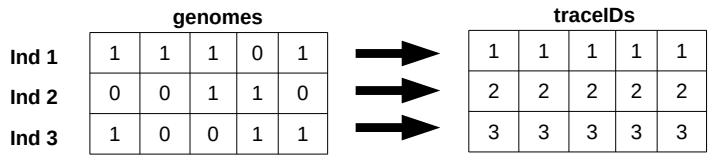

Figure 4.1.: Example of the genomes of a population in the Max Ones Problem (left) and the corresponding traceIDs (right) in initialization.

Figure (4.1) shows a made up example of a bit vector starting population. For better readability the tuples are displayed as two separate vectors with the data part on the right and the traceIDs on the left. The traceIDs are initialized with the corresponding individual number, so Ind 1 gets traceID 1 in each genome, Ind 2 with traceID 2 and so on. Mutated genes get assigned negative traceIDs counting down from -1 for each occurring mutation. Going through crossover and mutation operations the traceIDs will mix and show the origin of a given gene in the starting population. The following part first shows how the traceIDs behave in crossover and how mutation is handled.

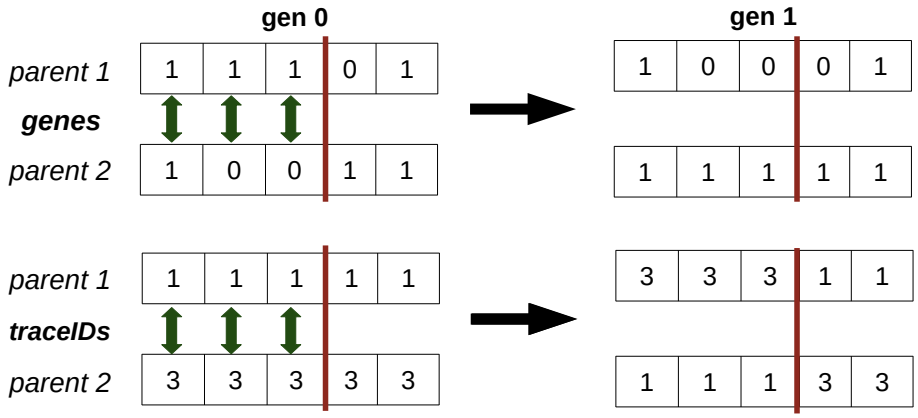

Figure 4.2.: Example of a crossover operation from a Max Ones problem where the genomes are on the top and the corresponding traceIDs on the bottom.

Figure (4.2) shows an example of a one point crossover operation on a bit vector. The top shows the data part of the tuple, the bottom shows the traceIDs. Just like a typical crossover operation, the genomes of two individuals are cut and the pieces are rearranged into two new individuals. Since the genes are a tuple of data and traceIDs, the genes in the offspring in gen 1 can be traced to the genes of the individuals in gen 0. Two point and n point crossover, as well as other crossover types, would work in a similar way, since the traceIDs are directly linked to the single genes. Crossover types that combine the genomes of two individuals instead of cutting and mixing are currently considered. In further generations, the traceIDs will get more mixed up than in this example, like in the figure (4.3), showing the first and

last generation in the Max Ones Problem.

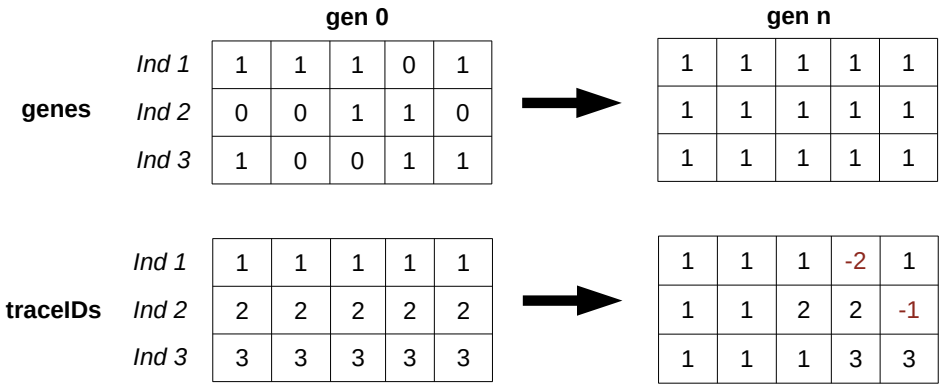

Figure 4.3.: Example of an initial population (left) and a final population (right) with the corresponding traceIDs in the Max Ones problem.

In figure (4.3) we can see an initial population of a Max Ones Problem, with the last generation on the right side and the traceIDs beneath them. Colored in red are negative traceIDs, showing the mutations. Since this thesis focused on uniform mutations, it was sufficient to assign negative values to mutated genes, since they are not influenced by the original gene. To differentiate the mutations, the negative values are counted down from -1 to the negative of the overall occurred mutations. Not assigning every mutation the same ID allows for the analysis of the impact of each occurring mutation instead of just all of the mutations combined, as well as computing an entropy over each gene which is needed for the impact metrics explained in the next section. Mutation types depending on the original gene data are not sufficiently tracked by this approach, since there is no link from the negative traceID to the original gene, which could be expanded in the future.

Through crossover and mutation the last generation of individuals of figure (4.3) are all optimal. The resulting last population of each generation is just a mix of the genomes from the initial generation, plus those newly generated by mutation. Looking at the traceIDs we can analyze what gene came from which starting individual. This way we can see that in genes 1 and 2, individual 1 is dominating through the whole generation, while genes 3 and 4 are more mixed

with different traceIDs. Besides analyzing such phenomenon, counting the traceIDs can be used to compute a more general impact metric of a starting individual on the last generation, which will be discussed in the following section.

4.2. Impact Metrics

To describe the impact from an individual of the initial population on the last generation using T-EA, this thesis proposes four different measures called impact metrics. In the following sections, first the counting-based impact (CI) is proposed, extending it to a fitness-based impact (FI), an entropy-based impact (EI) and finally the combination of both, the fitness-entropy-based impact (FEI).

4.2.1. Counting-based Impact

As already hinted in the traceIDs section, the simplest approach to calculate the impact is the counting-based impact (CI), presented in equation(4.1). This metric is calculated by summing the occurrence of the traceID of starting individual k. The function $t(x^P_{i,j})$ returns the traceID of gene j in individual i. The resulting sum is divided by the number of genes in the whole population $(n \cdot m)$ to normalize the values.

$$I_C(k) = \frac{1}{n * m} \sum_{i=0}^{n} \sum_{j=0}^{m} \left(t\left(x^P_{ij}\right) == k \right) \tag{4.1}$$

The CI represents a simple and normalized metric which directly shows the impact from the initial generation into the last. Figure (4.4) shows an example of a last generation of a Max Ones Problem. To compare the CI to the other three impact metrics, the figure also shows the fitness of the individuals as well as the entropy of the traceIDs, which is needed to calculate the other impact metrics. The resulting impacts can be seen in the table (4.1). Focusing on the results of the CI, we can see that individual 1 from the starting population has the highest impact with 35% of the genes having its traceID, followed in order by individual 3 with, individual 2 and finally individual 4 with the lowest impact of 10%.

	data					fitness	traceIDs				
Ind 1	0	0	0	0	0	**0**	1	1	3	3	3
Ind 2	1	1	1	1	1	**1**	2	2	2	2	2
Ind 3	0	0	0	0	0	**0**	1	1	4	4	3
Ind 4	0	0	0	0	0	**0**	1	1	3	3	1
						entropy	0.8	0.8	1.5	1.5	1.5

Figure 4.4.: Example of a generation in the Max Ones problem. In addition to the gene data, the fitness, the traceIDs as well as the entropy of the traceIDs are shown.

individuals	CI	FI	EI	FEI
traceID 1	0.35	0.28	0.3	0.24
traceID 2	0.25	0.4	0.25	0.4
traceID 3	0.3	0.24	0.34	0.27
traceID 4	0.1	0.08	0.11	0.09

Table 4.1.: The normalized impacts from figure (4.4). The values are rounded to two decimal places.

However, the CI sums up each occurrence with no regard to the fitness of an individual, which might not be an ideal representation, since individuals with a higher fitness have a bigger impact on the successive generations.
Besides taking fitness into account, it also might be preferable to look at the entropy of each genome when computing the impact. A traceID in a non dominated gene might have a higher impact than a traceID in an already dominated and therefore decided gene. As well as the FI, the EI is discussed in more detail in a later section.

4.2.2. Fitness-based Impact

To address the imbalance when counting every traceID the same in the CI, even though the fitness values might be different, the fitness-based impact (FI) takes fitness into account. FI is calculated similarly to CI, by summing up the occurrences of the traceID of an individual, like equation (4.2) shows. But instead of summing each occurrence of a traceID, it is summed up by $fd\left(X_i^P\right)$, the distance the fitness ($f(X_i^P)$) of individual X_i^P to the lowest fitness in generation P, $f_{min}(P)$, shown in equation (4.3). TraceIDs in individuals

with a higher fitness will therefore add a higher value than individuals with a lower fitness, resulting in a higher impact from those individuals.
Normalizing the fitness this way is required, since negative values can occur in constraint problems, like the Knapsack problems in this thesis, if the solution an individual represents is unfeasible. Multiplying by a negative value would therefore result in a negative impact, which should be avoided.
However, the resulting impact is still dependent on the overall fitness of the generation, making it unnormalized. Although it is possible to just use the unnormalized values, it is harder to compare results between generations with a high difference in the overall fitness, or to the other impact-metrics. To transform the FI back to a percentage-value between 0 and 1, we need to normalize it. Equation (4.4) shows the normalization process, dividing the calculated $I_F(k)$ by the sum of the impacts of all traceIDs.

$$I_F(k) = \sum_{i=0}^{n} \sum_{j=0}^{m} \left(t\left(x_{ij}^P\right) == k\right) \cdot fd\left(X_i^p\right) \tag{4.2}$$

$$fd\left(X_i^p\right) = f(X_i^p) - f_{min}(P) \tag{4.3}$$

$$I_F^{norm}(k) = \frac{I_F(k)}{\sum_{i=0}^{n} I_F(i)} \tag{4.4}$$

The resulting impact values from the CI example in figure (4.4) can be seen in table (4.1). Even though traceID 1 occurs most often and has the highest CI, its FI is just the second highest. TraceID 2, making up only 25% of the last generation, has the highest FI, since individual 2 in which it occurs has a higher fitness than the individuals of the other traceIDs. Having a clearly better fitness than the other individuals makes individual 2 more likely to survive into the next generation, meaning the traceIDs in that individual are more likely to survive into the next generation as well. This evolutionary advantage is clearly reflected in the FI of this example, presumably showing a more accurate distribution than the CI. The downsides of calculating the FI is that it is slightly more computationally expensive than the CI, since an additional normalization step is needed. It is also a more abstract impact-metric, since it is a little more complex than just counting the traceIDs.

To compare the impact-metrics outside of theoretical examples, the results of the impact-metrics are further compared in chapter 6.

4.2.3. Entropy-based Impact

The idea behind the entropy based impact calculation is a little more complicated than the FI. If a gene is dominated by a traceID, meaning the traceIDs for a specific gene are equal across all individuals, the impact will be counted many times, which leads to a high CI and possibly FI. Since this gene is already decided and not prone to change in following generations, the impact might be unproportionally high. Genes which are consisting of many different traceIDs on the other hand are not yet decided and have therefore a higher impact on the next generations than dominated and decided genes.

To balance that phenomen, the EI takes the Shannon-entropy of each gene into account, just like FI takes fitness into account. The entropy for each gene is a suitable metric for this. Equation (4.6) shows how the Shannon-entropy $H\left(\Psi_j^P\right)$ [36] is calculated. It is the negative sum of P_k times the logarithm of base 2 of $H\left(\Psi_j^P\right)$ is therefore low if there are few different traceIDs in one gene and high if there are many different traceIDs. Equation (4.5) shows how the EI is calculated. Just like in the FI, instead of counting up one for every traceID occurrence, the EI counts up by the entropy of the gene the traceID counted is located. Since $H\left(\Psi_j^P\right)$ can be zero if the gene is totally dominated by a traceID, a one is added, to ensure that the impact never counts 0. Like the FI, the EI needs to be normalized at the end for comparison reasons, shown in equation (4.5), since the entropy of the genes is not summing up to one.

$$I_E(k) = \sum_{i=0}^{n}\sum_{j=0}^{m}\left(t\left(x_{ij}^P\right) == k\right) \cdot \left(1 + H\left(\Psi_j^P\right)\right) \tag{4.5}$$

$$H\left(\Psi_j^P\right) = -\sum_{i=0}^{n} P_k \cdot log_2(P_k) \tag{4.6}$$

$$P_k = \frac{1}{n}\sum_{j=0}^{n}(t(x_{0,0}^P) == k) \tag{4.7}$$

$$I_E^{norm}(k) = \frac{I_E(k)}{\sum_{i=0}^{n} I_E(i)} \tag{4.8}$$

Looking at the EI in (4.1) from the example in figure (4.4), we can see that traceID 3 actually has the highest EI, compared to traceID 2 for the $I_F^{norm}(k)$ or traceID 1 for $I_C(k)$. Looking at the distribution of the traceIDs in the genes, with their respective entropy, we can see that traceID 3 is present in genome 3, 4 and 5. These genomes have a higher diversity in them than the first two, resulting in a higher entropy. Since traceID 3 is present in the genes with the higher entropy, it is valued higher when the impact is calculated. The EI is an interesting metric especially in cases where the entropy of the genes is not equal. If the entropy is similar over all genes, it will return similar values like the CI.

The disadvantage of including the entropy is the higher computation time compared to the other two approaches. Since the entropy of each gene needs to be calculated beforehand, the run time therefore is not linear any more with $O(n^2)$. The current implementation of the impact also does not take into account the case of two traceIDs having the exact same data-value in the same gene. If both traceIDs have the same data-value, they could potentially both survive, resulting in a higher entropy, even though the gene has the same data-value for every individual.

To analyze this phenomenon, as well as getting a better understanding of the usefulness of the EI, the results of I_E^{norm} need to be studied in more detail in chapter 6.

4.2.4. Fitness-Entropy-based Impact

The final impact, the combined FEI takes both the fitness of the individual as well as the entropy of each gene into account. The result can be seen in equation (4.9). Just like I_F or I_E, the combined impact needs to be normalized since it is this time dependent on both the overall fitness of the population, as well as the summed entropies of each genome.

$$I_{E\cdot F}(k) = \sum_{i=0}^{n}\sum_{j=0}^{m}\left(t\left(x_{ij}^{P}\right) == k\right)\cdot f\left(x_i^p\right)\cdot\left(1 + H\left(\Psi_j^P\right)\right) \tag{4.9}$$

$$I_{F\cdot E}^{norm}(k) = \frac{I_{E\cdot F}(k)}{\sum_{i=0}^{n} I_{E\cdot F}(i)} \tag{4.10}$$

Taking both fitness and entropy into account should result in the most differentiated impact of the four proposed. The results in table (4.1) from figure

(4.4) show that traceID 2, which has the highest FI, also has the highest combined impact. TraceID 3, which has the highest EI, has the second highest overall impact, with traceID 1, having the greatest CI, just reaching the third highest combined impact.

The advantage of $I_{F.E}^{norm}$ is, that it takes both corrections to the original impact-metric into account. Its disadvantage is the high computation-time from the entropy. Of course, more testing is needed to analyze the results further than the theoretical examples.

5. Experimental Setup

To evaluate the proposed tracable evolutionary algorithm (T-EA), this chapter describes the test setup for the evaluation. First, three hypothesis about evolutionary algorithms (EAs) are formulated, to test and compare the results of the four impact metrics. After that, the test configurations and experimental setup for the three hypothesis are discussed.

5.1. Hypotheses

To evaluate the T-EA, three hypotheses are used. The goal of these hypotheses is to show the capabilities from gaining knowledge from EAs through the traceIDs and to compare and test the four impact metrics in the process. The evaluation should answer some general assumptions about the relation between the starting fitness of the individuals and the result, regarding the impact of the individuals as well as the impact of mutation. For this reason, the following three hypotheses where formulated:

1. The ranking of the 5 highest fitness individuals from the initial population should match the mean impact ranking of the final generation over the 31 runs.

2. If the fitness is the same for all individuals of the initial population, the medium impact on the last generation over 31 runs will be relatively equal.

3. The better the best fitness individual of an initial population, the lower the mutation impact will be in the final generation.

To evaluate the hypotheses, the next section will focus on the test design and the configurations used for the testruns.

Parameters	Max Ones	0/1 Knapsack	(Un)bound Knapsack
generations	20		
population size	20		
breeding selection	tournament selection with a tournament size of 2		
crossover type	two piont crossover		
mutation type	bit flip mutation		uniform mutation
mutation probability	1%		5%
elites	2 per generation		

Table 5.1.: Overview of the general test configurations of the three problems.

5.2. Test design

To evaluate the hypotheses presented in the section before, the three problems explained in section 2.3 are used. The different parameter configurations as well as the different initial populations used are discussed in this section.

Table (5.1) shows the general test configurations for all tests. Every test is conducted over 20 generations with 20 individuals, with tournament selection with a tournament size of 2 as a breeding selection and two point crossover as the crossover operator. The difference in mutation type and probability from the (Un)bound Knapsack problem to the other two problems derives from the different encoding. In every run, 18 new individuals are generated and pass with 2 elites from the current generation as the offspring into the next generation, so no extra environmental selection is required in this case.

The genome size as well as the seeding of the initial population is varying for each hypothesis and will be discussed in the following sections.

Since every hypothesis has different requirements, different tests need to be used to evaluate them. For example hypothesis 2 needs starting populations with an equal fitness in each individual, while hypothesis 1 needs an initial population with a different initial fitness, allowing for ranking them by fitness. In the following parts, the different tests and evaluation techniques used for the three hypotheses are presented.

Hypothesis 1

To evaluate the first hypothesis, all three problems are used. This way, potential results specific to a problem or an encoding can be differentiated. Each problem is run with three different difficulties, resulting in nine different configurations. Tuning the difficulty is done by adjusting the genome length for

Problems	Easy	Medium	Hard
Max Ones	10	30	100
0/1 Knapsack	10	20	50
(Un)bound Knapsack	5	10	20

Table 5.2.: The genome length used for every difficulty level of the three problems.

each problem. Table (5.2) provides an overview of the genome lengths used. The knapsack configurations used are from the Universidad del Cauca [1] and can be also found in Appendix A.

To analyze the behavior of the impact over multiple runs, the population used for each problem needs to stay the same. To avoid distorted results due to an unproportionally high or low fitness in the starting population, a selection of five different initial populations is used. To get an approximation of which starting populations have a good or a bad fitness, 1000 initial populations were generated and sorted by their best fitness individual. From those, the two populations with each the highest and lowest best fitness are chosen. In addition, three other populations with equal distance between each other and the worst and best are chosen. Throughout this thesis, each of the initial populations, short pop, is referred to with a number, corresponding to its initial fitness. Pop 1 refers to the worst initial population, pop 2 to the second worst, pop 3 to the average, pop 4 to the second best and pop 5 to the best population the problem is initialized with. Therefore, each of the nine problems is run with five different starting populations, resulting in 45 tests overall. For all tests, the median over 31 testruns will be analyzed.

To evaluate the first hypothesis, the mean impact rankings over the 31 testruns of the top 5 initial fitness individuals of a given population are compared with each other. Since every problem is run in three difficulties with five different initial populations, 15 rankings are compared. This is totalling to 45 comparisons over different problems in different difficulties, which is a basis for a meaningful result.

Hypothesis 2

To compare the impact in same fitness initial populations, the randomly generated populations from hypothesis 1 can not be used. Instead, they need to be specifically built to have the same initial fitness. Building initial same fitness populations is a complex task in the Knapsack problems. Furthermore, the low amount of items used for these tests often results in a low diversity when

trying to achieve the same fitness. Therefore, to analyze hypothesis 2, only the Max Ones problem is used, since same fitness populations can achieve a higher diversity as well as being built more easily.

Like in table (5.2) for hypothesis 1, the Max Ones problem is run with three different difficulties through adjusting the genome length. The three difficulties are also run with five different initial populations, with a fitness of 0.1, 0.3, 0.5, 0.7 and 0.9 for each individual. To create the initial populations, the starting populations of the Max Ones problem from hypothesis 1 were randomly altered to fit the fitness values. All configurations are also analyzed by the median of 31 runs.

To evaluate the difference of the final impact of each traceID, the difference from the highest to the lowest mean impact in the last generation over the 31 runs was used.

Hypothesis 3

Analyzing hypothesis 3 does not require separate tests, but can be done by analyzing the results of the three tests described in hypothesis 1 as well as the same fitness Max Ones test used in hypothesis 2 as an interesting extreme case. Since the selected initial populations are already ranked by their best initial fitness, the mutation impact can directly be compared for populations of the same problem and difficulty.

To evaluate the hypothesis, the average mutation impact over 31 runs is used. The results of the five different populations used for each difficulty in each problem then can be compared, since they all are run with the same test configurations. As the initial populations 5 always have a higher best initial fitness than the other populations, because of the selection of the initial populations described in the explanation of hypothesis 1, the resulting mutation impact of them is assumed to be lower. This relation between the best initial fitness and the resulting mutation impact will be evaluated in the next chapter, to show if the assumption of the hypothesis 3 can be found true or false.

6. Evaluation

To evaluate the four impact metrics along with the tracable evolutionary algorithm (T-EA), this chapter first provides a proof of concept evaluation. For this, two populations of the easy 0/1 Knapsack are picked, first evaluating a single testrun from both, then the results of all 31 runs for both populations are evaluated. Secondly, the three hypotheses of this thesis are evaluated and discussed.

6.1. Proof of concept evaluation

To show the capabilities of the T-EA and bridge the gap between the evaluation of a single run briefly explained in chapter 4.1 and the evaluation of the hypotheses, this chapter first provides a proof of concept. First, an example evaluation of a single run, then an evaluation of multiple runs, before evaluating all tests for the hypotheses in the next section.

6.1.1. Evaluating a single run

To show the capabilities of T-EA in a single-run analysis, two runs, each from a different starting population, were picked. First, the 24th run from population 2 in the easy 0/1 Knapsack problem is discussed, followed by run 3 of population 5 of the easy 0/1 Knapsack problem. Both runs show different results, but still can be compared to each other since they are a result for the same problem and run with the same configurations, just with different initial populations.

Run 24 of population 2 from the easy 0/1 Knapsack problem

Figure (6.1) shows the data and results for testrun 24 of population 2 for the easy 0/1 Knapsack problem. This figure consist of multiple graphs showing different metrics and information. Graph (6.1.a) shows the fitness of the initial population, graph (b) the fitness over all generations, graph (6.1.c) the combined entropy per generation and graph (d) the combined entropy and fitness.

The combined entropy of a generation is the sum of the entropy of every gene in the population of that generation. The graphs (6.1.e), (6.1.f), (6.1.g) and (6.1.h) show the four different impact metrics.

Looking at graph (6.1.e), showing the results of the counting-based impact (CI), we can see every traceID starts with the same value, since all genes are still in their initial positions. This already changes in generation 1, with the amount of traceIDs found in this generation dropping from the initial 20 to only 10, with the mutation impact also appearing. Only about half of the initial 20 traceIDs passing into the first generation can be explained through the tournament selection operator used, and the elite size of two. It can also be noted that only four of the initial traceIDs survived to the last generation. While traceID 3 had the highest impact in the first generation, its impact drops over the next generations, rising again after generation 7 to the second highest CI at the end. TraceID 6, having the highest initial fitness, starts with a lower impact in generation 1. Even though its impact rises in generations 2, 3 and 4, it quickly drops again settling with the lowest impact reached in the last generation. The second highest initial fitness traceID 11 also has the second highest impact in the second generation. Then it gets the highest impact from the third generation until the end. TraceID 2 also survives to the last generation with the fourth highest CI, even though it has a negative initial fitness. The mutation impact with traceID m steadily rises until generation 7, then changes only slightly in the further generations, finally having the third highest CI after traceIDs 11 and 3. From generation 10 on, the CI is relatively similar, with only minor changes occurring.

Comparing the CI shown in graph (6.1.e) to graph (6.1.f), where the fitness-based impact (FI) metric is shown, some differences can be found. While the CI starts with the same impact values for all individuals, the FI already has impact differences in the first generation, since the initial fitness of the individuals is not the same. The values for generation 1 are also different, with the impact for the traceIDs 2, 8, 9, 15, 19 and 20 being noticeably lower in the FI than in the CI. On the other hand, the impact of the traceIDs 3, 6, 11 and 16 is higher in the FI than in the CI. The traceIDs which do not survive to the next generations are rated lower and the traceIDs which do survive to the next generation are rated higher, with the exception of traceID 2 and 16. However, the values for the FI and the CI get more similar in further generations, being nearly indistinguishable in the graphs from generation 10 on. This could be due to the individuals having a similar fitness from this generation, as shown in graph (6.1.b). In generation 9, the fitness graph reaches its highest value for the first time, with all individuals having a positive fitness value.

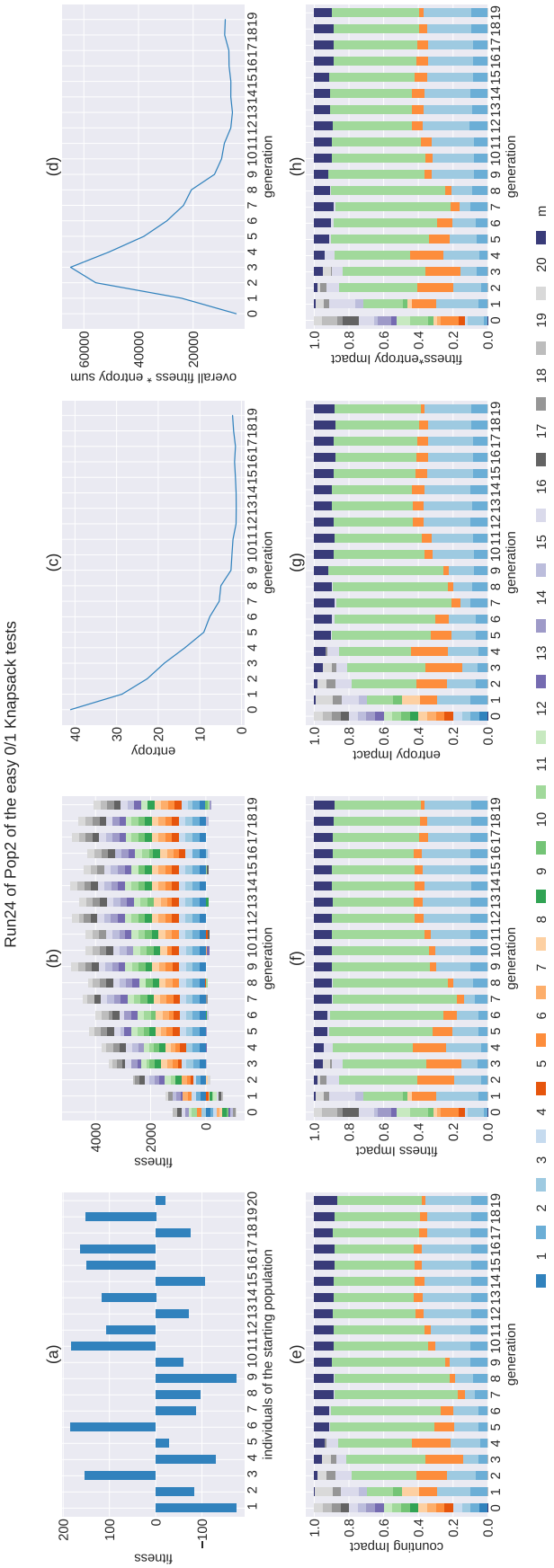

Figure 6.1.: Visualization of run 24 of population 2 from the 0/1 Knapsack problem. On the top, graph (a) shows the fitness of the initial population, graph (b) the fitness over all generations, each single individual being represented by an unique color, graph (c) the accumulated entropy per gene and graph (d) the summed entropy per gene multiplied by the summed fitness per generation. On the bottom, graph (e) shows the CI of the run, the impact values of the single individuals represented by different colors, graph (f) the FI, graph (g) the entropy-based impact (EI) and graph (h) the fitness-entropy-based impact (FEI).

The overall fitness of the later generations sometimes being lower could be due to the crossover or the mutation making the overall weight in the knapsack too heavy in some individuals. The mutation impact not rising higher in further generations is an indicator of the low probability for a mutation operation to improve an individual.

The EI, shown in graph (6.1.g), starts out like the CI with every traceID having the same impact in the initial population, since the entropy per gene is still equal for all genes in the first generation. This similarity also continues in the first generations. From generation 10 and on, however, the values of traceID 6 are a little higher than in the CI and the FI. This fact, mixed with the overall low impact of traceID 6 suggest it not dominating one single gene, but being mixed throughout many genes, increasing their entropy. Also, the impact of traceIDs 2 and m are not as constant as in the CI. However, the differences in the values are comparably low.

Graph (6.1.h), showing the FEI, predictably shows a mix of the developments found in the FI and EI. While the combined impact is similar to the FI in the early generations, it is more similar to the EI later on.
The four impact metrics being similar can be explained with graph (6.1.c). The exponentially dropping entropy per gene, which already reaches a relatively low value in generation 10 suggests a quick domination of one or two individuals in the tests, which indicates the selective pressure is to high. When all individuals in a generation have the same traceID in each gene, the entropy of every gene will be low for all genes, leading to the EI being similar to the CI. All individuals being similar also leads to similar fitness values for each individual, which itself leads to the FI being similar to the CI. If both the fitness and entropy factors are similar, the combined FEI also produces similar results to the other three metrics. This therefore can explain the similar results for the four impact metrics. The disadvantage of a low diversity in a population was discussed by Sudholt and Dirk in [39]. Besides not being able to compare the four impact metrics with each other, this also has negative effects on the performance of the evolutionary algorithm (EA). The low entropy reached in this run therefore could be an indicator changing the breeding and selection process to promote a better diversity throughout the run. This would not only have a positive effect on the performance but also provide different results for the four impact metrics, which would allow for a better comparison and discussion of the four metrics proposed.

Comparing the impact reached in the final generation with the initial fitness, table (6.1) shows a comparison between the ranking of the initial fitness and the ranking for each impact metric for run 24 of population 2 of the easy 0/1 Knapsack problem. TraceID 6 has the highest initial fitness, however

it has the lowest impact in the four metrics among the surviving traceIDs, even behind the before mentioned traceID 2, which only was the 15th best initial individual with a negative initial fitness. The highest impact traceID was traceID 11, which reached impact values around 50% for all of the metrics. TraceID 3, having the fourth best initial fitness, had the second highest impact with around 27% in the four metrics. The third highest impact in the four metrics was the mutation impact, with about 10% to 13%, followed by traceID 2 with around 9.5% and finally traceID 6 with around 2.3% impact in the four metrics. The third highest initial fitness was traceID 17, which was not found in the last generation.

traceID	6	11	3	2	m
initial fitness	1 (184)	2 (182)	4 (153)	15 (-82)	x
CI	5 (0.020)	1 (0.49)	2 (0.265)	4 (0.095)	3 (0.130)
FI	5 (0.022)	1 (0.496)	2 (0.272)	4 (0.094)	3 (0.117)
EI	5 (0.024)	1 (0.498)	2 (0.269)	4 (0.095)	3 (0.114)
FEI	5 (0.027)	1 (0.504)	2 (0.276)	4 (0.094)	3 (0.099)

Table 6.1.: Impact ranking of the remaining traceIDs in run 24 of population 2 from the easy 0/1 Knapsack problem. The traceIDs are sorted by their initial fitness. The corresponding impact values are shown in brackets.

As already discussed, the four impact values are very similar, with the highest difference having traceID 11 with 1% between the four metrics. This also leads to the four impact metrics having the same rank.
Comparing the initial fitness rank with the impact rank yields some unexpected results. TraceID 11 having the highest impact and traceID 3 the second highest in the last generation could be explained with the low initial fitness difference to the best initial fitness traceID 6. However, traceID 6 having the best initial fitness but the lowest recorded impact in the four metrics, and traceID 2 being found in the last generation even though it only has the 15th best initial fitness does not fit the hypothesis 1. Another possible reason could be found comparing the best fitness reached in this run, which is 246, with the best fitness reached in run 3 of population 5 found in figure (6.2.b), which is 295. That shows the testrun did not find the global optimum, with traceID 11 potentially being a deceptive individual, leading the EA into a local optimum. If this result is an outlier or an indicator for other factors then the initial fitness contributing to the final impact ranking needs to be evaluated in the multi-run tests and the evaluation of hypothesis 1.

Run 3 of population 5 from the easy 0/1 Knapsack problem

Figure (6.2) shows the same visualization as in the previous section, but for run 3 of population 5 from the easy 0/1 Knapsack problem. Graph (6.2.a) again shows the fitness of the initial population, graph (6.2.b) the fitness of the population per generation, graph (6.2.c) the entropy per gene and graph (6.2.d) the combined fitness and the entropy per gene over the generations. The graphs (6.2.e), (6.2.f), (6.2.g) and (6.2.h) show the CI, FI, EI and FEI

Like in pop 2, the CI shown in graph (6.2.e) starts with every individual having the same impact in the initial population. Equivalently to the previous run, the amount of traceIDs present in the second generation is reduced drastically to only 9, again about half of the initial 20. Only 3 different traceIDs, including mutation, survive to the last generation. The mutation impact this time first appears in generation 2. Starting in generation 1, traceID 16, which has the highest initial fitness, also has the highest CI, which is only growing in the further generations, already reaching an impact of over 90% in generation 5. The only other traceID from the initial population surviving to the end is traceID 14, reaching only an impact of about 5% from generation 5 onwards. The mutation impact reached a small peak in generation 3, then dropped down again, even disappearing completely in the generations 8, 16 and 18, only reaching a CI of 1.5%. Comparing the CI of this run to the run 24 of pop 2, we can see that the values seem to converge faster in this run, not having significant changes from generation 5 on, while the evaluated run of pop 2 took twice as long to reach that state. This quicker convergence could be a result of this population reaching a better optimum faster than the previous. If this is due to the best initial fitness of population 5 being higher than the best initial fitness of population 2, it would need to be checked in further tests.

Taking a look at graph (6.2.f) for the FI, the initial fitness again is not similar to the differences in the initial fitness of the individuals. Also, the impact in generation 1 shows increased impact values for the traceIDs 7, 14, 16 and 20, two of which are surviving to the last generation. Unlike in the run analyzed in population 2 though, we can see differences throughout the whole run in the mutation impact, which is lower in the FI than in the CI, hinting at mutation worsening the fitness of individuals more than improving it. Looking at the fitness over the generations in graph (6.1.b), we can also see that the fitness peaks in the generations 8, 11, 16 and 18, which includes the three generations where the mutation impact is 0%, with generation 11 also having a low mutation impact.

The entropy impact in graph (6.2.g) also starts with every traceID having the same impact in the beginning. Like in the evaluation of run 24 population 2,

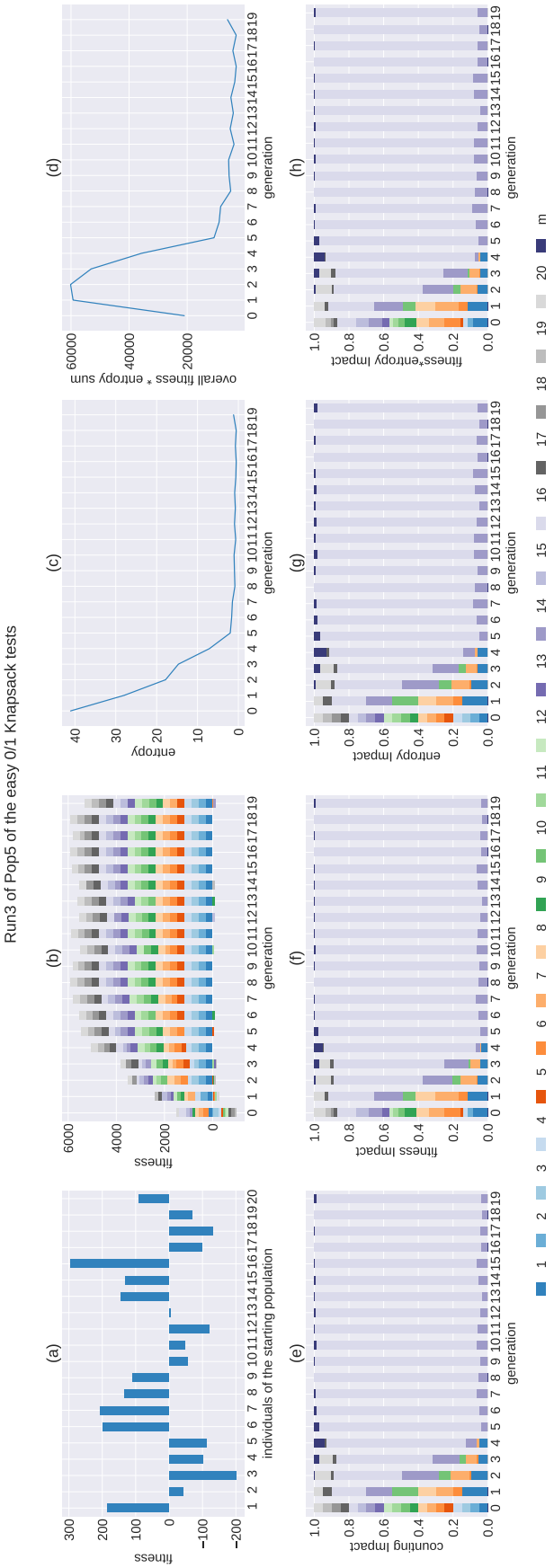

Figure 6.2.: Visualization of run 3 of population 5 from the 0/1 Knapsack problem. On the top, graph (a) shows the fitness of the initial population, graph (b) the fitness over all generations, each single individual being represented by an unique color, graph (c) the accumulated entropy per gene and graph (d) the summed entropy per gene multiplied by the summed fitness per generation. On the bottom, graph (e) shows the CI of the run, the impact values of the single individuals represented by different colors, graph (f) the FI, graph (g) the EI and graph (h) the FEI.

it has very similar values to the CI, especially in first generations. The values in the later generations are also similar, the impact of the traceIDs 14 and m being a little higher, but not in a significant way.

The combined FEI shown in graph (6.2.h) starts similarly to the FI, with the impact in the first generation, and the general development until generation 5, being relatively similar to the FI. Also like the FI, the FEI shows a smaller impact for the mutation. The impact for traceID 14 on the other hand is more similar to the EI.

In general, the difference in the four impact metrics is very small, with no significant differences in the later generations. In this run, the drop in the impact graph is even steeper than in the run analyzed in population 2. This could be due to the higher best initial fitness in this run, however, to get reliable results more runs need to be evaluated. The concern about the breeding process not allowing for enough diversity is now increased, since this second run also shows a quick drop in the entropy per gene, with the four impact metrics achieving very similar results.

Also equivalent to pop 2, the impact of generation 0 for the FI and FEI do not start with equal values. While the fitness difference from the best to the second best and third best individual is higher in pop 5, the bigger difference in impact is still boosted by the traceIDs representing just one individual. For this reason, traceID 16 already has an initial FI and FEI of over 90%. However, unlike in run 24 of pop 2, the impact value of traceID 16 remains high until the last generation.

traceID	16	14	m
initial fitness	1 (295.0)	5 (145.0)	x
CI	1 (0.945)	2 (0.040)	3 (0.015)
FI	1 (0.955)	2 (0.062)	3 (0.006)
EI	1 (0.919)	2 (0.062)	3 (0.019)
FEI	1 (0.933)	2 (0.060)	3 (0.007)

Table 6.2.: Impact ranking of the remaining traceIDs in run 3 of population 2 from the easy 0/1 Knapsack problem. The traceIDs are sorted by their initial fitness. The corresponding impact values are shown in brackets.

Taking a closer look at the results in the last generation for the four impact metrics, table (6.2) shows the initial fitness ranking of the traceIDs in the last generation, as well as the impact ranking of the four metrics. TraceID 16 having the highest initial fitness also has by far the highest impact in the four

metrics with values ranging from 93% to 95%. The second highest impact is on traceID 14, only having the fifth best initial fitness, with values from 4% to 6%. Mutation had the least impact at the end, only having an impact of 1.9% in the EI, and having a near zero impact in the FI and FEI. Like in the previous population, the initial fitness ranking was not really representative for the resulting impact values. While traceID 16 had by far the highest initial fitness, traceIDs 7, 6 and 5, having the 2nd, 3rd and 4th best initial fitness, did not survive to the last generation. The fitness difference to traceID 14 this time is not as high as in the previous run, where traceID 2 had a bad, negative initial fitness. However, even though the result might be not as extreme as last time, the general trend for hypothesis 1 still continues. For more reliable results, every run from every problem should be evaluated.

The very high impact values for traceID 16 could be a result of the initial fitness being significantly higher than the ones from the other traceIDs. However, if impact values this high are also found in the other testruns needs to be evaluated in the multi-run evaluation. The generally lower mutation impact could be a result of a much higher best initial fitness, which could be a hint for the result of hypothesis 3. However, the mutation impact being low for the FI and FEI also imply that the mutation is not producing good results. In a real world problem, this could be an indicator to make changes to the mutation operator. However, as no better result than the fitness of 295 reached in this testrun was found in the other results of the easy 0/1 Knapsack problem, the population could have reached the global optimum. As the fitness and entropy graphs showed, the population converged rapidly, meaning that mutation potentially was not able to have a positive impact after generation 4, which could also be a reason for its low impact.

Discussion

A few hypotheses can be made when analyzing the results from the two runs. Both runs show a fast drop in the entropy per gene, leading to the conclusion that the breeding mechanism could be improved by allowing more diversity. Run 24 of population 2 not reaching the global optimum further supports that argument. This also results in the four impact metrics being fairly similar, which can also be seen in the ranking in the last generation. The initial fitness ranking not being similar to the impact ranking in the last generation already hints at results for hypothesis 1, and raises the question of which factors may also have an influence on the impact of an individual other than its initial fitness. Besides these general findings, two very different developments in the impact of the individuals where observed. Whether these differences can be

explained through its initial populations or the fitness reached, and how the results of both populations compare to the other 30 testruns, needs to be evaluated in the next section.

Although the analyzed runs already hint at a result for hypothesis 1 and 3, no conclusive information can be gained by just analyzing one single run. While these single-run evaluation capabilities might be useful to gain more detailed information about the behavior of an EA, the randomness through selection, crossover and mutation varies the result. Therefore, to get reliable results, more than one run for each population in each test is required. For this reason, the next section shows an evaluation of the easy 0/1 Knapsack problem for more than the two runs shown in this example.

6.1.2. Evaluating multiple runs

To show that the traceIDs can also be used to provide information not only about a single testrun, this section first focuses on the evaluation of all testruns of population 2 from the easy 0/1 Knapsack problem. After this, the results of the testruns of population 5 from the easy 0/1 Knapsack problem are evaluated. Finally, the results of both are shortly discussed.

Multi-run evaluation of population 2 from the easy 0/1 Knapsack problem

To evaluate the result of the multiple runs, figure (6.4) shows the results of all runs in the final population both for run 2 and 5 as box plots. The left side shows the results for pop 2, with graph (6.4.a) again shows the initial fitness, while graph (6.4.b) shows the results of the CI, graph (6.4.c) the results of the FI, graph (6.4.d) the results of the EI and graph (6.4.e) the results of the FEI.

The results for the four impact metrics being fairly equal is not only found in the single-run evaluation, but also in the multi-run analysis of the box plots. As only the last generation is visualized, the results are all very similar, with just some small changes being found in the outliers, like for example traceID 5 or 8. Not only are the median and the quartiles similar across the four metrics, the majority of the outliers are also similar according to the four graphs. To understand the reason behind the similar values, figure (6.3) shows the average of the entropy of every gene in the last generation. Taking a look at the graph, the highest average entropy of a gene for this test is found in gene 4, with a value of 0.18, with the overall average being at about 0.125. These values suggest a very low diversity in the last generation for each run. This further

confirms the believe, that the diversity allowed by the breeding operators is too low in this experiments and the population gets dominated by the same individual in the last generation. Whether this effect is specific to the 0/1 Knapsack problem, to the difficulty of the problem or to the initial population employed will be evaluated in the following sections.

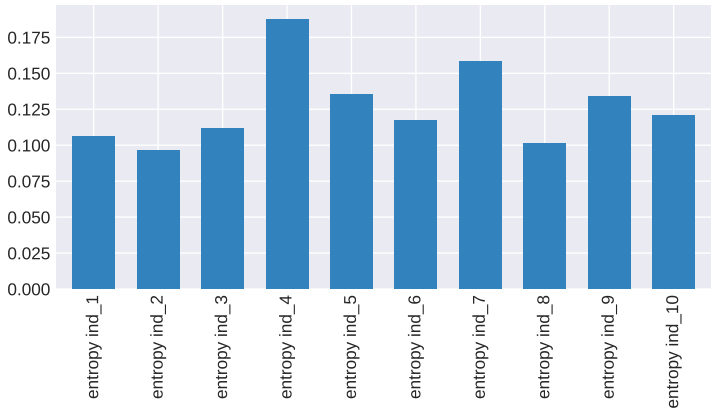

Figure 6.3.: The mean entropy per gene in the last generation of population 2 of the easy 0/1 Knapsack problem.

Since the four impact metrics are so similar, the further evaluation refers to the impact as the impact of all four metrics at the same time. In all four box plots, only the traceIDs 11, 6 and m had median values above 0%. Furthermore, the traceIDs 3, 17 and 19 had positive values for the upper quartile. The traceIDs 9 and 10 did not have any run with an impact over 0%, while the traceIDs 1, 7 and 18 only had one, with the one run of traceID 1 even reaching about 30% impact in all of the metrics. TraceID 11 having the highest median impact matches the result of the single-run evaluation. However, we can see that the single-run result is a high impact outlier, with an impact of about 50% in the four metrics, just above the upper quartile, the median lying around 30% in all four metrics. The ranking of the single-run evaluation for traceID 6 however does not match, only having the 5th highest impact in the single run, while having the second highest median impact, just above the mutation. However, the result still is inside the lower quartile. TraceID 3, which had the second highest impact in the single-run evaluated, overall has a median impact of 0%. The single-run result, like in the traceID 11, is again an outlying value just above the upper quartile. Also an outlying value was the result of traceID 2, which had a surprising impact in the run 24 of population 2, even though it was ranked low compared to the other traceIDs with regards to the initial fitness.

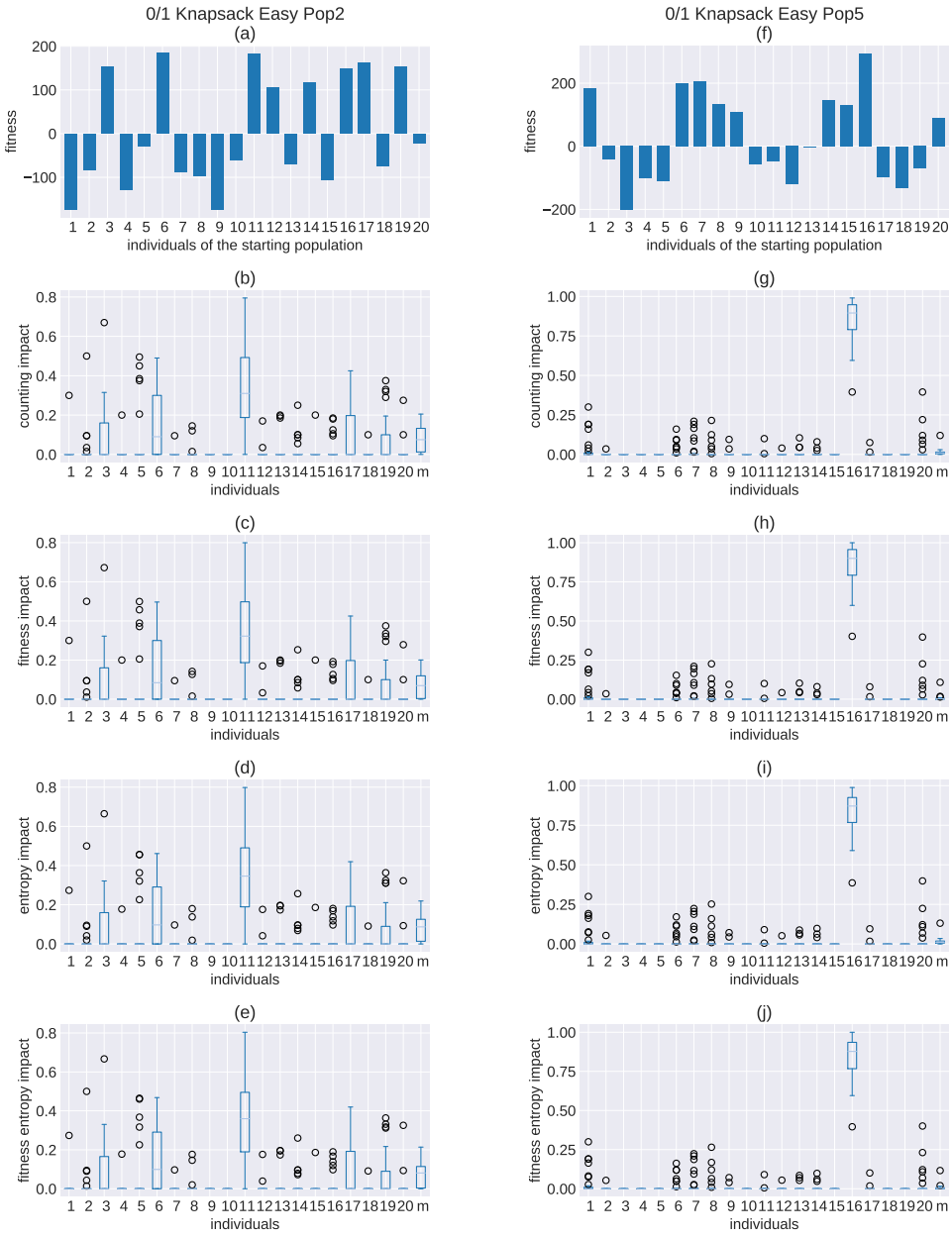

Figure 6.4.: Box Plots of population 2 (left graphs) and population 5 (right graphs) of the easy 0/1 Knapsack problem. The top graphs represents the fitness of the initial populations, with the following graphs showing the results of the four impact metrics of the last generation as a Box Plot.

In general, all results in the last generation are very spread. While traceID 11 has the highest median traceID with 30%, its lower quartile reaches values just below 20%, and the upper quartile values of just below 50% in all four metrics, with outlying values reaching from 0% impact to about 80% impact. The upper quartile of traceID 6 also reaches to about 30% impact, suggesting that there are cases where traceID 6 has a higher impact than traceID 11. While the fitness difference of these two traceIDs are comparably similar, more extreme outliers can also be found. The before mentioned traceID 2 has one outlier in every metric, showing an impact value of around 50%, which suggests it having the highest impact in the whole run. These high outliers can also be found in the traceIDs 3 and 5. Comparing the impact results for all of the 31 runs with the initial fitness ranking, a brought connection between the initial fitness value and the resulting impact can be made. While traceID 6 and 11 are switching the first and second places in the fitness and impact ranking, the difference in the initial fitness ranking is very small. Also traceID 17, having the third highest initial fitness, while having a median impact of 0%, is found to have the third highest upper quartile. This trend is also followed for the traceIDs 3 and 19, having the fourth and fifth highest initial impact and also the fourth and fifth highest quartiles. The lower impact traceIDs 9 and 10, having no recorded run over 0%; the traceIDs 1, 4, 15, 13 and 18, having only one; and the traceID 13 having two runs with an impact over 0%. One exception for this is the before mentioned traceID 2.

While a general connection between the initial fitness and the resulting impact can be assumed over these many runs, the spread of the results as well as the single-run evaluated suggest that this effect only exists for the evaluation of the average over the multiple runs. The resulting impact metrics in one single run therefore could be very different from the initial fitness ranking, with the higher initial fitness only suggesting a higher probability of a high impact. The relations of the median impact not matching with the relations in the initial fitness also suggests another factor than the initial fitness having an influence on the impact results in the last generation. As the box plots do not show the best fitness reached in the final generation, it has to be noted that this results did not reach the assumed global optimum of 295, with a mean fitness result of 273.2 and a standard deviation of 26.3. This further indicates traceID 11 leading the initial population into local optima. For more reliable information, these results need to be evaluated against the results of other testruns in the evaluation of the hypotheses.

Multi-run evaluation of population 5 from the easy 0/1 Knapsack problem

The box plots for population 5 of the easy 0/1 Knapsack problem are also visualized in figure (6.4). Graph (6.4.f) shows the initial fitness, while graph (6.4.g) shows the results of the CI, graph (6.4.h) the results of the FI, graph (6.4.i) the results of the EI and graph (6.4.j) the results of the FEI.

Immediately noticeable is that the four impact metrics seem very similar, like in the evaluation of population 2, even down to the outliers. This time, a difference between the CI/FI and the impact metrics considering entropy EI and FEI, can be found. Like in the evaluation of run 3, the impact of the dominating traceID 16 is a little lower for the EI and FEI, and the impact of the results of the other traceIDs are a little higher. Nevertheless, the changes in value are relatively small and do not change the results in a significant way. The reason for the similar values can again be attributed to a low diversity in the last generation, equalising the differences in the four impact metrics. This is proven by figure (6.5), showing the average entropy per gene in the last generation of the 31 runs of population 5. Compared to the previous run, the highest entropy of a single gene is a little higher, but with about 0.24, it is still very low. This consolidates the theory of the breeding operator not allowing enough diversity in the easy 0/1 Knapsack problem tests, proposed in the evaluation of the single runs of populations 2 and 5. Whether the same effects can also be found for the other difficulty levels and problems will be analyzed in the evaluation of the hypotheses.

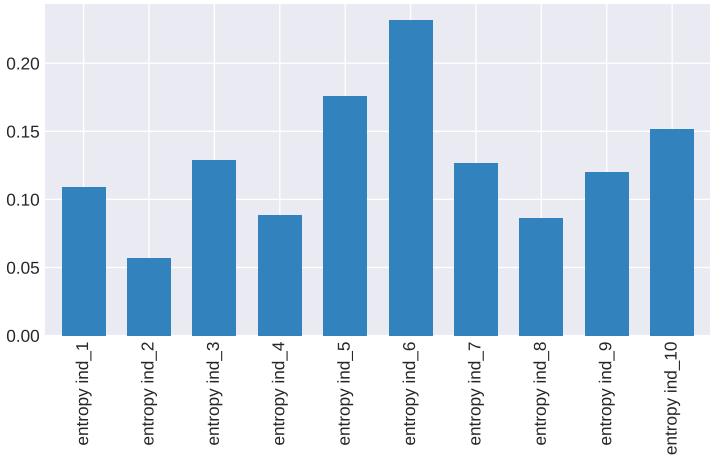

Figure 6.5.: The mean entropy per gene in the last generation of population 5 of the easy 0/1 Knapsack problem.

Further evaluating the four impact metrics shows only traceIDs 16 and m have a median impact above 0%, with the impact of mutation also being near 0%, not dropping presumably because of the reoccurring mutation. Furthermore, only traceID 1 has also an upper whisker, which however gets only up to circa 2% impact in all four metrics. The traceIDs 3, 4, 5, 10, 15, 18 and 19 did not have a single run with an impact value above 0%. The rest of the traceIDs managed only some outlying results above 0% impact.
Comparing the results to the previously evaluated single run, we can see that traceID 16 having this high of an impact in all four metrics was no outlying result. Furthermore, the low mutation impact recorded is also found in the other testruns. TraceID 14 having the second highest impact however was not as common, with only three of the 31 runs showing impact values above 0%.

The impact results in the last generation are less spread compared to the evaluation of population 2. The lower quartile of traceID 16 starts at around 80% for the CI and FI and stretches to 95% impact in the second quartile, with the lowest outlying run still reaching about 40% impact. The outliers of all other traceIDs are mostly in the range up to 20% impact, with some exceptions reaching up to about 40% impact. This low spread indicates traceID 16 having the highest impact in all runs this time, with just one outlier in traceID 20 potentially having a better EI and FEI in one run. The very high mean value of traceID 16 also suggests that the result of just two traceIDs surviving (three with the mutation impact) in the evaluated run 3 was not an outlying result.

Comparing the initial fitness ranking, the highest impact traceID has also the highest initial fitness. Since traceID 1 also has an upper quartile, it can be counted as the second highest impact individual, while only having the 4th highest initial fitness. However the traceIDs 6, 7 and 8 also have a high amount of outlier results above 0% impact. TraceID 14, having the 5th highest initial fitness, only has three runs with an above 0% impact, and traceID 15 having the 6th highest initial fitness does not have one single run with an above 0% impact. TraceID 20, on the other hand, has more outliers than expected, given its average initial fitness. In conclusion, the comparison between the initial fitness and the impact yields similar results like in population 2. The final impact of a traceID on average seems to be connected to the initial fitness, but other factors also seem to have an influence. Also the spread of solutions suggests individual results may differ from the average, even though the spread is lower than for population 2.

While population 2 did not find the global optimum in most testruns, the result of population 5 always reached the assumed global optimum of 295. This indicates the initial individual corresponding to traceID 16 already being very good, explaining its high impact. The impact of mutation being lower in

all four metrics than in the previous evaluation of population 2 might also be explained by this, as a quick conversion to the best result does not leave much room for mutation to improve the population.

Discussion

Comparing the results of the evaluation of the multiple runs to the two single runs analyzed in the previous section, both similarities as well as differences can be found. First of all, the results of the four impact metrics were very similar. The reason for this is a low diversity of individuals in the last generation, as the average entropy per gene showed. For this reason, no real comparison can be made between the four impact metrics. To be able to compare the four metrics, and also to improve the results of the EA, the breeding operator should be changed for future tests of the 0/1 Knapsack problem.
The results of the last generation in both single-run evaluations were found to be no outliers in the multi-run evaluation. Because of a high spread of results in both population 2 and 5, no single run can represent an average of all of the results. The spread of information being lower for population 5, which has a higher initial fitness might be one indicator for this phenomenon. However, a lower difference in the fitness of the top 5 initial individuals in population 2 could also be a reason.
On average, a connection between the initial fitness ranking and the achieved impact could be loosely drawn, however other factors seem to also have an influence on the final impact ranking than the initial fitness. Evaluating the other tests in hypothesis 1 might give some additional information to estimate the reason for a high or low impact. The mutation impact being lower in population 5 than in population 4 fits into the assumption of hypothesis 3. Whether this trend continues for the other populations and the other tests needs to be evaluated with the other testruns.

6.2. Evaluation of the hypotheses

The following section shows and discusses the results of the three hypotheses layed down previously. Similar to the multi-run evaluation, only the results of the last generation will be used to evaluate the hypothesis, as they are based on assumptions regarding the impact of the initial population on the final generation. The hypothesis are discussed separately.

Hypothesis 1

The first hypothesis to be evaluated is the ranking of the 5 highest fitness individuals matching with the impact ranking. This means, the best fitness initial individual (rank 1) should also have the highest impact, the second best fitness initial individual (rank 2) should have the second highest impact, and so on. To test this, the three problems are used as described in section (5.2). As every test configuration is run 31 times, and to avoid outlying solutions distorting the result, the mean impact ranking over those runs is used.

For the evaluation of the first hypothesis, the table (6.3) shows the results for the Max Ones problem, the table (6.4) for the 0/1 Knapsack problem and the table (6.5) for the (Un)bound Knapsack problem. These three tables are structured the same, showing the fitness rank in the first row, with the four impact metrics shown below. For the impact metrics, the amount of matching impact and fitness rankings is shown, as well as the amount of populations with a better or worse impact rank than their fitness rank. Since each problem is run with five populations in three difficulty levels, there are 15 test configurations per problem for which the mean impact ranking is compared. The results shown in the three tables are also shown in the appendix (B.2), were the fitness and impact rankings are presented separate for each population. Following the evaluation of hypothesis 1, first the results of the three problems are shown. Thereafter, the results will be compared and discussed, finally concluding if the assumption of hypothesis 1 is right.

fitness rank	**1**	**2**	**3**	**4**	**5**
CI rank matching	0 / 13 / 2	1 / 12 / 2	2 / 6 / 7	3 / 2 / 10	5 / 1 / 9
FI rank matching	0 / 13 / 2	1 / 12 / 2	2 / 6 / 7	3 / 2 / 10	5 / 1 / 9
EI rank matching	0 / 13 / 2	1 / 12 / 2	2 / 6 / 7	3 / 2 / 10	4 / 2 / 9
FEI rank matching	0 / 13 / 2	1 / 12 / 2	2 / 6 / 7	3 / 2 / 10	4 / 2 / 9

Table 6.3.: The four impact rankings matching with the fitness ranking of the Max Ones problem. Each cell shows the amount of times the impact rank was better / matching / worse than the initial fitness ranking for the 15 different test configurations.

Looking at the matching values for the Max Ones problem in table (6.3), we can see the values for the four impact metrics being nearly identical. Only the fifth best impact ranking shows a slight difference, with an additional impact rank matching for the EI and FEI, which had a better impact rank in the other two metrics. This hints at the trend also found in the two populations of the 0/1 Knapsack problem evaluated in the proof of concept evaluation, which showed the diversity in the last generation to be very small, resulting in similar

impact values. The reason for the four impact metrics being similar will be discussed in more detail after the results of the three hypotheses are shown. Conparing the results of all five fitness ranks with each other, we can see a clear decrease in matching ranks from fitness rank 1 to 5. While the best fitness traceID was found to have the highest impact in all four metrics in 13 of the 15 test configuration, and the second best fitness traceID to have the second highest impact in 12 of the 15 test configurations, the values drop significantly for the further ranks. While the third best initial fitness rank still had a matching impact ranking six times, the result for the fourth and fifth best fitness rank only show one and two cases of a matching fitness and impact ranking, with matching impact rankings almost being an exception. Instead, both the amount of better and worse impact rankings rise constatnly. While a lower matching percentage for the lower fitness ranked traceIDs seems probable, since they can not only have a lower but also a higher value, the extreme drop in the matching percentage raises the question for other factors having an influence on the final result. These reasons could include the fitness of the individuals being similar or the same, or certain individuals having a higher chance of producing good offspring in crossover. These possible factors will be further discussed after analyzing the other tests.

fitness rank	**1**	**2**	**3**	**4**	**5**
CI rank matching	0 / 12 / 3	1 / 10 / 4	5 / 4 / 6	5 / 5 / 5	5 / 3 / 7
FI rank matching	0 / 12 / 3	1 / 10 / 4	5 / 3 / 7	5 / 4 / 6	6 / 3 / 6
EI rank matching	0 / 12 / 3	1 / 11 / 3	4 / 5 / 6	5 / 5 / 5	6 / 3 / 6
FEI rank matching	0 / 12 / 3	1 / 11 / 3	4 / 5 / 6	5 / 5 / 5	6 / 3 / 6

Table 6.4.: The four impact rankings matching with the fitness ranking of the 0/1 Knapsack problem. Each cell shows the amount of times the impact rank was better / matching / worse than the initial fitness ranking for the 15 different test configurations.

The 0/1 Knapsack problem tests in table (6.4) show similar results to the Max Ones problem before. Again, the results for the four impact metrics are very similar, the differences this time are higher. Only the best fitness individual (rank 1) this time shows the same values in the four metrics. The differences in the other metrics are still very low though, with all but the third fitness rank only having one matching case more or less for each metric. The third fitness rank is showing a difference in matching cases of two for the four metrics in this case. As the proof of concept evaluation already showed for some of the 0/1 Knapsack tests, the reason for the impact metrics showing this similar results is that the diversity in the last generation is very low. Smaller changes between the impact metrics here could therefore be a result of the differences

overall being very small, resulting in small changes being enough to change the impact ranking. However, this will be discussed in more detail later in this section.
Comparing the initial fitness ranks with the impact rankings reveals similar results to the results from the Max Ones problem. The best initial fitness traceID this time had in 12 of the 15 test configurations also the highest mean impact in all four metrics. The second best fitness traceID managed to have a matching CI and FI in 10 and a matching EI and FEI in 11 cases. The fourth and fifth highest fitness ranks show matching impact values in around 5 cases, with the FI being an exception, showing only 3 and 4 matching values, with both better and worse rankings being found. The fifth fitness rank showing three matching cases in all four metrics.
While the results are similar to the Max Ones problems, they show a smoother drop in matching values in the top 5 initial fitness traceIDs. Also the 0/1 Knapsack problem rankings show more rankings being better, and especially in the fourth and fifth highest fitness rank fewer impact rankings being worse than the initial fitness rank. The general trend of the best two fitness ranks having a matching impact rank with the remaining ranks showing far fewer matching results is still found though.

fitness rank	**1**	**2**	**3**	**4**	**5**
CI rank matching	0 / 13 / 2	1 / 11 / 3	1 / 5 / 9	4 / 1 / 10	5 / 6 / 4
FI rank matching	0 / 13 / 2	1 / 11 / 3	1 / 5 / 9	4 / 1 / 10	5 / 6 / 4
EI rank matching	0 / 13 / 2	1 / 10 / 4	2 / 5 / 8	3 / 1 / 11	5 / 6 / 4
FEI rank matching	0 / 13 / 2	1 / 10 / 4	2 / 5 / 8	4 / 2 / 9	3 / 5 / 7

Table 6.5.: The four impact rankings matching with the fitness ranking of the (Un)bound Knapsack problem. Each cell shows the amount of times the impact rank was better / matching / worse than the initial fitness ranking for the 15 different test configurations.

Evaluating the (Un)bound Knapsack problem tests shown in table (6.5), again the values between all four metrics are very similar. Like in the rankings of the 0/1 Knapsack problem, only the highest fitness rankings shows equal impact rankings for all four metrics. The differences again are minor, with all but one having only a one case difference.
Comparing the differences between the fitness ranking and the impact ranking, again the highest fitness traceID is matching in the most cases, with 13 out of the 15 tests for the four impact metrics. The second highest fitness traceID is matching in 11 cases for the CI and EI and 10 cases for the other two metrics. The third fitness rank had a matching impact metric in 5 cases. The fourth fitness rank only once matching with all impact values, which is lower than for

the other two metrics. The fifth best fitness traceID having matching impact values in 6 cases, or 5 for the FEI, indicates the fourth rank having an outlying value.
In general, the results are in line with the other two tests though. Since the fourth highest fitness traceID is different in all initial populations, the low values can not be attributed to a specific initial population. However, the results of the Max Ones problem also showing low values for the fourth highest fitness traceID indicates the values having a higher variance in the lower initial fitness ranks than in the higher, where the results seem more similar.

Before comparing the results of the three tests with each other, the reason for the low differences of the impact metrics is evaluated here. The proof of concept evaluation of population 2 and 5 of the easy 0/1 Knapsack tests already showed a low diversity in the last generation by evaluating the entropy of the traceIDs. To check if this assumption is right, the accumulated mean entropy of the last generation from all tests used is shown in figure (6.6). The results in the graph show an interesting pattern, with the higher fitness initial populations 5 having a lower entropy in every configuration than the lower fitness initial populations 1. The highest entropy is in population 1 of the hard (Un)bound Knapsack problem. However, comparing the test configurations needs to take the amount of genes of each test configuration into account. While evaluating the accumulated entropys of the different problems and configurations further certainly could show some interesting information, the most important finding for hypothesis 1 is the mean entropy of every test run being very low for all genes. This directly proves the assumption of the breeding not allowing a diverse enough population. This means, a comparison between the four impact metrics will not yield many differences in the result. As the data of hypothesis 1 already showed, most rankings are very similar, with only minor differences found. While there can be differences found in the four impact metrics, especially in the early generations of the proof of concept evaluation of the single run showed, the last generation of both the single and multi-run evaluation already showed the data being very similar. For this hypothesis, the differences found in the four metrics come down to very similar mean impact values for some ranks, meaning small changes in the impact metrics due to fitness and entropy changes from occurring mutation where enough to switch some ranks. The only conclusion comparing the four metrics therefore can be that tests with a very similar population in the last generation also show little differences in the four impact metrics.

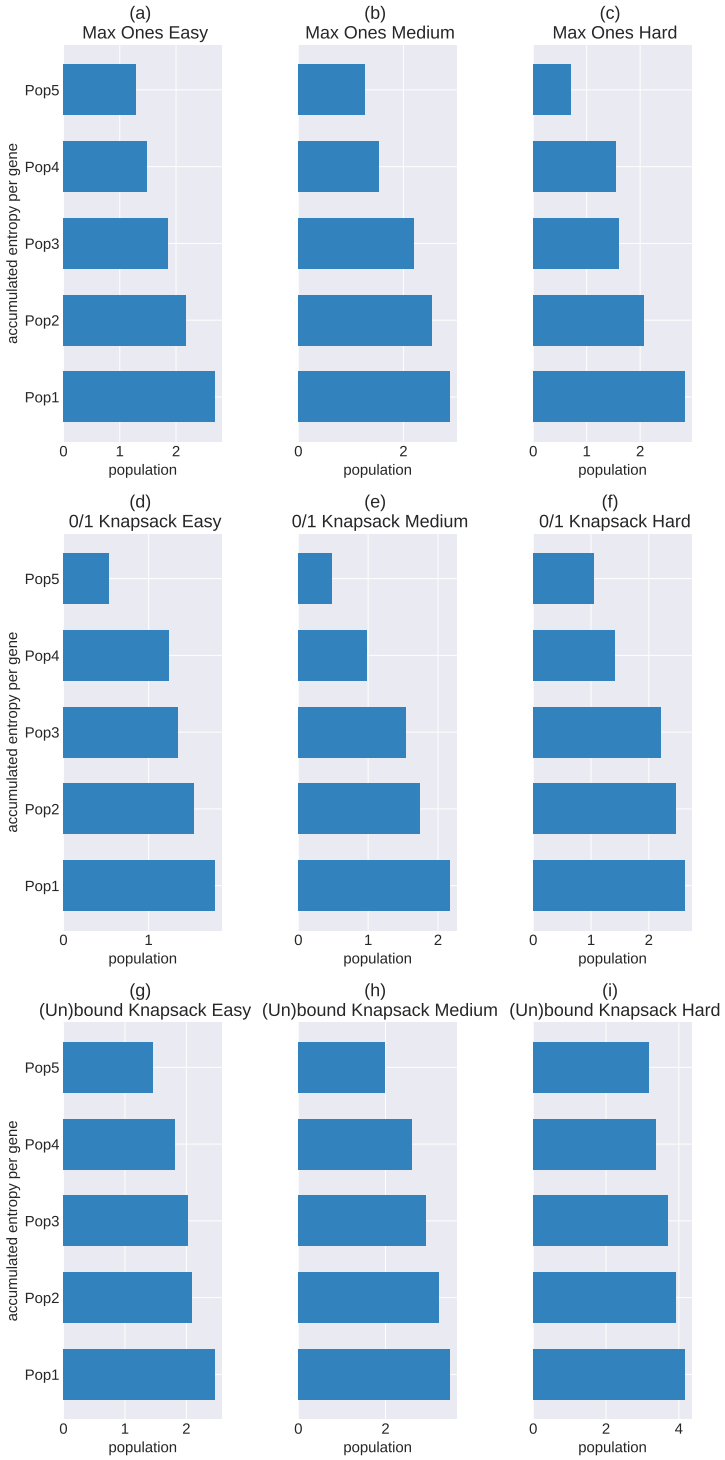

Figure 6.6.: The summed mean entropy in the last generation for the three problems used in hypothesis 1. Each column is showing a different problem type and each row a different difficulty level of the problems.

Besides the four impact metrics showing similar data, the results are also consistent over the three different problems. This indicates that the problem type and the specific data structure does not have a big influence. The biggest influence on the result needs to come from the common breeding operators used between the three problems. Since the impact from mutation is not taken into account in these rankings, the difference in mutation probability between the bit vector represented Max Ones and 0/1 Knapsack problems compared to the integer vector represented (Un)bound Knapsack problem also does not have a big influence. To understand why the probability of a matching fitness and impact rank is dropping with the fitness rank in all tests, it needs to be discussed what influences there might be for a gene to survive to the last generation or not. The most obvious influence is the fitness of the initial individual which the traceID is connected with. One reason for the best and second best individual having such high matching rate could be the elitism of the breeding process. Since the two best individuals always pass into the next generation by default, their traceIDs always survive into the second generation. The rest of the initial populations are picked at random for the tournament selection. The tournament size of 2 means for each crossover candidate, two random individuals from the current generation are picked, with the one with better fitness being selected for crossover. This means, even though a better fitness increases the chance of winning the tournament, the selection still has a random factor on it, which also has an influence on the impact of the initial individuals, especially if two traceIDs have a similar initial fitness. Since for every two parents, two offsprings are created, the two point crossover used does not specifically alter the impact of certain traceIDs in that way.
Besides the fitness of the initial individuals, a second possible influence on the impact of the last generation could be the combinability of individuals. The term combinability in this case describes the probability of an individual producing a good or bad offspring in crossover. A good combinability therefore means a higher chance of good offspring, while a bad combinability means a higher chance of producing lower fitness offspring. While the probability of such a combinability having a big influence on the result seems low for the used crossover operators, not only needing to match the right individuals but also the right gene parts, it is certainly possible for it to have an effect for traceIDs having a similar initial fitness. This could potentially explain the fact found in the proof of concept evaluation for pop 2, having only the second highest impact on the best initial fitness traceID.
Besides the initial fitness and the potential combinability, the third factor to be considered is the randomness through the breeding operator, already discussed. As the proof of concept evaluation showed, the results for all tests are highly spread. Since every test configuration is run 31 times, random influence could

therefore also be a reason for some mean impact rankings not matching with their initial fitness rank, especially in scenarios with a low impact difference.

Although there are many different reasons for the results shown in this section, a definitive conclusion can only be reached for hypothesis 1. While the assumption of the fitness ranking matching with the impact ranking mostly holds true for the best and second best initial fitness traceID, the following ranks having a higher chance of being better or worse than their initial fitness rank indicates another factor than the initial fitness having an influence on the final impact ranking. Therefore it can be concluded that the assumption of hypothesis 1, the ranking of the top 5 initial fitness individuals having a matching impact ranking over the 31 testruns, is wrong. While the first two initial fitness ranks show a high probability of matching fitness and impact ranks, there is a high spread found in the following initial fitness ranks, with not even a single one of the 45 testruns showing perfectly matching fitness and impact ranks.

Hypothesis 2

Hypothesis 2 assumes the mean impact difference to be low for initial populations with the same fitness for all individuals. Therefore, initial populations fitting this criteria were created for the Max Ones problem (referred to as same fitness Max Ones problem), as the three problems used in hypothesis 1 did not satisfy this requirement. Like in hypothesis 1, the mean impact over 31 runs for every population is used. This means, the difference between the highest and lowest mean impact of the traceIDs from a given population is evaluated. As the single-run proof of concept evaluation showed, not all traceIDs manage to survive to the last generation. To avoid the lowest impact value to always be 0%, distorting the results, the difference in mean impact was evaluated. The mutation impact was not considered for the evaluation, since it does not have an initial fitness and would also distort the impact comparison of the individuals.

To evaluate hypothesis 2, the impact difference for each metric is visualized in figure (6.8). The figure shows three graphs, graph (6.8.a) for the easy, graph (6.8.b) for the medium and graph (6.8.c) for the hard tests. Every difficulty shoes the differences in the four impact metrics per population.

The first thing to notice when evaluating the graphs is that the differences found in the four impact metrics again are very similar. Similarly to the prevouis test setups, the reason can be found when evaluating the accumulated mean entropy of the same fitness Max Ones problem. Figure (6.7) shows the

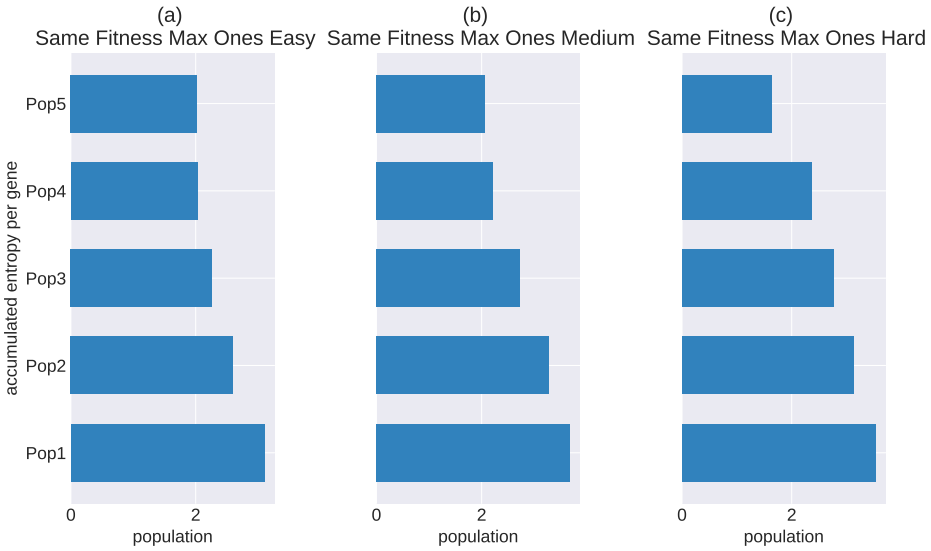

Figure 6.7.: The summed mean entropy in the last generation for the same fitness Max Ones problem used for hypothesis 2. Graph (a) shows the results of the easy difficulty, graph (b) the results of the medium difficulty and graph (c) the result of the hard difficulty.

accumulated mean entropy of the last generations of the 31 runs for each population in each difficulty level. The same trend as for the previous tests can be found, with the entropy of the initial populations 1 being lower than the entropy of the populations 5. While the values are about 0.5 higher than the entropy values of the Max Ones problem from hypothesis 1, they still are very low for the same fitness Max Ones problem, with the highest value being found in the population 5 from the hard difficulty. Again, when comparing the different difficulties, the gene count needs to be kept in mind. Although the initial population has the same fitness in every individual, the entropy of each test configuration still is very low. This means the last generation is still composed mostly of the same individual. While gene wise, every test reached the global optimum with a fitness of 1 in the end, with some individuals being an exeption because of an occurring bit flip mutation, the entropy per gene of the data being low is not suprising. However, the entropy of the traceIDs being low implies the dominance of one single individual throughout the run. Especially interesting is the initial populations having a lower entropy, even though the number of ones per gene is much higher than in the other initial populations.

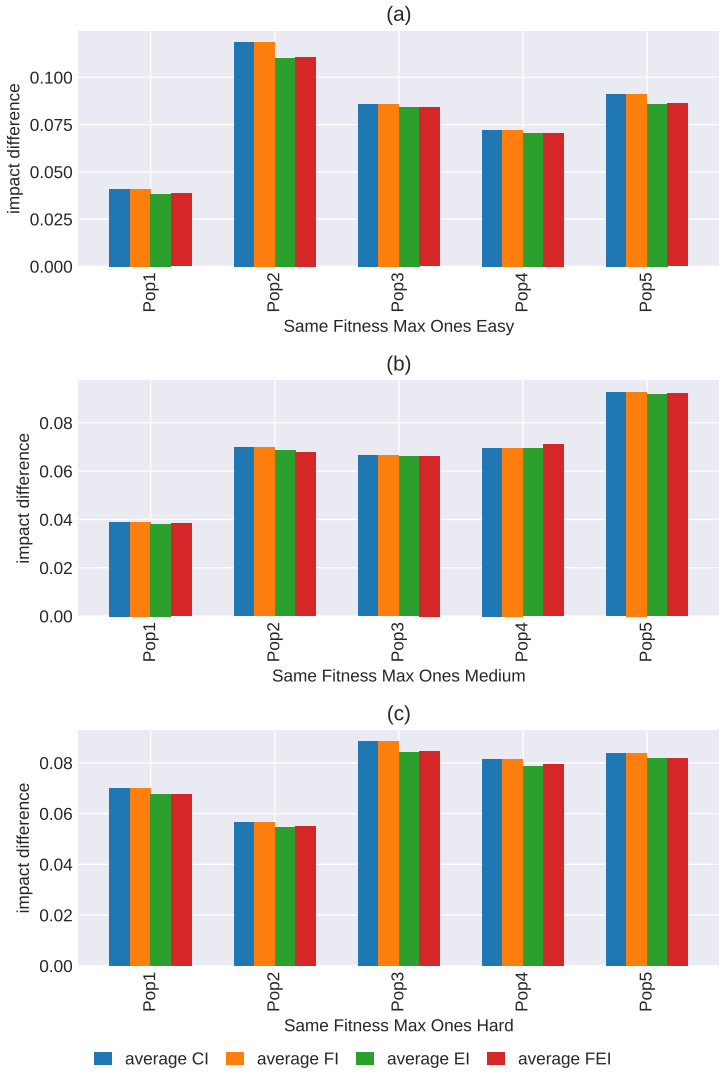

Figure 6.8.: The difference between the highest and the lowest mean impact for every population from the same fitness Max Ones problem. Graph (a) shows the results of the easy tests, graph (b) of the medium tests and graph (c) of the hard tests.

Since these similar entropy values can also be found for the same fitness initial populations, the reason for the dominance of a traceID can not be attributed to the initial fitness. This indicates that the first individual reaching an optimum quickly dominates the population. While only having one optimum might not be an issue for the Max Ones problem, only having one optimum, for other problems like the Knapsack problems also used in this thesis, the EA

might easily get stuck in a local optimum. Besides the implications on the performance loss of the EA, this also means a comparison between the four impact metrics does not yield interesting results, since the four impact metrics are all similar. As already mentioned, all tests of the same fitness Max Ones problem result in finding the optimum fitness of 1. This means the small differences in the values are due to the bit flip mutation worsening an optimal individual. The following evaluation of the mean impact differences therefore does not compare the four impact metrics with each other.

Taking a closer look at the results of the easy difficulty in figure (6.8), the highest impact difference can be found in population 2 with about 11% for the four metrics. The second highest value is found in population 5 with an impact difference of about 8% in the four metrics, closely followed by population 3. The fourth highest impact difference can be seen in population 4 with values just under 7.5%. Population 1 has the lowest impact difference of just under 4%.
The medium tests have a highest impact difference of about 9%, slightly lower than the easy tests. Population 2, 3 and 4 are all closely below 7%, with population 1 again having the lowest value of just under 4%.
The impact differences for the hard tests seem to be more similar, with population 3 having the highest impact difference of just under 8%, closely followed by population 4 and 5 at about 7.6% in the four metrics. Population 1 had the fourth highest difference just under 7.5%. Population 2 had the lowest impact difference with about 5%, which is a little higher compared to the other two difficulties.

While the impact differences overall are fairly low, the distance from the highest impact of 11% being more than twice as high than the lowest recorded mean impact difference of only 4% raises the question on the reason for the difference in the final values. There are four possible factors that could be the reason for a higher impact difference. Firstly, impact differences of individuals inside of one population could lead to a higher or lower mean impact difference. Secondly, the overall best initial fitness or medium initial fitness of a population compared to another population could also have an influence generating a higher fitness difference. Thirdly, the combinability of individuals could be a factor for a higher or lower mean impact difference. Finally, the randomness in the breeding process is also a factor for mean differences in the end. These four factors will now be discussed to find the reason for the differences found in figure (6.8).

The first possible reason, the impact differences inside of a population, can be ruled out for this test, since the populations all have the same fitness in all individuals by the nature of the test design.

To evaluate the second possible reason of fitness differences between the five initial populations used, a relation between the initial fitness and the resulting impact difference would need to be found. Since the best initial fitness and the mean fitness of a generation is the same for all populations, no separation needs to be done for this special case. Looking at the data of all four difficulties of the three tests in figure (6.8), no real trend can be found between the initial fitness differences of the populations and the resulting impact difference visualized. The lowest impact differences are primarily found in populations 1, which have the lowest initial fitness, for the easy and medium tests, and in population 2 for the hard tests. However, population 2 in the easy test having the highest impact difference also shows that high differences in the populations with a low initial fitness are also possible. In general, the highest fitness differences appear all over the spectrum of initial fitnesses with population 2 in the easy tests, population 5 in the medium tests and population 3 in the hard tests. This suggests the initial fitness not having an influence on a higher or lower fitness difference for this same initial fitness test.
The third possible factor for a higher or lower impact difference might be the combinability of the individuals of an initial population. Combinability refers to two individuals having a better chance to produce a good offspring in crossover. If there is an individual having a particularly good combinability with the rest of the population, the influence of such an individual should on average be higher than for other individuals, in theory leading to a bigger impact difference. If on the other hand, the combinability of an individual is lower with the rest of the population, it should have, on the other hand, a lower impact on average, also leading to a higher average fitness difference. Because the objective of the Max Ones problem is to maximise the amount of ones in the genes, the combinability is directly linked to the amount of ones present in an individual. Because every individual has the same fitness and therefore also the same amount of ones in its genes, the only other factor remaining should be the distribution of the ones inside of the individuals. However, for this theory, the combinability should be lower for the highest and lowest fitness initial populations, with a spiking impact difference for the medium fitness populations 3. For the high fitness starting populations, especially population 5 of each difficulty, where every individual has a fitness of 0.9 out of 1.0, nearly every recombination will result in the best fitness individual. That means the surviving individuals are chosen basically at random through the tournament selection, meaning the combinability of all individuals would be fairly equal, resulting in a similar impact for each individual over many runs. On the other hand, for the lower fitness initial populations, the recombination of individuals also produces fairly low fitness individuals, with mutations always having the highest impact. This can also be observed in the box plot visualizations for

the same fitness Max Ones problem in appendix (B.1.4). The medium impact individuals on the other hand should be more influenced by the combinability. While it can be observed that the initial populations 1 have a lower mean impact difference in the easy and medium tests, the hard tests show a higher impact difference in the first population. Also, the assumption that the initial populations 5 have a lower impact difference can not be found. While more research into the possibility of the combinability of individuals having an influence on the result needs further research, for now it can be assumed that combinability is not the reason for the differences in values found.
The first three possible influences to the differences in the data leaves variation through the randomness in the breeding procedure as the most probable factor.

Despite the differences found in the data, it can be concluded that in initial populations with the same fitness, the impact for each individual is similar at the end. Even though there are cases where some individuals had a higher impact than others, the difference between the best and worst impact was at most about 11%. Because of the high spread found in the proof of concept evaluation, which is also found in the box plots of the same fitness max ones tests in the appendix section B.1, the results of this evaluation can not necessarily reflect the results of one single run. Furthermore, not all traceIDs surviving to the last generation results in the impact difference being equal to the highest. The low impact difference therefore reflects a low medium impact, meaning the probability of one traceID making it to the last generation is fairly equal. While hypothesis 2 may not be true evaluating one individual testrun, it holds true when evaluating a multitude of runs.

Hypothesis 3

The third hypothesis assumes the impact of mutation lower for initial populations with a higher best fitness. To evaluate this, the mean mutation impact for every initial population from every problem is shown in four separate figures. The three problems also used in hypothesis 1 are presented in figure (6.9), showing the results for the normal Max Ones problem. Figure (6.10) shows the results of the 0/1 Knapsack problem and figure (6.11) the results of the (Un)bound Knapsack problem. The same fitness Max Ones problem from hypothesis 2 is also used in hypothesis 3 as shown in figure (6.12). The visualizations for all four problems are structured the same. The left row shows for each difficulty a bar chart, in which for every population the best initial fitness as well as the medium fitness per generation is visualized. The graphs (a) show the tests of the easy difficulty, the graphs (c) for the medium difficulty and the

graphs (e) for the hard difficulty. The right row also shows a bar chart for each difficulty. Each graph shows for all four metrics the mean mutation impact per population. In addition, a trend line for every impact metric is plotted, showing the trend from populations 1, with the lowest best initial fitness, to populations 5, with the highest best initial fitness. The graphs (b) shows the easy difficulty, graphs (d) the medium and graphs (f) the hard.

Since the same problems are used like in the previous two hypothesis, the following evaluation is also not expected to show significant differences between the four impact metrics. While this time some small differences in the mutation impact can be found, the values overall are very similar again.

Evaluating the four figures, we can see that the best initial fitness rises linearly from the populations 1 to the populations 5 for all problem types in all difficulties. This can be attributed to the selection mechanism mentioned in the experimental setup, picking the initial populations with the most equal distance between each other. Although the hypothesis 3 is based on the assumption of the best initial fitness having an influence at the resulting mutation impact of the four metrics, the average fitness of the initial populations also is visualized for comparison purposes.

Looking at figure (6.9) for the Max Ones problem, we can see that all three difficulties show a clear downward trend in the mutation impacts from population 1 to population 5. In graph (6.9.b), showing the easy tests, we can see that the four impact metrics are relatively similar. Pop 1, pop 2 and pop 3 show a little higher values for the entropy considering EI and FEI, while pop 4 has slightly higher CI and FI. However, the differences in these metrics are not very significant, with all four metrics showing the same development over the five populations. Pop 1 has the highest mutation impact of over 30% in all four metrics. The four impact values drop about 8% to 10% impact each turn to about 2% mutation impact for every metric in pop 5. The change from pop 2 to pop 3 is the only exception, with a drop of just 1% EI and FEI, and about 3% for the CI and FI. The reason for this lower change between pop 2 and pop 3 can not be explained with the initial fitness of population 2 or 3. While the best initial fitness has, due to the generation process of the populations, a fairly equal distance between each population, the mean initial fitness also rises steadily from population 1 to 5, and does not show a significantly lower change between population 2 and 3.
The medium tests in graph (6.9.d) start with a mutation impact of about 25% in pop 1. This time, the changes between population 1 to 2, and the changes between population 4 to 5 are similar, with a drop of about 3%. Also the changes between the populations 2 to 3 and 3 to 4 are similar with a drop of about 10% impact in all four metrics. This time, a potential reason for the

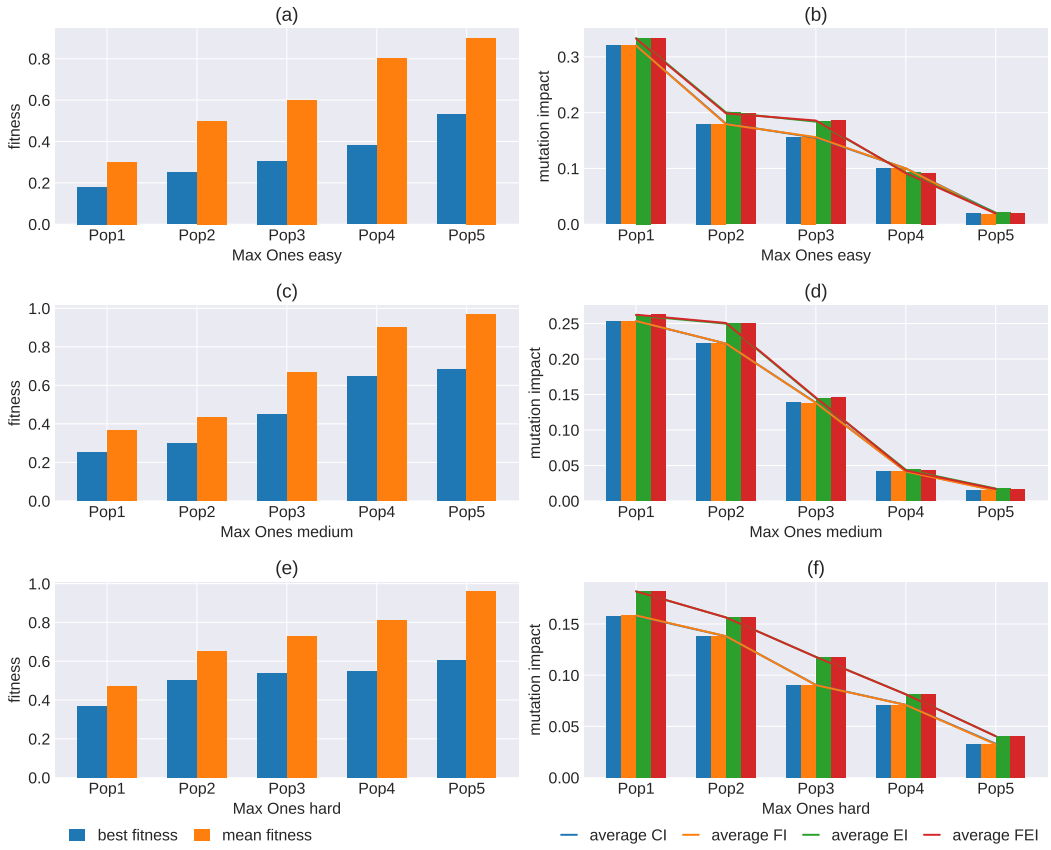

Figure 6.9.: Visualization of the Max Ones problem for hypothesis 3. The left graphs shows the best and average initial fitness of the five populations as a bar chart, graph (a) for the easy, graph (c) for the medium and graph (e) for the hard tests. The right side shows the mutation impact of the four impact metrics per population, also as a bar chart, with graph (b) showing the easy, graph (d) the medium and graph (f) the hard test results.

lower drop rates from pop 1 to 2 and pop 4 to 5 can be found in the corresponding fitness of the initial population in graph (6.9.c). Both the best initial fitness, as well as the medium fitness of the initial populations change slightly higher from pop 2 to 3 and pop 3 to 4. Comparing the results of the four impact metrics, again the entropy considering EI and FEI are a little higher than the CI and FI for pop1, pop2, pop3 and pop 4. Pop 5 again shows similar values for all four metrics.
The results of the hard difficulty of the Max Ones problem in graph (6.9.f) show a more equal change in mutation impact over the five populations. Compared to the other three difficulties, the impact metrics considering entropy also have higher values than the impact metrics not considering entropy. This time though, a clear difference can be found in all five populations. Starting with a mutation impact of about 19% for the EI and the FEI, the values of these two metrics drop about 4% for every population. The CI and FI start lower with about 15% mutation impact, also falling linearly. Comparing this result to the initial fitness of the populations of the hard tests in graph (6.9.e), we can see the drop in mutation impact, correlating more with the best initial fitness increasing linearly from pop 1 to pop 5. On the other hand, the average impact of the populations sees a jump from pop 1 to pop 2, changing much slower, which is not found in the mutation impact of the five populations.
All in all, the assumption of the mutation impact being lower for an individual with a better initial fitness holds true for all three difficulties of this Max Ones tests. However, even though the changes in best initial fitness are linear, both the easy and the medium tests had one outlying smaller change between two initial populations, which also could not be explained with the average fitness of the initial populations.

Evaluating the results of the 0/1 Knapsack tests, shown in figure (6.10), we can already find results not matching to the assumption of hypothesis 3. Starting with the easy difficulty shown in graph (6.10.b), while we still see a downward trend between each initial population, the change from pop 2 to pop 3 is very high compared to the changes found in the previous Max Ones tests, with the mutation impact being similar from pop 3 to pop 4, slightly dropping again in pop 5. The high change in average fitness between pop 2 and pop 3 could be an explanation for the high change in mutation impact between the two populations. However, the average fitness of another population does suggest no link between the average fitness and the impact through mutation. The average fitness of population 5 is actually lower than for pop 4 and pop 3, while having a smaller mutation impact in all four metrics than those populations. This means, that neither the best initial fitness, increasing linearly from pop 1 to pop 5, nor the average initial fitness can be the only factor for the impact of mutation. Comparing the four impact metrics with each other, they are

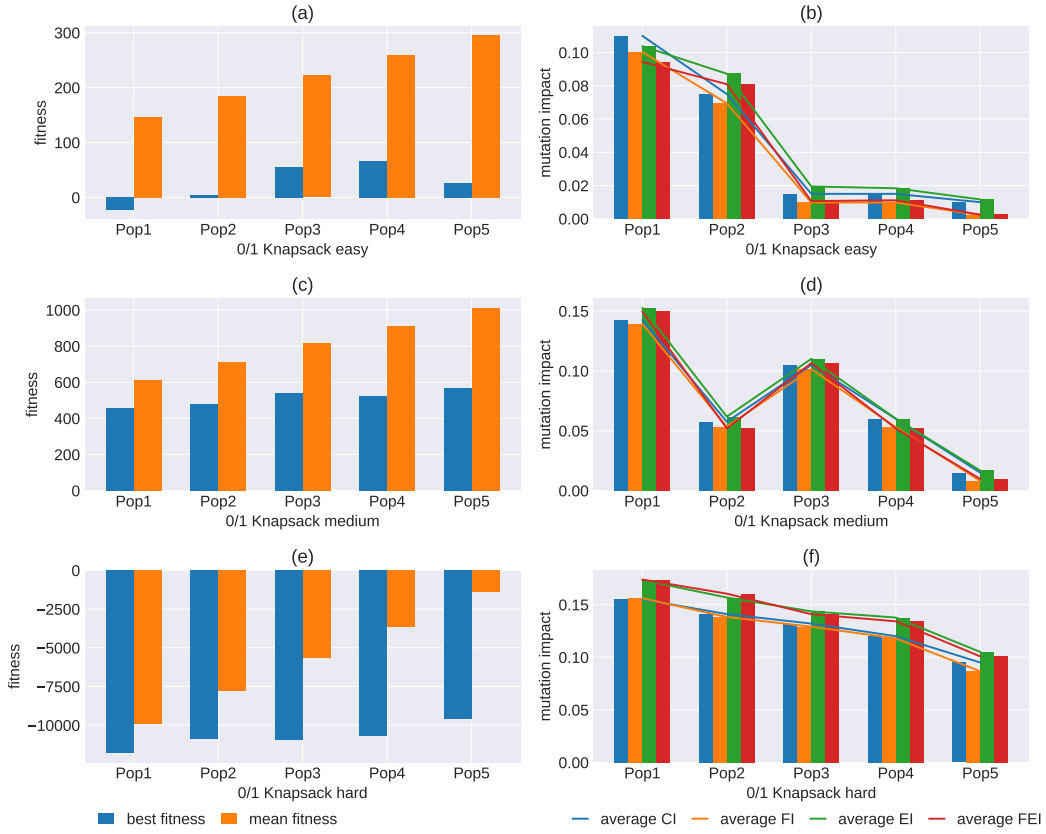

Figure 6.10.: Visualization of the 0/1 Knapsack problem for hypothesis 3. The left graphs shows the best and average initial fitness of the five populations as a bar chart, graph (a) for the easy, graph (c) for the medium and graph (e) for the hard tests. The right side shows the mutation impact of the four impact metrics per population, also as a bar chart, with graph (b) showing the easy, graph (d) the medium and graph (f) the hard test results.

again very similar. However, while in the Max Ones tests the CI always had similar values to the FI, and the EI always had similar values to the FEI, this time the four impact metrics show different results. This time, the FI seems to have the lowest value in all five populations, with pop 1 being an exception with the FEI being the lowest. The EI this time is always the highest impact metric, again with pop 1 being one exception with the CI being the highest.
The results of the medium tests in graph (6.10.d) surprisingly do not show a continuous downward trend in the mutation impact from one population to another. While the overall trend from pop 1 to pop 5 still shows a lower mutation impact for a higher best initial fitness, the result of pop 2 is an outlier with an unexpectedly low mutation impact of only about 5% for all four metrics compared to the about 15% mutation impact in pop 1 and about 10% mutation impact in pop 3. Comparing this result to the initial fitness in graph (6.10.c), no connection between the average fitness or the best initial fitness can be found. Comparing the four impact metrics, only minor changes are found, with the FI being the lowest in all populations and the EI being the highest.
Moving on to the hard tests in graph (6.10.f), no outlying results can be found this time, with an even decrease between all of the five populations. Similar to the Max Ones tests, the metrics considering entropy show higher values than the CI or FI in all five populations. Notable this time though is the overall small decrease in mutation impact from about 15% for the CI and FI in pop 1, to about 8% for both metrics in pop 5, with the EI and FEI having around a 2% higher mutation impact. This could be due to the average impact of the populations also changing relatively little from pop 1 to pop 5. However, other examples before, like the hard Max Ones tests, showed no such correlation.
The medium tests might prove the assumption of hypothesis 3 wrong. While randomness in the breeding process also can have an effect on the final result, it seems not probable in this case because of the clear difference between the values of pop 2 to pop 1 and pop 3. This leads to the conclusion of another factor having an influence on the mutation impact. However, even though there are outliers found in the 0/1 Knapsack problem, overall the best initial fitness seems to have an influence on the impact of the mutation, since all tests show a drop in mutation impact from pop 1 to pop 5.

Figure (6.11) shows the data of the (Un)bound Knapsack tests. Similar to the previous 0/1 Knapsack tests, not all difficulties show a clear downward trend in the mutation impact from the populations 1 to 5.
In the easy tests in graph (6.11.b), we can see population 1 starts with the highest average mutation impact of about 40%, falling to about 30% in population 2. However, from population 2 to population 3, the impact of all four values actually increased. The change from population 3 to population 4 is especially

notable, with it being one of the two times in the evaluation of hypothesis 3, where not all of the four impact metrics are increasing or decreasing at the same time. While the values of the CI and FI decrease slightly from pop 3 to pop 4, the values of the EI and FEI increase. The increase in EI shows, that many different negative traceIDs boost the entropy values, meaning no single mutation is dominating in the last generation. The FI being lower than the other four metrics indicates that the mutations do not have a positive impact on the fitness of the individuals. Population 5 finally has the lowest mutation impact of about 20%.

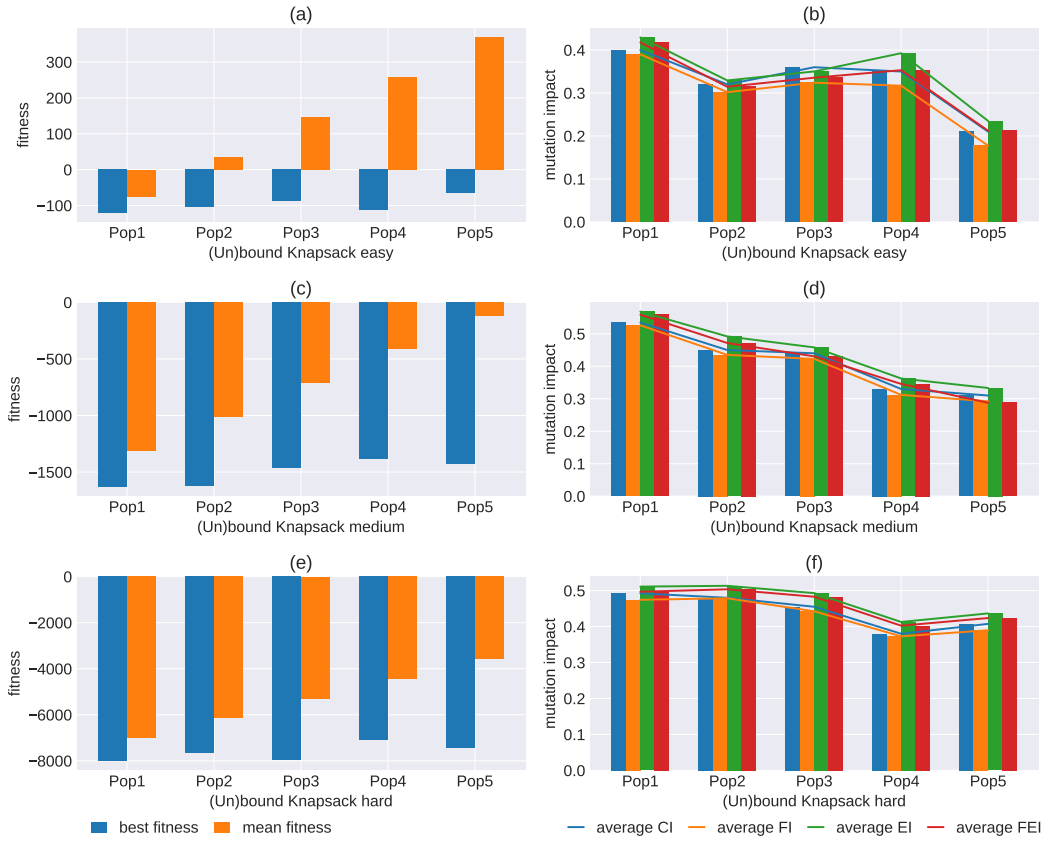

Figure 6.11.: Visualization of the (Un)bound Knapsack problem for hypothesis 3. The left graphs shows the best and average initial fitness of the five populations as a bar chart, graph (a) for the easy, graph (c) for the medium and graph (e) for the hard tests. The right side shows the mutation impact of the four impact metrics per population, also as a bar chart, with graph (b) showing the easy, graph (d) the medium and graph (f) the hard test results.

Comparing the results to the initial fitness in graph (6.11.a), we can see the mutation impact neither matching to the development of the best initial fitness of the populations, nor the average fitness. While the graph correlates for pop 1, pop 2 and pop 5, the populations 3 and 4 are not in line with the assumption of the hypothesis, this time even having different values for throughout the four impact metrics.
The medium tests visualized in graph (6.11.d) on the other hand are more streamlined, steadily decreasing from population 1 to population 5. Like in the easy difficulty, we can see the FI always has the lowest average mutation impact value. Comparing this to the CI, it again suggests that some mutations are slightly decreasing the fitness of an individual in the last generation. Since the mutation FI still starts with 50% impact in population 1, and still has about 30% average FI in pop 5, mutation overall had a big influence on the final result. The EI always being the highest average mutation impact of the four metrics suggests, on the other hand, that the entropy of the individuals with mutation in them is high, possibly meaning different mutation traceIDs boosting the diversity in the last generation. Comparing the results to the initial fitness in graph (6.11.c), the change in mutation impact from pop 2 to pop 3 being the smallest can neither be explained with the best initial fitness, nor the with the mean initial fitness.
The hard difficulty tests are shown in graph (6.11.f). Like in the other two difficulties, the FI is the lowest of the four impact metrics, and the EI the highest. The only exception found in all of the (Un)bound Knapsack tests is population 2, with the CI being lower by a small margin. The mutation impact change from pop 1 to pop 2 is the second time where not all of the four impact metrics rise or drop at the same time, with the CI having a higher value in pop 1 than in pop 2, while the other three metrics have a lower value. Overall the mutation is not showing a clear downward trend with the best initial fitness rising, with the lowest mutation impact value being the FI of pop 4 with a value of 38%. One reason for the fairly similar values across five populations in the hard test may be found in the initial fitness, shown in graph (6.11.e). Compared to the other populations, the changes in the best initial fitness as well as the average fitness shows a smaller relative range compared to the other tests.
The overall smaller changes in the mutation impact compared to the previous Max Ones or 0/1 Knapsack problem might be a result of the initial populations having a bad fitness compared to the other tests. In general, the average fitness of all populations is negative, even for the populations 5, with the best fitness. This indicates that even the better initial populations are not really good. While the average initial fitness sows negative values for all populations in all difficulties, even the best initial fitness shows negative values, except in

the easy tests. This might explain the rather high mutation impact in the populations 5 and potentially also the lower drop in mutation impact from pop 1 to pop 5.

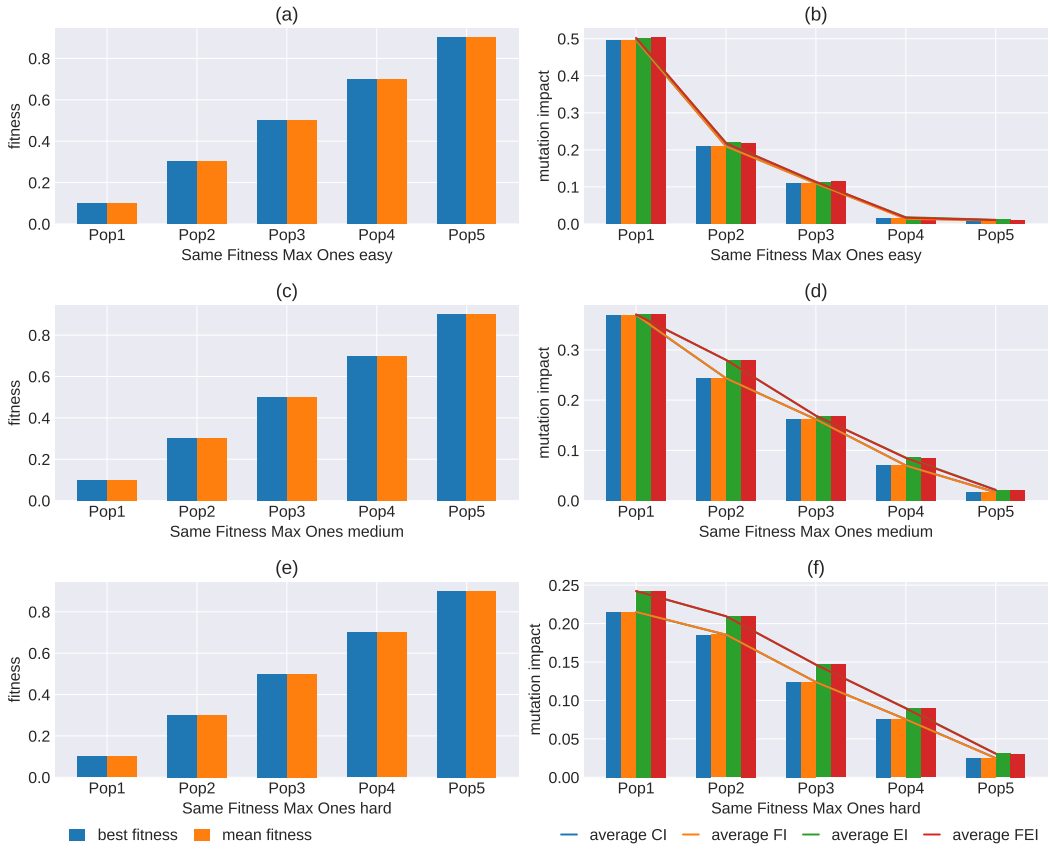

Figure 6.12.: Visualization of the same fitness Max Ones problem for hypothesis 3. The left graphs shows the best and average initial fitness of the five populations as a bar chart, graph (a) for the easy, graph (c) for the medium and graph (e) for the hard tests. The right side shows the mutation impact of the four impact metrics per population, also as a bar chart, with graph (b) showing the easy, graph (d) the medium and graph (f) the hard test results.

Finally, figure (6.12) shows the results of the same fitness Max Ones problem. As the fitness of every individual is the same, the average initial fitness and the best initial fitness of every population is the same on the left graphs (6.12.a), (6.12.c), and (6.12.e). Also, the height of the fitness values is the same for all three difficulties, with the populations 1 having an initial fitness of 0.1, the populations 2 an initial fitness of 0.3, the populations 3 an initial fitness of 0.5,

the populations 4 an initial fitness of 0.7 and finally the initial populations 5 an initial fitness of 0.9. While both the best initial fitness and the average initial fitness are the same in all populations, the resulting mutation impact of the three difficulties show two different trends.
The mutation impact in the easy tests, visualized in graph (6.12.b), drops exponentially. Starting in pop 1 with a value of about 50% mutation impact, the mutation impact of pop 5 does not even reach 1% mutation impact, with all four metrics being very similar over all populations.
The mutation impact of the medium tests on the other hand shows a more linear drop, starting at about 38% for all four values, dropping about 9% for every population, finally reaching about 2% for pop 5. Again, the four impact metrics are similar; however, in populations 2, 3 and 4 the EI and FEI show a slightly higher values than the other two metrics, comparable to the results of the other Max Ones problem.
The hard difficulty also shows this higher values for the metrics considering entropy, starting with a considerable impact difference of about 3% between the two groups. This difference however gets smaller in pop 4, being similar again in pop 5. All four metrics this time also show a relatively linear downward trend from pop 1 to pop 2.
While two of the three tests showed a linear trend from the lower initial fitness pop 1 to the higher initial fitness pop 5, with the mutation impact dropping continuously, the results of the easy tests still raise questions as to the reason for the different results, since this time, initial populations all had the same initial fitness. Comparing the result to the easy 0/1 Knapsack tests, which also showed a quick drop in the mutation impact of all four metrics from pop 1 to pop 5, may lead to the assumption of the easy difficulty with a lower gene count could be a reason for this quicker drop. However, the easy (Un)bound Knapsack problem and the Max Ones problem both do not show this trend. While the higher values in the (Un)bound Knapsack problem could be explained with a bad initial fitness in all five populations, the Max Ones problem still does not fit this explanation.

The first thing to note for the evaluation of hypothesis 3 is that some differences between the four impact metrics can be found. This can largely be attributed to the way the traceIDs for mutated genes are assigned. While the mutation is shown as a whole in all figures, every mutation initially gets its own negative ID, to be able to separate them when calculating the values for entropy and fitness. Every occurring mutation therefore has its own ID, with the impact of all negative traceIDs later being summed to the mutation impact visualized. This explains the effect found that the values for the CI and FI are lower than the impact of the entropy based EI and FEI, found in the graphs (6.9.a), (6.9.b), (6.9.c), (6.10.c), (6.12.b) and (6.12.c). Since mutations occur

between each generation, they can never be phased out of a gene completely. Other graphs showing a smaller FI than the other metrics, suggest the newly occurring mutation is not improving the individuals, together with the low entropy in the last generation indicating a local optimum. This is especially noticeable for both Knapsack problems in the figures (6.10) and (6.11). While the entropy of the last generation from every problem run suggests a very similar population, some differences still can be found for the mutation impact. However, no significantly big differences could be found throughout the run, with the differences being explainable with the method of assigning traceIDs for mutation.
Moving on and comparing the four different tests with each other, we can see some interesting trends. A higher mutation impact, especially in the better fitness initial populations, can be found in the tests with a negative mean or best fitness in the initial population. This can not only be seen in the (Un)bound knapsack tests in figure (6.11), but also in the hard 0/1 Knapsack tests in figure ((6.10).f). While the initial fitnesses of the Max Ones , 0/1 Knapsack and (Un)bound Knapsack problems can hardly be compared, even the populations 5 having a negative fitness is a good indicator for a bad initial population. Therefore, the impact of mutation being higher in these cases is in line with the assumption of hypothesis 3. While in general a downward trend could be found from pop 1 to pop 5 in all tests, the outliers as well as the general development of the mutation impact could not be explained just with the best initial or average fitness of the corresponding populations. While there is an inherent random factor throughout in the results because of the breeding process, more extreme outliers, like pop 2 in the medium 0/1 Knapsack problem, hint at other factors also having a possible effect in the development of an EA. The average impact of the initial population does not seem to be one such factor, with neither the two knapsack tests nor the two Max Ones tests showing a strong correlation. One possible factor could be the combinability of individuals, already discussed in hypothesis 2. If individuals of a population do not combine well with each other, there is a heavy reliance on mutation to improve the population. On the other hand, if the individuals of a population do combine well with each other, there is a smaller reliance on the mutation to improve the population. When converging quickly to the global optimum, mutation could even worsen the otherwise good solutions, as the proof of concept evaluation for run 3 of population 5 from the easy 0/1 Knapsack problem showed. However, assigning a combinability to individuals of a population is heavily dependent on the intended solution and not trivial for complex problems like the Knapsack problems. Further research into the theory is needed to provide information about its impact on the result of an EA.

However, the general assumption of a higher best initial fitness in an initial population leading to a lower impact in mutation holds true for the tests conducted, with only the before mentioned population 2 of the medium 0/1 Knapsack problems being an exception. While there might be other factors contributing to the influence of mutation, the best initial fitness seems to have a major influence on the impact of mutation, with all difficulties of all problems showing a lower mutation impact for a higher initial fitness. The bad initial populations of the Knapsack problems having a comparably higher mutation impact further confirms this hypothesis. As with the hypothesis before, since the proof of concept evaluation showed a large spread in the results in the final population, this finding might not reflect the results of one single run.

7. Conclusion and future Work

Closing the thesis, the conclusion of the proof of concept evaluation as well as the three hypothesis is reached. An outlook on possible future expansions and directions for tracable evolutionary algorithms (T-EAs) is also given.

7.1. Conclusion

To track the impact of the initial generation throughout the generations of an evolutionary algorithm (EA), the T-EA was proposed, tracking genes to individuals from the initial generation by attaching a traceID to every gene. Alongside, four metrics of measuring the impact where presented. The fitness-based impact (FI) extends the basic counting-based impact (CI) by taking the fitness of the individual the gene is in into account, while the entropy-based impact (EI) is dependent on the entropy from the gene counted. Combining both, the fitness-entropy-based impact (FEI) is based both on fitness and entropy.

To evaluate the proposed T-EA and its four metrics for measuring the impact, three tests where designed, spanning not only different problems but also two different representations, the bit vector and the integer vector. All tests where not only run in three difficulty levels, but also with 5 different initial populations, to ensure comparable and robust results spanning different starting conditions. The goal of these three hypotheses was to show the capabilities of information gain through the tracking of gene heritage, as well as answering general assumptions about the inter workings of EAs.

Before evaluating the three hypotheses, a proof of concept evaluation of population 2 and 5 from the easy 0/1 Knapsack problem was done, with the goal of showing the capabilities as well as gaining information on a more basic level. First, a single run of both populations was evaluated. In both of those runs, a quick domination of one single individual was found, with the result of the last population being very similar. Therefore, the results of the four impact metrics were also very similar in the later generations. However, some differences in the initial generations could be found. The early generations still showed

some differences though, with the FI favoring better fitness individuals whose genes survived to the last generation. The EI on the other hand showed higher results for lower impact traceIDs. However, due to the quick drop in entropy over the generations and the later populations featuring very similar individuals, no conclusive comparison could be done for the metrics. Another effect to be noted was the low amount of genes of the initial population surviving to the last generation. This effect might very well be linked to the very similar individuals in the last generation and should be revisited in a future test with a better breeding operator.
Moving on to the results of the multi-run evaluation of the populations 2 and 5, the results of all four impact metrics where found to be similar even in very outlying impact results, proving the breeding operator didn't provide enough diversity. Besides this, the results also were found to be very spread, especially for the lower initial fitness individuals of a population, with the mean impact of only 3 traceIDs in population 2, and only 2 traceIDs in population 5 exceeding 0% impact. The high spread of results shows the findings in this thesis to be applicable more on the average of many testruns, showing a probable result, but not neccecarily being comparable to one single testrun. Furthermore, for population 2 the initially best fitness traceID was not found to have the highest impact at the end.

While the proof of concept evaluation of the two selected populations already showed hints to the results of the hypotheses, the evaluation of all of the tests brought more reliable results outside of the easy difficulty and the 0/1 Knapsack problem. The first finding for all three hypotheses was again the results for the four impact metrics being very similar. This again could be attributed to one single individual dominating the last population in all tests. This proved the similar results not being a reason of the easy difficulty, but of the breeding operator not allowing for enough diversity throughout the testrun. Because of the similar results, the comparison of the four impact metrics was complicated, with only the values for hypothesis 3 finding some differences, explainable through the assignment of traceIDs for mutated genes.
The evaluation of hypothesis 1 showed the top 5 initial fitness ranking not matching with the impact ranking of the four metrics. While for the fitness ranks 1 and 2 the ranks matched up in the majority of cases, this was not found for the further fitness ranks. While no strong correlation between the initial fitness could be found for hypothesis 1, the other two hypotheses showed different values. The impact differences being low for the same fitness Max Ones problem proved hypothesis 2 to be true, and the initial fitness having an influence at the resulting impact. Variations in the data still raise the questions for other factors having an influence on the result. Evaluating the third hypothesis also showed an influence that the best initial fitness individual

of a population has on the impact of mutation. The impact of mutation overall was found to be lower in populations with a higher best initial fitness, not only comparing a specific difficulty in a specific problem, but also overall. This was shown espacially by the two Knapsack problems when having all negative initial populations. However, irregularities in the data also raised the question for other influences on the impact on the survival of a gene other than the fitness of its initial individual. Possible influences found include a possible better chance of survival through a better combinability with other individuals or the configuration of the breeding operator itself with factors like the elite size. Of course, all results need to be viewed with the test configurations in mind. Changing the breeding operator will have a significant influence on the result.

Since the concept of tracking the influence of the initial population is a novel approach, there are still a lot of areas to be explored with the current state of the framework. Furthermore, the T-EA can also be expanded to include more representations and breeding operators. Therefore, the next section is closing the thesis by showing the potential future development of the framework.

7.2. Future Work

Describing the possible future work can be split into two parts. First, other interesting areas to be tested with the existing implementation of T-EAs are explored. Secondly, possible extensions of the framework will be discussed.

Since the topic of this thesis is in a quite novel field, there is a lot of potential for future work to be done. While the data used in this thesis covers different problems of different difficulty levels, the test configuration largely was the same for all test, with the mutation probability being the only difference. Comparing the influence of different breeding operators and initial configurations could be investigated. Also, as the four impact metrics showed very similar results due to the low entropy found in the last generation, further tests are needed for a conclusive comparison. Not only can different breeding operators be evaluated, but also other different compositions of initial populations. Related to that, the combinability of individuals mentioned in the thesis also needs further research as a possible factor for the survival of genes in the EA. Finally, it could be evaluated if there is a certain distribution of the impact results found when evaluating multiple testruns.

Because the T-EA framework currently only supports bit vector and integer vector representations, the implementation of other data types is needed to cover a wider range of problems. Besides other representation options, the

framework currently only supports gene swapping crossover operators, as well as random mutation. The heritage of a gene not being picked from a single individual, but being a combination of the values of both can not be represented by a single traceID, but is a combination of two. Updating the structure of T-EA will therefore be necessary. Saving more than one traceID with an accompanying influence percentage could enable tracking for this type of crossover operators.
Furthermore, mutation operators not randomly assigning new values, but altering the existing value by some amount, are also not sufficiently tracked with the current data structure. A possible expansion would be to track both the original traceID as well as the difference to the original gene value from the current one.
Challenges of the data visualization also need to be overcome, especially for problems with a high number of individuals in the initial population. Since the problems used in this thesis all had only 20 individuals in the initial population, only 20 traceIDs needed to be evaluated. Since in real world scenarios, the number of individuals in the initial population can be much bigger, solutions need to be developed to clearly show the data. Possible solutions could include intelligent filtering or grouping. The same is true regarding the number of generations visualized.
Finally, evaluating the data over many runs does not take into account the quality of the solution. As the FI is taking the fitness of the inidivual into account, including such information, like the best fitness reached in a run, when building a mean impact over a number of runs could also be researched.

Bibliography

[1] *0/1 Knapsack Problem datasets, Universidad del Cauca*. URL: http://artemisa.unicauca.edu.co/~johnyortega/instances_01_KP/.

[2] Musrrat Ali, Millie Pant, and Ajith Abraham. "Unconventional initialization methods for differential evolution". In: *Applied Mathematics and Computation* 219.9 (Jan. 2013), pp. 4474–4494. DOI: 10.1016/j.amc.2012.10.053.

[3] Jeff Clune et al. "Evolving coordinated quadruped gaits with the HyperNEAT generative encoding". In: *2009 IEEE Congress on Evolutionary Computation*. IEEE, May 2009. DOI: 10.1109/cec.2009.4983289.

[4] Carlos A Coello. "A Survey of Constraint Handling Techniques used with Evolutionary Algorithms". In: ().

[5] Paulo Cortez and Mark J. Embrechts. "Using sensitivity analysis and visualization techniques to open black box data mining models". In: *Information Sciences* 225 (Mar. 2013), pp. 1–17. DOI: 10.1016/j.ins.2012.10.039.

[6] Dipankar Dasgupta et al. "On the use of informed initialization and extreme solutions sub-population in multi-objective evolutionary algorithms". In: *2009 IEEE Symposium on Computational Intelligence in Milti-Criteria Decision-Making*. IEEE, Mar. 2009. DOI: 10.1109/mcdm.2009.4938829.

[7] Anupam Datta, Shayak Sen, and Yair Zick. "Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems". In: *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, May 2016. DOI: 10.1109/sp.2016.42.

[8] Na Dong et al. "An opposition-based chaotic GA/PSO hybrid algorithm and its application in circle detection". In: *Computers & Mathematics with Applications* 64.6 (Sept. 2012), pp. 1886–1902. DOI: 10.1016/j.camwa.2012.03.040.

[9] Filip Karlo Dosilovic, Mario Brcic, and Nikica Hlupic. "Explainable artificial intelligence: A survey". In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, May 2018. DOI: 10.23919/mipro.2018.8400040.

[10] Andre Esteva et al. "Dermatologist-level classification of skin cancer with deep neural networks". In: *Nature* 542.7639 (Jan. 2017), pp. 115–118. DOI: 10.1038/nature21056.

[11] Tobias Friedrich and Markus Wagner. "Seeding the initial population of multi-objective evolutionary algorithms: A computational study". In: *Applied Soft Computing* 33 (Aug. 2015), pp. 223–230. DOI: 10.1016/j.asoc.2015.04.043.

[12] Raluca D. Gaina, Simon M. Lucas, and Diego Perez-Liebana. "Population seeding techniques for Rolling Horizon Evolution in General Video Game Playing". In: *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, June 2017. DOI: 10.1109/cec.2017.7969540.

[13] Wei-feng Gao and San-yang Liu. "A modified artificial bee colony algorithm". In: *Computers & Operations Research* 39.3 (Mar. 2012), pp. 687–697. DOI: 10.1016/j.cor.2011.06.007.

[14] Yu Gao and Yong-Jun Wang. "A Memetic Differential Evolutionary Algorithm for High Dimensional Functions' Optimization". In: *Third International Conference on Natural Computation (ICNC 2007)*. IEEE, 2007. DOI: 10.1109/icnc.2007.60.

[15] Jason Gauci and Kenneth Stanley. "Generating large-scale neural networks through discovering geometric regularities". In: *Proceedings of the 9th annual conference on Genetic and evolutionary computation - GECCO '07*. ACM Press, Jan. 2007, pp. 997–1004. DOI: 10.1145/1276958.1277158.

[16] Tony Van Gestel et al. "Linear and non-linear credit scoring by combining logistic regression and support vector machines". In: *The Journal of Credit Risk* 1.4 (2005), pp. 31–60. DOI: 10.21314/jcr.2005.025.

[17] Maoguo Gong et al. "Immune algorithm with orthogonal design based initialization, cloning, and selection for global optimization". In: *Knowledge and Information Systems* 25.3 (Oct. 2009), pp. 523–549. DOI: 10.1007/s10115-009-0261-8.

[18] A. L. Gutiérrez et al. "Comparison of different PSO initialization techniques for high dimensional search space problems: A test with FSS and antenna arrays". In: *Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP)*. Apr. 2011, pp. 965–969.

[19] Lisa Anne Hendricks et al. "Generating Visual Explanations". In: (Mar. 28, 2016). arXiv: http://arxiv.org/abs/1603.08507v1 [cs.CV].

[20] Johan Huysmans et al. "An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models". In: *Decision Support Systems* 51.1 (Apr. 2011), pp. 141–154. DOI: 10.1016/j.dss.2010.12.003.

[21] Borhan Kazimipour, Xiaodong Li, and A. K. Qin. "A review of population initialization techniques for evolutionary algorithms". In: *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, July 2014. DOI: 10.1109/cec.2014.6900618.

[22] Borhan Kazimipour, Xiaodong Li, and A. K. Qin. "Initialization methods for large scale global optimization". In: *2013 IEEE Congress on Evolutionary Computation*. IEEE, June 2013. DOI: 10.1109/cec.2013.6557902.

[23] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Springer Berlin Heidelberg, 2004. DOI: 10.1007/978-3-540-24777-7.

[24] Shuhei Kimura and Koki Matsumura. "Genetic algorithms using low-discrepancy sequences". In: *Proceedings of the 2005 conference on Genetic and evolutionary computation - GECCO '05*. ACM Press, 2005. DOI: 10.1145/1068009.1068225.

[25] Rudolf Kruse et al. *Computational Intelligence*. Springer London, 2016. DOI: 10.1007/978-1-4471-7296-3.

[26] Yiu-Wing Leung and Yuping Wang. "An orthogonal genetic algorithm with quantization for global numerical optimization". In: *IEEE Transactions on Evolutionary Computation* 5.1 (2001), pp. 41–53. DOI: 10.1109/4235.910464.

[27] Manuel López-Ibáñez et al. "The irace package: Iterated racing for automatic algorithm configuration". In: *Operations Research Perspectives* 3 (2016), pp. 43–58. DOI: 10.1016/j.orp.2016.09.002.

[28] H. Maaranen, K. Miettinen, and M.M. Mäkelä. "Quasi-random initial population for genetic algorithms". In: *Computers & Mathematics with Applications* 47.12 (June 2004), pp. 1885–1895. DOI: 10.1016/j.camwa.2003.07.011.

[29] Millie Pant, Radha Thangaraj, and Ajith Abraham. "Particle Swarm Optimization: Performance Tuning and Empirical Analysis". In: *Foundations of Computational Intelligence Volume 3*. Springer Berlin Heidelberg, 2009, pp. 101–128. DOI: 10.1007/978-3-642-01085-9_5.

[30] Lukas Pastorek and Michael O'Neill. "Historical Markings in Neuroevolution of Augmenting Topologies Revisited". In: *Theory and Practice of Natural Computing*. Ed. by Carlos Martín-Vide, Roman Neruda, and Miguel A. Vega-Rodríguez. Cham: Springer International Publishing, 2017, pp. 243–254. ISBN: 978-3-319-71069-3.

[31] L. Peng et al. "A novel differential evolutionwith uniform design for continuous global optimization". In: *Journal ofComputers* 7.1 (2012), pp. 3–10.

[32] Lei Peng and Yuanzhen Wang. "Differential Evolution using Uniform-Quasi-Opposition for Initializing the Population". In: *Information Technology Journal* 9.8 (Aug. 2010), pp. 1629–1634. DOI: 10.3923/itj.2010.1629.1634.

[33] Shahryar Rahnamayan, Hamid R. Tizhoosh, and Magdy M.A. Salama. "A novel population initialization method for accelerating evolutionary algorithms". In: *Computers & Mathematics with Applications* 53.10 (May 2007), pp. 1605–1614. DOI: 10.1016/j.camwa.2006.07.013.

[34] Shahryar Rahnamayan, Hamid R. Tizhoosh, and Magdy M.A. Salama. "Opposition versus randomness in soft computing techniques". In: *Applied Soft Computing* 8.2 (Mar. 2008), pp. 906–918. DOI: 10.1016/j.asoc.2007.07.010.

[35] Sartaj Sahni. "Approximate Algorithms for the 0/1 Knapsack Problem". In: *Journal of the ACM (JACM)* 22.1 (Jan. 1975), pp. 115–124. DOI: 10.1145/321864.321873.

[36] C. E. Shannon. "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27.3 (July 1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[37] M. Sharma and S. Tyagi. "Novel knowledge based selective tabuinitialization in genetic algorithm". In: *International Journal* 3.5 (2013).

[38] Kenneth O. Stanley and Risto Miikkulainen. "Evolving Neural Networks through Augmenting Topologies". In: *Evolutionary Computation* 10.2 (June 2002), pp. 99–127. DOI: 10.1162/106365602320169811.

[39] Dirk Sudholt. *The Benefits of Population Diversity in Evolutionary Algorithms: A Survey of Rigorous Runtime Analyses*. Ed. by Benjamin Doerr and Frank Neumann. Cham: Springer International Publishing, 2020, pp. 359–404. ISBN: 978-3-030-29414-4. DOI: 10.1007/978-3-030-29414-4_8. URL: https://doi.org/10.1007/978-3-030-29414-4_8.

[40] Nguyen Quang Uy et al. "Initialising PSO with randomised low-discrepancy sequences: the comparative results". In: *2007 IEEE Congress on Evolutionary Computation*. IEEE, Sept. 2007. DOI: 10.1109/cec.2007.4424717.

[41] Hui Wang et al. "A New Population Initialization Method Based on Space Transformation Search". In: *2009 Fifth International Conference on Natural Computation*. IEEE, 2009. DOI: 10.1109/icnc.2009.371.

[42] Muhanad Tahrir Younis, Shengxiang Yang, and Benjamin Passow. "Meta-Heuristically Seeded Genetic Algorithm for Independent Job Scheduling in Grid Computing". In: *Applications of Evolutionary Computation*. Springer International Publishing, 2017, pp. 177–189. DOI: 10.1007/978-3-319-55849-3_12.

[43] Matthew D. Zeiler and Rob Fergus. "Visualizing and Understanding Convolutional Networks". In: *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 818–833. DOI: 10.1007/978-3-319-10590-1_53.

[44] Zhi-Hua Zhou, Yuan Jiang, and Shi-Fu Chen. "Extracting Symbolic Rules from Trained Neural Network Ensembles". In: *AI Commun.* 16.1 (Jan. 2003), pp. 3–15. ISSN: 0921-7126.

[45] James Zou and Londa Schiebinger. "AI can be sexist and racist — it's time to make it fair". In: *Nature* 559.7714 (July 2018), pp. 324–326. DOI: 10.1038/d41586-018-05707-8.

Declaration of Authorship

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

Tobias Benecke Magdeburg, 22.05.2020

A. Knapsack configurations

Conf 1	
Max Weight	80
Cost	**Weight**
33	15
24	20
36	17
37	8
12	31

Conf 2	
Max weight	269
Cost	**Weight**
55	95
10	4
47	60
5	32
4	23
50	72
8	80
61	62
85	65
87	46

Conf 3	
Max weight	878
Cost	**Weight**
44	92
46	4
90	43
72	83
91	84
40	68
75	92
35	82
8	6
54	44
78	32
40	18
77	56
15	83
61	25
17	96
75	70
29	48
75	14
63	58

Conf 4	
Max Weight	599
Cost	**Weight**
94	485
506	326
416	248
922	421
649	322
237	795
457	43
815	845
446	955
422	252
791	9
359	901
667	122
598	94
7	738
544	574
334	715
766	882
994	367
893	984
633	299
131	433
428	682
700	72
614	874
874	138
720	856
419	145
794	995
196	529
997	199
116	277
908	97
539	719
707	242
569	107
537	122
931	70
726	98
487	600
772	645
513	267
81	972
943	895
58	213
303	748
764	487
536	923
724	29
789	674

Figure A.1.: Configurations of the Knapsack Problems. Conf 1 is used for the easy (Un)bound Knapsack, Conf 2 for the easy 0/1 Knapsack and the medium (Un)bound Knapsack, Conf 3 for the medium 0/1 Knapsack and the hard (Un)bound Knapsack, and Conf 4 for the hard 0/1 Knapsack.

B. Additional Plots

B.1. Box Plots

B.1.1. Max Ones problem

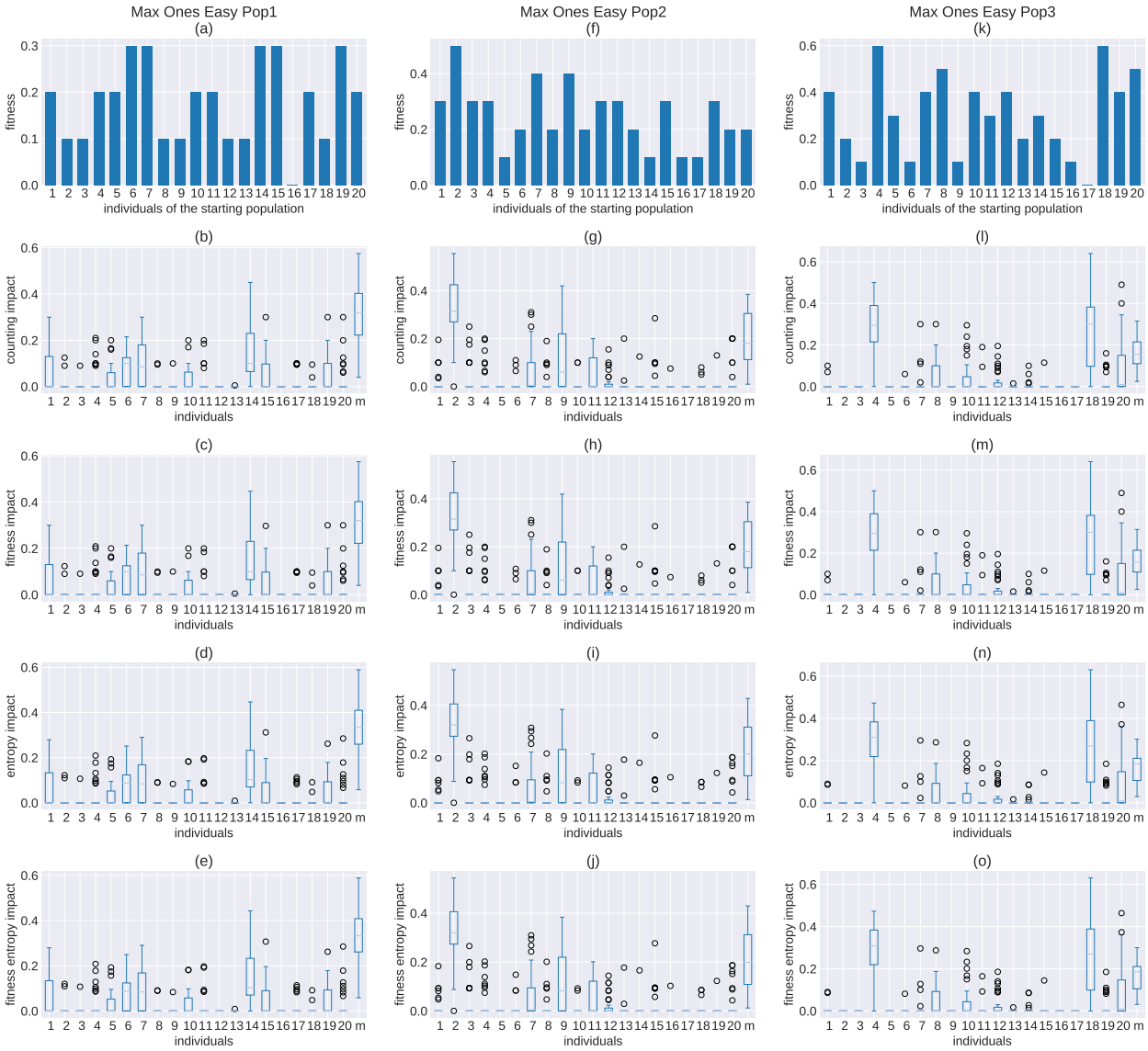

Figure B.1.: Box Plots of populations 1, 2 and 3 from the easy Max Ones problem tests.

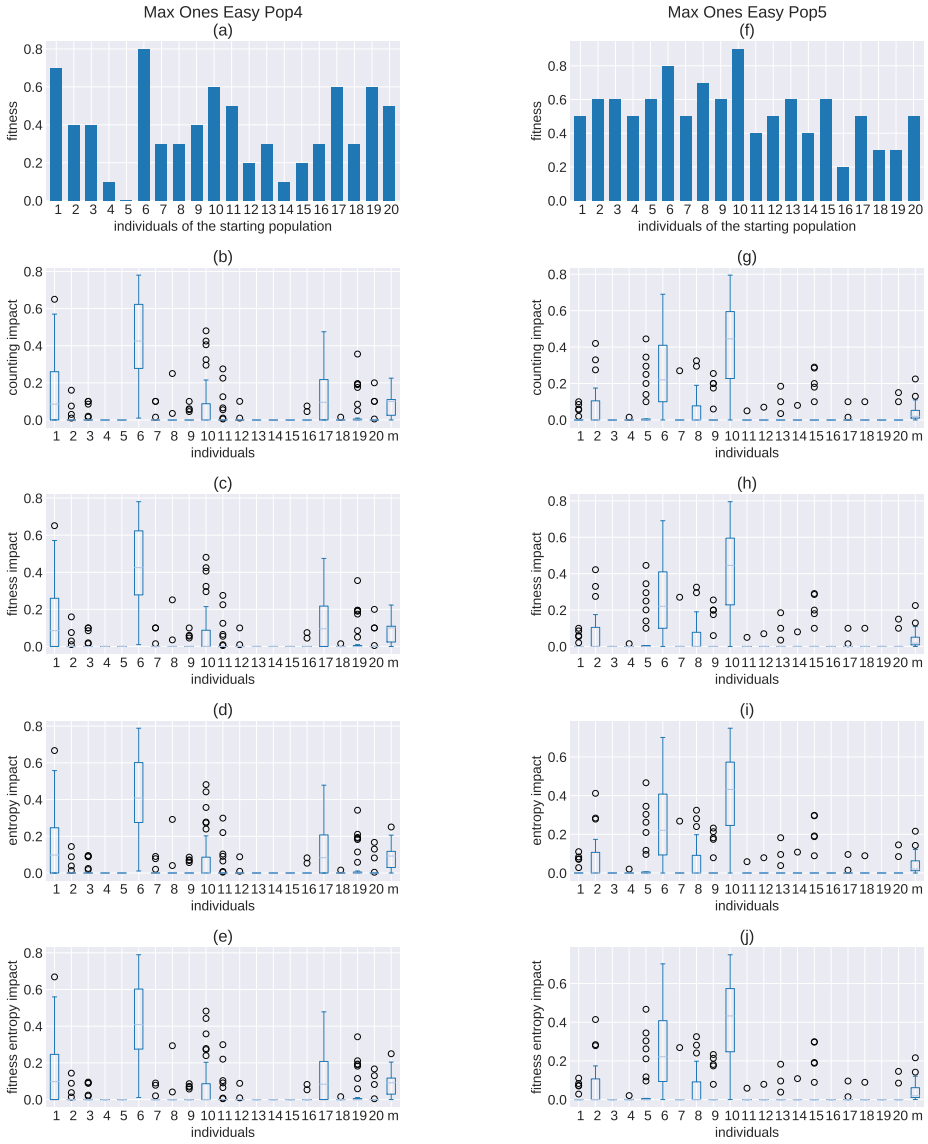

Figure B.2.: Box Plots of populations 4 and 5 from the easy Max Ones problem tests.

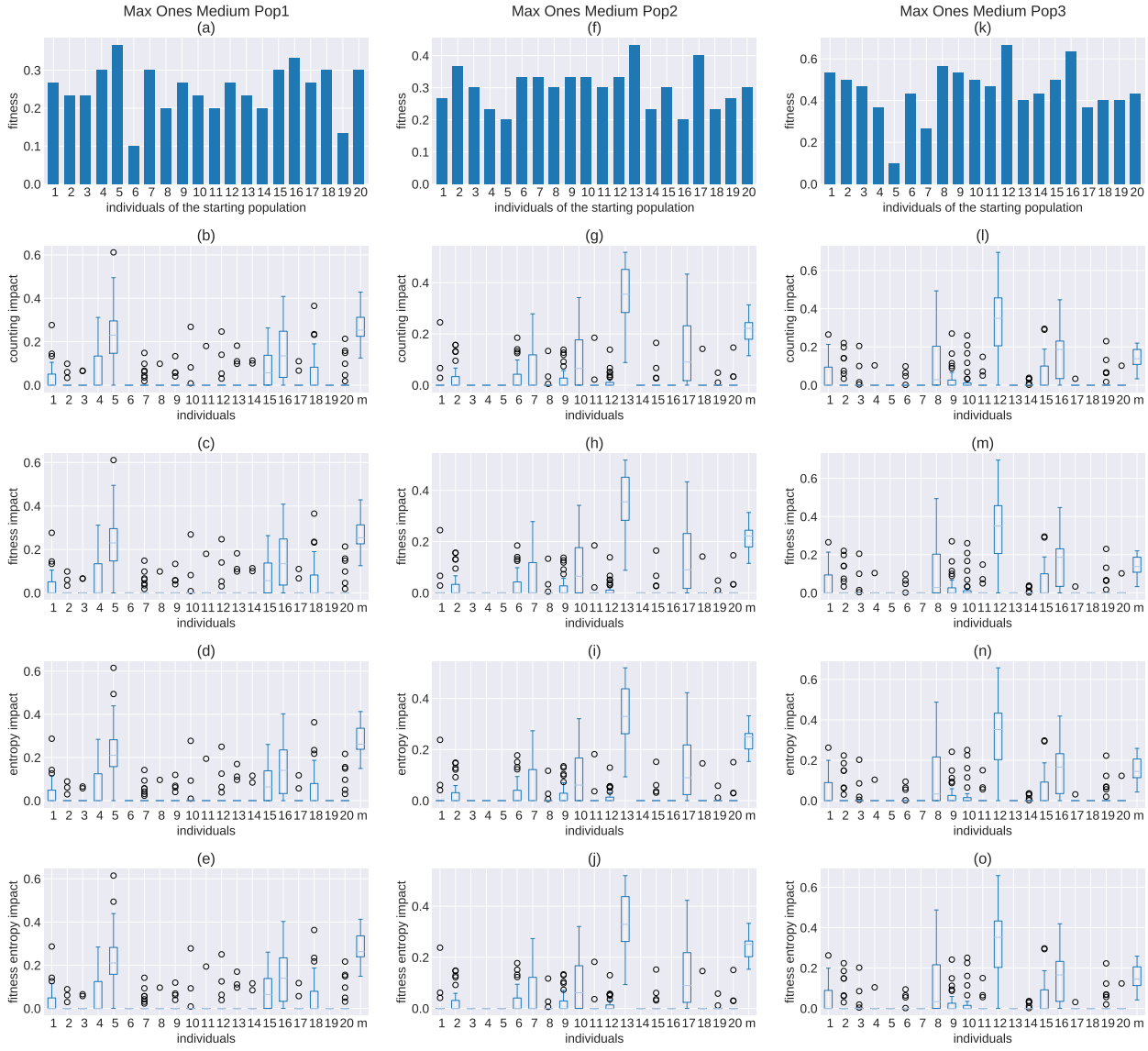

Figure B.3.: Box Plots of populations 1, 2 and 3 from the medium Max Ones problem tests.

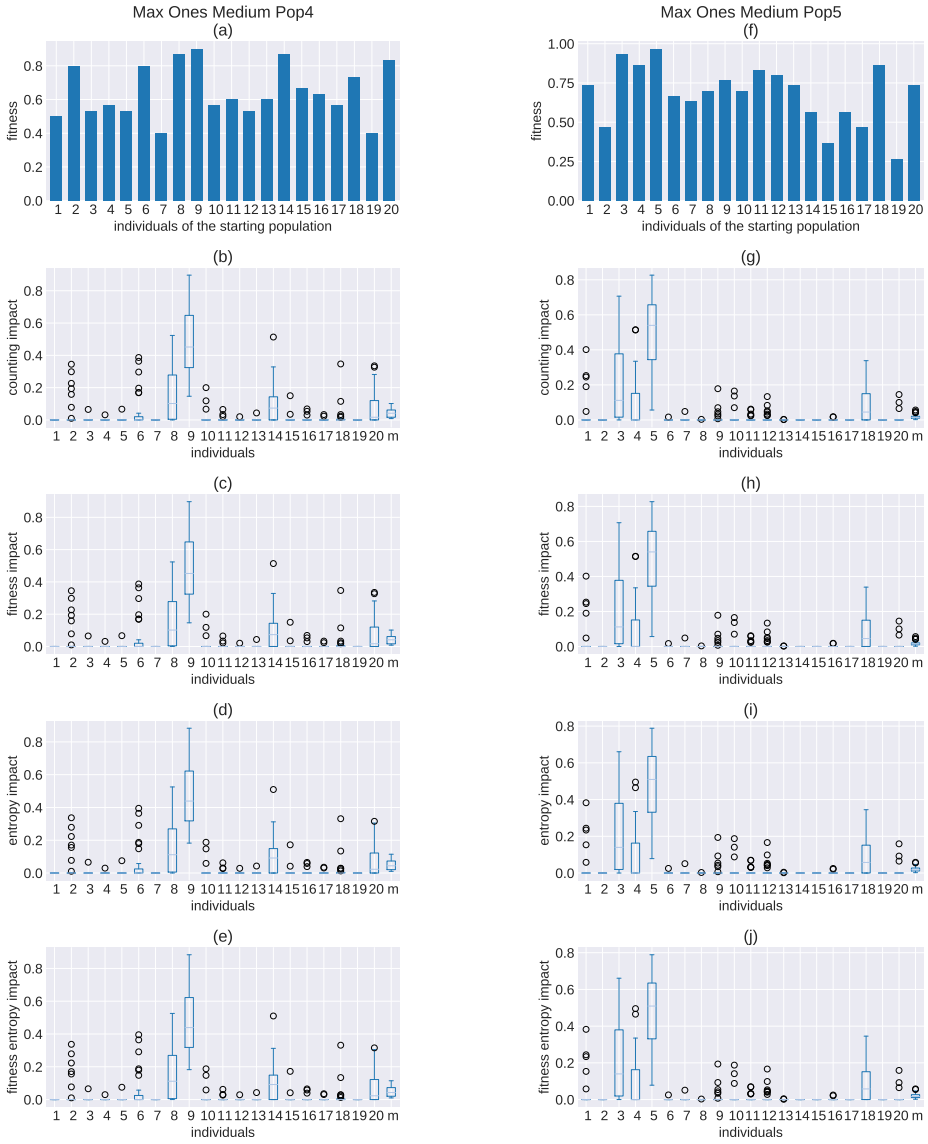

Figure B.4.: Box Plots of populations 4 and 5 from the medium Max Ones problem tests.

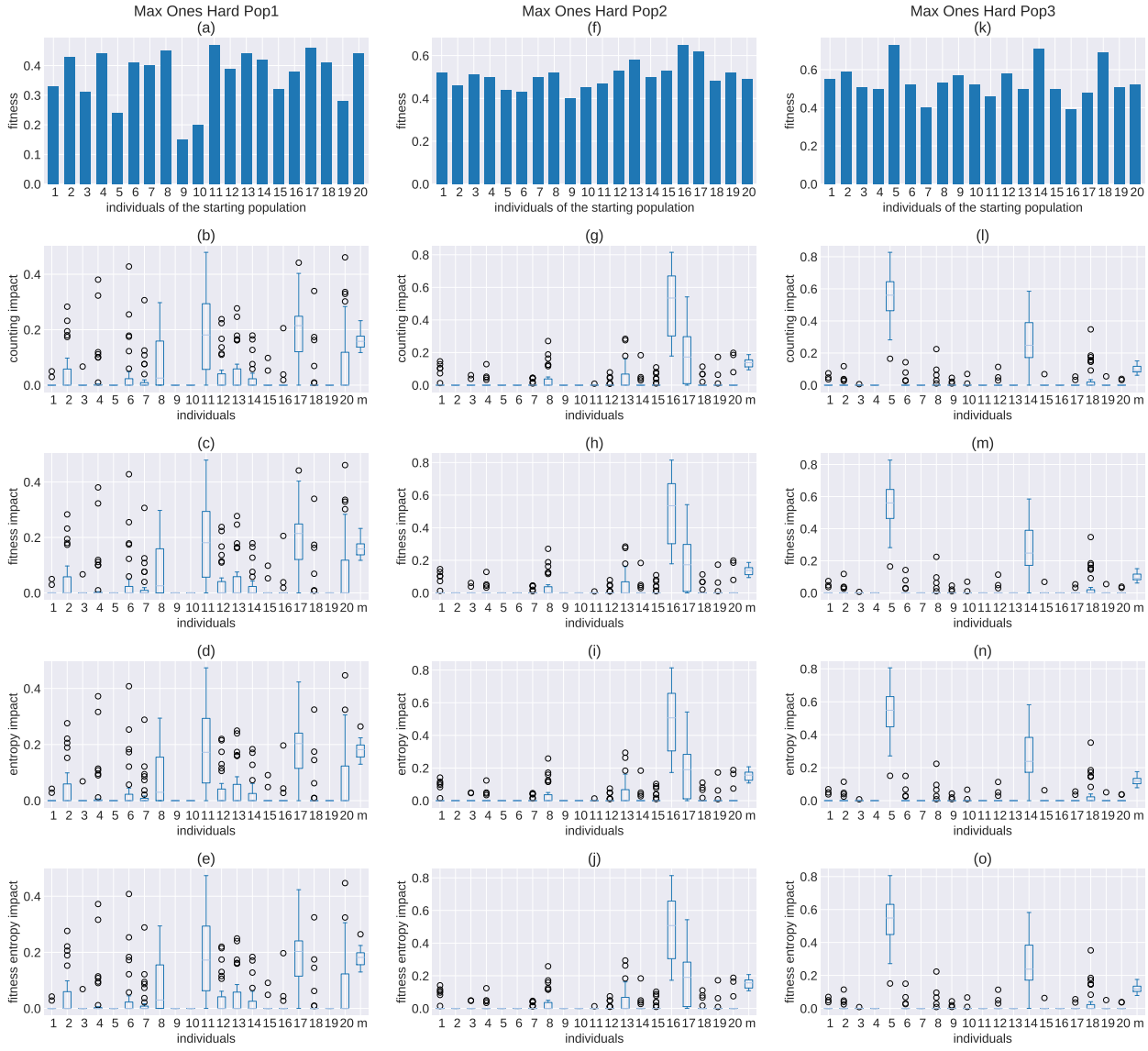

Figure B.5.: Box Plots of populations 1, 2 and 3 from the hard Max Ones problem tests.

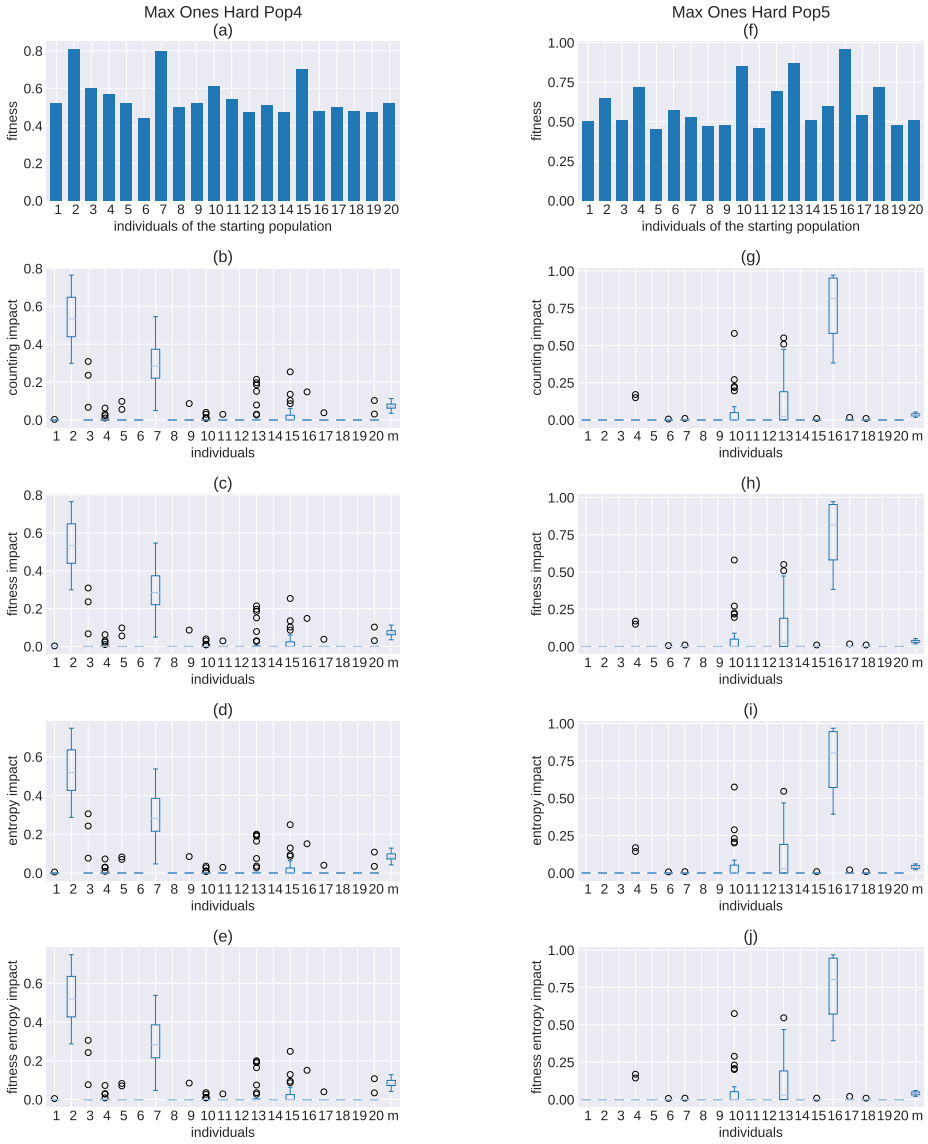

Figure B.6.: Box Plots of populations 4 and 5 from the hard Max Ones problem tests.

B.1.2. 0/1 Knapsack problem

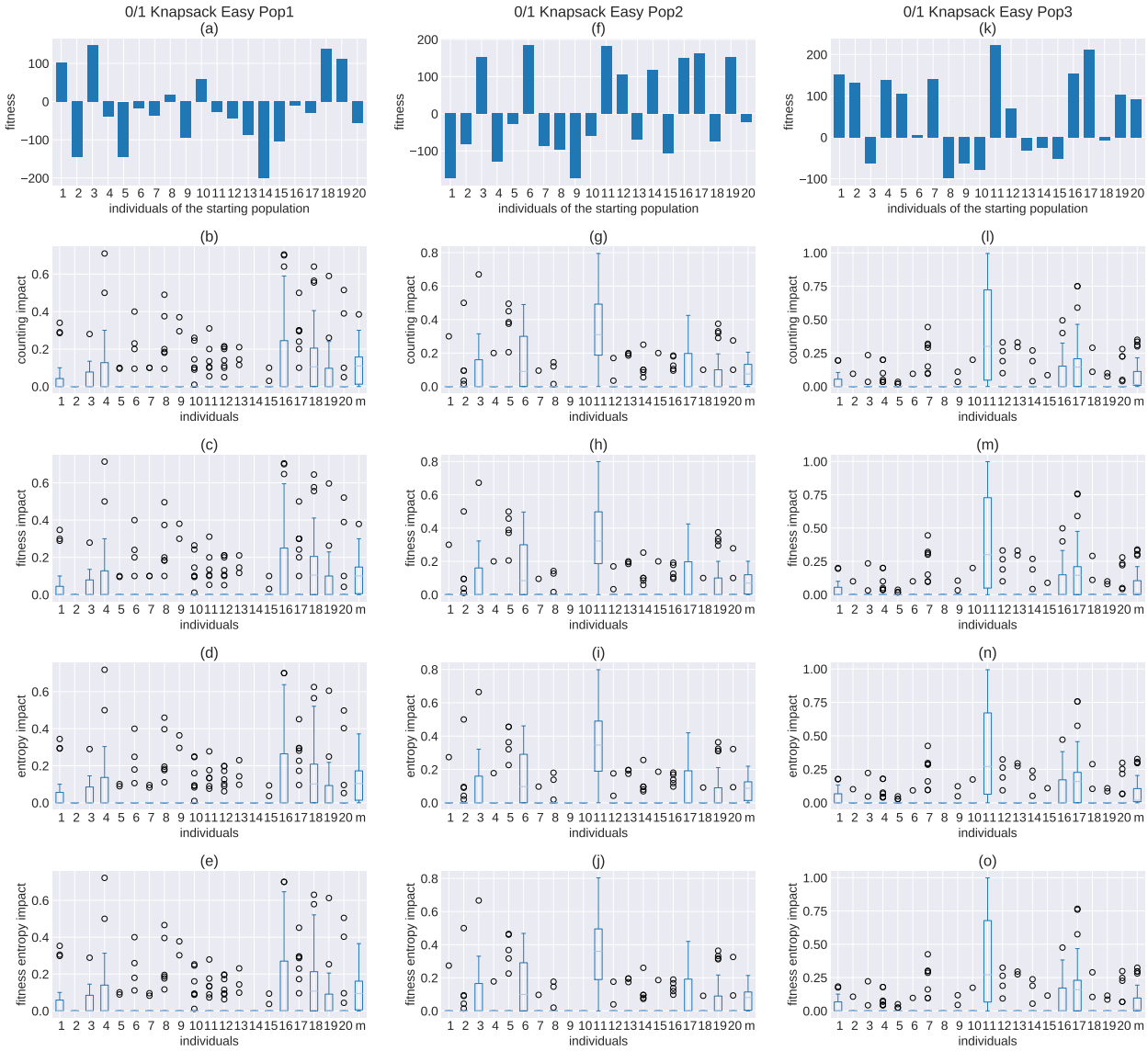

Figure B.7.: Box Plots of populations 1, 2 and 3 from the easy 0/1 Knapsack problem tests.

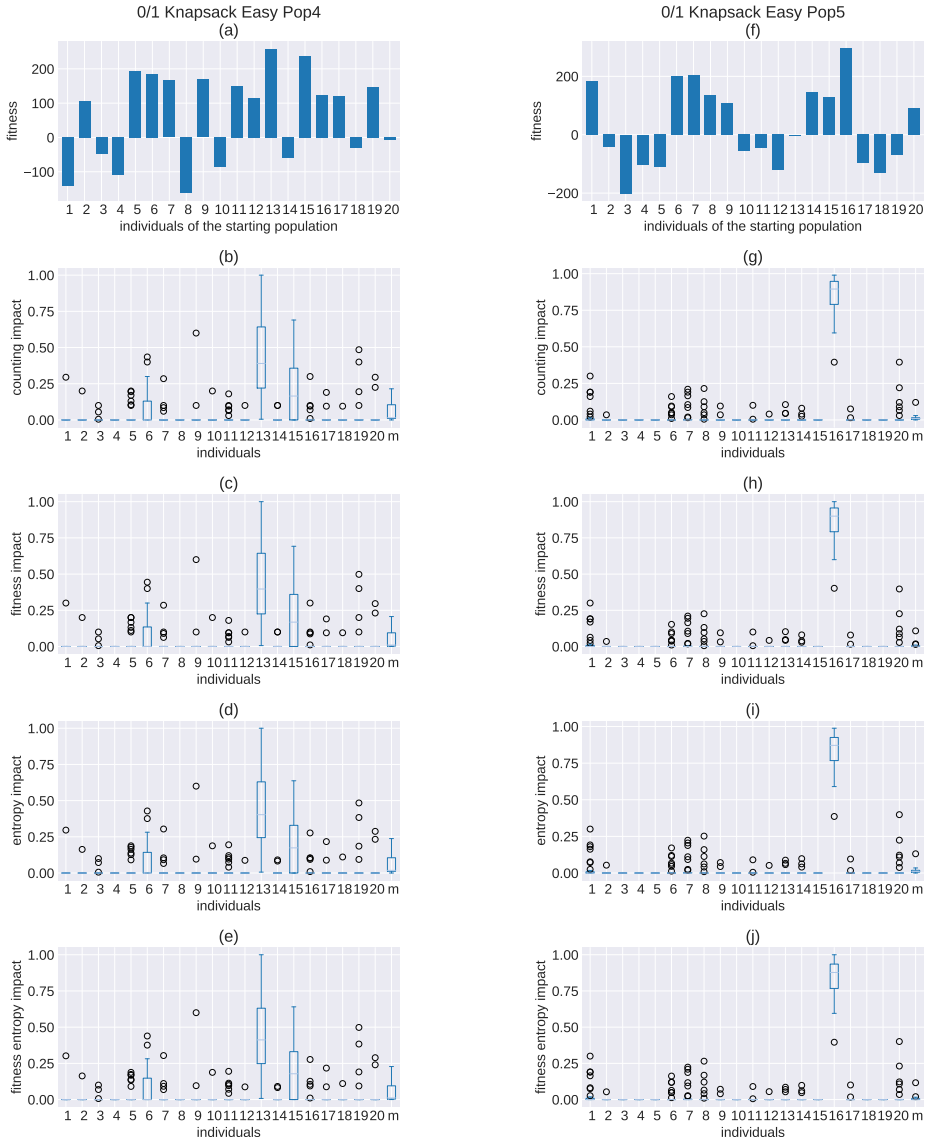

Figure B.8.: Box Plots of populations 4 and 5 from the easy 0/1 Knapsack problem tests.

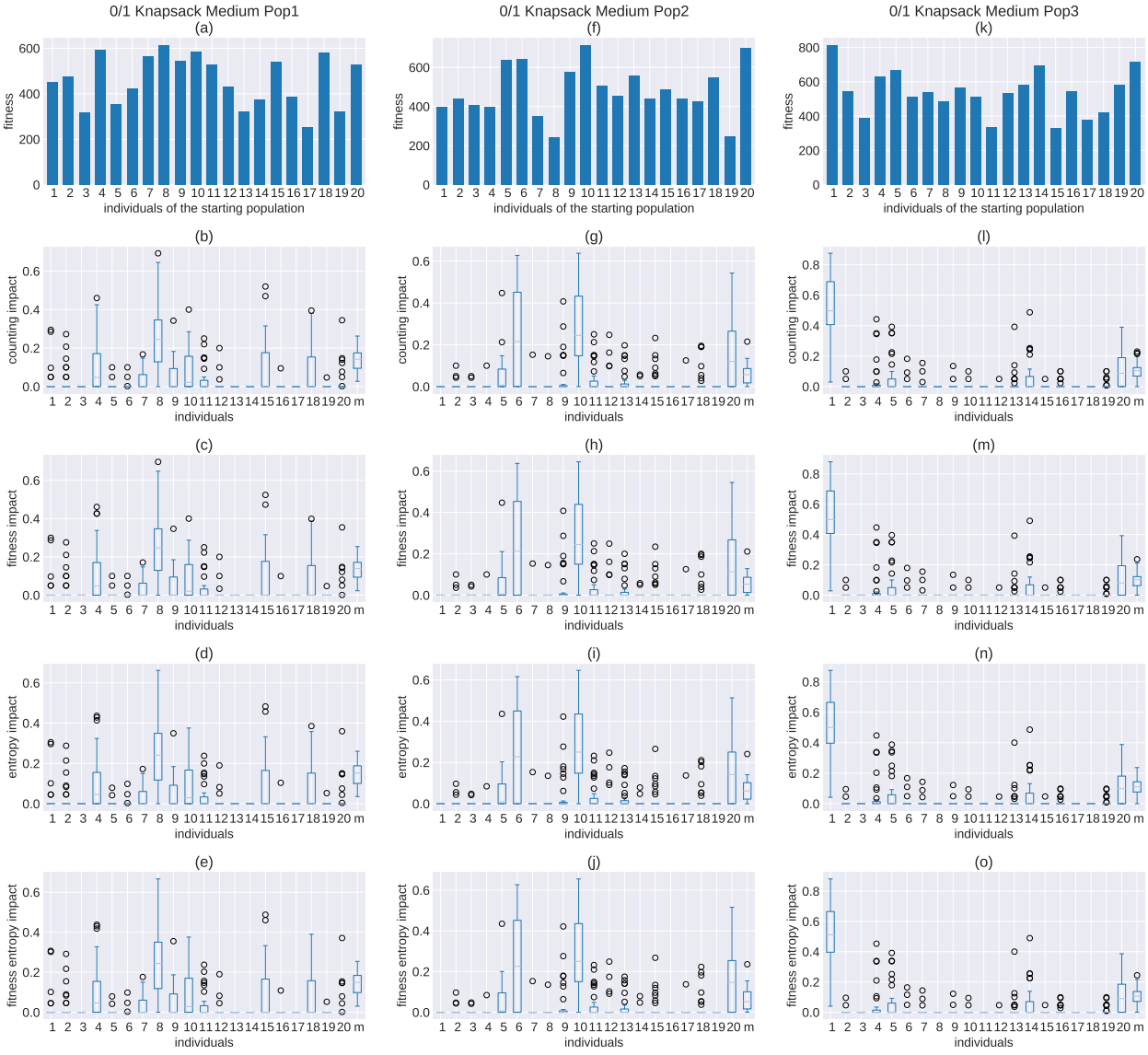

Figure B.9.: Box Plots of populations 1, 2 and 3 from the medium 0/1 Knapsack problem tests.

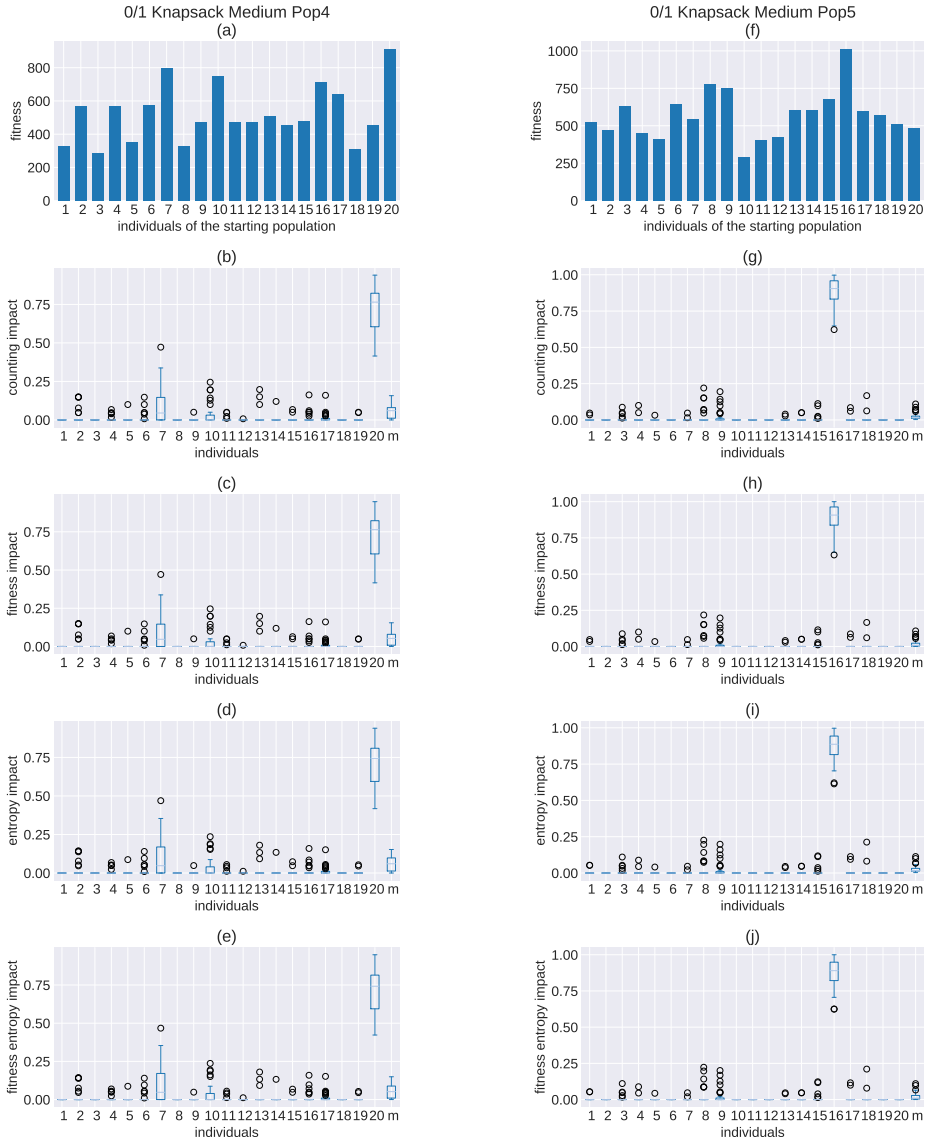

Figure B.10.: Box Plots of populations 4 and 5 from the medium 0/1 Knapsack problem tests.

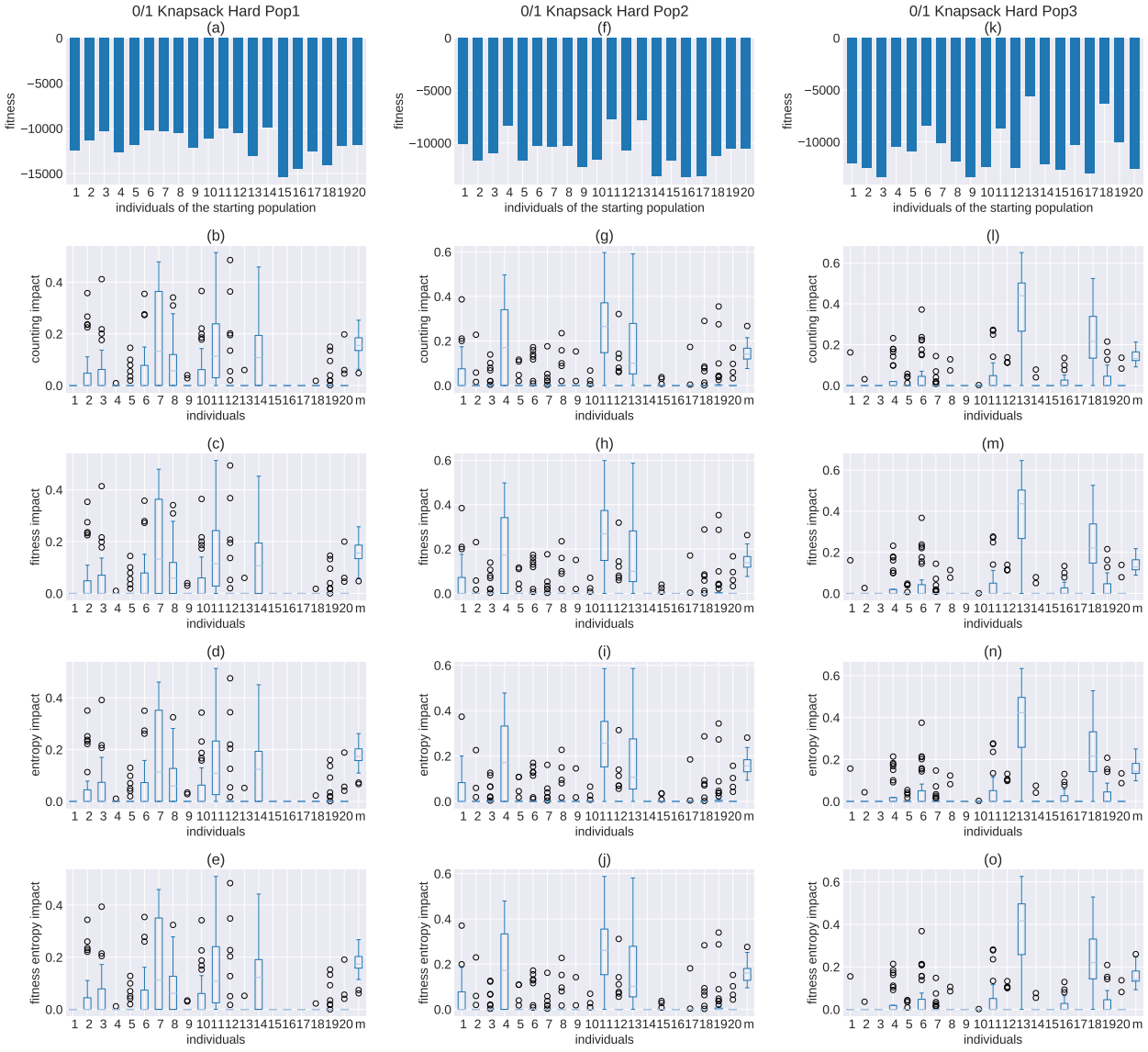

Figure B.11.: Box Plots of populations 1, 2 and 3 from the hard 0/1 Knapsack problem tests.

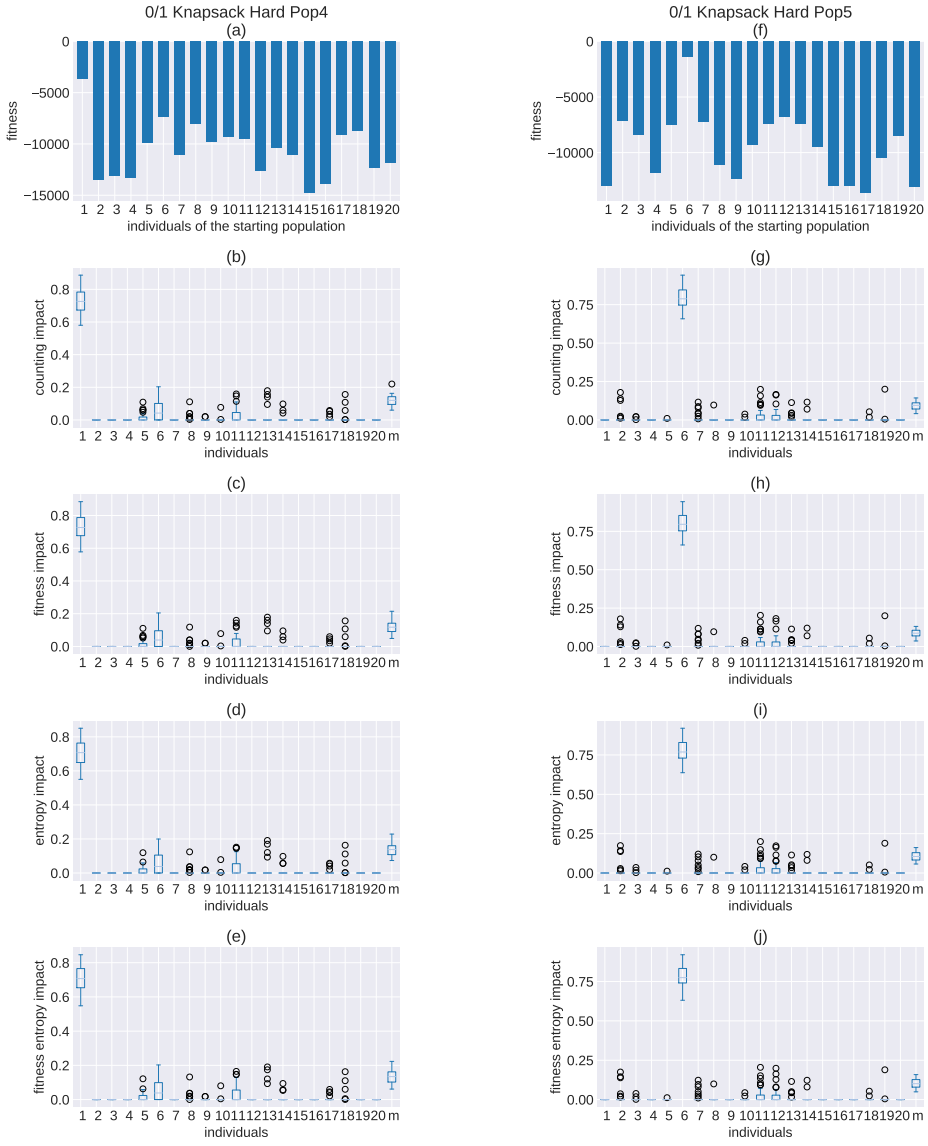

Figure B.12.: Box Plots of populations 4 and 5 from the hard 0/1 Knapsack problem tests.

B.1.3. (Un)bound Knapsack problem

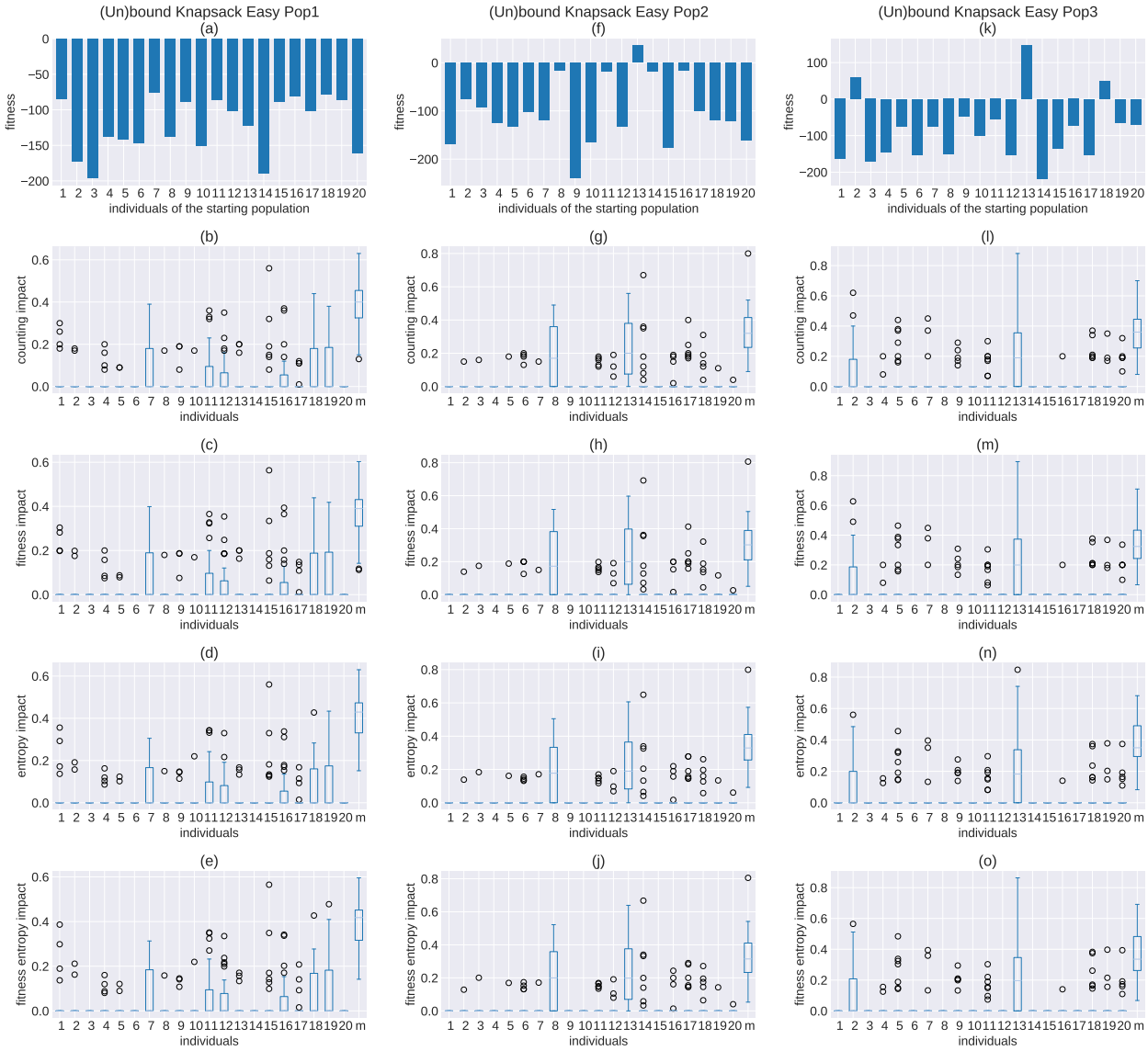

Figure B.13.: Box Plots of populations 1, 2 and 3 from the easy (Un)bound Knapsack problem tests.

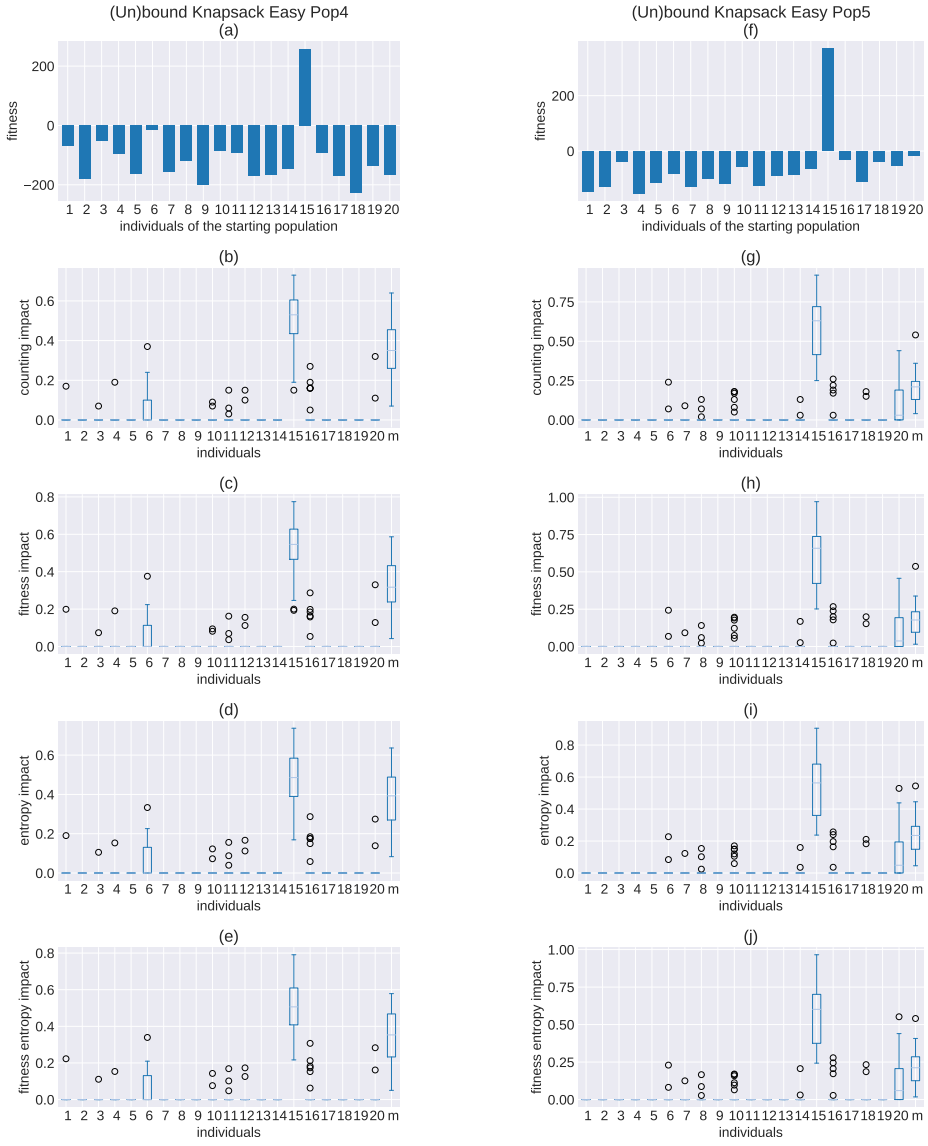

Figure B.14.: Box Plots of populations 4 and 5 from the easy(Un)bound Knapsack problem tests.

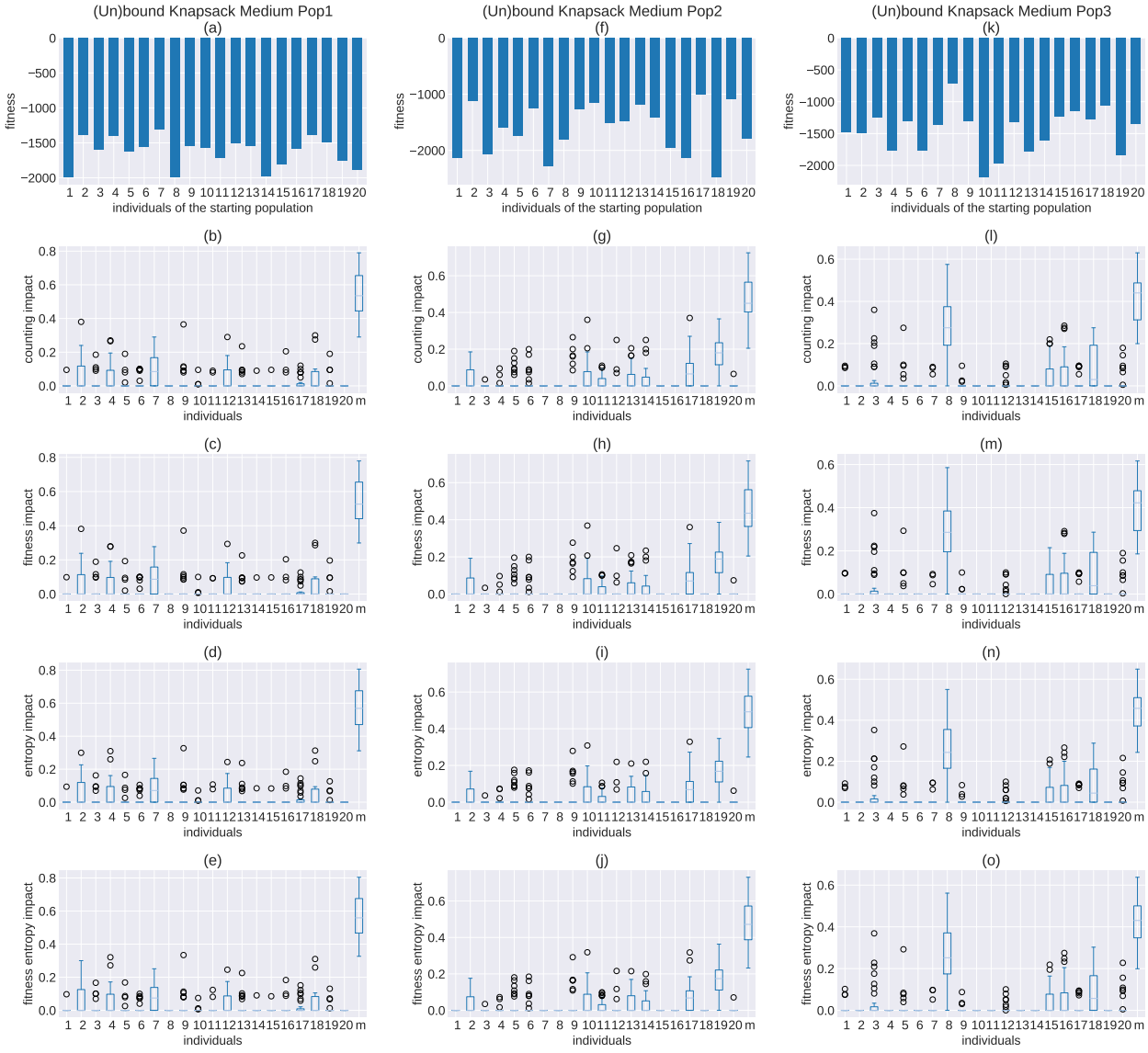

Figure B.15.: Box Plots of populations 1, 2 and 3 from the medium (Un)bound Knapsack problem tests.

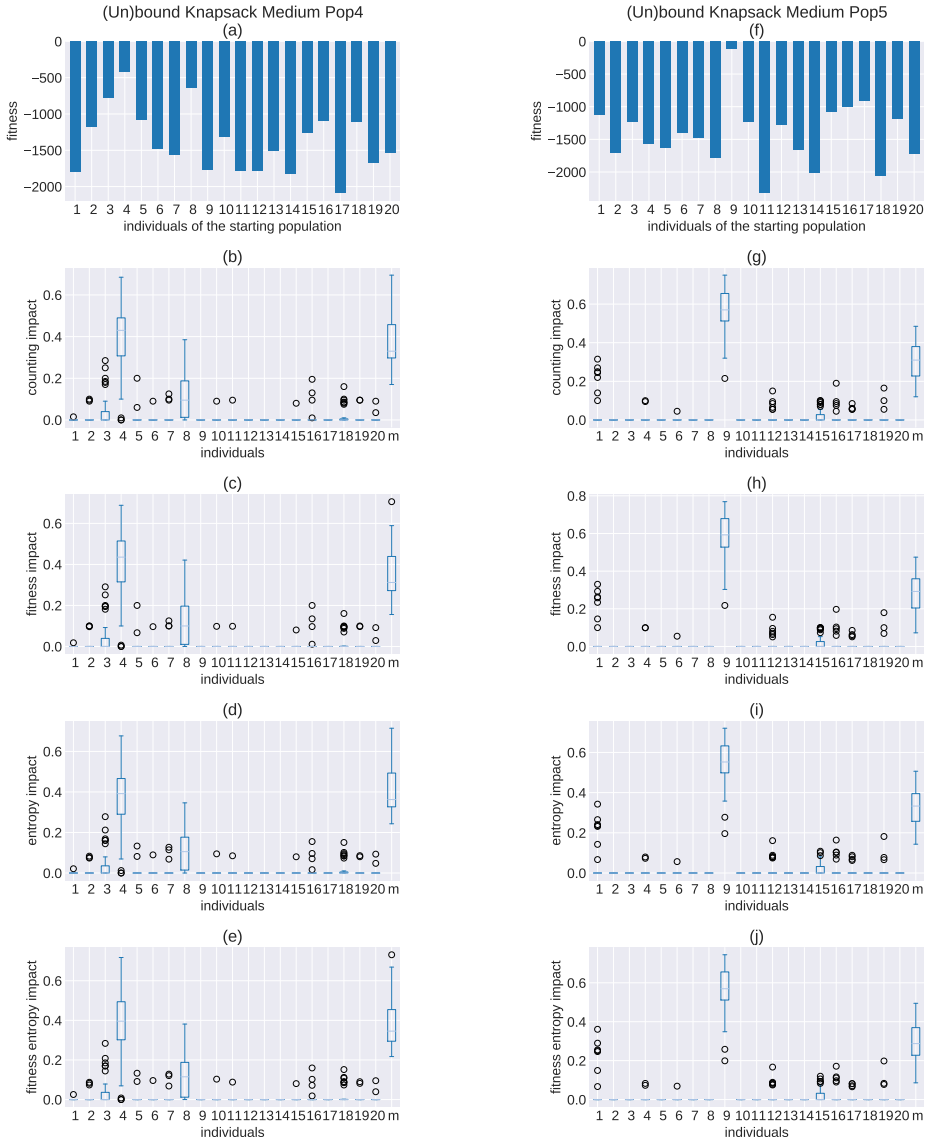

Figure B.16.: Box Plots of populations 4 and 5 from the medium (Un)bound Knapsack problem tests.

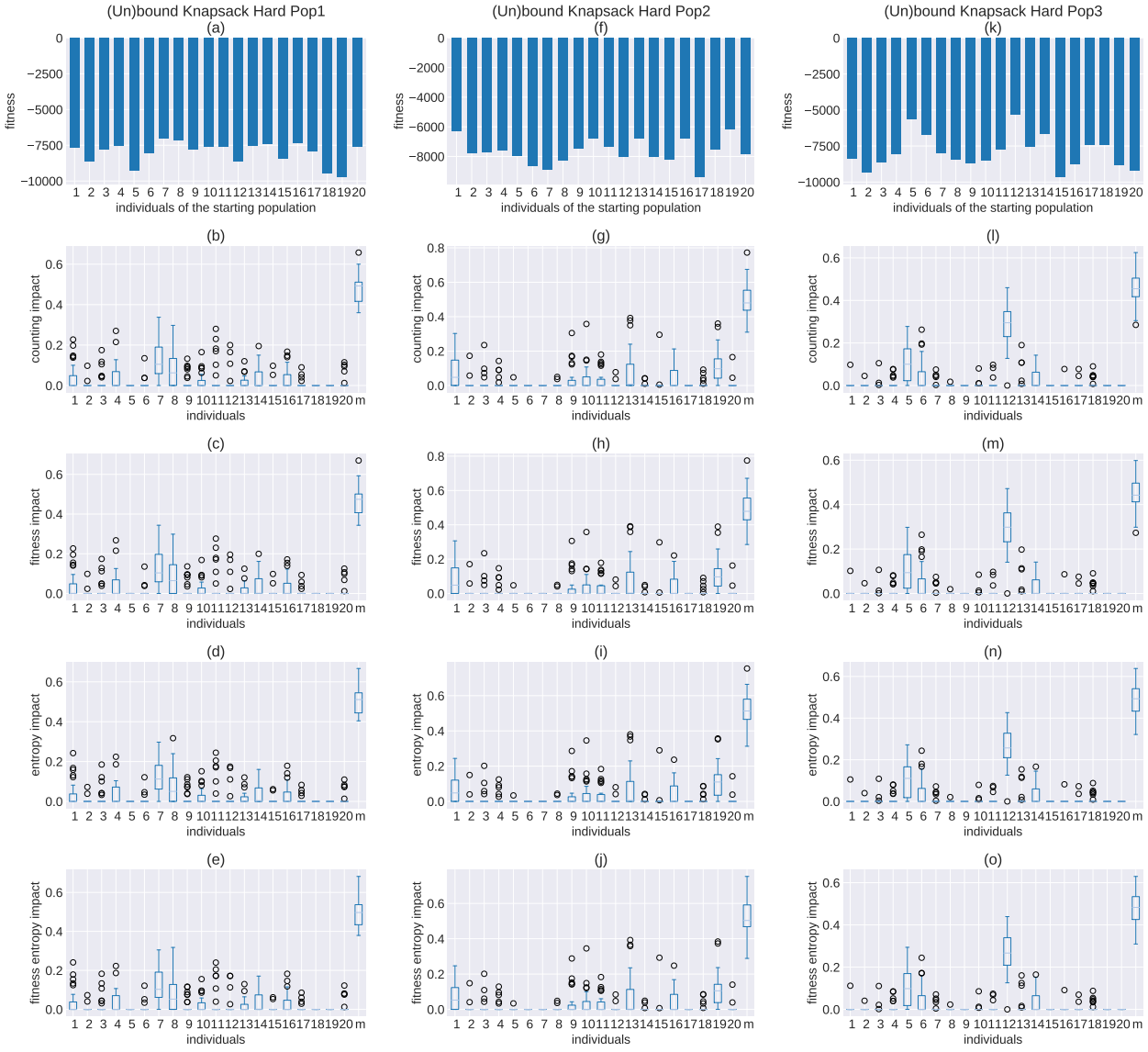

Figure B.17.: Box Plots of populations 1, 2 and 3 from the hard (Un)bound Knapsack problem tests.

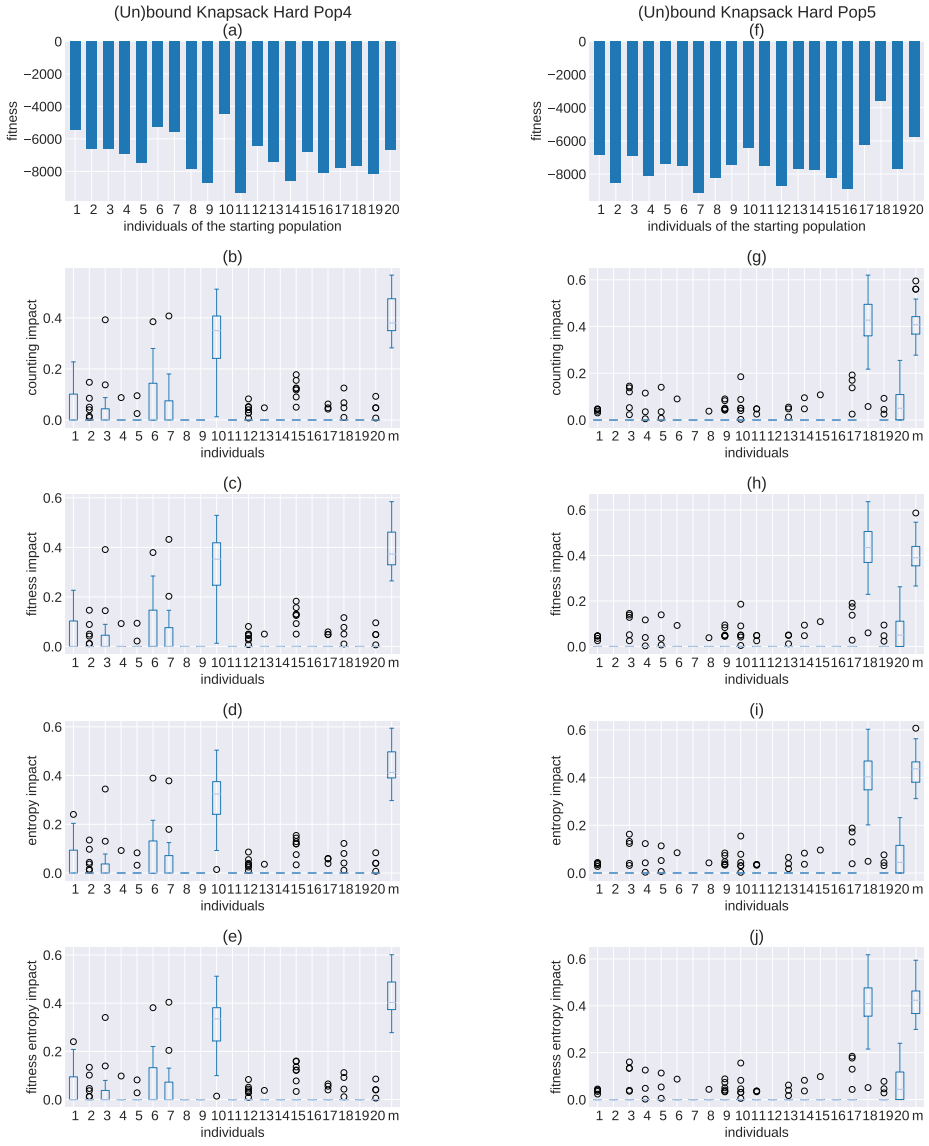

Figure B.18.: Box Plots of populations 4 and 5 from the hard (Un)bound Knapsack problem tests.

B.1.4. Same Fitness Max Ones Problem tests

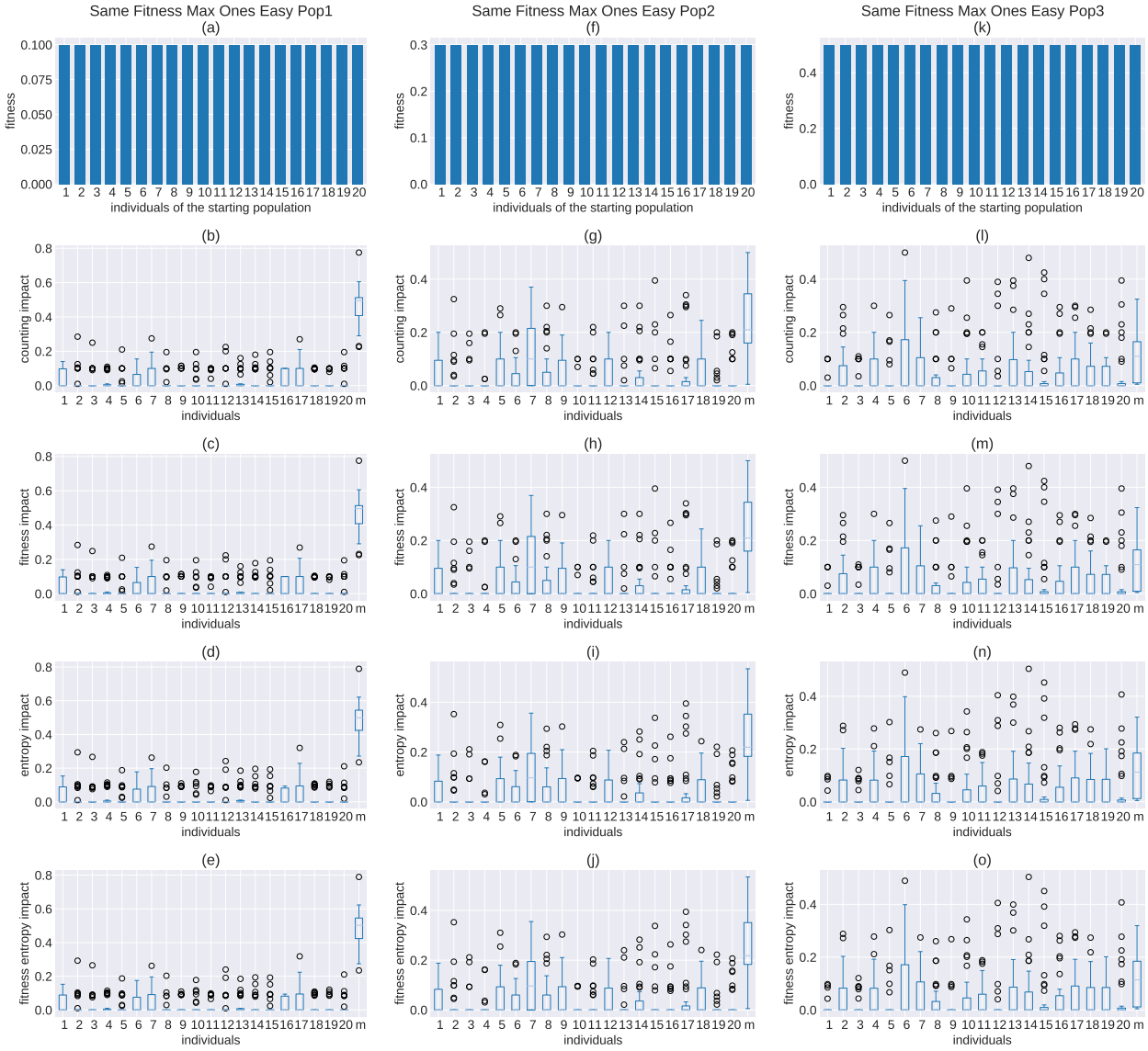

Figure B.19.: Box Plots of populations 1, 2 and 3 from the easy same fitness Max Ones problem tests.

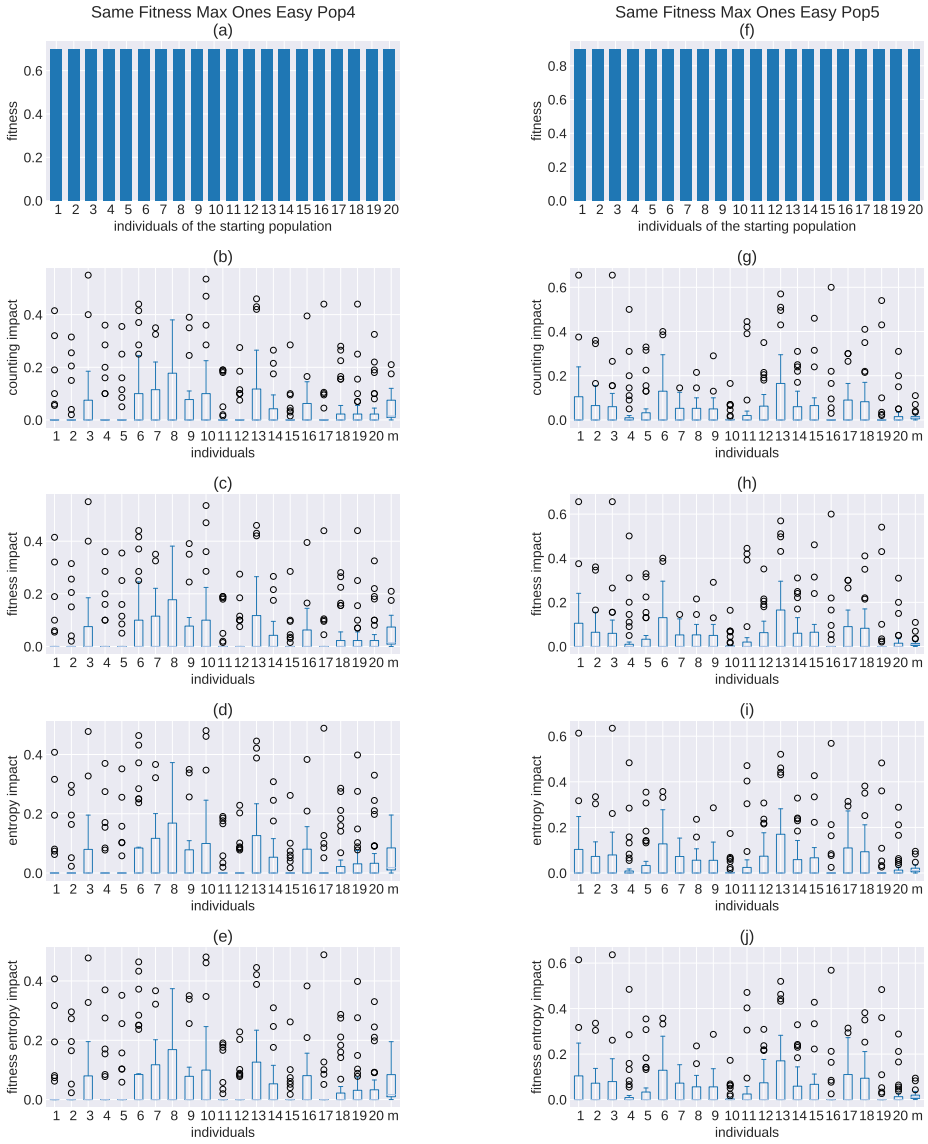

Figure B.20.: Box Plots of populations 4 and 5 from the easy same fitness Max Ones problem tests.

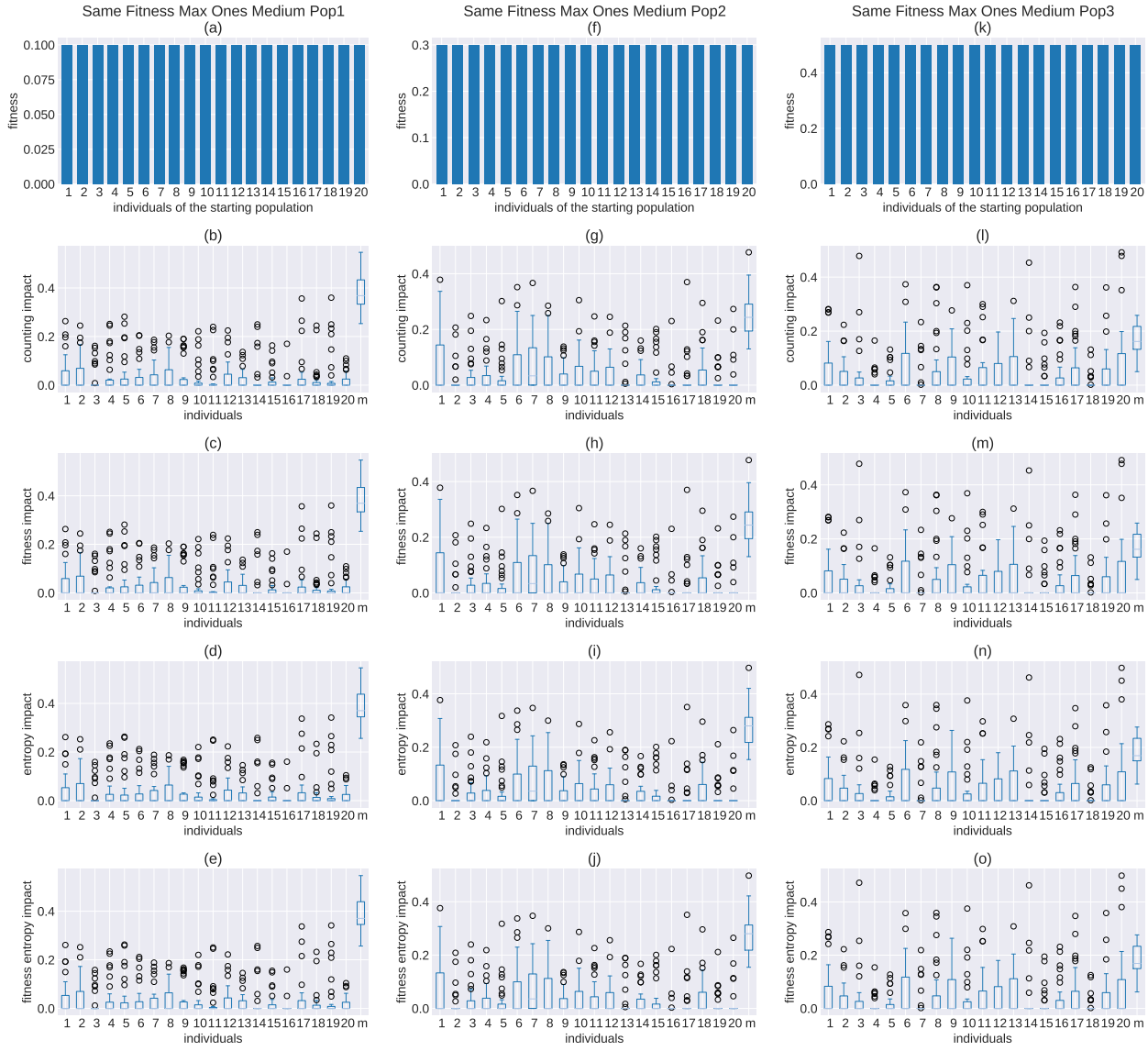

Figure B.21.: Box Plots of populations 1, 2 and 3 from the medium same fitness Max Ones problem tests.

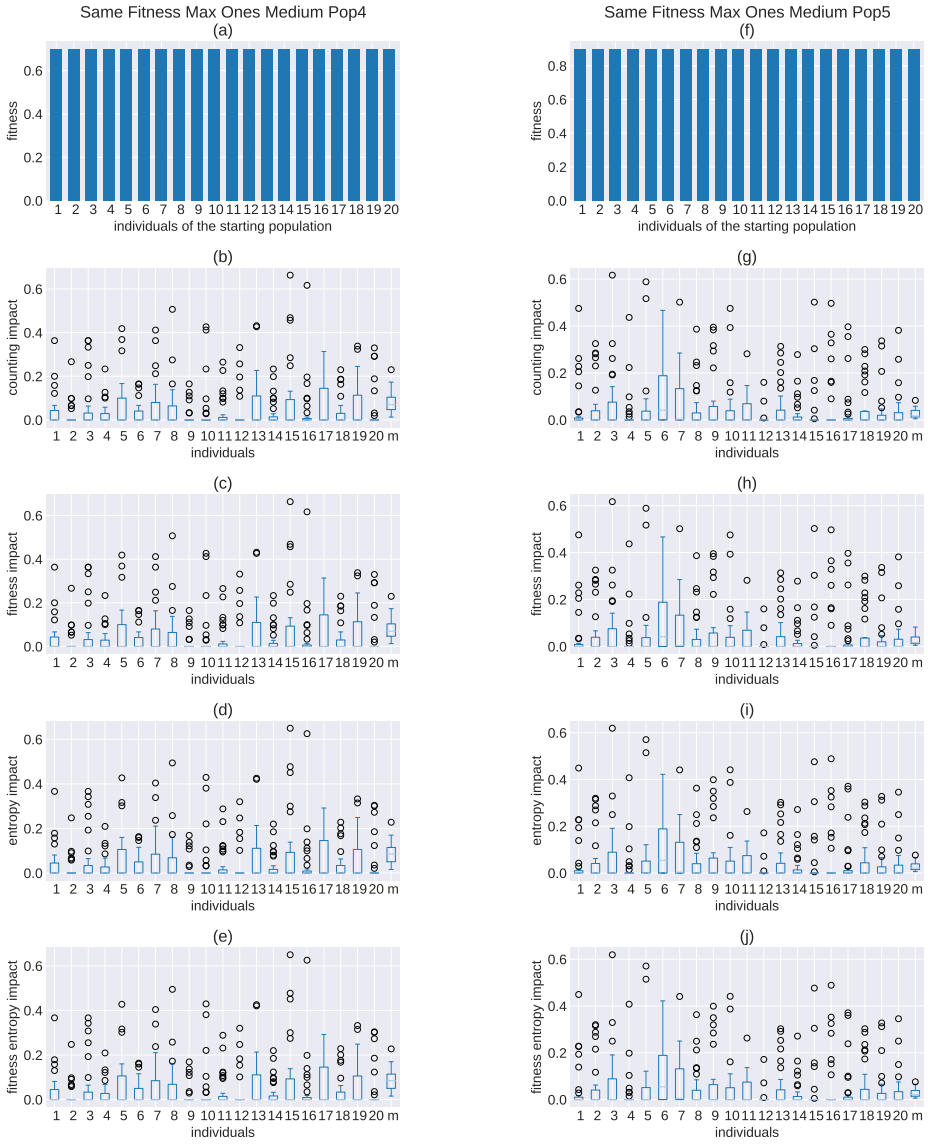

Figure B.22.: Box Plots of populations 4 and 5 from the medium same fitness Max Ones problem tests.

Figure B.23.: Box Plots of populations 1, 2 and 3 from the hard same fitness Max Ones problem tests.

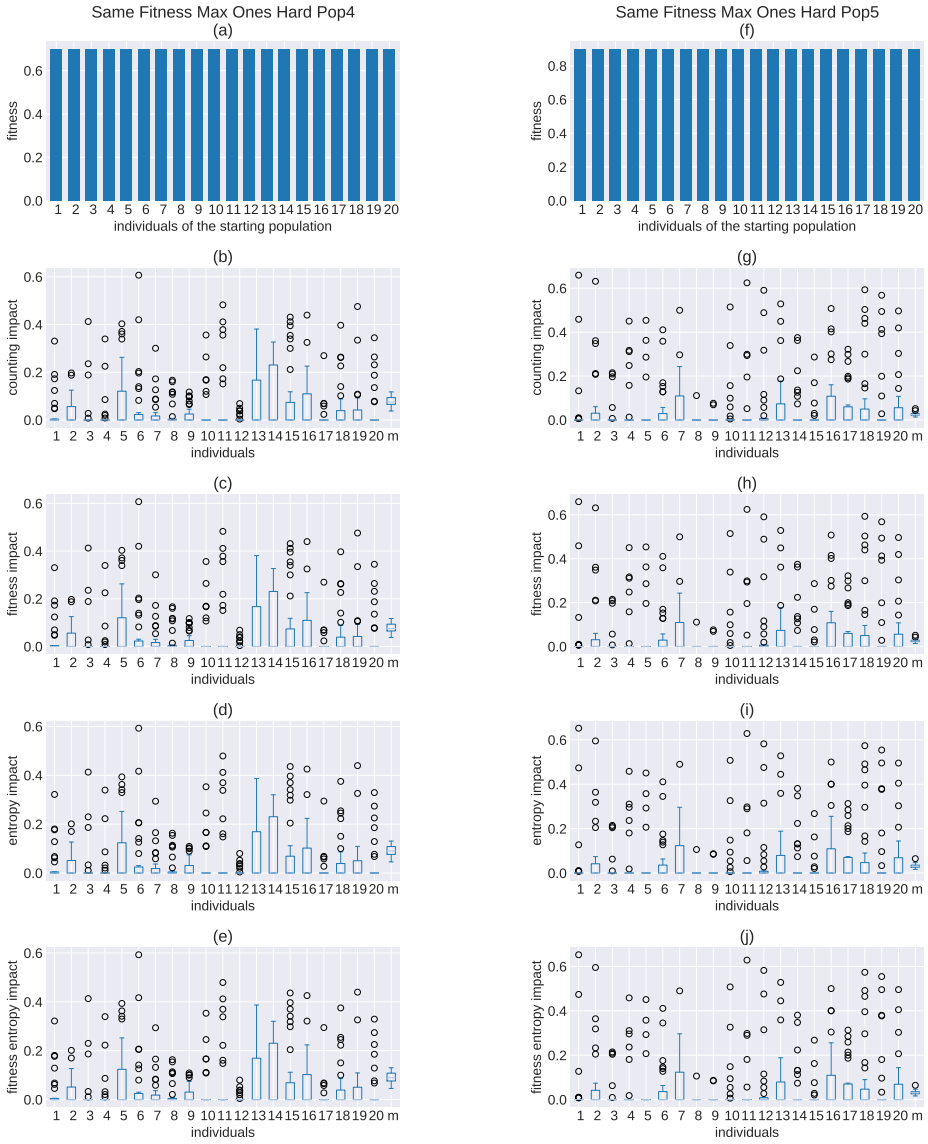

Figure B.24.: Box Plots of populations 4 and 5 from the hard same fitness Max Ones problem tests.

B.2. Hypothesis 1 additional plots

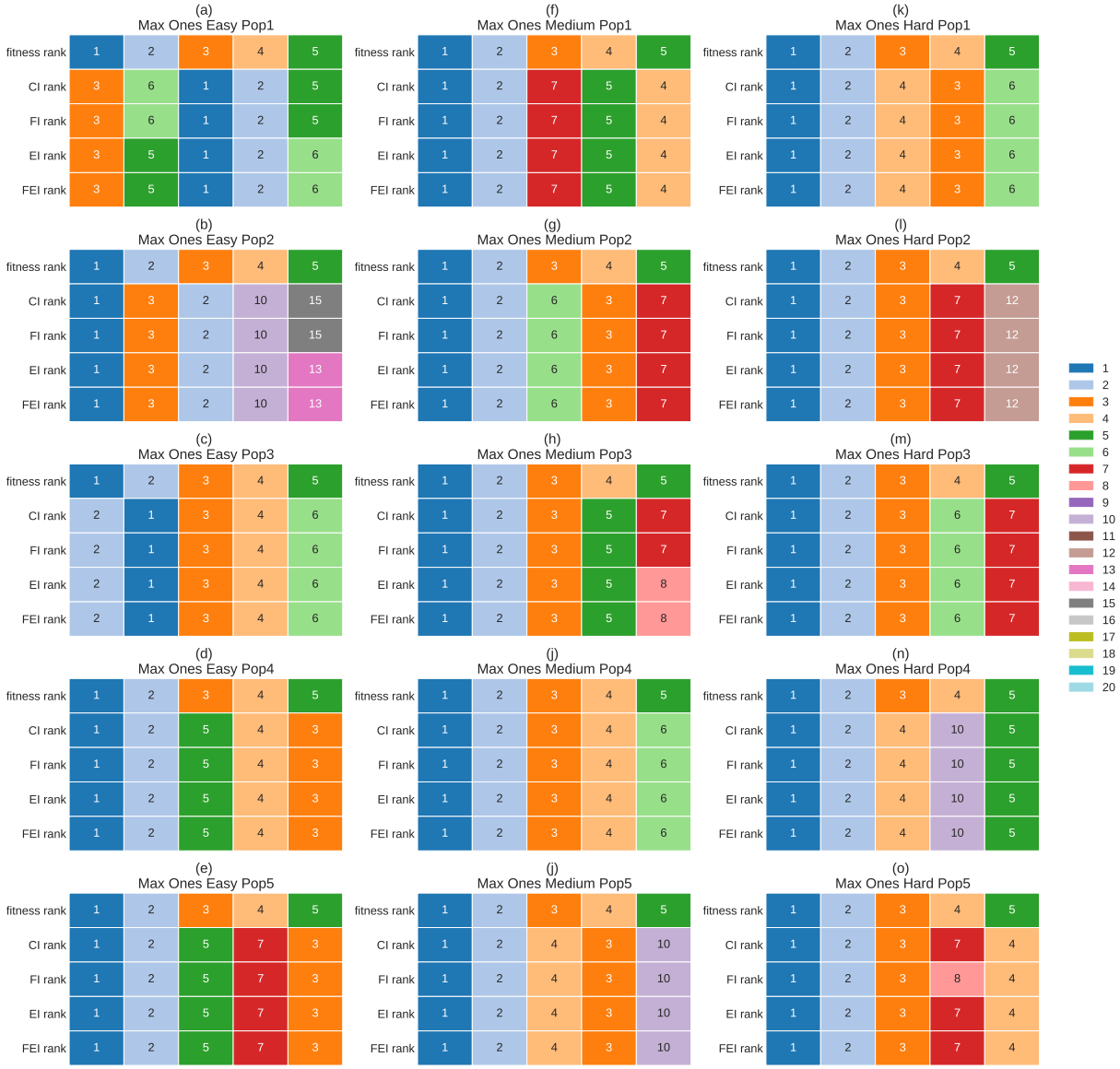

Figure B.25.: Top 5 Max Ones problem fitness vs impact ranking. The left column shows the easy tests, the middle column the medium and the right the hard tests. Each graph represents the initial fitness rank on the top, with the ranking of the four impact metrics below.

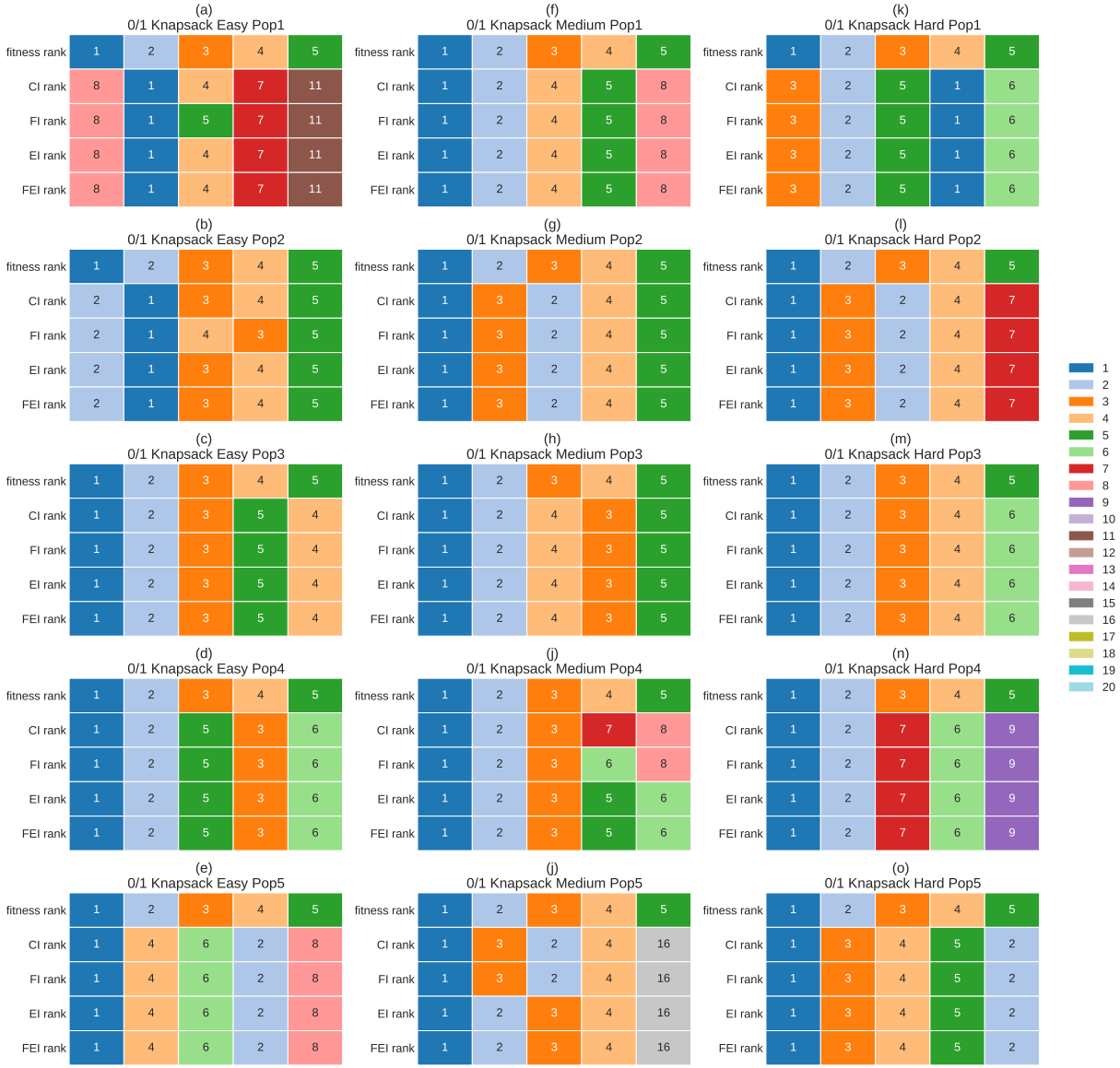

Figure B.26.: Top 5 0/1 Knapsack problem fitness vs impact ranking. The left column shows the easy tests, the middle column the medium and the right the hard tests. Each graph represents the initial fitness rank on the top, with the ranking of the four impact metrics below.

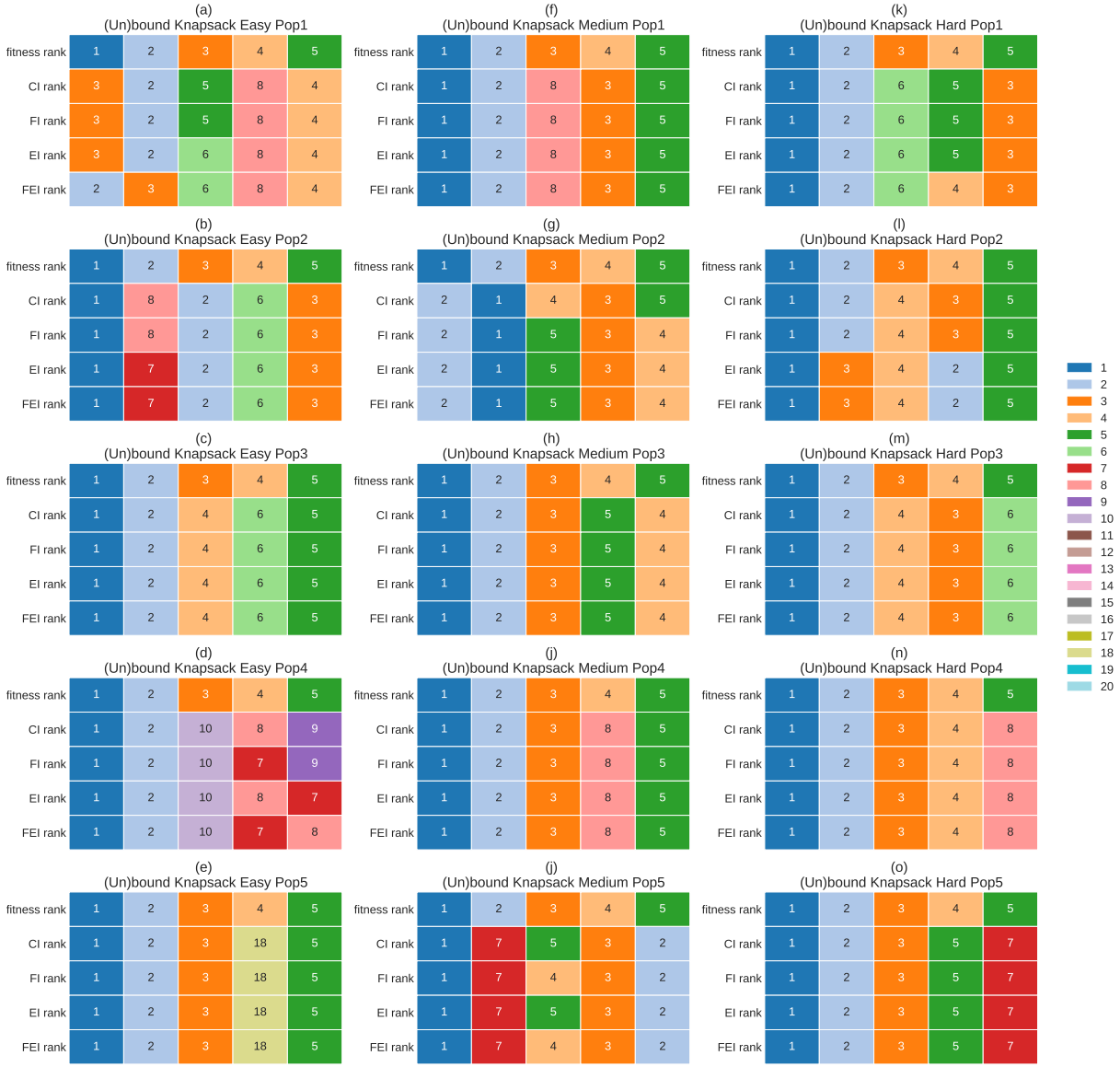

Figure B.27.: Top 5 (Un)bound Knapsack problem fitness vs impact ranking. The left column shows the easy tests, the middle column the medium and the right the hard tests. Each graph represents the initial fitness rank on the top, with the ranking of the four impact metrics below.

B.3. Hypothesis 2 additional plots

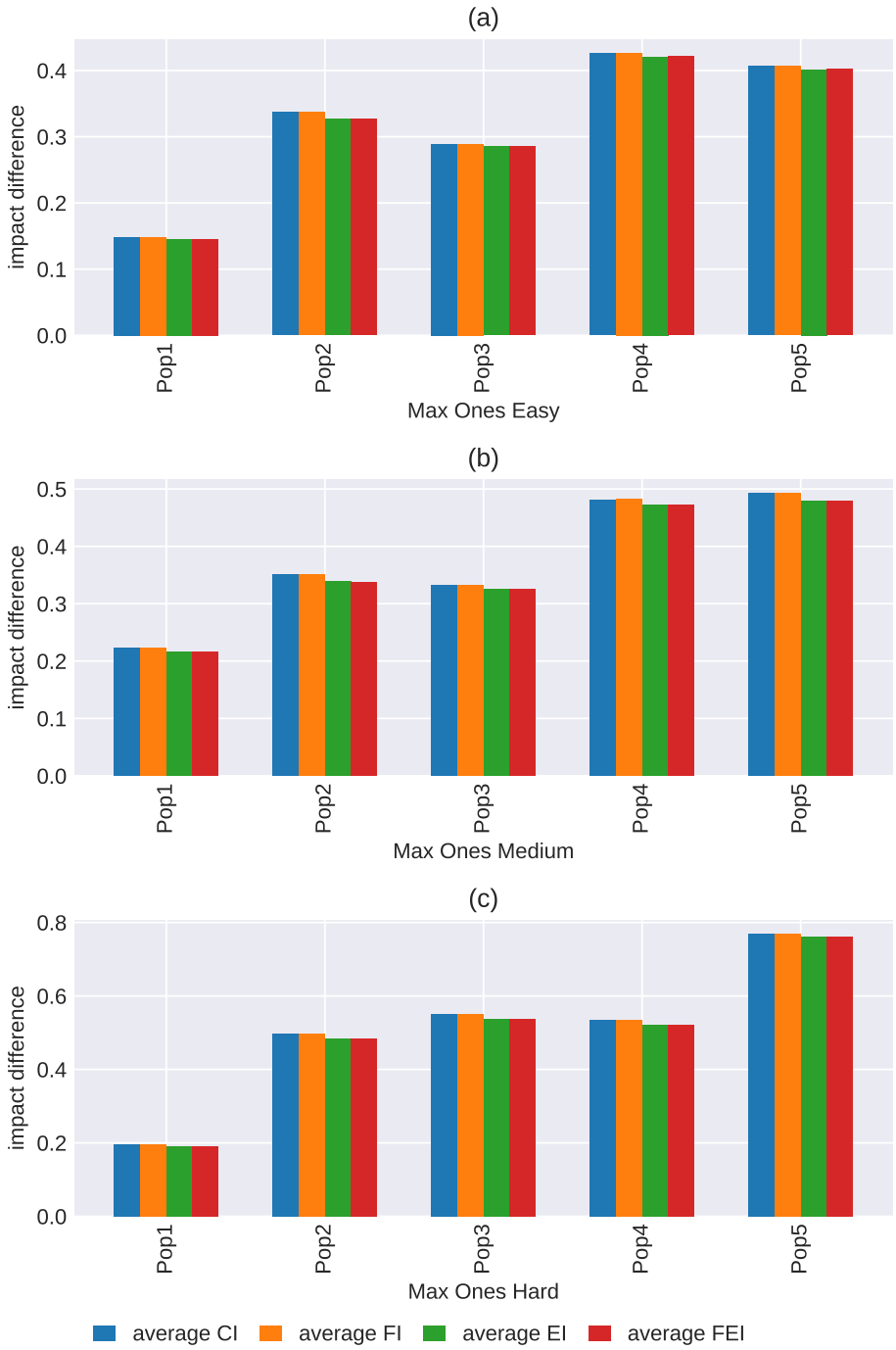

Figure B.28.: The difference between the highest and the lowest mean impact for every population from the Max Ones problem (used for hypothesis 1). Graph (a) shows the results of the easy tests, graph (b) of the medium tests and graph (c) of the hard tests.

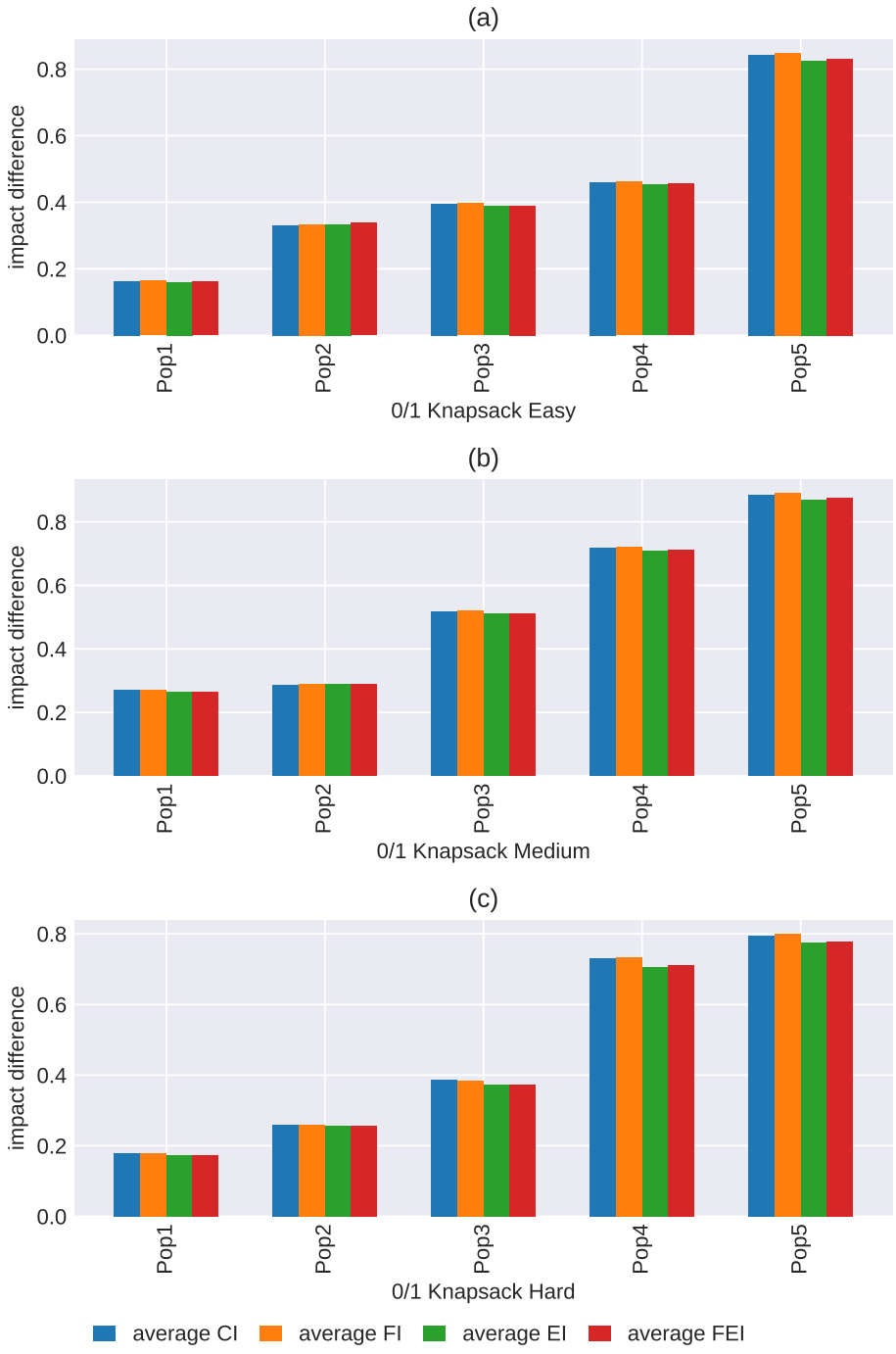

Figure B.29.: The difference between the highest and the lowest mean impact for every population from the 0/1 Knapsack problem. Graph (a) shows the results of the easy tests, graph (b) of the medium tests and graph (c) of the hard tests.

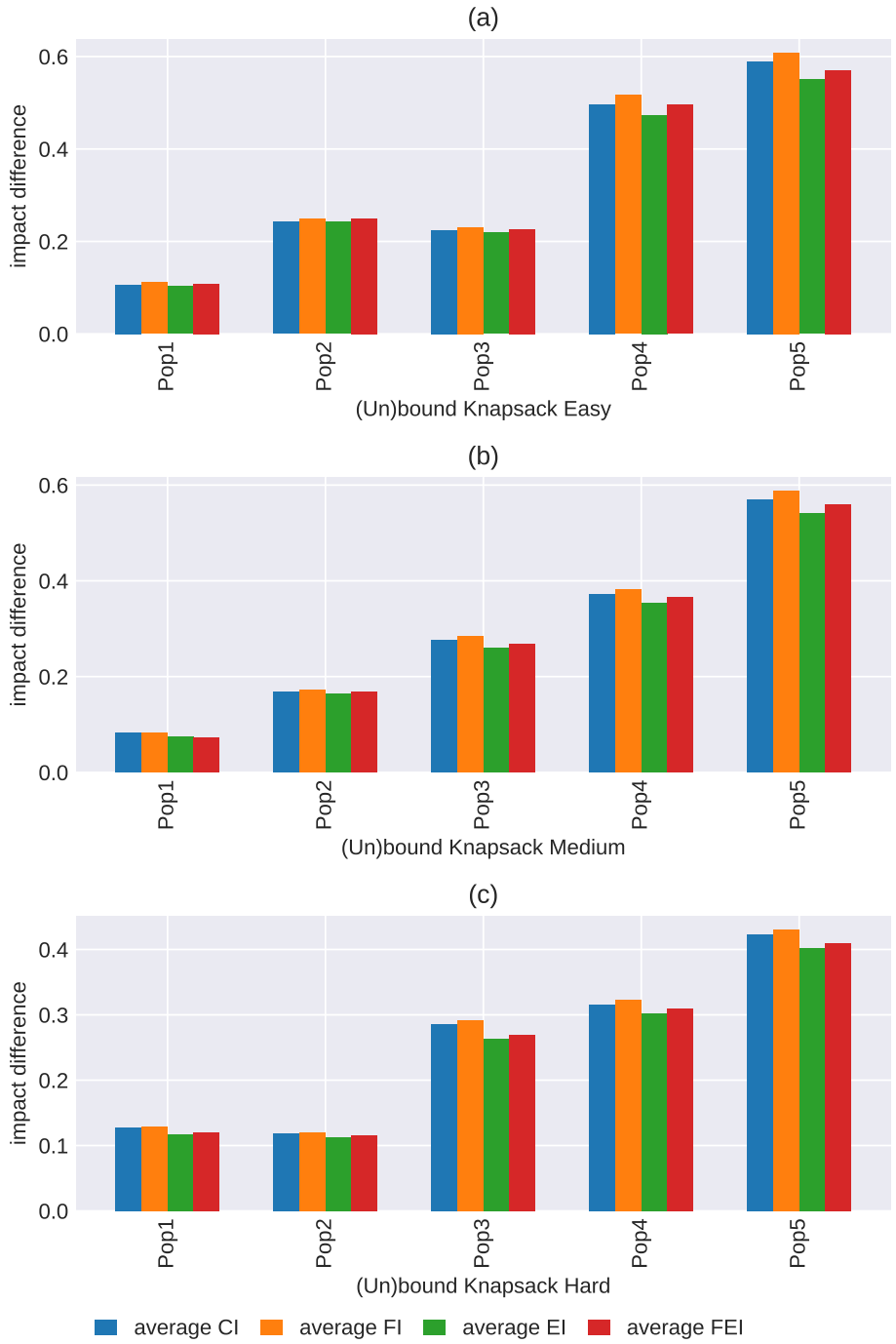

Figure B.30.: The difference between the highest and the lowest mean impact for every population from the (Un)bound Knapsack problem. Graph (a) shows the results of the easy tests, graph (b) of the medium tests and graph (c) of the hard tests.